Distributed information fusion in sensor networks

Boris Nikolai Oreshkin

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

December 2009

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

This thesis addresses the problem of design and analysis of distributed in-network signal processing algorithms for efficient aggregation and fusion of information in wireless sensor networks. The distributed in-network signal processing algorithms alleviate a number of drawbacks of the centralized fusion approach. The single point of failure, complex routing protocols, uneven power consumption in sensor nodes, inefficient wireless channel utilization, and poor scalability are among these drawbacks. These drawbacks of the centralized approach lead to reduced network lifetime, poor robustness to node failures, and reduced network capacity. The distributed algorithms alleviate these issues by using simple pairwise message exchange protocols and localized in-network processing. However, for such algorithms accuracy losses and/or time required to complete a particular fusion task may be significant. The design and analysis of fast and accurate distributed algorithms with guaranteed performance characteristics is thus important. In this thesis two specific problems associated with the analysis and design of such distributed algorithms are addressed.

For the distributed average consensus algorithm a memory based acceleration methodology is proposed. The convergence of the proposed methodology is investigated. For the two important settings of this methodology, optimal values of system parameters are determined and improvement with respect to the standard distributed average consensus algorithm is theoretically characterized. The theoretical improvement characterization matches well with the results of numerical experiments revealing significant and well scaling gain. The practical distributed on-line initialization scheme is devised. Numerical experiments reveal the feasibility of the proposed initialization scheme and superior performance of the proposed methodology with respect to several existing acceleration approaches.

For the collaborative signal and information processing methodology a number of theoretical performance guarantees is obtained. The collaborative signal and information processing framework consists in activating only a cluster of wireless sensors to perform target tracking task in the cluster head using particle filter. The optimal cluster is determined at every time instant and cluster head hand-off is performed if necessary. To reduce communication costs only an approximation of the filtering distribution is sent during hand-off resulting in additional approximation errors. The time uniform performance guarantees accounting for the additional errors are obtained in two settings: the subsample approximation and the parametric mixture approximation hand-off.

Sommaire

Cette thèse aborde le problème de la conception et l'analyse d'algorithmes distribuès servant à l'agrégation efficace et la fusion de l'information dans des reséaux capteurs sans fil. Ces algorithmes distribuès servent à addresser un bon nombre d'inconvénients qu'ont les approches de fusion centralisée telles que le point de défaillance unique, les protocoles de routage complexe, la consommation de puissance inégale dans les noeuds de capteurs, l'utilisation inefficace des voies de transmission sans-fil et l'extensibilité limitée. Ces inconvénients de l'approche centralisée ont comme effet de réduire la durée de vie du reséau, la robustesse des noeuds face aux défaillances et la capacité du réseau. Les algorithmes distribuès atténuent ces problèmes en utilisant des simples protocoles de messageries entre les noeuds ainsi que du traitement d'information localisé. Toutefois, pour ces algorithmes, les pertes de précision et/ou de temps nécessaire pour effectuer une tâche peuvent être importantes. C'est pourquoi la conception et l'analyse d'algorithmes distribuès rapide et précis est importante. Dans cette thèse, deux problèmes spécifiques associés à l'analyse et le conception de tels algorithmes sont abordés.

En ce qui concerne l'algorithme de consensus sur la moyenne distribuè, une méthode d'accélération fondé sur la mémoire est proposée et sa convergence analysée. Pour les deux paramètres importants de cette méthodologie, les valeurs optimales pour le système sont déterminées et l'amélioration par rapport à l'algorithme de consensus de base est caractérisée de façon théorique. Cette caractérisation correspond aux rêsultats d'expériences numériques et révèlent des gains importants et extensibles. Le régime distribuè d'initialisation en ligne est conçu. Des expériences numériques révèlevent la faisabilité du régime d'initilisation proposé ainsi qu'un rendement supérieur à plusieurs approches existantes.

Pour la méthodologie de traitement de signaux et d'information collaborative, un certain nombre de garanties théoriques de performance sont obtenues. Ce cadre de travail consiste à activer seulement une grappe de capteurs sans fil pour effectuer les tâches de pistage d'objet au niveau deu chef de groupe en utilisant un filtre particulaire. La grappe optimale est déterminée à chaque intervale de temps et le transfert du titre de chef de groupe est réalisé au besoin. Pour réduire les coûts de communication, seulement une approximation de la distribution du filtre est envoyé pendant le transfert de responsabilités ce qui entraîne des erreurs supplémentaires. Les garanties de performance uniformes dans le temps tenant compte de ces erreurs supplémentaires sont obtenues dans deux contextes.

Acknowledgments

I have received such immeasurable support from so many people, and in so many ways, that it seems impossible to enumerate them.

First and foremost, my deepest gratitude must go to Dr. Mark Coates, my thesis advisor and mentor. He has not only incredible vision and intelligence and boundless energy as a professor, but he is also a very gentle and kind soul. I am indebted to him for all the support, inspiration, and advice that made my research at McGill University an interesting and exciting experience. I am also very grateful to his family, Dr. Milica and little George, for their warmth, encouragement, and support, both intellectual and emotional.

My greatest thanks must also go to the members of my dissertation committee, Dr. Tal Arbel and Dr. Michael Rabbat. The successful realization of this dissertation and my academic achievements would not have been possible without their insightful comments, guidance, and support. It was a great pleasure and honor for me to write my thesis under their supervision. Especially, I would like to thank Mike Rabbat for his friendship, advice and constructive feedback on my research.

I am very grateful to Dr. Can Aysal, whose brilliant ideas were crucial for the effective progress of this dissertation, and Dr. Yvan Pointurier, who had a great impact on my work and who has been a true friend to me all these years.

I would like to express my sincere gratitude to the Faculty of Engineering and Department of Electrical and Computer Engineering at McGill University for all the wonderful courses I took and for the much appreciated financial support. The McGill Engineering Doctoral Award (MEDA – formerly known as the DDSRRA) helped me immensely in my endeavors.

My research would not have been possible without the generous financial support of the National Scientific and Engineering Research Council of Canada (NSERC) through the Discovery Grants program and the MITACS (Mathematics in Information Technology and Complex Systems) Networked Centres of Excellence.

Millions of thanks to my friends and colleagues, especially to Dennis, Mohammad, Konstantin, Frederic, Daniel, Fariba, Alex, Deniz, Hong, Abhay, and Xuan. Special thanks to Fred for translating the abstract into French; and to Fred, Abhay, and Deniz for proofreading the manuscript. My graduate school experience at McGill University was a very challenging, but extremely interesting and enjoyable experience and I am very grateful to

my colleagues for their friendship and understanding, which I hope will last forever.

I would also like to thank my dear parents. Without them I certainly would not come this far in my education. They not only provided me with the best possible opportunities, but also with all the support they could give. I am deeply grateful to them for everything. And of course exceptional thanks to my beloved wife, Katya, for all her love, patience and encouragement; for believing in me, supporting me, and helping me move forward.

Contents

List of Figures

List of Acronyms

CSIP	Collaborative Signal and Information Processing
DOI	Decentralized Orthogonal Iterations
GML	Greedy Likelihood Maximization
i.i.d.	independent and identically distributed
KL	Kullback-Leibler
MC	Monte Carlo
MD	Maximum Degree
MH	Metropolis-Hastings
MI	Mutual Information
MSE	Mean Squared Error
PDA	Personal Data Assistant
pdf	probability density function
QEP	Quadratic Eigenvalue Problem
RGG	Random Geometric Graph
RMSE	Root Mean Squared Error
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Motivation

Wireless sensor networks can be very effective mechanisms for acquiring, delivering, and processing complex data flows generated by a large variety of physical processes and environment sensing applications. Wireless sensor network (WSN) based architectures have great potential as the foundation of cost efficient and scalable solutions for sensing applications ranging from medical data dissemination in hospitals to fire control during disaster relief operations and military sensing and control on the battlefield [1]. There are still many important open problems related to building efficient WSN information fusion protocols [2, 3].

The centralized data fusion protocol assumes that a centralized entity (fusion center) gathers data captured by a WSN and performs data processing operations. The sensor nodes thus play the role of sensing and communication devices that acquire and route data to the fusion center. The internal signal processing capabilities available in most modern sensor nodes are thus not used in the centralized fusion approach. Moreover, the need for routing potentially large volumes of data acquired by the nodes in the network induces the following issues inherent to the centralized approach. Any node has to be able to transfer its data to the fusion center. Complex routing protocols are thus needed to establish the required data flows and account for the node failures and channel instability. In the centralized scenario, each node is responsible for routing data from a subset of nodes in the network. This results in the single point of failure issue and uneven power consumption in sensor nodes. Uneven power consumption in sensor nodes leads to early death of overloaded

nodes and reduces network lifetime. Since each node is responsible for routing data from a subset of nodes in the network, capacity of the network does not scale well with growing network size.

Distributed algorithms alleviate these drawbacks by reducing communication protocols to simple one-hop information exchanges with immediate neighbors and localizing sensing and signal processing operations. Distributed protocols often propagate the solution based on the acquired data instead of the raw data that are used to obtain the solution in the centralized fusion center. Clearly, this methodology relies on the processing capabilities available in the sensor nodes to obtain the distributed solution that is otherwise calculated at the centralized fusion center. The distributed solutions obtained by the WSN are often suboptimal. The nature of many distributed algorithms is asymptotic. Thus additional errors are inevitably introduced by resorting to the more robust and simple distributed solutions and the time required to complete a fusion task may by significant. It is therefore important to design and analyze fast and accurate distributed algorithms with guaranteed performance characteristics.

In this thesis we address two specific problems related to the design and analysis of fast and localized WSN information fusion algorithms with theoretical guarantees of performance. The first problem we address is the acceleration of the distributed average consensus protocol that is known to suffer from the poor scalability of averaging time. The second problem we consider is obtaining the theoretical approximation performance guarantees for the leader node particle filter using occasional intermittent approximations of two kinds: subsample approximation and parametric mixture approximation.

1.2 Synopsis

Chapter 2 provides WSN related background information and terminology. It outlines important WSN application domains, reviews recent publications describing different aspects of WSN applications, and discusses challenges associated with these applications. The last part of Chapter 2 introduces a number of foundational concepts related to the distributed processing in wireless sensor networks and discusses WSN application domains where distributed in-network processing can be used.

Chapter 3 presents the statement of the average consensus problem and reviews relevant literature on this topic. It presents important open issues inherent to the decentralized

solution of the consensus problem and outlines the approach that will be taken in Chapter 4 to alleviate these issues.

Chapter 4 describes the proposed technique to accelerate the convergence of asymptotic distributed consensus algorithms. The first part of Chapter 4 presents a general memory-based framework for consensus acceleration based on mixture of local prediction and the outcome of the conventional consensus iteration, discusses convergence properties of this framework, quantifies the rate of convergence and establishes the existence of convergent solution. The second part of Chapter 4 analyzes a technique for distributed optimization of consensus matrix based on the proposed methodology and its approximate distributed implementation. Finally, the last section of Chapter 4 analyzes accelerated consensus with short node memory based on the proposed framework and presents an efficient distributed routine for its initialization. In this last section, the value of the optimal parameter is determined in the short node memory framework, the rate of convergence of the improved algorithm is quantified and several results quantifying the improvement obtained using the proposed technique are presented.

Chapter 5 presents the overview of communication constrained collaborative WSN based distributed tracking methods and issues related to their analysis. Important concepts related to non-linear filter analysis are presented in conjunction with the analysis of collaborative particle filter based WSN framework performed in Chapter 6.

Chapter 6 investigates the performance of a collaborative WSN based target tracking application. Within the framework of this application nodes in the WSN form a two-tier clustered architecture. Only one cluster chosen based on the mutual information criterion is active at any time instant and intermittent approximations of tracking statistics obtained via in-network processing are used for cluster hand-offs. The approximation errors are investigated for this framework in two distinct settings: (1) the intermittent approximation is obtained as a sub-sample of a particle filter approximation and (2) parametric approximation is obtained from the particle approximation using a greedy mixture estimation algorithm. For the first case we obtain a number of inequalities characterizing approximation/sampling errors and exponential inequalities revealing tail behavior of the algorithm. For the second scenario we obtain inequalities characterizing approximation/sampling errors and formulate conditions for unbiased intermittent parametric approximation. It turns out that the frequency of intermittent approximations plays a vital role in the error characterization for both approximation scenarios under consideration. We conclude the chapter

with simulations that illustrate important practical aspects of the investigated scenarios.

Chapter 7 concludes the thesis by summarizing the problems studied in the thesis and results obtained in the thesis and discussing the future work.

1.3 Contributions

The original contributions of this thesis can be briefly outlined as follows.

1. The memory based methodology for the acceleration of the distributed average consensus algorithm based on the mixture of predictor and the outcome of standard consensus iteration.

2. The theoretical proof of the existence of the convergent configuration of the proposed memory based acceleration methodology.

3. The optimal value of the mixing parameter for the simplest configuration of the proposed memory based acceleration methodology.

4. The distributed suboptimal initializations of the mixing parameter.

These results have been obtained in collaboration with Dr. Tuncer C. Aysal and appear in[1]

T. C. Aysal, B. N. Oreshkin and M. J. Coates, Accelerated distributed average consensus via localized node state prediction, *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1563–1576, Apr. 2009.

B. N. Oreshkin, T. C. Aysal and M. J. Coates, Distributed average consensus with increased convergence rate, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Las Vegas, NV, pp. 2285–2288, Apr. 2008.

5. For the proposed memory based acceleration methodology, the upper bound on the growth rate of the limiting ε-averaging time for asymptotically small values of ε has

[1]Dr. Tuncer Aysal proposed to use a predictor to improve the performance of the distributed average consensus algorithm and played a supervisory role. Prof. Mark Coates played a supervisory role. Boris Oreshkin formulated the linear predictor based accelerated distributed average consensus algorithm and conducted mathematical analysis and numerical simulations.

been obtained. The parameter ε quantifies the largest acceptable network-wide ℓ_2 deviation from the average after one round of distributed average consensus.

6. The study of the predictive accelerated average consensus with short node memory and the value of the optimal mixing parameter for the short memory case.

7. The quantification of the convergence rate of the predictive accelerated average consensus with short node memory.

8. The asymptotically optimal configuration of the predictor weights for the predictive accelerated average consensus with short node memory.

9. The quantification of the average asymptotic convergence rate improvement achieved by the accelerated average consensus with short node memory.

10. The practical distributed on-line mixing parameter initialization scheme.

These results have been submitted or will appear in[2]

B. N. Oreshkin, M. J. Coates and M. G. Rabbat, Optimization and Analysis of Distributed Averaging with Memory, in *Proc. 47 Ann. Allerton Conf. Comm. Control Comp.*, Allerton, IL, Oct. 2009.

B. N. Oreshkin, M. J. Coates and M. G. Rabbat, Optimization and Analysis of Distributed Averaging with Short Node Memory, *IEEE Trans. Signal Process.*, under review.

11. The formulation of the Feynman-Kac formulae describing the propagation of distribution flows in the distributed wireless sensor network target tracking scenario based on the collaborative signal and information processing methodology.

12. The analysis of local weak-sense L_p approximation errors of the subsample approximation leader node particle filter.

13. The analysis of local weak-sense L_p approximation errors of the parametric mixture approximation leader node particle filter.

[2]Prof. Mark Coates and Michael Rabbat played supervisory roles. Boris Oreshkin conducted mathematical analysis and numerical simulations.

14. The time uniform weak-sense L_p error bounds for the leader node particle filter with intermittent subsample and parametric mixture approximations.

15. The exponential inequalities characterizing the probabilities of the large deviations of distribution flows for the subsample approximation leader node particle filter .

16. The conditions that guarantee the asymptotically unbiased approximation performance of the leader node particle filter with parametric mixture approximation.

These results appear in or have been submitted to[3]

B. N. Oreshkin and M. J. Coates, Analysis of error propagation in particle filters with approximation, *Ann. Appl. Probab.*, under review.

B. N. Oreshkin and M. J. Coates, Particle filters with approximation steps, *Int. Workshop Comp. Adv. Multi-Sensor Adapt. Process.*, Aruba, Dutch Antilles, Dec. 2009, to appear.

B. N. Oreshkin and M. J. Coates, Weak sense L_p error bounds for leader-node distributed particle filters, in *Proc. Int. Conf. Info. Fusion*, Cologne, Germany, Jul. 2008.

[3]Prof. Mark Coates played a supervisory role. Boris Oreshkin conducted mathematical analysis and numerical simulations.

Chapter 2

Wireless Sensor Networks

2.1 Background and Definitions

Wireless Sensor Networks have great potential to form the basis of universal and efficient sensing and fusion architectures. Due to relatively recent advances in science and technology it has become possible to build small sensor nodes with on-board sensing, processing, and communication capabilities. On the other hand, the development of tiny electrochemical elements with relatively large capacities has enabled the creation of autonomous sensor nodes that can operate in scenarios with minimal human intervention. We adopt the following definition for a sensor node.

Definition 2.1. *A sensor node is a device with on-board sensing, processing, and communication capabilities equipped with an autonomous power source.*

In the minimal human intervention scenario a set of sensors nodes comprises an autonomous network that can be used to monitor certain physical phenomena, interpret its observations, and dissimilate related information flows. According to one of the definitions from [4] *a network* is 'a complex, interconnected group or system'. Thus, strictly speaking, to form a network these devices should be interconnected by physical links to be able to process information collaboratively since resources of a single sensor may not suffice to accomplish complex sensing, processing, and control tasks [5–7]. There are multiple ways to connect sensors: wires, radio transmission, optical cables, laser links, etc. Many of them require careful installation and tuning (e.g. laser links), others need the establishment of a costly infrastructure (according to [8] $200 additional expenditures per sensor can be

incurred in wired networks). In many scenarios sensors are deployed without prior preparation of an infrastructure. Limited time and other resources can prevent building such an infrastructure. In such situations using wireless radio links to build a collaborative sensor network is an 'inevitable requirement' [1].

A set of sensors comprises the heart and the basic working horse of the collaborative network. However, in most situations this set per se is not enough to make the network usable. Sensors often serve only as sources (and/or processors, routers) with respect to the in-network information flows. Thus it is also important to define the notion of a requestor of information or a certain network activity.

Definition 2.2. *A sink is the destination node that consumes either raw or preprocessed and aggregated sensor measurements and often requests certain network activities or operations.*

Sensors and sinks are basic elements of a sensor network. Based on the above discussion we can introduce the following definition of a WSN consisting of these basic elements.

Definition 2.3. *A wireless sensor network is a set of geographically spread sensors and sinks equipped with radio transceivers and interconnected via wireless radio channels in order to observe certain physical phenomena, interpret these observations and broadcast the interpretations to sinks.*

There exist a considerable number of alternative definitions (see e.g. [1, 2, 9, 10] and references therein). The differences in the definitions arise mainly due to explicitly including actuators [2] or not including sinks [1] and explicitly considering inherent sensor density [9] or geographical [10] constraints. The above definition combines many WSN features generally found and accepted in the literature, while some other aspects (e.g. possible presence of actuators in the network) are omitted since they lie outside of the scope of this thesis. Another important feature of Definition 2.3 is that sinks are considered to be a part of the network (cf. [2]). While sinks are often both requestors and receivers of WSN information, physically they may or may not be part of the network. Indeed, within the distributed signal processing framework, sensor nodes themselves are often sinks due to the algorithmic structure of distributed computations. On the other hand, firefighters reading data from a temperature sensing WSN are clearly not part of the WSN. However, when a firefighter requests data from the WSN located in a forest (e.g. using a PDA [1]) this PDA becomes a part of the WSN. Thus sinks can often be considered part of a WSN. While connected

to WSN, sinks can provide it with feedback and instructions necessary for its configuration or translate WSN data flows to the outside world (e.g. Internet or remote users [1]).

While Definition 2.3 and the preceding discussion clearly reflect important physical aspects of WSNs, they cannot be employed to describe and model WSNs mathematically. The most important WSN modeling aspect is its connectivity. A common approach to connectivity modeling is based on graph theory. In particular, the following definition of a graph can be adopted [11, 12]:

Definition 2.4. *A graph $G = (V, E)$ consists of a set of vertices V and a set E of two-element subsets of V defining the adjacency relations between vertices. E is called a set of edges since $(u, v) \in E$ holds for two vertices $u, v \in V$ if and only if there is an edge between v and u (v is adjacent to u).*

There is a clear correspondence between the definition of a graph and the definition of a WSN [2]. Vertices in a graph correspond to the sensor nodes and edges correspond to wireless radio links in the related WSN. A number of graph based models for WSNs are presented in [11]. The generality and complexity of these models vary with their ability to represent complex connectivity and propagation effects such as fading, shadowing, interference, and multi-hop transmissions. Many of the graphical models in [11] are hard to analyze. A simpler, but more tractable, *random geometric graph* (RGG) model was proposed in [13] in the context of wireless network capacity analysis. This model has often been successfully used to analyze different performance aspects of WSN based algorithms (see e.g. [14]) due to a large number of available theoretical results describing the connectivity and information diffusion properties of WSN under this model. The definition of the RGG model is given below (see [13–15] for details).

Definition 2.5. *A random geometric graph $G^d(n, r)$ is a graph of n nodes with connectivity radius r obtained by placing n vertices on a d-dimensional cube uniformly at random and connecting any two vertices if and only if they are within distance r of each other.*

2.2 Challenges and Applications

The applications of WSNs include several major areas [1, 16]: event detection and classification, tracking, estimation and function approximation, and reporting periodic measurements.

Event detection and classification includes a large number of diverse environment tracking, industry and military applications. In these applications, sensors often make local decisions about a phenomenon being observed and then either transmit these local decisions to a sink or make a collaborative network-wide decision (e.g. using a decentralized protocol) and communicate it to a sink. For example, in [17, 18] an experimental heterogeneous two-level network was deployed along roadway in Los Alamos to investigate the potential of distributed sensor networks for detecting radiological dispersal devices transported in vehicles. A general framework for the software design of distributed event detection algorithms was developed in [19] with the emphasis on the detection of explosion events using nodes with temperature, light, and acoustic sensors. Performance of collaborative learning event detection algorithms applied to fence monitoring has been investigated for varying compression levels of feature extraction algorithm [20, 21]. Another intrusion detection application is the cooperative algorithm developed by Krontiris et. al [22]. Other interesting examples include a flood detection application with a deployment installed in Honduras [23], medical emergency detection using custom-built motes [24], object detection in sparsely sampled networks using a grid of TelosB nodes [25], and the Debris-flow warning system [26].

Tracking has wide applicability in the environment monitoring, security, and network self-organization applications. In a typical tracking scenario nodes generally observe relevant parameters of a target (e.g. range, bearing, velocity, etc.) and transmit this information to the cluster head (if WSN clustering is employed), fusion center (in the case of centralized processing) or collaboratively track a target and disseminate tracking statistics. A centralized 25-node WSN grid acoustic array prototype using MicaZ nodes was described in [27]. Mobile node tracking using anchor nodes and radio-interferometry principle [28] can be used to obtain improved node location information. This information can often be used to construct efficient routing protocols [2] or is necessary for performing accurate multi-sensor tracking and fusion. The medical asset tracking application was reported in [29], where mobile nodes are attached to medical equipment to decrease the time required to find necessary equipment in a critical situation. In [30] the Debris-flow tracking system is designed, including the prototype sensors with on-board accelerometer and omnidirectional antenna that can be employed on the riverbank to analyze vibration information and track Debris-flow movements. A target tracking application based on a WSN consisting of binary nodes encoding the relative target movement direction was presented in [31]. He et al. address [32] a practical real-time security application based on the VigilNet [33] platform.

Using field experiments and extensive simulations they provide guidelines for building the WSN based real-time security systems using target tracking, detection, and classification modules. Onel et al. [34] investigate theoretical aspects of surveillance network deployment. In particular, they determine the size of the network necessary to ensure required performance characteristics (such as breach probability and coverage) and design Mutual Information (MI) based metric for sensor scheduling.

Estimation and function approximation can be applied to learn environmental conditions, patterns, and important features of spatio-temporal fields induced by observed phenomena. In this scenario, each node in a dense WSN observes noisy measurements of a spatio-temporal process. After that, using an inference routine (that can be either distributed or centralized) important parameters or features of the spatio-temporal field are estimated or interpolated and the resulting approximation is learned over the entire network and/or communicated to a sink. A classical function approximation application is the poisonous gas plume boundary estimation [9]. A practical WSN boundary estimation algorithm was developed by Duttagupta [35] using a kernel-based regressor of the boundary obtained from aggregates transmitted by cluster heads at every time step. The parameters of the kernel regressor are updated from one time instant to another using the Kalman filter, thus forming the spatio-temporal field representing the moving boundary front (and associated confidence bounds). Zhao and Nehorai applied WSN based parameter learning (surface fitting) for centralized [36] and distributed [37] estimation to localize moving sources with applications to security, explosive detection, and pollution control. Measurement prediction based on an estimate of a model of the observed physical phenomenon can be used to minimize the amount of data transmitted by the WSN [38, 39]. Path-loss exponent estimation [40] can be used for improving the performance of received signal strength based localization techniques [41, 42]. Received signal strength based location estimators are critical ingredients in such WSN applications as precision agriculture, water quality monitoring, intrusion detection, inventory monitoring [42], and emergency resource estimation [43] during disaster relief operations. Acoustic source direction of arrival estimation in WSN can be used for habitat monitoring and smart environment applications [44].

The distributed consensus problem [45] belongs to the class of distributed function approximation and estimation problems. Protocols based on the distributed consensus paradigm can be applied to sensor fusion [46] formation control for multi-robot systems [47], distributed load balancing [48], etc.

Reporting periodic measurements is a typical monitoring application where each sensor is expected to produce and convey to the sink a continuous or triggered flow of monitoring information [16]. Remote sensors often form a group and their data are relayed through a wireless link to a processing center. A classical example from this area is the pipeline monitoring application [49] where there is a need for high sampling rate, synchronous data logging. Werner-Allen et al. [50] use this framework to monitor volcanic activity. Here, as well as in the Debris-flow monitoring [30] event detectors are employed to trigger data flows, however, after triggering, measurement data flows become continuous and periodic. This concept of storing/transmitting only those readings that are informative is especially useful when data volumes acquired by the WSN are large or sensors are deployed for long-term unattended operation. For example, in the parking lot monitoring application [51] a network of acoustic sensor nodes detects certain events (e.g. slamming door) localizes these events, and provides event coordinates to a camera network. The camera network uses movement detection to improve the localization estimate, logs the event and reports the event to a human operator.

Challenges that WSN designers have to face are tightly connected to the Quality-of-Service metrics that their WSN designs have to satisfy within the framework of intended application. It can be seen from the discussion of WSN applications that in many scenarios WSNs are deployed for long-term unattended operation. Thus one of the first challenges in WSN applications is *maintainability*. *Self-configuration* is another desirable property of a WSN related to its ability to initialize without manual intervention at the time of deployment. Maintainability is related to the *adaptivity* of the WSN. A self-configured WSN should be able to adapt to changing environmental conditions, failing nodes, and depleting energy sources to be able to maintain its operability over long periods of time exhibiting *fault tolerance*. These concepts are closely related to one of the most important WSN Quality-of-Service metrics: network lifetime. In the literature, there is agreement on the following general definition of WSN lifetime [52]:

Definition 2.6. *Network lifetime is the time span from the deployment to the instant when the network is considered non-functional.*

However, the instant when network is considered non-functional can be defined in different ways based on the number of operational nodes remaining, sensor coverage, connectivity, or estimation error (see Dietrich and Dressler [53] and Verdone et al. [2]). Often maximiza-

tion of the network lifetime is directly related to minimizing the energy consumption within and evenly distributing the consumption among WSN nodes. Thus one of the important challenges in WSN designs is their *energy-efficient operation*.

In order to be able to design energy-efficient WSN protocols one needs to know the energy consumption characteristics of the nodes. These characteristics are application-dependant, however it is still possible to define general trends. A sensor node typically consists of a sensing device, processor, and transceiver. For a typical Mica2 node, the transceiver draws 27 mA on transmit, 10 mA on receive operation, and 1 μA during sleep mode [54]. Typical passive sensing devices (temperature, humidity, pressure, etc.) draw from 0.01 to 1.5 mA — and in many cases their consumption can even be ignored[1] [1]. On the other hand, MPR400 processor used in the Mica2 node draws 8 mA during active operation and < 15 μA in the sleep mode [55]. Thus in many passive sensing applications communication and computation costs dominate power consumption. On the other hand, it is known that the ratio of energy consumption necessary to transmit a single bit and calculate a single instruction is often greater than 1000 [1]. Thus communication costs often dominate power consumption in WSNs. It does not mean that computation costs can be neglected. However, it does imply that performing reasonably complex in-network signal processing operations reducing communication load can result in significant energy savings, often leading to extended WSN network lifetime.

Another important aspect of WSN design is its *scalability*. For example, it was shown [13] that the scalability of the capacity of the flat centralized WSN under the RGG model can be poor. A concise definition of scalability can be found in [1].

Definition 2.7. *Scalability is the ability to maintain performance characteristics irrespective of the size of the network.*

Scalability issues include growing required sensor memory or processing power, number of iterations or amount of energy spent per WSN task, capacity of the network, etc. In practice, it turns out [1] that one of the effective means of alleviating scalability problems in WSNs is the deployment of local or distributed algorithms (these concepts will be clarified in the next section). Distributed algorithms can also eliminate the *single point of failure* problem inherent to the centralized WSN algorithms [56]. On the other hand, local

[1] However, for active sensing applications (such as radar, sonar, or lidar) sensing power consumption can be the dominating power load.

and distributed WSN algorithms often employ *collaboration and in-network processing* (e.g. different forms of aggregation) to reduce the amount of data being transmitted. Collaboration and in-network processing also help to address another important challenge in WSN architectures: the capabilities of a single sensor may not suffice to perform required WSN tasks.

Finally, in the WSN based tracking applications an important aspect of the Quality-of-Service is the estimation performance captured by the tracking or prediction error. As a rule, in collaborative target tracking scenarios only a subset of nodes that are close (in some sense) to the target are activated [57]. This solution resolves scalability and, to certain extent, network lifetime issues. However, there is a trade-off between the number of nodes activated at any given point and the estimation accuracy. Typically a larger number of active nodes provides more information about target characteristics thus allowing for more accurate tracking. However, larger number of active nodes requires more energy to be spent on sensing (this amount can be considerable in the case of active sensing) and communication of measurements. Thus an important challenge in applications of this kind is to maintain the trade-off between tracking accuracy and sensing and communication energy costs.

2.3 Decentralized in-Network Processing in Wireless Sensor Networks

It was mentioned in the previous section that distributed (decentralized) WSN algorithms can alleviate such important WSN issues as poor scalability or insufficient network lifetime and eliminate single point of failure sources. To understand the nature of this class of algorithms it is important to introduce related definitions. The following definition of a distributed algorithm is based on the description provided by Lynch [58].

Definition 2.8. *A distributed algorithm is the algorithm operating on a network of processors that run concurrently and independently and each of them has no global information.*

Thus the important features of a distributed algorithm are that every node runs its own part of the common code, independently of the others, and accesses only a limited amount of the global information (information available to a centralized controller if it gatherers all the data available in the WSN). Ideally, a decentralized algorithm achieves performance

identical to the centralized algorithm at least asymptotically (when either time, energy, or the number of nodes in the WSN tends to infinity). Due to the partial data access property of a decentralized algorithm, many distributed algorithms are also local. A local algorithm has the following definition [1].

Definition 2.9. *Local algorithm is an algorithm in which a node has accesses only to its own information and information in its neighborhood.*

A sensor neighborhood is often defined as follows [11]:

Definition 2.10. *The neighborhood of a node is the set of nodes (neighbors) with which the node can establish bidirectional single-hop wireless communication.*

Typical examples of decentralized algorithms include decentralized detection [59] or decentralized parameter estimation [60] where each node makes local decisions or locally quantizes its measurements and communicates the result to a fusion center. Graphical model based belief propagation algorithms [61] naturally fit the distributed WSN fusion framework since they are based on pairwise message exchanges. Distributed consensus algorithms [45] represent the simplest instance of a message passing algorithm (in fact, consensus propagation is a special case of belief propagation [62]). Finally, the collaborative WSN target tracking strategy [57] that activates only the informative subset of nodes and uses local approximations to simplify the message passing procedure combines many important WSN data acquisition and fusion concepts: locality, decentralization, and active data acquisition (only those nodes that 'count' are activated).

Chapter 3

Distributed Consensus and Agreement

In the distributed consensus problem, each node initially has a value, e.g., captured by a sensor, and the goal is to calculate the group consensus value that is a function of all initial values at the nodes in the network. The constraint is that information can only be exchanged locally. Thus at every time instant every node only knows its own value and the values of its neighbors, with whom it can communicate directly.

Distributed consensus algorithms have received considerable attention in the literature. De Groot [63], Borkar and Varaiya [64], and Tsitsiklis [65] were among the first scholars that studied distributed consensus (agreement) problems. Historically, the consensus problem first appeared in management science and statistics [45] and then migrated to the control community. Since then consensus based distributed agreement and estimation has been an active research area in the control, distributed computing, and signal processing communities. This chapter outlines the general consensus framework, links it to a more specific average consensus problem and reviews relevant literature addressing the fast (accelerated) consensus framework.

3.1 Distributed Consensus Framework

Within the distributed consensus framework n individuals (agents) forming a set of vertices $V = \{1, \ldots, n\}$ act together as a team to reach an agreement regarding a value of some parameter (estimation problem) or the assignment of probabilities in a probability

distribution (detection problem). Agents are connected via a graph $G(V, E)$ defined by a set of vertices V and a set of edges E. The edge between agents i, j is denoted (i, j) and two agents i and j are connected if and only if $(i, j) \in E$. The distributed consensus problem is non-trivial if each agent i, $i = 1, \ldots, n$ is only connected to a subset $\mathcal{N}_{[i]} \subset V$ such that $\mathcal{N}_{[i]} = \{j : (i, j) \in E\}$ (the corresponding graph is not *complete*). $\mathcal{N}_{[i]}$ is called a closed neighborhood; an open neighborhood $\mathcal{N}_i$ of agent i, on the other hand, excludes agent i: $\mathcal{N}_i = \{j : (i, j) \in E, j \neq i\}$. Using the connectivity established via graph $G(V, E)$ agents work collaboratively to reach common objective generated by the *agreement function*. Agent i has an initial scalar measurement $x_i(0)$ and these measurements can be stacked to form a vector $\mathbf{x}(0) = [x_1(0), \ldots, x_n(0)]^\mathsf{T}$. The *agreement function* is a generic continuous and permutation invariant function of these initial values $\mathbf{x}(0)$, $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}$. The goal of a distributed consensus algorithm is to reach the agreement $\mathcal{A}(\mathbf{x}(0))$ iteratively only using *local* information. As follows from Definition 2.9, for every agent i its local information is enclosed in its neighborhood $\mathcal{N}_{[i]}$. The set of values comprising ith agent's local information at iteration t is denoted $\mathbf{x}_{\mathcal{N}_{[i]}}(t) = \{x_j(t) : j \in \mathcal{N}_{[i]}\}$. The agreement protocol is said to reach the agreement asymptotically if $\|\mathbf{x}(t) - \mathcal{A}(\mathbf{x}(0))\mathbf{1}\| \to 0$ as $t \to \infty$. The concise definition of a distributed consensus problem corresponding to the above description can be adopted from [66].

Definition 3.1 (Consensus Problem)**.** *For a given agreement function, determine a (distributed stationary) protocol, that makes the agents asymptotically reach consensus for an arbitrary initial state.*

For agreement functions of the following general structure

$$\mathcal{A}(\mathbf{x}(0)) = f\left(\sum_{i=1}^{n} g(x_i(0))\right), \tag{3.1}$$

$f, g : \mathbb{R} \to \mathbb{R}$, the design of distributed consensus algorithms is particularly simple. In particular, assuming that

$$\frac{\mathrm{d}g}{\mathrm{d}x_i}(a) \neq 0, \forall a \in \mathbb{R} \tag{3.2}$$

the following protocol leads to a convergent solution [66]:

$$x_i(t) = x_i(t-1) + \alpha \frac{1}{\mathrm{d}g/\mathrm{d}x_i(x_i(t-1))} \sum_{j \in \mathcal{N}_i} \phi(\vartheta[x_j(t-1)] - \vartheta[x_i(t-1)]). \tag{3.3}$$

Here $\phi, \vartheta : \mathbb{R} \to \mathbb{R}$, $\phi(\cdot)$ is continuous, locally Lipschitz, odd and strictly increasing; $\vartheta(\cdot)$ is differentiable; and $\alpha > 0$. protocol (3.3) thus defines a sufficiently general framework for distributed consensus computations. For example, arithmetic, geometric, harmonic, and order-p means are all calculable within this framework [66]. Stability conditions leading to asymptotic convergence of the protocol are discussed in [66].

3.2 Distributed Average Consensus

3.2.1 Distributed Average Consensus Protocol

Protocol (3.3) defines a general framework for distributed consensus computations. A simpler *distributed average consensus* problem boils down to the distributed computation of arithmetic means. This problem arises, e.g., in the case of distributed parameter estimation in white Gaussian noise, or distributed detection when it is necessary to calculate the log-likelihood ratios represented by sums. The distributed average consensus problem can thus be defined as follows.

Definition 3.2 (Average Consensus Problem)**.** *For the arithmetic mean agreement function, determine a (distributed stationary) protocol, that makes the agents asymptotically reach consensus for an arbitrary initial state.*

The arithmetic mean agreement function is simply

$$\mathcal{A}_{\mathrm{am}}(\mathbf{x}(0)) = \frac{1}{n} \sum_{i=1}^{n} x_i(0). \tag{3.4}$$

The protocol (3.3) for the arithmetic mean agreement function can be inferred from the structure of the agreement function. In particular, $f(x) = 1/n$, $g(x) = x$, and $\vartheta(x_i) = \vartheta(x_j) = \vartheta_{i,j}$ lead to the following distributed average consensus update rule:

$$x_i(t) = x_i(t-1) + \alpha \sum_{j \in \mathcal{N}_i} \vartheta_{i,j}(x_j(t-1) - x_i(t-1)). \tag{3.5}$$

This update rule can be written in the matrix form

$$\mathbf{x}(t) = \mathbf{W}\mathbf{x}(t-1). \tag{3.6}$$

Matrix $\mathbf{W}$ is often called the consensus weight matrix. It follows from (3.5) that the consensus weight matrix is constrained by the network topology $G(V, E)$ as $\mathbf{W} \in \mathcal{W}(V, E)$, where the topology constrained set of potential consensus weight matrices is defined as follows:

$$\mathcal{W}(V, E) = \{\mathbf{W} \in \mathbb{R}^{n\times n} : W_{i,j} = 0 \quad \text{if} \quad (i, j) \notin E\} \tag{3.7}$$

The general sparse structure of the distributed average consensus weight matrix can thus be defined as follows:

$$W_{i,j} = \begin{cases} W_{i,j} = 0 & \text{if} \quad (i, j) \notin E \\ W_{i,j} = 1 - \alpha \sum_{j\in\mathcal{N}_i} \vartheta_{i,j} & \text{if} \quad i = j \\ W_{i,j} = \alpha\vartheta_{i,j} & \text{otherwise} \end{cases}. \tag{3.8}$$

To make consensus protocol (3.6) completely decentralized, the distributed weight matrix construction rule needs to be established. Several such construction rules based on the relationship to local node degrees (the degree d_i of node i is the number of its immediate neighbors, $d_i = |\mathcal{N}_i|$) are known in the distributed average consensus literature. In particular, the Maximum Degree (MD) weight design scheme [67] is obtained by setting $\alpha = 1/d_{\max}$ and $\vartheta_{i,j} = 1$, where $d_{\max} = \max_{i\in V} d_i$ is the maximum degree of the graph. Sometimes the MD scheme is initialized by setting $\alpha = 1/n$, i.e. using the maximal possible degree [68, 69]. The Metropolis-Hastings (MH) weight design scheme [67] is obtained by setting $\alpha = 1$ and $\vartheta_{i,j} = 1/\max(d_i, d_j)$.

3.2.2 Randomized Gossip Protocol

The distributed average consensus protocol outlined in the previous section belongs to the family of *synchronous* protocols. In the synchronous setting, all nodes wake up at time instant t, exchange data with their immediate neighbors and perform the consensus update (3.5). The randomized gossip protocol, on the other hand, is the *asynchronous*

protocol. In a typical asynchronous setting only one node, i, wakes up at time instant t and selects one of its neighbors, j, randomly, according to a prescribed probability distribution (see e.g. Boyd et al. [14] for details). These two nodes then exchange information and update their values, $x_i(t)$ and $x_j(t)$, to their average $(x_i(t-1)+x_j(t-1))/2$. The relationship between the performance of the distributed average consensus and the performance of the randomized gossip protocols was investigated by Denantes et al. in [70]. The study in [70] revealed that these protocols have similar performance in terms of the normalized time to achieve required level of accuracy in a static network scenario. In the case of network with link failures the distributed average consensus protocol outperformed gossiping protocol in terms of the number of messages transmitted (communication costs) for the same prescribed accuracy level. However, this increased robustness of the distributed average consensus protocol comes at the cost of the need for establishing synchronization. In the rest of the thesis we concentrate exclusively on the synchronous distributed averaging scenario.

3.2.3 Convergence of Distributed Average Consensus

The convergence conditions for the distributed average consensus protocol (3.6) were studied by Xiao and Boyd [71]. In the distributed average consensus framework the agreement function $\mathcal{A}_{\text{am}}(\mathbf{x}(0))$ is equal to any element of the average vector $\overline{\mathbf{x}}(0)$:

$$\overline{\mathbf{x}}(0) = \frac{\mathbf{1}\mathbf{1}^{\mathsf{T}}}{n}\mathbf{x}(0), \tag{3.9}$$

Where $\mathbf{1}$ is the vector of all ones having an appropriate size (e.g. in the above equation it has the same dimension as $\mathbf{x}(0)$, $n \times 1$). The average vector is the result of applying the ideal weight matrix (weight matrix obtained on a complete graph), $\mathbf{J} = \mathbf{1}\mathbf{1}^{\mathsf{T}}/n$, to the vector of initial states $\mathbf{x}(0)$. Thus the distributed average consensus protocol asymptotically converges to its agreement function $\mathcal{A}_{\text{am}}(\mathbf{x}(0))$ if and only if the following holds for $\mathbf{W} \in \mathcal{W}(V, E)$ [71]:

$$\lim_{t\to\infty} \mathbf{W}^t = \mathbf{J}. \tag{3.10}$$

The necessary and sufficient conditions for this relation to hold can be summarized in the following theorem that is cited here from [71] without a proof. Before stating this result we introduce the definitions of the spectral radius of a matrix [72], the asymptotic convergence

rate $r_{\text{asym}}(\mathbf{W})$ and the step-wise convergence rate $r_{\text{step}}(\mathbf{W})$.

Definition 3.3. *Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the (real or complex) eigenvalues of a matrix $\mathbf{W} \in \mathbb{C}^{n\times n}$. Then its spectral radius $\rho(\mathbf{W})$ is defined as:*

$$\rho(\mathbf{W}) = \max_{i=1,\ldots,n} |\lambda_i|$$

The asymptotic convergence rate

$$r_{\text{asym}}(\mathbf{W}) = \sup_{\mathbf{x}(0)\neq\overline{\mathbf{x}}(0)} \lim_{t\to\infty} \left(\frac{\|\mathbf{x}(t) - \overline{\mathbf{x}}(0)\|_2}{\|\mathbf{x}(0) - \overline{\mathbf{x}}(0)\|_2} \right)^{1/t} \tag{3.11}$$

defines the asymptotic convergence time

$$\tau_{\text{asym}}(\mathbf{W}) \triangleq \frac{1}{\log(1/r_{\text{asym}}(\mathbf{W}))}, \tag{3.12}$$

which, asymptotically, corresponds to the number of iterations required to reduce the error $\|\mathbf{x}(t) - \overline{\mathbf{x}}(0)\|_2$ by a factor of e^{-1} [71]. The step-wise convergence rate

$$r_{\text{step}}(\mathbf{W}) = \sup_{\mathbf{x}(t)\neq\overline{\mathbf{x}}(0)} \frac{\|\mathbf{x}(t+1) - \overline{\mathbf{x}}(0)\|_2}{\|\mathbf{x}(t) - \overline{\mathbf{x}}(0)\|_2} \tag{3.13}$$

quantifies the amount of guaranteed, worst-case, error contraction attained at every time step.

Theorem 3.1 (Xiao and Boyd [71], Theorem 1)**.** *The equation* (3.10) *holds if and only if*

$$\mathbf{1}^T\mathbf{W} = \mathbf{1}^T, \tag{3.14}$$

$$\mathbf{W1} = \mathbf{1}, \tag{3.15}$$

$$\rho(\mathbf{W} - \mathbf{J}) \leq 1, \tag{3.16}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix. Moreover,

$$r_{asym}(\mathbf{W}) = \rho(\mathbf{W} - \mathbf{J}) \tag{3.17}$$

$$r_{step}(\mathbf{W}) = \|\mathbf{W} - \mathbf{J}\|_2 \tag{3.18}$$

(Here $\|\cdot\|_2$ denotes the spectral norm, or maximum singular value.)

The conditions of Theorem 3.1 are intuitive. The first condition ensures stationarity of the protocol, i.e. the agreement function is preserved at any iteration:

$$\mathcal{A}_{\text{am}}(\mathbf{x}(i)) = \mathcal{A}_{\text{am}}(\mathbf{x}(j)), \quad \forall i, j \geq 0. \tag{3.19}$$

The second condition ensures that the agreement function generates the stationary point of the protocol, $\mathcal{A}_{\text{am}}(\mathbf{x}(0))\mathbf{1}$. Finally, the last condition ensures that the associated Markov chain is irreducible and aperiodic [71] (if $\mathbf{W}$ is term-wise non-negative). The distributed MD and MH weight design schemes mentioned earlier satisfy the conditions of Theorem 3.1 and thus lead to convergent distributed average consensus protocols if the underlying graph $G(V, E)$ is *connected* [68] (in a connected graph there is a path from any vertex to any other vertex).

Theorem 3.1 also identifies the asymptotic convergence rate $r_{\text{asym}}(\mathbf{W})$ and the step-wise convergence rate $r_{\text{step}}(\mathbf{W})$. Using the definition of the step-wise convergence rate $r_{\text{step}}(\mathbf{W})$, the error at time t can be upper bounded as follows:

$$\|\mathbf{x}(t) - \overline{\mathbf{x}}(0)\|_2 \leq r^t_{\text{step}}(\mathbf{W})\|\mathbf{x}(0) - \overline{\mathbf{x}}(0)\|_2. \tag{3.20}$$

This relation leads to an upper bound for another important characteristic of the speed of a distributed average consensus algorithm, namely *averaging time*.

Definition 3.4 (Averaging time)**.** *Averaging time is the smallest total time required to achieve the prescribed level of accuracy ε while performing the distributed averaging operation for the worst-case initialization:*

$$T_{ave}(\mathbf{W}, \varepsilon) \triangleq \sup_{\mathbf{x}(0) \neq \overline{\mathbf{x}}(0)} \inf_{t \geq 0} \{t : ||\mathbf{x}(t) - \overline{\mathbf{x}}(0)||_2 \leq \varepsilon ||\mathbf{x}(0) - \overline{\mathbf{x}}(0)||_2\}. \tag{3.21}$$

For a symmetric convergent weight matrix $\mathbf{W}$ (e.g. symmetric weight matrices are constructed by the MD and MH algorithms on the undirected graph $G(V, E)$, such that $(i, j) \in E \Leftrightarrow (j, i) \in E$) the spectral norm is also the spectral radius [71, 72] and thus using (3.20) and definition of the averaging time the following upper bound on T_{ave} can be

obtained [14]:

$$T_{\text{ave}}(\mathbf{W}, \varepsilon) \leq \frac{\log \varepsilon^{-1}}{\log r_{\text{step}}^{-1}(\mathbf{W})} = \frac{\log \varepsilon^{-1}}{\log \rho(\mathbf{W} - \mathbf{J})^{-1}} \tag{3.22}$$

3.3 Accelerated Distributed Average Consensus

Boyd et al. [14] have shown that for important network topologies — such as the two-dimensional grid or random geometric graph, which are commonly used to model connectivity in wireless networks — the memoryless distributed averaging (3.6) protocol can be prohibitively slow.

In the current literature two main approaches to accelerating the convergence of consensus algorithms can be identified: optimizing the weight matrix $\mathbf{W}$ [14, 68, 71, 73], and incorporating memory into the distributed averaging algorithm [69, 74–78].

3.3.1 Memoryless Weight Matrix Optimization

The spectral radius of the weight matrix governs the asymptotic convergence rate in memoryless distributed averaging algorithms, so optimizing the weight matrix corresponds to minimizing the spectral radius, subject to connectivity constraints [14, 68, 71]. In the case of symmetric weight matrix $\mathbf{W}$ minimizing the spectral radius is also equivalent to maximizing the step-wise convergence rate and minimizing the averaging time. Xiao et al. [71] formulate the following fastest distributed linear averaging optimization problem:

$$\begin{aligned} \text{minimize} \quad & r_{\text{step}}(\mathbf{W}) \\ \text{subject to} \quad & \mathbf{W} \in \mathcal{W}(V, E), \quad \mathbf{W}^{\mathsf{T}} = \mathbf{W}, \quad \mathbf{W}\mathbf{1} = \mathbf{1}. \end{aligned} \tag{3.23}$$

It turns out that this symmetric fastest distributed linear averaging problem can be cast as a semi-definite program with the linear matrix inequality constraint:

$$\begin{aligned} \text{minimize} \quad & s : s \geq 0 \\ \text{subject to} \quad & -s\mathbf{I} \preceq \mathbf{W} - \mathbf{J} \preceq s\mathbf{I} \\ & \mathbf{W} \in \mathcal{W}(V, E), \quad \mathbf{W}^{\mathsf{T}} = \mathbf{W}, \quad \mathbf{W}\mathbf{1} = \mathbf{1}, \end{aligned} \tag{3.24}$$

where $\preceq$ denotes inequality with respect to the cone of symmetric positive semidefinite matrices. The solution of this convex problem is the weight matrix with fastest asymptotic convergence rate. The semi-definite program above can be solved efficiently using existing numerical solvers. Furthermore, Boyd et al. [14] describe a subgradient decentralized algorithm solving this problem using decentralized orthogonal iterations (DOI) [79]. Although elegant, this approach involves substantial initialization costs: every iteration of the subgradient algorithm involves performing expensive DOI operation. Moreover, the improvement does not scale in grid or random geometric graph topologies [14] (the averaging time is improved by a constant factor independent of network size).

3.3.2 Memory Based Consensus Acceleration

A more promising research direction is based on using local node memory. The idea of using higher-order eigenvalue shaping filters was discussed in [77]. Distributed average consensus protocols of the following form were considered:

$$\mathbf{x}(t) = \sum_{k \geq 1} \mathbf{W}(t,k)\mathbf{x}(t-k). \tag{3.25}$$

Here $\mathbf{W}(t,k)$ is a proper $n \times n$ matrix kernel. In the first order memory based consensus protocol the kernel of the following form was proposed:

$$\mathbf{W}(t,k) = \mathbf{I} + \alpha\mathbf{A}, \tag{3.26}$$

where $\mathbf{A}$ is the connectivity matrix ($\mathbf{A}_{i,j} = 1$ if $(i,j) \in E$ and $\mathbf{A}_{i,j} = 0$ otherwise). The optimal value of α was determined:

$$\alpha = -\frac{2}{\lambda_2(\mathbf{A}) + \lambda_n(\mathbf{A})}. \tag{3.27}$$

For the second-order framework the following heuristic eigenvalue shaping filter was proposed:

$$H(z) = \frac{(1+c)z^{-1}}{1 + cz^{-2}}. \tag{3.28}$$

The eigenvalues of the second-order system resulting from (3.26) and (3.28) satisfy

$$\widetilde{\lambda} = \begin{cases} \sqrt{c} & \text{if } |\lambda| \leq \lambda_0 \\ \frac{(1+c)|\lambda| + \sqrt{\lambda^2(1+c)^2 - 4c}}{2} & \text{if } \lambda_0 \leq |\lambda| \leq 1, \end{cases} \tag{3.29}$$

where $\lambda_0 = 2\sqrt{c}/(1+c)$. The problem of finding the optimal c was left open, and convergence speed improvement was not analyzed.

In [76] Cao et al. proposed a memory-based acceleration framework for gossip algorithms where updates are a weighted sum of previous state values (memory registers) and gossip exchanges. However, Cao et al. [76] provide no solutions or directions for weight vector design or optimization.

Johansson and Johansson [75] advocate a similar scheme for distributed consensus averaging. The general update rule for the M-tap memory based accelerator described by Johansson and Johansson [75] is

$$\begin{cases} \mathbf{X}(t) = \mathbf{\Phi}\mathbf{X}(t-1) \\ \mathbf{y}(t) = \ [\mathbf{I} \quad \mathbf{1}^{\mathsf{T}} \otimes \mathbf{0}] \ \mathbf{X}(t) \end{cases} . \tag{3.30}$$

Here $\mathbf{X} \in \mathbb{R}^{nM \times 1}$ is the network-wide delay line and $\mathbf{X}(0) = \mathbf{1} \otimes \mathbf{z}$ for some initialization $\mathbf{z} \in \mathbb{R}^{n \times 1}$, $\mathbf{\Phi} \in \mathbb{R}^{nM \times nM}$ is the weight matrix of the memory based protocol, and $\mathbf{y} \in \mathbb{R}^{n \times 1}$ is the output of the distributed average consensus protocol. The Kronecker product $\otimes$ is defined as usual for two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} A_{1,1}\mathbf{B} & A_{1,2}\mathbf{B} & \dots & A_{1,n}\mathbf{B} \\ A_{2,1}\mathbf{B} & A_{2,2}\mathbf{B} & \dots & A_{2,n}\mathbf{B} \\ \dots & \dots & \dots & \dots \\ A_{m,1}\mathbf{B} & A_{m,2}\mathbf{B} & \dots & A_{m,n}\mathbf{B} \end{bmatrix} . \tag{3.31}$$

Matrix $\mathbf{\Phi}$ has the following block structure:

$$\mathbf{\Phi} = \begin{bmatrix} \beta_1\mathbf{W} & \beta_2\mathbf{I} & \dots & \beta_{M-1}\mathbf{I} & \beta_M\mathbf{I} \\ \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \dots & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{0} \end{bmatrix}. \tag{3.32}$$

Johansson and Johansson [75] investigate convergence conditions for this consensus protocol. For the asymptotic output $\mathbf{y}(t)$, $t \to \infty$ to reach the desired agreement function, $\mathcal{A}_{\text{am}}(\mathbf{z})\mathbf{1}$, the following conditions have to be satisfied:

$$\lim_{t\to\infty} \mathbf{\Phi}^t = \frac{1}{n}\mathbf{1}(\alpha_1\mathbf{1}^\mathsf{T} \dots \alpha_M\mathbf{1}^\mathsf{T}), \quad \sum_{i=1}^{M} \alpha_i = 1. \tag{3.33}$$

Indeed, if the previous condition is satisfied then we have

$$\lim_{t\to\infty} \mathbf{y}(t) = \lim_{t\to\infty} \mathbf{\Phi}^t(\mathbf{1}\otimes\mathbf{z}) = \frac{1}{n}\mathbf{1}\sum_{i=1}^{M}\alpha_i\mathbf{1}^\mathsf{T}\mathbf{z} = \mathcal{A}_{\text{am}}(\mathbf{z})\mathbf{1}. \tag{3.34}$$

The main result stating convergence conditions for the memory based accelerated distributed average consensus developed by Johansson and Johansson [75] can thus be summarized in the following theorem that is cited here without a proof.

Theorem 3.2 (Memory based average consensus convergence conditions, Johansson and Johansson [75] Theorem 1)**.** *The iteration* (3.30) *satisfies* (3.33) *if and only if* $\mathbf{\Phi}$ *and* $\boldsymbol{\alpha}$ *fulfill the following conditions.*

1. $\mathbf{\Phi}\mathbf{1} = \mathbf{1}$
2. $\mathbf{g}^T\mathbf{\Phi} = \mathbf{g}^T$, $\mathbf{g}^T = (\alpha_1\mathbf{1}^T \dots \alpha_M\mathbf{1}^T)$, $\sum_{i=1}^{M}\alpha_i = 1$
3. $\rho(\mathbf{\Phi} - \frac{1}{n}\mathbf{1}\mathbf{g}^T) < 1$

It can be shown [75] that the general matrix (3.32) satisfies the conditions specified in Theorem 3.2 if, for some β_i, such that $\sum_{i=1}^{M}\beta_i = 1$, the α_i satisfy the recursion $\alpha_i = \alpha_1\beta_i + \alpha_{i+1}$, where $\alpha_{M+1} = 0$. To optimize the performance of the general matrix Johansson

and Johansson [75] formulate the spectral radius optimization problem constrained by bilinear inequalities and use standard solvers to find a numerical solution for the optimal weight vector $\boldsymbol{\beta}$. The drawbacks of this approach are threefold. First, the formulated problem is non-convex. Second, the formulated problem is computationally demanding and it exhibits poor scalability with respect to both the network size and the memory size. Finally, Johansson and Johansson [75] have not presented a decentralized approach for initializing matrix $\boldsymbol{\Phi}$ (choosing the appropriate vector $\boldsymbol{\beta}$).

Recently, polynomial filtering was introduced for consensus acceleration by Kokiopoulou and Frossard [69]. In contrast to the approach of Johansson and Johansson [75], polynomial filtering consists in *periodically* updating the local state value with the weighted linear combination of the previous local states. The period is equal to the length of the polynomial filter. In between these memory-based updates, the state vector is updated with an ordinary consensus protocol. The order-M polynomial filtering based distributed average consensus protocol thus has the following form:

$$\begin{cases} \mathbf{x}(t) = \mathbf{W} \sum_{i=0}^{M} \alpha_i \mathbf{x}(t - M - 1 + i) & \text{if} \quad \text{mod } (t, M+1) = 0 \\ \mathbf{x}(t) = \mathbf{W}\mathbf{x}(t-1) & \text{otherwise} \end{cases} \tag{3.35}$$

It is obvious from the previous expression that at time instances when $\text{mod}(t, M+1) = 0$ holds true the performance of the polynomial filter based consensus is determined by the following accelerated matrix:

$$\boldsymbol{\Phi}[\boldsymbol{\alpha}, \mathbf{W}] = \sum_{i=0}^{M} \alpha_i \mathbf{W}^i. \tag{3.36}$$

Kokiopoulou and Frossard then formulate the following spectral radius optimization problem to identify optimal vector $\boldsymbol{\alpha}$ for a given $\mathbf{W}$:

$$\begin{aligned} \text{minimize} \quad & s : s \geq 0 \\ \text{subject to} \quad & -s\mathbf{I} \preceq \boldsymbol{\Phi}[\boldsymbol{\alpha}, \mathbf{W}] - \mathbf{J} \preceq s\mathbf{I} \\ & \boldsymbol{\Phi}[\boldsymbol{\alpha}, \mathbf{W}]\mathbf{1} = \mathbf{1}, \end{aligned} \tag{3.37}$$

The optimal weight vector can then be efficiently determined using numerical solvers since

Kokiopoulou and Frossard [69] showed that the problem is convex. Kokiopoulou and Frossard [69] also proposed a heuristic initialization of weight vector $\boldsymbol{\alpha}$ based on the Newton's interpolating polynomial. This initialization is suitable for accelerating distributed averaging consensus on dynamic topologies, however the performance boost obtained with this initialization is significantly smaller than that obtained with the optimal one.

The drawbacks of the polynomial filtering approach of Kokiopoulou and Frossard [69] are as follows. The weight vector optimization program is significantly more complex than that developed by Xiao et al. [71] for the fastest distributed linear averaging problem. The distributed solution for the weight optimization problem has not been developed. Although simulations definitely reveal that the polynomial filtering approach yields performance improvement, this performance improvement has not been analyzed and its scaling properties are not well understood. Finally, there is no proof that the heuristic initialization based on the Newton's interpolating polynomial always yields a convergent consensus protocol (or, alternatively, conditions under witch this protocol is convergent are not established).

An extreme approach to consensus acceleration is the methodology proposed by Sundaram and Hadjicostis in [78]. Based on the notion of observability in linear systems, the algorithm achieves consensus in a finite number of iterations. Each node records the entire history of values $\{x_i(t)\}_{t=0}^{T}$, and after enough iterations, inverts this history to recover the network average. In order to carry out the inversion, each node needs to know a topology-dependent set of weights. This leads to complicated initialization procedures for determining these weights. Another drawback is that the memory required at each node does not scale well: it grows with the network size.

In the next chapter the proposed methodology for the acceleration of distributed average consensus protocol will be presented. The proposed methodology avoids or alleviates the major drawbacks of existing approaches. It scales well with the size of the network, providing substantial gains even for very large networks. The local memory requirements are small and independent of the network size. We identify analytical expressions for the optimal parameter settings for the proposed method and derive bounds on the convergence rate improvement it provides. We also specify a fast algorithm for distributed initialization that has very low overhead in terms of both computational and communication requirements.

Chapter 4

Predictor Based Accelerated Distributed Average Consensus

This chapter presents the general predictor-based distributed average consensus acceleration framework, studies the convergence conditions for this framework, identifies the characterization for the averaging time for the general predictor-based accelerated algorithm and presents analytical optimization and convergence rate improvement analysis results for the important case of the accelerated consensus with short node memory. The proof of each presented result appears at the end of the corresponding section.

The main results of this chapter are as follows. For the general acceleration methodology we have the following contributions.

1. The memory based methodology for the acceleration of the distributed average consensus algorithm based on the mixture of predictor and the outcome of standard consensus iteration.

2. The theoretical proof of the existence of the convergent configurations of the proposed memory based acceleration methodology.

3. For the proposed memory based acceleration methodology, the upper bound on the growth rate of the limiting ε-averaging time for asymptotically small values of ε has been obtained.

For the memoryless weight matrix optimization based on the proposed methodology we have the following results.

1. The optimal value of the mixing parameter for the memoryless acceleration methodology.
2. The distributed suboptimal initializations of the mixing parameter.
3. The numerical experiments studying the convergence properties of the memoryless weight matrix optimization.

For the proposed methodology with short node memory we have the following results.

1. The study of the predictive accelerated average consensus with short node memory and the value of the optimal mixing parameter for the short memory case.
2. The quantification of the convergence rate of the predictive accelerated average consensus with short node memory.
3. The asymptotically optimal configuration of the predictor weights for the predictive accelerated average consensus with short node memory.
4. The quantification of the average asymptotic convergence rate improvement achieved by the accelerated average consensus with short node memory.
5. The practical distributed on-line mixing parameter initialization scheme.

4.1 Predictor Based Accelerated Average Consensus

4.1.1 Problem Formulation

We define a graph $G = (\mathcal{V}, \mathcal{E})$ as a 2-tuple, consisting of a set $\mathcal{V}$ with $|\mathcal{V}| = N$ vertices, where $|\cdot|$ denotes the cardinality, and a set $\mathcal{E}$ with $|\mathcal{E}|$ edges. We denote an edge between vertices i and j as an unordered pair $(i, j) \in \mathcal{E}$. The presence of an edge between two vertices indicates that they can establish bidirectional noise-free communication with each other. We assume that transmissions are always successful and that the topology is fixed. We assume also connected network topologies; the connectivity pattern of the graph is given by the $n \times n$ adjacency matrix $\mathbf{A} = [A_{ij}]$, where

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} . \tag{4.1}$$

Denote the neighborhood of node i by $\mathcal{N}_i = (j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$, and the degree of node i by $d_i \triangleq |\mathcal{N}_i|$.

We consider the set of nodes of a network (vertices of the corresponding graph), each with an initial real valued scalar $x_i(0)$, where $i = 1, 2, \ldots, n$. Let $\mathbf{1}$ denote the vector of ones with dimension defined by the context. The goal is to develop a distributed iterative algorithm that computes, at every node in the network, the value $\overline{\mathbf{x}} = (n)^{-1}\mathbf{1}^{\mathsf{T}}\mathbf{x}(0)$. In this chapter we focus on a particular class of iterative algorithms reducing to the following recursion

$$\mathbf{x}(t+1) = \mathbf{W}\mathbf{x}(t) \tag{4.2}$$

where $\mathbf{x}(t)$ denotes the state vector. The weight matrix, $\mathbf{W}$, needs to satisfy the following necessary and sufficient conditions to ensure asymptotic average consensus [80]:

$$\mathbf{W}\mathbf{1} = \mathbf{1}, \quad \mathbf{1}^{\mathsf{T}}\mathbf{W} = \mathbf{1}^{\mathsf{T}}, \quad \rho(\mathbf{W} - \mathbf{J}) < 1 \tag{4.3}$$

Algorithms have been identified for generating weight matrices that satisfy the required convergence conditions if the underlying graph is connected, e.g. Maximum-degree and Metropolis weights [68, 80].

In the next section, we describe our approach to accelerate the consensus algorithm. The approach is based on the observation that in the standard consensus procedure [71] the individual node state values converge in a smooth fashion. This suggests that it is possible to predict with good accuracy a future local node state based on past and current values. Combining such a prediction with the consensus operation thus has the potential to drive the overall system state closer to the true average at a faster rate than the standard consensus algorithm. Effectively, the procedure bypasses redundant states.

The next section describes the proposed acceleration methodology. The general form of the acceleration method is first discussed and two important results characterizing the convergence of the proposed methodology are presented. The primary parameter in the proposed algorithm is the mixing parameter, α, which determines how much weight is given to the predictor and how much to the consensus operator. For the general case, we derive sufficient conditions on this parameter to ensure convergence to the average and characterize the limiting averaging time necessary to compute the average within the prescribed accuracy level.

4.1.2 Acceleration Methodology

Computational resources available at the nodes are often scarce and it is desirable that the algorithms designed for distributed signal processing are computationally inexpensive. We are therefore motivated to use a linear predictor, thereby retaining the linear nature of the consensus algorithm.

In the proposed acceleration, we modify the state-update equations at a node to become a linear combination of the predictor and the value derived by application of the consensus weight matrix:

$$x_i(t) = \alpha x_i^{\mathrm{P}}(t) + (1-\alpha) x_i^{\mathrm{W}}(t) \tag{4.4a}$$

$$x_i^{\mathrm{W}}(t) = W_{ii} x_i(t-1) + \sum_{j \in \mathcal{N}_i} W_{ij} x_j(t-1) \tag{4.4b}$$

$$x_i^{\mathrm{P}}(t) = \theta_M x_i^{\mathrm{W}}(t) + \sum_{j=1}^{M-1} \theta_j x_i(t-M+j) \tag{4.4c}$$

Here $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_M]$ is the vector of predictor coefficients. The schematic diagram depicting the proposed predictor-based consensus is shown in Fig. 4.1.

The network-wide equations can be expressed in matrix form by defining

$$\mathbf{W}_M[\alpha] \triangleq (1-\alpha+\alpha\theta_M)\mathbf{W} + \alpha\theta_{M-1}\mathbf{I} \tag{4.5}$$

$$\mathbf{X}(t-1) \triangleq [\mathbf{x}(t-1)^{\mathsf{T}}, \mathbf{x}(t-2)^{\mathsf{T}}, \mathbf{x}(t-3)^{\mathsf{T}}, \ldots, \mathbf{x}(t-M+2)^{\mathsf{T}}, \mathbf{x}(t-M+1)^{\mathsf{T}}]^{\mathsf{T}} \tag{4.6}$$

where $\mathbf{I}$ is the identity matrix of the appropriate size and

$$\boldsymbol{\Phi}_M[\alpha] \triangleq \begin{bmatrix} \mathbf{W}_M[\alpha] & \alpha\theta_{M-2}\mathbf{I} & \alpha\theta_{M-3}\mathbf{I} & \ldots & \alpha\theta_2\mathbf{I} & \alpha\theta_1\mathbf{I} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{4.7}$$

Here all the components of the block matrix are $n \times n$. We adopt the convention that for

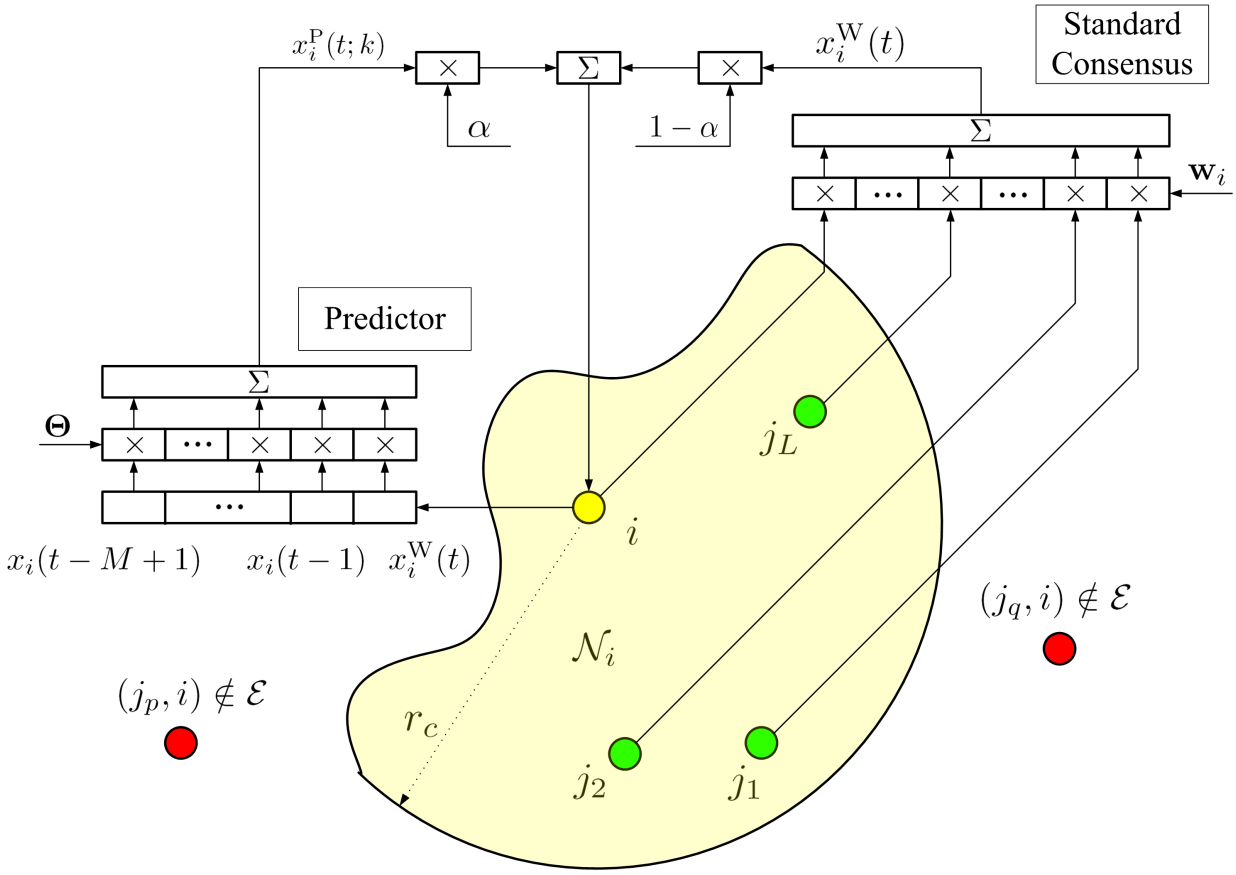


Fig. 4.1 The schematic diagram depicting the proposed predictor-based consensus

$t<0$, $\mathbf{x}(t)=\mathbf{x}(0)$. The update equation is then simply

$$\mathbf{X}(t)=\boldsymbol{\Phi}_M[\alpha]\mathbf{X}(t-1). \tag{4.8}$$

We adopt a time-invariant extrapolation procedure. The advantage of this approach is that the coefficients can be computed off-line as they do not depend on the data. We employ the best linear least-squares k-step predictor that extrapolates the current state $x_i(t)$ of the ith node k time steps forward. Choosing higher k implies a more aggressive prediction component to the algorithm. The prediction coefficients become (see the detailed derivation in Appendix A.1):

$$\boldsymbol{\theta}=\mathbf{B}^{\dagger\mathsf{T}}\mathbf{c} \tag{4.9}$$

where

$$\mathbf{B} \triangleq \begin{bmatrix} -M+1 & \dots & -1 & 0 \\ 1 & \dots & 1 & 1 \end{bmatrix}^{\mathsf{T}}, \tag{4.10}$$

$\mathbf{c} \triangleq [k, 1]^{\mathsf{T}}$ and $\mathbf{B}^{\dagger}$ is the Moore-Penrose pseudoinverse of $\mathbf{B}$. Appendix A.1 provides general expressions for the parameters $\boldsymbol{\theta}$.

It can be seen from (4.9) that $x_i^{\mathrm{P}}(t)$ is a linear combination of M previous local consensus values. Thus the consensus acceleration mechanism outlined in equations (4.4a–4.4c) is fully local if it is possible to find an optimum value of α in (4.4a) that does not require any global knowledge.

4.1.3 Convergence of Predictor-based Consensus

In this section we provide a result that characterizes a range of α values that achieve convergence to the consensus for arbitrary, finite, values of M. Let λ_i denote the ith ranked eigenvalue. The main result is the following theorem:

Theorem 4.1. *If* $\mathbf{W}$ *is symmetric, satisfies the conditions for asymptotic consensus (4.3),* $\sum_{i=1}^{M} \theta_i = 1$*, and*

$$0 \leq \alpha < \min\left(\frac{1-|\lambda_n|}{-|\lambda_n| + |\theta_M||\lambda_n| + \sum_{j=1}^{M-1} |\theta_j|}, \frac{1-|\lambda_2|}{-|\lambda_2| + |\theta_M||\lambda_2| + \sum_{j=1}^{M-1} |\theta_j|} \right) \tag{4.11}$$

then the general accelerated consensus algorithm achieves asymptotic convergence.

The first set of conditions in the theorem is satisfied by the choice of $\mathbf{W}$ and predictor weights we have outlined. The condition (4.11) specifies the bounds on the mixing parameter α. This is only a sufficient condition for convergence, but it does indicate that there is a range of values of α for every M that leads to asymptotic convergence. Significant improvements in the rate of convergence are generally achieved by α values outside the identified range due to the conservative nature of the proof.

While the previous result establishes the existence of a convergent solution for the proposed acceleration methodology, the next result quantifies the averaging time necessary to attain the prescribed accuracy level. As defined in Section 3.2.3, the ε-averaging time is the time required to achieve the prescribed level of accuracy ε while performing the

distributed averaging operation:

$$T_{\text{ave}}(\mathbf{W},\varepsilon) \triangleq \sup_{\mathbf{X}(0)\neq\mathbf{0}} \inf_{t\geq 0} \left\{ t : ||\mathbf{W}^t\mathbf{X}(0) - \mathbf{J}\mathbf{X}(0)||_2 \leq \varepsilon ||\mathbf{X}(0) - \mathbf{J}\mathbf{X}(0)||_2 \right\}, \tag{4.12}$$

In the case where $\mathbf{W}$ is symmetric, $\rho(\mathbf{W}-\mathbf{J})$ also defines an upper bound on the averaging time (see Section 3.2.3). The update matrix we propose, (4.7), is not symmetric and it may not even be contracting. The results of [71] do not apply for such matrices, and the spectral radius $\rho(\mathbf{W}-\mathbf{J})$ cannot, in general, be used to specify an upper bound on averaging time. In fact, since $\mathbf{\Phi}_M[\alpha]$ is not symmetric, $\mathbf{\Phi}_M[\alpha]^t$ does not even converge to $\mathbf{J}$ as $t\to\infty$, as in the memoryless setting. We can, however, establish a result for the *limiting* ε-averaging time, which is the averaging time for asymptotically small ε.

Before stating our first result we must introduce some notation. For now, assume we are given a matrix $\mathbf{\Phi}\in\mathbb{R}^{n\times n}$ with $\overline{\mathbf{\Phi}} = \lim_{t\to\infty}\mathbf{\Phi}^t$. We will address conditions for existence of the limit below. For a given initialization vector $\mathbf{X}(0)\in\mathbb{R}^n$, let $\widetilde{\mathbf{X}}(0) = \overline{\mathbf{\Phi}}\mathbf{X}(0)$, and define the set of non-trivial initialization vectors $\mathcal{X}_{0,\mathbf{\Phi}} \triangleq \{\mathbf{X}(0)\in\mathbb{R}^n : \mathbf{X}(0)\neq\widetilde{\mathbf{X}}(0)\}$. Since we have not yet established that $\widetilde{\mathbf{X}}(0) = \overline{\mathbf{X}}(0) \triangleq \mathbf{J}\mathbf{X}(0)$, we keep the discussion general and use the following definition of the averaging time:

$$T_{\text{ave}}(\mathbf{\Phi},\varepsilon) \triangleq \sup_{\mathbf{X}(0)\in\mathcal{X}_{0,\mathbf{\Phi}}} \inf_{t\geq 0} \left\{ t : ||\mathbf{X}(t) - \widetilde{\mathbf{X}}(0)||_2 \leq \varepsilon ||\mathbf{X}(0) - \widetilde{\mathbf{X}}(0)||_2 \right\}. \tag{4.13}$$

We now prove a result relating the spectral radius and the ε-averaging time for general non-symmetric averaging matrices $\mathbf{\Phi}$, which we will then apply to our particular construction, $\mathbf{\Phi}_M[\alpha]$.

Theorem 4.2. *Let $\mathbf{\Phi}\in\mathbb{R}^{n\times n}$ be given, with limit $\lim_{t\to\infty}\mathbf{\Phi}^t = \overline{\mathbf{\Phi}}$, and assume that $\rho(\mathbf{\Phi}-\overline{\mathbf{\Phi}})>0$. Then we have for any $\varepsilon\in(0,1]$:*

$$\lim_{\varepsilon\to 0}\frac{T_{ave}(\mathbf{\Phi},\varepsilon)}{\log\varepsilon^{-1}} < \frac{1}{\log\rho(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{-1}}. \tag{4.14}$$

According to this result, the averaging time required to approach the average within ε-accuracy grows at the rate at most $1/\log\rho(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{-1}$ as $\varepsilon\to 0$ for operators $\mathbf{\Phi}$. Minimizing the spectral radius is thus a natural optimality criterion for such operators. In order to apply the above result, we must establish that $\mathbf{\Phi}_M[\alpha]$ satisfies the conditions of Theorem 4.2. In doing so, we will also show that for $\mathbf{\Phi} = \mathbf{\Phi}_M[\alpha]$, the limit $\overline{\mathbf{\Phi}}\mathbf{X}(0) = \mathbf{J}\mathbf{X}(0)$, so our approach

indeed converges to the average consensus.

We will demonstrate the applicability of Theorem 4.2 under the assumption $\sum_{i=1}^{M}\theta_i = 1$. To demonstrate the applicability of Theorem 4.2 to the matrix $\mathbf{\Phi}_M[\alpha]$ we can use Theorem 3.2 (a result due to Johanson and Johanson [75], Theorem 1). According to Theorem 3.2, the necessary and sufficient conditions for the consensus algorithm of the form $\mathbf{\Phi}_M[\alpha]$ to converge to the average are (JJ1) $\mathbf{\Phi}_M[\alpha]\mathbf{1} = \mathbf{1}$; (JJ2) $\mathbf{g}^\mathsf{T}\mathbf{\Phi}_M[\alpha] = \mathbf{g}^\mathsf{T}$ for some vector $\mathbf{g}^\mathsf{T} = [\beta_1\mathbf{1}^\mathsf{T}, \ldots, \beta_{M-1}\mathbf{1}^\mathsf{T}]$ with weights satisfying $\sum_{i=1}^{M-1}\beta_i = 1$; and (JJ3) $\rho(\mathbf{\Phi}_M[\alpha] - \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T}) < 1$. If these conditions hold then we also have $\overline{\mathbf{\Phi}}_M[\alpha] = \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T}$ [75] implying $\widetilde{\mathbf{X}}(0) = \overline{\mathbf{X}}(0)$.

Condition (JJ1) is easily verified after straightforward algebraic manipulations using the definition of $\mathbf{\Phi}_M[\alpha]$ in (4.7), the assumption that $\sum_{i=1}^{M}\theta_i = 1$, and recalling that $\mathbf{W}$ satisfies $\mathbf{W1} = \mathbf{1}$ by design.

To address condition (JJ2), we can write the linear system induced by this condition, $\mathbf{g}^\mathsf{T}\mathbf{\Phi}_M[\alpha] = \mathbf{g}^\mathsf{T}$, and the requirement $\sum_{i=1}^{M-1}\beta_i = 1$:

$$\begin{aligned}
\beta_1(1 + \alpha(\theta_M - 1)) + \beta_2 &= \beta_1 \\
\alpha\theta_{M-2}\beta_1 + \beta_3 &= \beta_2 \\
\alpha\theta_{M-3}\beta_1 + \beta_4 &= \beta_3 \\
&\cdots \\
\alpha\theta_2\beta_1 + \beta_{M-1} &= \beta_{M-2} \\
\alpha\theta_1\beta_1 &= \beta_{M-1} \\
\beta_1 + \beta_2 + \ldots + \beta_{M-1} &= 1
\end{aligned} \tag{4.15}$$

Rearranging the above system we obtain

$$\begin{aligned}
\beta_{M-1} &= \alpha\theta_1\beta_1 \\
\beta_{M-2} &= \alpha(\theta_1 + \theta_2)\beta_1 \\
\beta_{M-3} &= \alpha(\theta_1 + \theta_2 + \theta_3)\beta_1 \\
&\cdots \\
\beta_1 + \beta_2 + \ldots + \beta_{M-1} &= 1
\end{aligned} \tag{4.16}$$

Thus setting β_i according to the solution of the above system,

$$\beta_i = \begin{cases} \frac{1}{1+\alpha\sum_{j=1}^{M-2}(M-1-j)\theta_j} & \text{if} \quad i=1 \\ \frac{\alpha\sum_{j=1}^{M-i}\theta_j}{1+\alpha\sum_{j=1}^{M-2}(M-1-j)\theta_j} & \text{if} \quad i\geq 2 \end{cases}, \tag{4.17}$$

shows that (JJ2) can be satisfied by choosing the values of β_i as above. It is also easy to verify condition (JJ2) by plugging these values into the definition of $\mathbf{g}$, and using the same properties of $\mathbf{\Phi}_M[\alpha]$, the θ_i's, and $\mathbf{W}$ as previously.

Condition (JJ3) holds for convergent matrices $\mathbf{\Phi}_M[\alpha]$. Theorem 4.1 establishes the existence of non-trivial convergent configurations of the proposed matrix, implying $\rho(\mathbf{\Phi}_M[\alpha]-\overline{\mathbf{\Phi}})<1$ (stronger results are provided for the matrix $\mathbf{\Phi}_3[\alpha]$ studied in Section 4.3). Thus we conclude that for the properly configured matrix $\mathbf{\Phi}_M[\alpha]$ Theorem 3.2 holds implying $\widetilde{\mathbf{X}}(0)=\overline{\mathbf{\Phi}}\mathbf{X}(0)=\mathbf{J}\mathbf{X}(0)$ and Theorem 4.2 holds establishing the limiting scaling law (4.14) for the ε-averaging time of the memory based consensus acceleration methodology.

4.1.4 Proofs

Proof of Theorem 4.1

We commence by introducing an operator $\mathbf{V}$:

$$\mathbf{V}=\frac{\mathbf{W}_M[\alpha]}{1-\sum_{i=1}^{M-2}\alpha\theta_i}=\frac{(1-\alpha+\alpha\theta_M)\mathbf{W}+\alpha\theta_{M-1}\mathbf{I}}{1-\sum_{i=1}^{M-2}\alpha\theta_i} \tag{4.18}$$

Denoting $c_i=\alpha\theta_i$, we can write the first component of network-wide state recursion $\mathbf{X}(t)=\mathbf{\Phi}_M[\alpha]\mathbf{X}(t-1)$ as:

$$\mathbf{x}(t)=(1-c_1-c_2-\ldots-c_{M-2})\mathbf{V}\mathbf{x}(t-1)+c_{M-2}\mathbf{x}(t-2)+\ldots+c_1\mathbf{x}(t-M+1) \tag{4.19}$$

where we set $\mathbf{x}(t)=\mathbf{x}(0)$ for any $t<0$. Let us denote $S=(1-c_1-c_2-\ldots-c_{M-2})||\mathbf{V}-\mathbf{J}||$ and $\beta=S+|c_1|+|c_2|+\ldots+|c_{M-2}|$. Here, as before, $\mathbf{J}$ denotes the averaging operator. The following lemma provides the platform for the proof of the theorem, identifying sufficient conditions on α that guarantee $|\beta|<1$.

Lemma 4.1. *If* $\mathbf{W}$ *is symmetric, satisfies the conditions for asymptotic consensus (4.3),*

$\sum_{i=1}^{M} \theta_i = 1$, *and*

$$0 \leq \alpha < \min\left(\frac{1-|\lambda_n|}{-|\lambda_n| + |\theta_M||\lambda_n| + \sum_{j=1}^{M-1}|\theta_j|}, \frac{1-|\lambda_2|}{-|\lambda_2| + |\theta_M||\lambda_2| + \sum_{j=1}^{M-1}|\theta_j|}\right) \tag{4.20}$$

then $|\beta| < 1$.

Proof. Using the triangle inequality and the definitions of S and $\mathbf{V}$, we can formulate a bound on $|\beta|$:

$$|\beta| = \left|(1 - c_1 - c_2 - \ldots - c_{M-2})||\mathbf{V} - \mathbf{J}||_2 + \alpha \sum_{j=1}^{M-2}|\theta_j|\right| \tag{4.21}$$

$$\leq ||(1 - \alpha + \alpha\theta_M)\mathbf{W} + \alpha\theta_{M-1}\mathbf{I}||_2 + \alpha \sum_{j=1}^{M-2}|\theta_j|. \tag{4.22}$$

The last inequality is true since by our assumption $\mathbf{W}$ is symmetric and thus $||\mathbf{V} - \mathbf{J}||_2 = \rho(\mathbf{V} - \mathbf{J})$. We have for the first eigenvector of $\mathbf{W}$, $\mathbf{1}$:

$$\begin{aligned}(\mathbf{V} - \mathbf{J})\mathbf{1} &= \frac{(1 - \alpha + \alpha\theta_M)\mathbf{W} + \alpha\theta_{M-1}\mathbf{I} - (1 - \alpha\sum_{i=1}^{M-2}\theta_i)\mathbf{J}}{1 - \alpha\sum_{i=1}^{M-2}\theta_i}\mathbf{1} \\ &= \frac{(1 - \alpha + \alpha\theta_M) + \alpha\theta_{M-1} - (1 - \alpha\sum_{i=1}^{M-2}\theta_i)}{1 - \alpha\sum_{i=1}^{M-2}\theta_i} \\ &= \mathbf{0}.\end{aligned}$$

On the other hand, for any other eigenvector of $\mathbf{W}$, $\mathbf{v} \perp \mathbf{1}$, and its respective eigenvalue λ we have

$$\begin{aligned}(\mathbf{V} - \mathbf{J})\mathbf{v} &= \frac{(1 - \alpha + \alpha\theta_M)\mathbf{W} + \alpha\theta_{M-1}\mathbf{I} - (1 - \alpha\sum_{i=1}^{M-2}\theta_i)\mathbf{J}}{1 - \alpha\sum_{i=1}^{M-2}\theta_i}\mathbf{v} \\ &= \frac{\left[(1 - \alpha + \alpha\theta_M)\lambda + \alpha\theta_{M-1} - 0\right]\mathbf{v}}{1 - \alpha\sum_{i=1}^{M-2}\theta_i},\end{aligned}$$

showing that the transition in (4.22) is valid. Thus if we ensure that (4.22) is true, we guarantee that $|\beta| < 1$. We can reformulate this inequality using the symmetry of $\mathbf{W}$, $\mathbf{I}$, $\mathbf{J}$

and the definition of the spectral radius of a matrix:

$$|(1-\alpha+\alpha\theta_M)\lambda_i+\alpha\theta_{M-1}|+\alpha\sum_{j=1}^{M-2}|\theta_j|<1,\quad \forall|\lambda_i|<1 \tag{4.23}$$

Again applying the triangle inequality, we see that this relationship is satisfied if:

$$|1-\alpha||\lambda_i|+\alpha\left(|\theta_M||\lambda_i|+\sum_{j=1}^{M-1}|\theta_j|\right)<1,\quad \forall|\lambda_i|<1 \tag{4.24}$$

Upon expansion of the modulus $|1-\alpha|$, and with algebraic manipulation, we arrive at:

$$0<\alpha<\mathcal{J}(\lambda_i)\triangleq\frac{1-|\lambda_i|}{-|\lambda_i|+|\theta_M||\lambda_i|+\sum_{j=1}^{M-1}|\theta_j|},\quad \forall|\lambda_i|<1 \tag{4.25}$$

Now, let us examine the properties of the upper bound $\mathcal{J}(\lambda_i)$ in (4.25). After some algebraic manipulations the derivative of $\mathcal{J}(\lambda_i)$ for the two cases $\lambda_i>0$ and $\lambda_i<0$ takes the following form:

$$\begin{aligned}\frac{\partial}{\partial\lambda_i}\mathcal{J}(\lambda_i:\lambda_i>0)&=\frac{1-\sum_{j=1}^{M}|\theta_j|}{\left(-\lambda_i+|\theta_M|\lambda_i+\sum_{j=1}^{M-1}|\theta_j|\right)^2}\\ \frac{\partial}{\partial\lambda_i}\mathcal{J}(\lambda_i:\lambda_i<0)&=\frac{\sum_{j=1}^{M}|\theta_j|-1}{\left(\lambda_i-|\theta_M|\lambda_i+\sum_{j=1}^{M-1}|\theta_j|\right)^2}\end{aligned} \tag{4.26}$$

Taking into account the fact that $\sum_{j=1}^{M}|\theta_j|\geq 1$ we can make the following conclusion:

$$\begin{aligned}\frac{\partial}{\partial\lambda_i}\mathcal{J}(\lambda_i:\lambda_i>0)&\leq 0,\quad \forall\lambda_i\\ \frac{\partial}{\partial\lambda_i}\mathcal{J}(\lambda_i:\lambda_i<0)&\geq 0,\quad \forall\lambda_i\end{aligned} \tag{4.27}$$

Thus $\mathcal{J}(\lambda_i)$ is nondecreasing when $\lambda_i<0$ and non-increasing when $\lambda_i>0$. Hence if for any λ_j, λ_k, and λ_i satisfying $|\lambda_j|>|\lambda_i|$ and $|\lambda_k|>|\lambda_i|$ there exists an α^* such that

$$0<\alpha^*<\min\left(\mathcal{J}(\lambda_j),\mathcal{J}(\lambda_k)\right) \tag{4.28}$$

then $0 < \alpha^* < \mathcal{J}(\lambda_i)$ and (4.11) follows. To ensure that such α^* always exists for any $|\lambda_j| < 1$ and $|\lambda_k| < 1$ we note that $\mathcal{J}(\lambda_j) > 0, \ \forall |\lambda_j| < 1$. This follows because $1 - |\lambda_j| > 0$. Moreover,

$$-|\lambda_i| + |\theta_M||\lambda_i| + \sum_{j=1}^{M-1} |\theta_j| \geq -|\lambda_i| + |\lambda_i| \sum_{j=1}^{M} |\theta_j| \tag{4.29}$$

$$\geq -|\lambda_i| + |\lambda_i| = 0. \tag{4.30}$$

□

We now present the proof of Theorem 4.1.

Proof of Theorem 4.1. We first show that if the conditions of the theorem hold, then the average is preserved at each time step. To do this, it is necessary and sufficient to show that $\left(\left(1 - \sum_{i=1}^{M-2} c_i\right)\mathbf{V} + \sum_{i=1}^{M-2} c_i \mathbf{I}\right)\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^\mathsf{T}\left(\left(1 - \sum_{i=1}^{M-2} c_i\right)\mathbf{V} + \sum_{i=1}^{M-2} c_i \mathbf{I}\right) = \mathbf{1}^\mathsf{T}$. We have:

$$\left(\left(1 - \sum_{i=1}^{M-2} c_i\right)\mathbf{V} + \sum_{i=1}^{M-2} c_i \mathbf{I}\right)\mathbf{1} = (1 - \alpha + \alpha\theta_M)\mathbf{W}\mathbf{1} + \alpha\theta_{M-1}\mathbf{I}\mathbf{1} + \alpha \sum_{i=1}^{M-2} \theta_i \mathbf{1} \tag{4.31}$$

$$= \left(1 - \alpha + \alpha \sum_{i=1}^{M} \theta_i\right)\mathbf{1} = \mathbf{1} \tag{4.32}$$

The proof of the condition $\mathbf{1}^\mathsf{T}\left(\left(1 - \sum_{i=1}^{M-2} c_i\right)\mathbf{V} + \sum_{i=1}^{M-2} c_i \mathbf{I}\right) = \mathbf{1}^\mathsf{T}$ is analogous and omitted.

We now show that $\mathbf{x}(t)$ converges to the average $\mathbf{J}\mathbf{x}(0)$. Our method of proof is induction. We show that $||\mathbf{x}(t) - \mathbf{J}\mathbf{x}(0)||_2 \leq \beta^{\ell+1}||\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)||_2$ where $\ell = \lfloor (t-1)/(M-1) \rfloor$. Lemma 1 implies that if the assumptions of the theorem are satisfied then $|\beta| < 1$, so the limit as t and consequently ℓ approaches infinity is 0. Initially, we show that the result holds for $\ell = 0$, or equivalently, $t = 1 \ldots M-1$. We have, using the triangle inequality and

employing the fact that $(\mathbf{V}-\mathbf{J})\mathbf{J}=\mathbf{0}$:

$$\begin{aligned}
||\mathbf{x}(1)-\mathbf{J}\mathbf{x}(0)||_2 &= \left|\left|\left(1-\sum_{i=1}^{M-2}c_i\right)(\mathbf{V}\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0))+\sum_{i=1}^{M-2}c_i(\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0))\right|\right|_2 \\
&= \left|\left|\left(1-\sum_{i=1}^{M-2}c_i\right)(\mathbf{V}-\mathbf{J})(\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0))+\sum_{i=1}^{M-2}c_i(\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0))\right|\right|_2 \\
&\leq \left(\left(1-\sum_{i=1}^{M-2}c_i\right)||\mathbf{V}-\mathbf{J}||_2+\sum_{i=1}^{M-2}|c_i|\right)||\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0)||_2 \\
&= \beta||\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0)||_2
\end{aligned}$$

Similarly:

$$||\mathbf{x}(2)-\mathbf{J}\mathbf{x}(0)||_2 = \left|\left|\left(1-\sum_{i=1}^{M-2}c_i\right)(\mathbf{V}\mathbf{x}(1)-\mathbf{J}\mathbf{x}(0))+\sum_{i=1}^{M-2}c_i(\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0))\right|\right|_2 \tag{4.33}$$

$$\leq \left(1-\sum_{i=1}^{M-2}c_i\right)||\mathbf{V}-\mathbf{J}||_2||\mathbf{x}(1)-\mathbf{J}\mathbf{x}(0)||_2+\sum_{i=1}^{M-2}|c_i||\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0)||_2 \tag{4.34}$$

$$\leq ||\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0)||_2\left(\beta\left(1-\sum_{i=1}^{M-2}c_i\right)||\mathbf{V}-\mathbf{J}||_2+\sum_{i=1}^{M-2}|c_i|\right) \tag{4.35}$$

$$\leq \beta||\mathbf{x}(0)-\mathbf{J}\mathbf{x}(0)||_2, \tag{4.36}$$

where to obtain the last inequality we use the fact that $|\beta|<1$. Using the same observations

we can show the following for any t such that $2 < t < M$:

$$||\mathbf{x}(t) - \mathbf{J}\mathbf{x}(0)||_2 = \Bigg|\Bigg| \left(1 - \sum_{i=1}^{M-2} c_i\right) (\mathbf{V}\mathbf{x}(t-1) - \mathbf{J}\mathbf{x}(0)) \tag{4.37}$$

$$+ \sum_{i=2}^{t-1} c_{M-i}(\mathbf{x}(t-i) - \mathbf{J}\mathbf{x}(0)) + \sum_{i=t}^{M-1} c_{M-i}(\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)) \Bigg|\Bigg| \tag{4.38}$$

$$\leq \left(1 - \sum_{i=1}^{M-2} c_i\right) ||\mathbf{V} - \mathbf{J}||_2 ||\mathbf{x}(t-1) - \mathbf{J}\mathbf{x}(0)||_2 \tag{4.39}$$

$$+ \sum_{i=2}^{t-1} |c_{M-i}| ||\mathbf{x}(t-i) - \mathbf{J}\mathbf{x}(0)||_2 + \sum_{i=t}^{M-1} |c_{M-i}| ||\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)||_2 \tag{4.40}$$

$$\leq ||\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)||_2 \left(\beta \left(1 - \sum_{i=1}^{M-2} c_i\right) ||\mathbf{V} - \mathbf{J}||_2 + \beta \sum_{i=2}^{t-1} |c_{M-i}| + \sum_{i=t}^{M-1} |c_{M-i}|\right) \tag{4.41}$$

$$\leq ||\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)||_2 \left(\left(1 - \sum_{i=1}^{M-2} c_i\right) ||\mathbf{V} - \mathbf{J}||_2 + \sum_{i=1}^{M-2} |c_i|\right) \tag{4.42}$$

$$\leq \beta ||\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)||_2, \tag{4.43}$$

By almost identical manipulations, we can show that if the result holds for $\ell - 1$ and $t = (\ell-1)(M-1)+1, \ldots, (\ell-1)(M-1)+M-1$, then it holds for ℓ and $t = \ell(M-1)+1, \ldots, \ell(M-1)+M-1$. □

Proof of Theorem 4.2

The following definition will be used below:

$$||\mathbf{\Phi}||_{\mathbf{X}(0)} \triangleq \frac{||\mathbf{\Phi}(\mathbf{X}(0) - \widetilde{\mathbf{X}}(0))||_2}{||\mathbf{X}(0) - \widetilde{\mathbf{X}}(0)||_2}. \tag{4.44}$$

The limit $\lim_{t\to\infty} \mathbf{\Phi}^t = \overline{\mathbf{\Phi}}$ exists if and only if (see [81]) $\mathbf{\Phi}$ can be expressed in the form

$$\mathbf{\Phi} = \mathbf{T} \begin{bmatrix} \mathbf{I}_\kappa & \mathbf{0} \\ \mathbf{0} & \mathbf{Z} \end{bmatrix} \mathbf{T}^{-1} \tag{4.45}$$

where $\mathbf{I}_\kappa$ is the identity matrix of dimension κ, $\mathbf{Z}$ is a matrix with $\rho(\mathbf{Z}) < 1$ and $\mathbf{T}$ is an invertible matrix. It follows that in the limit we have [75],

$$\overline{\mathbf{\Phi}} = \lim_{t\to\infty} \mathbf{\Phi}^t = \mathbf{T} \begin{bmatrix} \mathbf{I}_\kappa & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{T}^{-1}. \tag{4.46}$$

By linear algebra, $\mathbf{\Phi}\overline{\mathbf{\Phi}} = \overline{\mathbf{\Phi}}\mathbf{\Phi} = \overline{\mathbf{\Phi}}$ and $\mathbf{\Phi}^t\overline{\mathbf{\Phi}} = \overline{\mathbf{\Phi}}$. Using these facts it is straightforward to show that $(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t = \mathbf{\Phi}^t - \overline{\mathbf{\Phi}}$, implying that $(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t(\mathbf{X}(0) - \widetilde{\mathbf{X}}(0)) = \mathbf{X}(t) - \widetilde{\mathbf{X}}(0)$. Taking the norm of both sides we have

$$||\mathbf{X}(t) - \widetilde{\mathbf{X}}(0)||_2 = ||(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t(\mathbf{X}(0) - \widetilde{\mathbf{X}}(0))||_2 = ||(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t||_{\mathbf{X}(0)}||(\mathbf{X}(0) - \widetilde{\mathbf{X}}(0))||_2, \tag{4.47}$$

and therefore $T_{\text{ave}}(\mathbf{\Phi}, \varepsilon) = \sup\limits_{\mathbf{X}(0)\in\mathcal{X}_{0,\mathbf{\Phi}}} \inf\limits_{t\in\mathbb{N}} \left\{t : ||(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t||_{\mathbf{X}(0)} \le \varepsilon\right\}$. Given any $\mathbf{X}(0) \in \mathcal{X}_{0,\mathbf{\Phi}}$, it follows that $\|(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t\|_{\mathbf{X}(0)} \le \sup_{\mathbf{X}\in\mathcal{X}_{0,\mathbf{\Phi}}} \|(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t\|_{\mathbf{X}}$. Moreover,

$$\sup_{\mathbf{X}\in\mathcal{X}_{0,\mathbf{\Phi}}} \|(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t\|_{\mathbf{X}} \le \varepsilon \tag{4.48}$$

implies that

$$\|(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t\|_{\mathbf{X}(0)} \le \varepsilon. \tag{4.49}$$

Let $\mathcal{C}(\mathbf{\Phi}, \varepsilon)$ denote the set of t for which (4.48) holds, and let $\mathcal{B}(\mathbf{X}(0), \mathbf{\Phi}, \varepsilon)$ denote the set of t for which (4.49) holds. Since (4.48) implies (4.49), $\mathcal{C}(\mathbf{\Phi}, \varepsilon) \subseteq \mathcal{B}(\mathbf{X}(0), \mathbf{\Phi}, \varepsilon)$, and consequently, $\inf \mathcal{B}(\mathbf{X}(0), \mathbf{\Phi}, \varepsilon) \le \inf \mathcal{C}(\mathbf{\Phi}, \varepsilon)$, from which it follows that $T_{ave}(\mathbf{\Phi}, \varepsilon) \le t^*(\varepsilon)$, with

$$t^*(\varepsilon) \triangleq \inf \mathcal{C}(\mathbf{\Phi}, \varepsilon) = \inf \left\{ t : \sup_{\mathbf{X}\in\mathcal{X}_{0,\mathbf{\Phi}}} \|(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t\|_{\mathbf{X}} \le \varepsilon \right\}. \tag{4.50}$$

By the definition of $t^*(\varepsilon)$ in (4.50):

$$\left[\sup_{\mathbf{X}(0)\in\mathcal{X}_{0,\mathbf{\Phi}}} ||(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^{t^*(\varepsilon)}||_{\mathbf{X}(0)}^{1/t^*(\varepsilon)} \right]^{t^*(\varepsilon)} \le \varepsilon, \tag{4.51}$$

and so, noting that by the definition of the induced norm, $\sup\limits_{\mathbf{X}(0)\in\mathcal{X}_{0,\mathbf{\Phi}}} ||(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^t||_{\mathbf{X}(0)} =$

$||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^t||_2$, after taking the logarithm on both sides of (4.51), we have[1]

$$t^*(\varepsilon) \geq \frac{\log(\varepsilon)}{\log ||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{t^*(\varepsilon)}||_2^{1/t^*(\varepsilon)}}. \tag{4.52}$$

Since [72] $\rho(\mathbf{\Phi}-\overline{\mathbf{\Phi}}) \leq ||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{t^*(\varepsilon)}||_2^{1/t^*(\varepsilon)}$ for any $t^*(\varepsilon)$, it follows that $t^*(\varepsilon) \geq \log(\varepsilon)/\log\rho(\mathbf{\Phi}-\overline{\mathbf{\Phi}})$, from which it is clear that $t^*(\varepsilon) \to \infty$ as $\varepsilon \to 0$.

Now, by the definition of $t^*(\varepsilon)$ in (4.50) we also have $||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{t^*(\varepsilon)-1}||_2 > \varepsilon$, implying

$$(t^*(\varepsilon)-1)\log ||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{t^*(\varepsilon)-1}||_2^{1/(t^*(\varepsilon)-1)} > \log(\varepsilon), \tag{4.53}$$

and thus

$$T_{\text{ave}}(\mathbf{\Phi},\varepsilon) \leq t^*(\varepsilon) < \frac{\log(\varepsilon)}{\log ||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{t^*(\varepsilon)-1}||_2^{1/(t^*(\varepsilon)-1)}} + 1. \tag{4.54}$$

Dividing through by $|\log(\varepsilon)|$, taking the limit as $\varepsilon \to 0$, and moving the limit on the right under the log, we obtain

$$\begin{aligned}\lim_{\varepsilon\to 0}\frac{T_{\text{ave}}(\mathbf{\Phi},\varepsilon)}{|\log(\varepsilon)|} &< \lim_{\varepsilon\to 0}\frac{-1}{\log ||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{t^*(\varepsilon)-1}||_2^{1/(t^*(\varepsilon)-1)}} + \lim_{\varepsilon\to 0}\frac{1}{|\log(\varepsilon)|} \\ &< \frac{-1}{\log\lim_{\varepsilon\to 0} ||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{t^*(\varepsilon)-1}||_2^{1/(t^*(\varepsilon)-1)}}.\end{aligned} \tag{4.55}$$

Since $t^*(\varepsilon) \to \infty$ as $\varepsilon \to 0$ (see above), we may employ Gelfand's formula [72], $\lim_{t\to\infty} ||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^t||^{1/t} = \rho(\mathbf{\Phi}-\overline{\mathbf{\Phi}})$, to complete the proof:

$$\lim_{\varepsilon\to 0}\frac{T_{\text{ave}}(\mathbf{\Phi},\varepsilon)}{|\log(\varepsilon)|} < \frac{1}{\log\rho(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{-1}}. \tag{4.56}$$

[1] Since we are interested in asymptotic behavior of the type $\varepsilon \to 0$, there is no loss of generality in supposing that ε is sufficiently small so that the following holds: $\log(\varepsilon) < 0$, $\log \sup_{\mathbf{X}(0)\in\mathcal{X}_{0,\mathbf{\Phi}}} ||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{t^*(\varepsilon)}||_{\mathbf{X}(0)}^{1/t^*(\varepsilon)} < 0$, and $\log \sup_{\mathbf{X}(0)\in\mathcal{X}_{0,\mathbf{\Phi}}} ||(\mathbf{\Phi}-\overline{\mathbf{\Phi}})^{t^*(\varepsilon)-1}||_{\mathbf{X}(0)}^{1/(t^*(\varepsilon)-1)} < 0$

4.2 Memoryless Distributed Matrix Optimization

In this section we analyze the case when the algorithm (4.4) is based only on the current node states. For this case, we derive the mixing parameter that leads to the optimal improvement of worst-case asymptotic convergence rate. Evaluating the optimal value requires knowledge of the second-largest and smallest eigenvalues, which can be difficult to determine. We therefore derive a bound on the optimal α value which requires less information; setting α to this bound results in close-to-optimal performance.

The predictor under consideration is a one-step extrapolator based on the current node state and the result of the standard consensus operator, i.e., $k = 1$ and $M = 2$. According to the expression for the predictor weights provided in the previous section, $\boldsymbol{\theta} = [-1, 2]^{\mathsf{T}}$, so $x_i^{\mathrm{P}}(t)$ can be expressed as follows:

$$x_i^{\mathrm{P}}(t) = 2x_i^{\mathrm{W}}(t) - x_i(t-1). \tag{4.57}$$

We can estimate the gradient of the state with respect to time as $\widehat{\nabla} x_i(t) \triangleq x_i^{\mathrm{W}}(t) - x_i(t-1)$. Thus (4.57) can be rewritten as:

$$x_i^{\mathrm{P}}(t) = x_i^{\mathrm{W}}(t) + \widehat{\nabla} x_i(t). \tag{4.58}$$

The one-step predictor hence updates the current state in the gradient direction, to within estimation error.

Substituting (4.57) into (4.4a) we obtain the following expression for $x_i(t)$:

$$\begin{aligned} x_i(t) &= \alpha(2x_i^{\mathrm{W}}(t) - x_i(t-1)) + (1-\alpha)x_i^{\mathrm{W}}(t) && (4.59)\\ &= x_i(t-1)((1+\alpha)W_{ii} - \alpha) + (1+\alpha)\sum_{j\in\mathcal{N}_i} W_{ij}x_j(t-1). && (4.60) \end{aligned}$$

This can be written in matrix form as:

$$\mathbf{x}(t) = \boldsymbol{\Phi}_2[\alpha]\mathbf{x}(t-1) \tag{4.61}$$

where $\boldsymbol{\Phi}_2[\alpha]$ is the weight matrix (as a function of α):

$$\boldsymbol{\Phi}_2[\alpha] = (1+\alpha)\mathbf{W} - \alpha\mathbf{I} \tag{4.62}$$

It is obvious from the previous equation that the predictor based weight matrix $\mathbf{\Phi}_2[\alpha]$ has the same eigenvectors as $\mathbf{W}$ and its eigenvalues are related to the eigenvalues of the original matrix $\mathbf{W}$ via the relationship:

$$\lambda_i[\alpha] = (1+\alpha)\lambda_i - \alpha \qquad i = 1, \ldots, n \tag{4.63}$$

The following proposition describes some properties of the weight matrix $\mathbf{\Phi}_2[\alpha]$. We show that if the original weight matrix $\mathbf{W}$ satisfies the conditions necessary for asymptotical convergence, then $\mathbf{\Phi}_2[\alpha]$ also guarantees asymptotical convergence to consensus under some mild conditions.

Proposition 4.1. *Suppose* $\mathbf{W}$ *satisfies the necessary conditions for the convergence of the standard consensus algorithm. Moreover, let* $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ *denote the eigenvalues associated with eigenvectors* $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ *and let* $\lambda_i[\alpha]$ *denote the ranked eigenvalues of* $\mathbf{\Phi}_2[\alpha]$*. Then* $\mathbf{\Phi}_2[\alpha]$ *satisfies the required convergence conditions if*

$$0 \leq \alpha < \frac{1+\lambda_n}{1-\lambda_n}. \tag{4.64}$$

The proof of Proposition 4.1 implies that the eigenvalues of the predictor based weight matrix $\mathbf{\Phi}_2[\alpha]$ experience a left shift with respect to the eigenvalues of the original weight matrix $\mathbf{W}$ when $\alpha > 0$. Moreover, it is easy to show that the ordering of the eigenvalues does not change during the shift:

$$\lambda_i \leq \lambda_j \Rightarrow \lambda_i[\alpha] \leq \lambda_j[\alpha] \tag{4.65}$$

for all $i, j, \alpha \geq 0$, where i and j are associated with some eigenvectors $\mathbf{u}_i$, $\mathbf{u}_j$ of $\mathbf{W}$. The second largest and the smallest eigenvalues of matrix $\mathbf{\Phi}_2[\alpha]$ always correspond to the second largest and the smallest eigenvalues of matrix $\mathbf{W}$, and their values are always smaller. Using this property, together with the definition of spectral radius, it is possible to formulate the problem of optimizing the mixing parameter to achieve the fastest asymptotic worst-case convergence rate as a convex optimization problem. In the following subsection, we outline this formulation and provide the closed-form solution.

4.2.1 Optimization of The Mixing Parameter

Recall that α is the mixing parameter that determines the relative influences of the standard consensus iteration and the predictor in (4.4a). In the following, we consider optimization of α to achieve the fastest possible worst-case asymptotic convergence rate for the $M = 2$, $k = 1$ case of the accelerated consensus algorithm. The optimal value of the mixing parameter is provided in the following theorem.

Theorem 4.3. *The $M = 2, k = 1$ case of the proposed accelerated consensus algorithm has the fastest asymptotic worst-case convergence rate if the value of the mixing parameter α equals the following optimum value:*

$$\alpha^* = \frac{\lambda_n + \lambda_2}{2 - \lambda_n - \lambda_2} \tag{4.66}$$

where λ_i denotes the eigenvalues of the weight matrix $\mathbf{W}$.

As expected, the optimal mixing parameter α^* satisfies the following:

$$\alpha^* < \frac{1 + \lambda_n}{1 - \lambda_2 + 1 - \lambda_n} \tag{4.67}$$

$$< \frac{1 + \lambda_n}{1 - \lambda_n} \tag{4.68}$$

where both the first and second lines follow from the fact that $0 \leq \lambda_2 < 1$, respectively. We can conclude that the optimal mixing parameter satisfies the required convergence conditions for all cases.

Remark 4.1. *Algebraic manipulations lead to the following equality:*

$$|\lambda_n[\alpha^*]| = \frac{\lambda_2 - \lambda_n}{2 - \lambda_2 - \lambda_n} = \lambda_2[\alpha^*]. \tag{4.69}$$

The optimal mixing parameter thus induces a shift in the eigenvalues so that the magnitudes of the second-largest and smallest eigenvalues of $\mathbf{\Phi}_2[\alpha]$ are balanced. A similar effect is observed in the optimization conducted in [71]. It should be noted however, that even with the optimal choice of α the proposed algorithm for $M = 2$ case cannot outperform the global optimization proposed in [71].

4.2.2 Convergence Rate Analysis

To see to what extent the proposed one-step extrapolation algorithm yields performance improvement over the conventional consensus procedure, we consider the ratio of the spectral radius of the corresponding matrices. This ratio gives the lower bound on performance improvement:

$$\gamma[\alpha] \triangleq \frac{\rho(\mathbf{W}-\mathbf{J})}{\rho(\mathbf{\Phi}_2[\alpha]-\mathbf{J})} = \frac{\lambda_2}{\max\{\lambda_2[\alpha], |\lambda_n[\alpha]|\}} \tag{4.70}$$

The following proposition considers the provided convergence rate improvement over the standard consensus algorithm when the optimal mixing parameter is utilized.

Proposition 4.2. *In the optimal case,* i.e., *when $\alpha = \alpha^*$, the performance improvement factor is given by*

$$\gamma[\alpha^*] = \frac{\lambda_2(2-\lambda_2-\lambda_n)}{\lambda_2-\lambda_n}. \tag{4.71}$$

Proof. Substituting α^* into (4.70) and taking into account the fact that $|\lambda_n[\alpha^*]| = \lambda_2[\alpha^*]$, after some algebraic manipulations, yield the expression for $\gamma[\alpha^*]$. □

Proposition 4.2 provides the expression for the asymptotic convergence rate resulting from the matrix optimization based on the proposed methodology. Although this expression indicates that the gain can be significant, there is no guarantee that any gain is achieved. For example, if $\lambda_2 = -\lambda_n$ we have $\gamma[\alpha^*] = 1$ indicating the absence of gain. In Section 4.3 we analyze a much more general instance of the predictor based accelerated average consensus. The guaranteed and significant performance gain is proved for this more general setting.

4.2.3 Suboptimal Choice of Mixing Parameter

Although (4.66) provides an expression for the optimum mixing factor resulting in the fastest asymptotic convergence rate, the calculation of this optimum value requires knowledge of the second and the last eigenvalues of matrix $\mathbf{W}$. This in turn either requires knowledge of $\mathbf{W}$ or some centralized mechanism for calculation and distribution of the eigenvalues of $\mathbf{W}$. In many practical situations such information may not be available. Therefore it is of interest to derive a suboptimum setting for α that results in less performance gain but requires considerably less information at each node.

Proposition 4.3. *The memoryless (M=2) predictor based distributed average consensus has asymptotic worst-case convergence rate faster than that of conventional consensus if the value of mixing parameter is in the following range:*

$$0 < \alpha \leq \alpha^*. \tag{4.72}$$

Proof. The asymptotic worst-case convergence rate of algorithm (4.4) is faster than that of conventional consensus algorithm if and only if $\gamma[\alpha] > 1 \Rightarrow \rho(\mathbf{\Phi}_2[\alpha] - \mathbf{J}) < \rho(\mathbf{W} - \mathbf{J})$. We can rewrite this condition in the following form:

$$\begin{cases} \dfrac{\lambda_2(1+\alpha) - \alpha}{\lambda_2} < 1 \\ \dfrac{\alpha(1-\lambda_n) - \lambda_n}{\lambda_2} < 1 \end{cases} \tag{4.73}$$

indicating that

$$\begin{cases} \alpha(\lambda_2 - 1) < 0 \\ \alpha < \dfrac{\lambda_2 + \lambda_n}{1 - \lambda_n} \end{cases}. \tag{4.74}$$

Observing that $\lambda_2 - 1 < 0$, dividing the first part of (4.74) by $\lambda_2 - 1$ and subtracting the same expression from the denominator of the second part we obtain the tightened version of (4.74):

$$0 < \alpha < \frac{\lambda_n + \lambda_2}{2 - \lambda_n - \lambda_2} \tag{4.75}$$

Finally, noting that the right hand side of this expression is equal to α^* concludes the proof. □

We strive to identify a setting for α that guarantees an improvement in the convergence rate but does not require global knowledge of the weight matrix. Based on Proposition 4.3, if we lower-bound α^*, then setting α to this lower-bound will guarantee improvement in convergence rate. In order to lower-bound α^*, we need to lower-bound $\lambda_2 + \lambda_n$. The next proposition provides such a bound in terms of the trace of the weight matrix $\mathbf{W}$.

Proposition 4.4. *If the weight matrix* $\mathbf{W}$ *satisfies the convergence conditions and its eigenspectrum is a convex function of the eigenvalue index, namely,*

$$\lambda_i \leq \frac{n-i}{n-2}\lambda_2 + \frac{i-2}{n-2}\lambda_n, \quad 2 \leq i \leq n \tag{4.76}$$

then

$$\lambda_2 + \lambda_n \geq \xi \triangleq \frac{2(\mathrm{tr}(\mathbf{W}) - 1)}{n-1} \tag{4.77}$$

where $\mathrm{tr}(\cdot)$ *denotes the trace of its argument.*

Proof. Recall that the sum of eigenvalues of a matrix is equal to its trace:

$$\sum_{i=1}^{n} \lambda_i = \mathrm{tr}(\mathbf{W}) \tag{4.78}$$

Noting that $\lambda_1 = 1$ and rearranging the summation give

$$\sum_{i=2}^{n} \lambda_i = \mathrm{tr}(\mathbf{W}) - 1. \tag{4.79}$$

Since, by assumption, the eigenspectrum is a convex function of the eigenvalue index, we have:

$$\frac{(\lambda_2 + \lambda_n)(n-1)}{2} \geq \sum_{i=2}^{n} \lambda_i \tag{4.80}$$

Substituting (4.79) into (4.80) results in the desired bound. □

Proposition 4.4 leads to an upper bound for a setting of the mixing parameter α in order to achieve convergence at an improved rate:

$$\alpha \leq \frac{\xi}{2 - \xi} \triangleq \Lambda(\xi). \tag{4.81}$$

The advantage of this setting is that it is much simpler to calculate the trace $\mathrm{tr}(\mathbf{W})$ in a distributed fashion than derive the eigenvalues λ_2 and λ_n, as required for determining the optimum mixing parameter. The lower bound depends linearly on the average of the diagonal terms of the matrix $\mathbf{W}$, which can be calculated using a standard consensus procedure. Although the result is useful and leads to a simpler mechanism for setting the mixing parameter, the convexity assumption is strong.

Under the assumption of the positivity of the eigenvalues of $\mathbf{W}$ another bound can be obtained.

Proposition 4.5. *If the weight matrix* $\mathbf{W}$ *satisfies the convergence conditions and its*

eigenvalues satisfy $\lambda_i \geq 0, \forall i \geq 1$, *then we have*

$$\lambda_2 + \lambda_n \geq \frac{\mathrm{tr}(\mathbf{W}) - 1}{n-1}. \tag{4.82}$$

Proof. The proof follows the lines of the proof of Proposition 4.4 and uses the fact that under the current assumption $\lambda_2 + \lambda_n \geq \lambda_i, \forall i \geq 2$ and thus $(\lambda_2 + \lambda_n)(n-1) \geq \sum_{i=2}^{n} \lambda_i$. □

4.2.4 Random Geometric Graphs: Choice of the Mixing Parameter

We now consider the special, but important, case of random geometric graphs, which can act as good topological models of wireless sensor networks, one of the promising application domains for consensus algorithms. For this case, we show that there exists an asymptotic upper bound $\Lambda(\xi_\infty)$ for α that can be calculated off-line. The random geometric graph is defined as follows: n nodes (vertices) are distributed in an area $\mathcal{D}$ according to a point process with known spatial distribution $p_{x,y}(x, y)$. Two nodes i and j are connected, i.e. $A_{ij} = 1$, if the Euclidean distance $r_{i,j}$ between them is less then some predefined connectivity radius r_c. The indicator function $I\{r_{i,j}^2 \leq r_c^2\} = 1$ whenever $r_{i,j}^2 \leq r_c^2$ holds.

We consider weight matrices $\mathbf{W}$ constructed according to a rule of the following form:

$$\begin{cases} W_{ij} = I\{r_{i,j}^2 \leq r_c^2\}\mathcal{L}(d_i, d_j), & i \neq j \\ W_{ij} = 1 - \sum_{j=1, j\neq i}^{N} W_{ij}, & i = j \end{cases} \tag{4.83}$$

where $\mathcal{L}(d_i, d_j)$ is some function of the local connectivity degrees d_i and d_j of nodes i and j satisfying:

$$\begin{aligned} \sum_{j=1}^{n} I\{r_{i,j}^2 \leq r_c^2\}\mathcal{L}(d_i, d_j) &= 1 \\ |\mathcal{L}(d_i, d_j)| &< 1 \end{aligned} \tag{4.84}$$

Let us introduce random variables $\zeta_{i,n}$ defined by:

$$\zeta_{i,n} = \sum_{j=1, j\neq i}^{n} I\{r_{i,j}^2 \leq r_c^2\}\mathcal{L}(d_i, d_j)\ . \tag{4.85}$$

Assume that $\mathcal{L}(d_i, d_j)$ is chosen so that these variables are identically distributed with mean

$\mathbb{E}\{\zeta_{i,n}\} = \mathbb{E}\{\zeta\}$ and covariance structure satisfying

$$\sum_{n\in\mathbb{N}^+} \frac{\sigma_n}{n}\sqrt{R_{n-1}} + \frac{\sigma_n^2}{n^2} < \infty \tag{4.86}$$

where R_n and σ_n are defined as follows:

$$R_n \triangleq \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}\{(\zeta_{i,n} - \mathbb{E}\{\zeta_{i,n}\})(\zeta_{j,n} - \mathbb{E}\{\zeta_{j,n}\})\} \tag{4.87}$$

$$\sigma_n^2 \triangleq \mathbb{E}\{(\zeta_{i,n} - \mathbb{E}\{\zeta_{i,n}\})^2\} \tag{4.88}$$

For such a graph and weight matrix, the following theorem provides an asymptotic upper bound on the value of mixing parameter α in terms of the expectation $\mathbb{E}\{\zeta\}$.

Theorem 4.4. *Let* $\mathbf{W}$ *be the* $n \times n$ *weight matrix constructed according to (4.83). Suppose* $\mathcal{L}(d_i, d_j)$ *is chosen so that the random variables* $\zeta_{i,n}$ *defined by (4.85) are identically distributed with finite mean* $\mathbb{E}\{\zeta\}$ *and covariance structure satisfying (4.86). Then the lower bound on* $\lambda_2 + \lambda_n$ *given by Proposition 4.4 almost surely converges to*

$$\xi_\infty \triangleq \lim_{n\to\infty}\xi = 2(1 - \mathbb{E}\{\zeta\}) \quad a.s. \tag{4.89}$$

and defines an asymptotic upper bound on α *as* $n \to \infty$ *given by the following expression:*

$$\alpha \leq \Lambda(\xi_\infty) = \frac{\xi_\infty}{2 - \xi_\infty} \quad a.s. \tag{4.90}$$

The above result relies on the assumption that $\mathcal{L}(d_i, d_j)$ satisfies the conditions discussed above. The following proposition states that this assumption holds for the popular max-degree weight design scheme [68]. The max-degree weights are very simple to compute and are well suited for distributed implementation. In order to determine the weights, the nodes need no information beyond their number of neighbors.

Proposition 4.6. *If the weights in the weight matrix* $\mathbf{W}$ *are determined using the max-degree weight approach, then assumptions of Theorem 4.4 hold and the asymptotic bound* ξ_∞ *on* $\lambda_2 + \lambda_n$ *satisfies:*

$$\xi_\infty^{MD} = 2(1 - p) \quad a.s. \tag{4.91}$$

and

$$\Lambda(\xi_{\infty}^{MD}) = \frac{1-p}{p}. \tag{4.92}$$

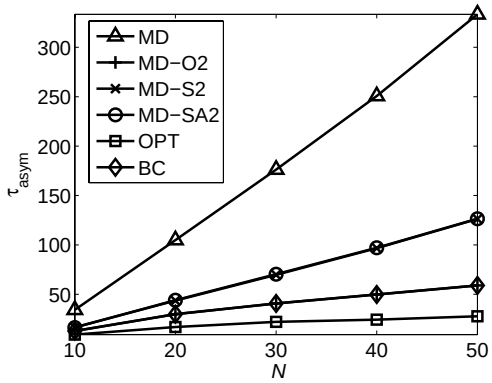
where p is the probability that two arbitrary nodes in a network are connected.

We note that p can be analytically derived for a given connectivity radius if the spatial distribution of the nodes is uniform (see Appendix A.2). Proposition 4.6 implies that for a random geometric graph with max-degree weights α should be chosen to satisfy $\alpha \leq (1-p)/p$. This result indicates that for highly connected graphs, which have a large value of p, a small α is desirable. For these graphs, standard consensus achieves fast mixing, so the prediction becomes less important and should be assigned less weight. In the case of a sparsely connected graph (small p), a large α is desirable. For these graphs, the convergence of standard consensus is slow because there are few connections, so the prediction component of the accelerated algorithm should receive more weight.

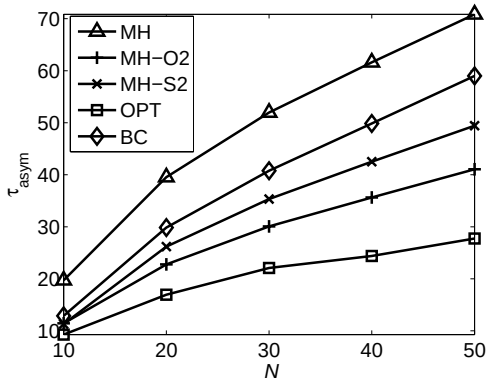
4.2.5 Numerical Examples

In our simulation experiments, we consider a set of n nodes uniformly distributed on the unit square. The nodes establish bidirectional links to each other if the Euclidean distance between them is smaller than the connectivity radius, $\sqrt{\log n/n}$. Initial node measurements are generated as $\mathbf{x} = \theta + \mathbf{n}$ where $\theta = 1$ and $\mathbf{n}$ is Gaussian distributed with $\sigma = 1$. Then, we regularize the data such that the average of all the values, $\overline{\mathbf{x}}$, equals to 1. All simulation results are generated based on 500 trials (a different random graph is generated for each trial).

First, we compare the asymptotic convergence time (3.12) results of the algorithm we propose for the theoretically analyzed $M = 2$ and $k = 1$ case, against the algorithms presented in [71]. Fig. 4.2 compares the convergence times of the algorithms for the $M = 2$ and $k = 1$ case as a function of the number of nodes in the network. In Fig. 4.2(a), the maximum-degree weight matrix is used as the consensus operator for the standard and accelerated consensus algorithms; in Fig. 4.2(b), the MH weight matrix acts as the consensus operator. It is clear from Fig. 4.2 that although our algorithm is extremely simple and does not require any global optimization, it achieves performance improvements approaching those of the optimum algorithm from [71]. It outperforms the best constant algorithm when used in conjunction with the MH weight matrix. When MD is utilized in the proposed algorithm, its asymptotic convergence time is very similar to that of the



(a) The convergence times for algorithms based on maximum-degree weight matrices as a function of the number of nodes in the network. Following algorithms were simulated. Standard Consensus (MD): △; Accelerated consensus with optimal α (MD-O2): +; Accelerated consensus with suboptimal α (MD-S2): ×; Accelerated consensus with asymptotic suboptimal α (MD-SA2): ∘; Best Constant [71] (BC): ♢; and Optimal Weight Matrix [71] (OPT): □



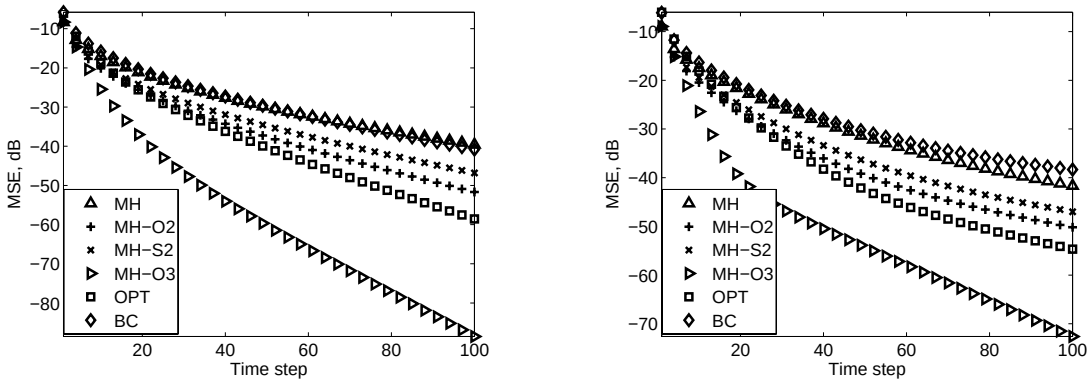
(b) The convergence times for algorithms based on MH weight matrices as a function of the number of nodes in the network. The following algorithms were simulated: Standard Consensus (MH): △; Accelerated consensus with optimal α (MH-O2): +; Accelerated consensus with suboptimal α (MH-S2): ×; Best Constant [71] (BC): ♢; and Optimal Weight Matrix [71] (OPT): □

Fig. 4.2 The asymptotic convergence time versus the number of nodes in the network. In (a), the standard and accelerated consensus algorithms are derived from the maximum-degree weight matrix; in (b) they are derived from the MH weight matrix.

optimized best constant approach from [71] for the optimal choice of α. Fig. 4.2(a) also suggests that the asymptotic upper bound on α derived for a random geometric graph with maximum-degree weight matrix is applicable when n is as low as 20. The two curves corresponding to the bound based on the trace of weight matrix (4.81) (represented by ×) and the asymptotic upper bound developed in Proposition 4.6, (4.90) (represented by ∘) are almost indistinguishable.

Since the performance of all algorithms is superior when the MH weight matrix is employed, the remainder of our simulations focus on this case. Fig. 4.3 shows the mean squared error (MSE) as a function of time for the standard, accelerated, best-constant and optimal weight matrix consensus algorithms. Three versions of the accelerated algorithm are depicted, including the $M = 2, k = 1$ case with optimal and suboptimal α, and the $M = 3, k = 1$ case. The number of nodes is 25 in Fig. 4.3(a) and 50 in Fig. 4.3(b).

The formal methodology for choosing α in the setting $M = 3$ will be developed in



(a) MSE as a function of time step when the number of nodes $n = 25$

(b) MSE as a function of time step when the number of nodes $n = 50$

Fig. 4.3 Mean-squared-error (MSE) versus time step for the proposed and standard consensus algorithm. The left panel depicts the results when the number of nodes in the network $n = 25$, and the right panel depicts the results when $n = 50$. The following algorithms were simulated. Standard consensus (MH): $\triangle$; Accelerated consensus, $M = 2$ with optimal α (MH-O2): $+$; Accelerated consensus $M = 2$ with suboptimal α (MH-S2): $\times$; Accelerated consensus $M = 3$ (MH-O3): $\triangleright$; Best Constant (BC): $\Diamond$; and optimal weight matrix (OPT): $\square$

Section 4.3. For the comparative purposes we present the following results based on the empirical selection of α: for each trial, we evaluate the MSE for each value of α ranging from 0 to 1 at intervals of 0.1 and chose the α that results in the lowest MSE at time step 50. We have observed that the random generation of the data has very little influence on the α value that is selected; it is the random graph and hence the initial $\mathbf{W}$ matrix that governs the optimal value of α. This suggests that it is possible to develop a data-independent procedure to choose an optimal α value. Fig. 4.3 indicates that the accelerated consensus algorithm with $M = 2$ and $k = 1$ achieves step-wise MSE decay that is close to that obtained using the optimal weight matrix developed in [71]. The accelerated algorithm with $M = 3$ and $k = 1$ significantly outperforms the optimal weight matrix [71] in terms of step-wise MSE decay. The $M = 3$ case permits much more accurate prediction, which leads to the significant improvement in performance.

In the following Section 4.3 we analyze the case $M = 3$: we derive the analytical expression for the asymptotically worst-case optimal α and analyze the improvement in the performance that can be obtained using the proposed methodology. It turns out that

in the more interesting case $M = 3$, we can obtain a significant gain that scales well with the size of the network and we can derive theoretical guarantees of performance improvement.

4.2.6 Proofs

Proof of Proposition 4.1

In order to ensure asymptotical convergence, we need to prove the following properties:

$$\mathbf{\Phi}_2[\alpha]\mathbf{1} = \mathbf{1}, \quad \mathbf{1}^\mathsf{T}\mathbf{\Phi}_2[\alpha] = \mathbf{1}^\mathsf{T}, \quad \rho(\mathbf{\Phi}_2[\alpha] - \mathbf{J}) < 1 \tag{4.93}$$

It is clear from (4.62) that $\mathbf{\Phi}_2[\alpha]$ has the same eigenvectors as $\mathbf{W}$. Eigenvalues of $\mathbf{\Phi}_2[\alpha]$ are connected to the eigenvalues via (4.63) and we conclude that $\lambda_1[\alpha] = 1$. Thus the two leftmost equations in (4.93) hold if $\mathbf{W}$ satisfies asymptotic convergence conditions. Now, let us consider the spectral radius of $\mathbf{\Phi}_2[\alpha] - \mathbf{J}$ defined as $\rho(\mathbf{\Phi}_2[\alpha] - \mathbf{J})$:

$$\rho(\mathbf{\Phi}_2[\alpha] - \mathbf{J}) \triangleq \max\{\lambda_2[\alpha], |\lambda_n[\alpha]|\}. \tag{4.94}$$

For $\alpha \geq 0$, the eigenvalues experience a left shift since $\lambda_i[\alpha] = \alpha(\lambda_i - 1) + \lambda_i$ and $(\lambda_i - 1)$ is always negative. It is also straightforward to see that $\lambda_i < \lambda_j \Rightarrow \lambda_i[\alpha] < \lambda_j[\alpha]$, $\forall i, j$. This implies that $\lambda_n[\alpha] < \lambda_2[\alpha] < 1$, so to ensure that $\rho(\mathbf{W} - \mathbf{J}) < 1$, we just need to make sure that $\lambda_n[\alpha] = \alpha(\lambda_n - 1) - \alpha > -1$. Rearrangement leads to $\alpha < \frac{1+\lambda_n}{1-\lambda_n}$, the condition expressed in (4.64). $\square$

Proof of Theorem 4.3

We need to show that $\alpha = \alpha^*$ is the global minimizer of $\rho(\mathbf{\Phi}_2[\alpha] - \mathbf{J})$. Hence we define the following optimization problem:

$$\alpha^* = \arg\min_{\alpha} \rho(\mathbf{\Phi}_2[\alpha] - \mathbf{J}) = \arg\min_{\alpha} \max_{i \neq 1}(|\lambda_{(i)}[\alpha]|). \tag{4.95}$$

However, this problem can be converted into a simpler one:

$$\alpha^* = \arg\min_{\alpha} \max(|\lambda_n - \alpha(1 - \lambda_n)|, |\lambda_2 - \alpha(1 - \lambda_2)|) \tag{4.96}$$

since $\lambda_n[\alpha]$ is the smallest and $\lambda_2[\alpha]$ is the largest eigenvalue of $(\mathbf{\Phi}_2[\alpha]-\mathbf{J})$. Let us introduce $f(\alpha) = \lambda_n - \alpha(1-\lambda_n)$ and $g(\alpha) = \lambda_2 - \alpha(1-\lambda_2)$. Clearly $|f(\alpha)|$ and $|g(\alpha)|$ are piecewise linear convex functions, with knots occurring where $f(\alpha) = 0$ and $g(\alpha) = 0$. Let these knots be $\alpha_f = \lambda_n/(1-\lambda_n)$ and $\alpha_g = \lambda_2/(1-\lambda_2)$. Since $\lambda_n < \lambda_2$ the magnitude of slope of $f(\alpha)$ exceeds that of $g(\alpha)$ and $\alpha_f < \alpha_g$. Consider $h(\alpha) = \max(|f(\alpha)|, |g(\alpha)|)$, which is also piecewise linear and convex with knots α^- and α^+ occurring where $f(\alpha^-) = g(\alpha^-)$ and $f(\alpha^+) = -g(\alpha^+)$ respectively. Since $h(\alpha)$ is piecewise linear and convex with $h(\infty) = \infty$ and $h(-\infty) = \infty$ its global minimum occurs at one of the knots. It follows that the knots of $h(\alpha)$ satisfy $\alpha^- < \alpha_f < \alpha^+ < \alpha_g$. The fact that $|g(\alpha)|$ is decreasing if $\alpha < \alpha_g$ implies $g(\alpha^+) < g(\alpha^-)$. Hence the global minimum of $h(\alpha)$ occurs at $\alpha = \alpha^+$. Thus solving $f(\alpha^+) = -g(\alpha^+)$ for α^+ gives the solution for $\alpha^* = \alpha^+$ $\square$

Proof of Theorem 4.4

By the construction of the weight matrix $\mathbf{W}$ (4.83) we can transform the expression for ξ (4.77) as follows:

$$\begin{aligned}
\xi &= \frac{2}{n-1}(\mathrm{tr}(\mathbf{W}) - 1) \\
&= \frac{2n}{n-1}\frac{1}{n}\sum_{i=1}^{n}\left(1 - \sum_{j=1, j\neq i}^{n} W_{ij}\right) - \frac{2}{n-1} \\
&= \frac{2(n-1)}{n-1} - \frac{2n}{n-1}\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} I\{r_{i,j}^2 \leq r_c^2\}\mathcal{L}(d_i, d_j) \\
&= 2 - \frac{2}{n-1}\sum_{i=1}^{n}\zeta_{i,n}. \qquad (4.97)
\end{aligned}$$

In order to obtain the last equality we have used the definition (4.85). Note that $I\{r_{i,j}^2 \leq r_c^2\}$ is a Bernoulli random variable; denote the probability $\mathbb{P}\{I\{r_{i,j}^2 \leq r_c^2\} = 1\}$ by p. Note that an analytical expression for p can be derived if the nodes are uniformly distributed (see Appendix A.2); for other distributions, numerical integration can be employed to determine p.

We require that $\mathcal{L}(\cdot)$ is such that the $\zeta_{i,n}$ are identically distributed with finite mean and (4.86) holds. It is straightforward to show that both mean and variance of random

variables $\zeta_{i,n}$ are bounded under our assumptions (4.84) on $\mathcal{L}(d_i, d_j)$:

$$\begin{aligned} |\mathbb{E}\{\zeta_{i,n}\}| &= \left| \mathbb{E}\left\{ \sum_{j=1, j\neq i}^{n} I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \right\} \right| \\ &\leq \sup \left| \sum_{j=1, j\neq i}^{n} I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \right| = 2. \end{aligned} \tag{4.98}$$

The transition involves moving the expectation outside the modulus, replacing it by a supremum, and then application of the bounds in (4.84). Moreover,

$$\begin{aligned} \sigma_n^2 &= \mathbb{E}\{(\zeta_{i,n} - \mathbb{E}\{\zeta_{i,n}\})^2\} \\ &\leq \mathbb{E}\{\zeta_{i,n}^2\} \\ &= \mathbb{E}\left\{ \sum_{j=1, j\neq i}^{n} I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \sum_{k=1, k\neq i}^{n} I\{r_{i,k}^2 \leq r_c^2\} \mathcal{L}(d_i, d_k) \right\} \\ &\leq \sup \left| \sum_{j=1, j\neq i}^{n} I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \right| \sup \left| \sum_{k=1, k\neq i}^{n} I\{r_{i,k}^2 \leq r_c^2\} \mathcal{L}(d_i, d_k) \right| = 4. \end{aligned} \tag{4.99}$$

Taking into account (4.98) and (4.99), we can consider the following centered square integrable random process:

$$\chi_{i,n} = \zeta_{i,n} - \mathbb{E}\{\zeta_{i,n}\}, i = 1 \ldots n \tag{4.100}$$

We note that if the correlation function of this random process satisfies ergodicity assumptions implied by (4.86), we can invoke the Strong Law of Large Numbers stated by Poznyak [82] (Theorem 1) to show that

$$\frac{1}{n} \sum_{i=1}^{n} \chi_{i,n} \longrightarrow 0. \quad \text{a.s.} \tag{4.101}$$

In turn, this along with the assumption $\mathbb{E}\{\zeta_{i,n}\} = \mathbb{E}\{\zeta\}$ implies that according to (4.100)

$$\frac{1}{n-1} \sum_{i=1}^{n} \zeta_{i,n} \longrightarrow \mathbb{E}\{\zeta\}. \quad \text{a.s.} \tag{4.102}$$

Combining (4.102) with (4.97) leads us to the following conclusion:

$$\begin{aligned}\xi_\infty &= \lim_{n\to\infty}\left\{2 - \frac{2}{n-1}\sum_{i=1}^{n}\zeta_{i,n}\right\} \\ &= 2(1-\mathbb{E}\{\zeta\}) \quad \text{a.s.}\end{aligned} \tag{4.103}$$

Finally, noting (4.81) concludes the proof. □

Proof of Proposition 4.6

Recall that the maximum degree weight design scheme employs the following settings: $\mathcal{L}(d_i, d_j) \triangleq n^{-1}$ for $(i,j) \in \mathcal{E}$ and $i \neq j$ and $W_{ii} = 1 - d_i/n$ (see Section 3.2.1, page 19; and [68, 69]). With these choices, $\zeta_{i,n}$ takes the following form:

$$\zeta_{i,n} = \frac{1}{n}\sum_{j=1, j\neq i}^{n} I\{r_{i,j}^2 \leq r_c^2\} \tag{4.104}$$

Taking the expectation of $\zeta_{i,n}$ gives us:

$$\mathbb{E}\{\zeta_{i,n}\} = \frac{1}{n}\sum_{j=1,j\neq i}^{n}\mathbb{E}\{I\{r_{i,j}^2 \leq r_c^2\}\} \;=\; \frac{n-1}{n}p, \qquad 0 \leq p \leq 1 \tag{4.105}$$

Now, consider the *double averaged* [82] correlation function (4.87) of the random process defined in (4.100)

$$\begin{aligned}R_n &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}\{(\zeta_{i,n} - \mathbb{E}\{\zeta_{i,n}\})(\zeta_{j,n} - \mathbb{E}\{\zeta_{j,n}\})\} \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}\left\{\left(\frac{1}{n}\sum_{k=1,k\neq i}^{n} I\{r_{i,k}^2 \leq r_c^2\} - \frac{n-1}{n}p\right)\left(\frac{1}{n}\sum_{\ell=1,\ell\neq j}^{n} I\{r_{j,\ell}^2 \leq r_c^2\} - \frac{n-1}{n}p\right)\right\} \\ &= \frac{1}{n^4}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1,k\neq i}^{n}\sum_{\ell=1,\ell\neq j}^{n}\mathbb{E}\left\{I\{r_{i,k}^2 \leq r_c^2\}I\{r_{j,\ell}^2 \leq r_c^2\}\right\} - \frac{(n-1)^2}{n^2}p^2\end{aligned}$$

Let us examine the quadruple sum in (4.106). There are four possible cases to analyze:

1. $i = j$ and $k = \ell$: The number of occurrences of this event is $n(n-1)$. The expectation

can be evaluated:

$$\mathbb{E}\left\{I\{r_{i,k}^2 \leq r_c^2\}^2\right\} = p^2 + p(1-p) = p \tag{4.106}$$

2. $i = j$ and $k \neq \ell$: The number of occurrences of this event is equal to $n(n-1)(n-2)$. It is not necessary to evaluate the expectation directly. It is sufficient to note that this expectation corresponds to the probability of three arbitrary nodes in the network being connected. This probability is less than or equal to the probability of two arbitrary nodes being connected. For some p' such that $0 \leq p' \leq p(1-p)$, we have:

$$\mathbb{E}\left\{I\{r_{i,k}^2 \leq r_c^2\}I\{r_{i,\ell}^2 \leq r_c^2\}\right\} = p^2 + p' \tag{4.107}$$

3. $i \neq j$ and $k = \ell$: This case is analogous to the preceding case.

4. $i \neq j$ and $k \neq \ell$: The number of occurrences of this event is equal to $n(n-1)(n^2 - 3n + 3)$. The expectation is easy to evaluate using the independence of the random variables involved. The expectation corresponds to the probability of two independent randomly selected pairs of nodes being connected:

$$\mathbb{E}\left\{I\{r_{i,k}^2 \leq r_c^2\}I\{r_{j,\ell}^2 \leq r_c^2\}\right\} = p^2 \tag{4.108}$$

The above analysis leads to the following bound on the double averaged correlation function:

$$\begin{aligned}
R_n &= \frac{1}{n^4}\left(n(n-1)(p^2+p(1-p)) + 2n(n-1)(n-2)(p^2+p')\right) - \frac{(n-1)^2}{n^2}p^2 \\
&= \frac{(n-1)^2}{n^2}p^2 - \frac{(n-1)^2}{n^2}p^2 + \frac{n(n-1)}{n^4}p(1-p) + \frac{2n(n-1)(n-2)}{n^4}p' \\
&< \frac{p(1-p)}{n^2} + \frac{2p'}{n} \\
&< \frac{p(1-p)}{n^2} + \frac{2p(1-p)}{n} \\
&= p(1-p)\frac{2n+1}{n^2}
\end{aligned}$$

Now we can use (4.109) and (4.99) to show that the series (4.86) converges. Indeed,

$$\sum_{n\in\mathbb{N}^+} \frac{\sigma_n}{n}\sqrt{R_{n-1}} + \frac{\sigma_n^2}{n^2} < \sum_{n\in\mathbb{N}^+} \frac{2\sqrt{2}\sqrt{p(1-p)}}{\sqrt{n}(n-1)} + \frac{4}{n^2} \ . \tag{4.109}$$

The series on the right hand side of (4.109) converges, which implies the convergence of the series in (4.86).

Since (4.86) is satisfied, we can apply Theorem 4.4 with (4.105) to derive (4.91). The result (4.92) follows immediately from the definition of $\Lambda(\xi)$ in (4.81). □

4.3 Accelerated Average Consensus with Short Node Memory

The simplest case of local memory is two taps (a single tap is equivalent to storing only the current value, as in standard distributed averaging), and this is the case we consider in this section. For two taps of memory, prediction at node i is based on the previous state value $x_i(t-1)$, the current value $x_i(t)$, and the value achieved by one application of the original averaging matrix, i.e. $x_i^{\mathrm{W}}(t+1) = W_{ii}x_i(t) + \sum_{j\in\mathcal{N}_i} W_{ij}x_j(t)$. The state-update equations at a node become a combination of the predictor and the value derived by application of the consensus weight matrix (this is easily extended for predictors with longer memories; see [74, 75]). In the two-tap memory case, we have:

$$x_i(t+1) = \alpha x_i^{\mathrm{P}}(t+1) + (1-\alpha)x_i^{\mathrm{W}}(t+1) \tag{4.110a}$$

$$x_i^{\mathrm{W}}(t+1) = W_{ii}x_i(t) + \sum_{j\in\mathcal{N}_i} W_{ij}x_j(t) \tag{4.110b}$$

$$x_i^{\mathrm{P}}(t+1) = \theta_3 x_i^{\mathrm{W}}(t+1) + \theta_2 x_i(t) + \theta_1 x_i(t-1). \tag{4.110c}$$

Here $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]$ is the vector of predictor coefficients.

The network-wide equations can then be expressed in matrix form by defining

$$\mathbf{W}_3[\alpha] \triangleq (1-\alpha+\alpha\theta_3)\mathbf{W} + \alpha\theta_2\mathbf{I}, \tag{4.111}$$

$$\mathbf{X}(t) \triangleq [\mathbf{x}(t)^{\mathsf{T}}, \mathbf{x}(t-1)^{\mathsf{T}}]^{\mathsf{T}}, \tag{4.112}$$

where $\mathbf{I}$ is the identity matrix of the appropriate size, and

$$\mathbf{\Phi}_3[\alpha] \triangleq \begin{bmatrix} \mathbf{W}_3[\alpha] & \alpha\theta_1\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}.$$

Each block of the above matrix has dimensions $n \times n$. We also define $\mathbf{x}(-1) = \mathbf{x}(0)$ so that $\mathbf{X}(0) = [\mathbf{x}(0)^\mathsf{T}\mathbf{x}(0)^\mathsf{T}]^\mathsf{T}$. The update equation is then simply $\mathbf{X}(t+1) = \mathbf{\Phi}_3[\alpha]\mathbf{X}(t)$.

For the two-tap memory case, the predictor coefficients based on the least-squares design described in Section 4.1.2 are identified as $\boldsymbol{\theta} = \mathbf{B}^{\dagger\mathsf{T}}\mathbf{c}$, where

$$\mathbf{B} \triangleq \begin{bmatrix} -2 & -1 & 0 \\ 1 & 1 & 1 \end{bmatrix}^\mathsf{T}, \tag{4.113}$$

$\mathbf{c} \triangleq [1,1]^\mathsf{T}$. This choice of predictor coefficients satisfies the technical conditions on $\boldsymbol{\theta}$ in Theorems 4.1 and 4.2: $\theta_1 + \theta_2 + \theta_3 = 1$. This choice of predictor coefficients also implies $\theta_3 \geq 1, \theta_2 \geq 0$.

4.3.1 Convergence of the Accelerated Distributed Average Consensus with Short Node Memory

The scaling properties of the averaging time for the general weight matrices were identified in Theorem 4.2. In order to apply this result, we must establish that $\mathbf{\Phi}_3[\alpha]$ satisfies the conditions of Theorem 4.2. In doing so, we will also show that (i) for $\mathbf{\Phi} = \mathbf{\Phi}_3[\alpha]$, the limit $\overline{\mathbf{\Phi}}_3[\alpha]\mathbf{X}(0) = \mathbf{J}\mathbf{X}(0)$, so our approach indeed converges to the average consensus, and (ii) that the limiting averaging time is characterized by a function of $\rho(\mathbf{\Phi}_3[\alpha] - \mathbf{J})$, which motivates choosing α to optimize this expression. (Recall, in this setting $\mathbf{J}$ is the $2n \times 2n$ matrix with all entries equal to $1/2n$.) Note that in the following proposition, the conditions on $\boldsymbol{\theta}$ are technical conditions that ensure convergence is achieved. Three factors motivate our belief that these are not overly-restricting: (i) these conditions are satisfied if we employ the least-squares predictor weights design strategy of Section 4.1.2; (ii) the conditions are relatively natural for a linear predictor that is based on an estimate of slope; (iii) in Section 4.3.3 we show that the choice of weights does not have a significant effect on the convergence properties. The condition on α is necessary for $\mathbf{\Phi}_3[\alpha]^t$ to have a limit as $t \to \infty$, as will be established in Section 4.3.7 (see page 83).

Proposition 4.7. *Let $\mathbf{\Phi}_3[\alpha]$ be defined as in (4.113) and assume that $\mathbf{W}$ satisfies conditions* (4.3) *for asymptotic convergence, $\theta_1+\theta_2+\theta_3=1$, $\theta_3\geq 1$, $\theta_2\geq 0$, and $\alpha\in[0,-\theta_1^{-1})$. Then:*

(a) $\overline{\mathbf{\Phi}}_3[\alpha]=\lim_{t\to\infty}\mathbf{\Phi}_3[\alpha]^t$ *exists, with* $\overline{\mathbf{\Phi}}_3[\alpha]\mathbf{X}(0)=\mathbf{J}\mathbf{X}(0)$ *for all* $\mathbf{X}(0)$ *defined in* (4.112),

(b) $\rho(\mathbf{\Phi}_3[\alpha]-\overline{\mathbf{\Phi}}_3[\alpha])>0$, *and*

(c) $\lim\limits_{\varepsilon\to 0}\frac{T_{ave}(\mathbf{\Phi}_3[\alpha],\varepsilon)}{\log\varepsilon^{-1}}<\frac{1}{\log\rho(\mathbf{\Phi}_3[\alpha]-\mathbf{J})^{-1}}$.

4.3.2 Optimal Mixing Parameter

According to the previous result, the averaging time required to approach the average within ε-accuracy grows at the rate at most $1/\log\rho(\mathbf{\Phi}_3[\alpha]-\mathbf{J})^{-1}$ as $\varepsilon\to 0$. Minimizing the spectral radius $\rho(\mathbf{\Phi}_3[\alpha]-\mathbf{J})$ is thus a natural optimality criterion. The following theorem establishes the optimal setting of α for a given weight matrix $\mathbf{W}$, as a function of $\lambda_2(\mathbf{W})$, the second largest eigenvalue of $\mathbf{W}$.

Theorem 4.5 (Optimal mixing parameter)**.** *Suppose $\mathbf{W}$ satisfies conditions* (4.3) *for asymptotic convergence and assume $\theta_3+\theta_2+\theta_1=1$ and $\theta_3\geq 1$, $\theta_2\geq 0$. Suppose further that $|\lambda_n(\mathbf{W})|\leq|\lambda_2(\mathbf{W})|$, where the eigenvalues $\lambda_1(\mathbf{W}),\dots,\lambda_n(\mathbf{W})$ are labeled in decreasing order. Then the solution of the optimization problem*

$$\alpha^\star=\arg\min_{\alpha}\rho(\mathbf{\Phi}_3[\alpha]-\mathbf{J}) \tag{4.114}$$

is given by the following:

$$\alpha^\star=\frac{-((\theta_3-1)\lambda_2(\mathbf{W})^2+\theta_2\lambda_2(\mathbf{W})+2\theta_1)-2\sqrt{\theta_1^2+\theta_1\lambda_2(\mathbf{W})\left(\theta_2+(\theta_3-1)\lambda_2(\mathbf{W})\right)}}{\left(\theta_2+(\theta_3-1)\lambda_2(\mathbf{W})\right)^2} \tag{4.115}$$

Here the conditions on $\boldsymbol{\theta}$ are the same as in Proposition 4.7. The condition on the weight matrix, $|\lambda_n(\mathbf{W})|\leq|\lambda_2(\mathbf{W})|$, significantly reduces the complexity of the proof. Most distributed algorithms for constructing weight matrices (e.g., MH or max-degree) lead to $\mathbf{W}$ that satisfy the condition, but they are not guaranteed to do so. We can ensure that the condition is satisfied by applying a completely local adjustment to any weight matrix. The mapping $\mathbf{W}\mapsto 1/2(\mathbf{I}+\mathbf{W})$ transforms any stochastic matrix $\mathbf{W}$ into a stochastic matrix

with all positive eigenvalues [14]; this mapping can be carried out locally, without any knowledge of the global properties of $\mathbf{W}$, and without affecting the order-wise asymptotic convergence rate as $n \to \infty$.

4.3.3 Convergence Rate Analysis

We begin with our main result for the convergence rate of two-tap predictor-based accelerated consensus. Theorem 4.6 indicates how the spectral radius of the accelerated operator $\mathbf{\Phi}_3[\alpha]$ is related to the spectral radius of the foundational weight matrix $\mathbf{W}$ (in terms of upper bounds on these quantities). Since the asymptotic convergence time is governed by the spectral radius, this relationship characterizes the improvement in convergence rate that can be obtained.

Theorem 4.6 (Convergence rate)**.** *Suppose the assumptions of Theorem 4.5 hold. Suppose further that the original matrix* $\mathbf{W}$ *satisfies* $\rho(\mathbf{W}-\mathbf{J}) \leq 1-\Psi(n)$ *for some function* $\Psi : \mathbb{N} \to (0,1)$ *of the network size* n*. Then the matrix* $\mathbf{\Phi}_3[\alpha^\star]$ *satisfies* $\rho(\mathbf{\Phi}_3[\alpha^\star]-\mathbf{J}) \leq 1-\sqrt{\Psi(n)}$.

Proof. According to the discussion in the proof of Theorem 4.5 (see page 83), we have

$$\begin{aligned}
&\rho(\mathbf{\Phi}_3[\alpha^\star]-\mathbf{J}) \\
&\quad = \lambda_2^*(\mathbf{\Phi}_3[\alpha^\star])| = (\alpha^\star|\theta_1|)^{1/2} \\
&\quad = \left[\frac{-((\theta_3-1)\lambda_2^2(\mathbf{W})+\theta_2\lambda_2(\mathbf{W})+2\theta_1) - 2\sqrt{\theta_1^2+\theta_1\lambda_2(\mathbf{W})\left(\theta_2+(\theta_3-1)\lambda_2(\mathbf{W})\right)}}{\left(\theta_2+(\theta_3-1)\lambda_2(\mathbf{W})\right)^2}|\theta_1|\right]^{1/2}.
\end{aligned}$$

In order to prove the claim, we consider two cases: $\lambda_2(\mathbf{W}) = 1-\Psi(n)$, and $\lambda_2(\mathbf{W}) < 1-\Psi(n)$.

First, we suppose that $\lambda_2(\mathbf{W}) = 1-\Psi(n)$ and show that $\rho(\mathbf{\Phi}_3[\alpha^\star]-\mathbf{J})^2-(1-\sqrt{\Psi(n)})^2 \leq 0$. Denoting $\Psi(n) = \delta$ and substituting $\lambda_2(\mathbf{W}) = 1-\delta$ and $\theta_1 = 1-\theta_2-\theta_3$, we obtain

$$\begin{aligned}
\rho(\mathbf{\Phi}_3[\alpha^\star]-\mathbf{J})^2 - (1-\sqrt{\Psi(n)})^2 &= -\left(\sqrt{\delta}-1\right)^2 \\
&\times \frac{(\theta_3-1)\left(\delta^2-\delta\right)+2\sqrt{\delta}\left(\theta_3+\theta_2-1\right)-2\sqrt{\delta\left(\theta_2+(2-\delta)\left(\theta_3-1\right)\right)\left(\theta_3+\theta_2-1\right)}}{[(2-\delta)\delta+1](1-\theta_3)-(1+\delta)\theta_2-2\sqrt{\delta\left(\theta_3+\theta_2-1\right)\left((\theta_3-1)(2-\delta)+\theta_2\right)}}.
\end{aligned}$$

It is clear from the assumptions that the expressions under square roots are non-negative. Furthermore, the denominator is negative since $1-\theta_3 < 0$, $\theta_2 > 0$ and $\delta \in (0,1)$. Finally,

note that $(\theta_3 - 1)(\delta^2 - \delta) \leq 0$ and $2\sqrt{\delta}(\theta_3 + \theta_2 - 1) \geq 0$. Thus, to see that the numerator is non-positive, observe that

$$[\sqrt{\delta}(\theta_3 + \theta_2 - 1)]^2 - \left[\sqrt{\delta(\theta_2 + (2-\delta)(\theta_3 - 1))(\theta_3 + \theta_2 - 1)}\right]^2 = (\delta - 1)\delta(\theta_3 - 1)(\theta_3 + \theta_2 - 1) \leq 0. \quad (4.116)$$

Thus, we have $\rho(\mathbf{\Phi}_3[\alpha^\star] - \mathbf{J})^2 - (1 - \sqrt{\Psi(n)})^2 \leq 0$, implying that $\rho(\mathbf{\Phi}_3[\alpha^\star] - \mathbf{J}) \leq 1 - \sqrt{\Psi(n)}$ if $\lambda_2(\mathbf{W}) = 1 - \Psi(n)$.

Now suppose $\lambda_2(\mathbf{W}) < 1 - \Psi(n)$. We have seen in Lemma 4.3 that $\alpha_i^*[\lambda_i(\mathbf{W})]$ is an increasing function of $\lambda_i(\mathbf{W})$, implying $\alpha_2^*[\lambda_2(\mathbf{W})] \leq \alpha_2^*[1 - \Psi(n)]$. Since $\rho(\mathbf{\Phi}_3[\alpha^\star] - \mathbf{J}) = (\alpha^\star|\theta_1|)^{1/2} = (\alpha_2^*[\lambda_2(\mathbf{W})]|\theta_1|)^{1/2}$ is an increasing function of $\alpha_2^*[\lambda_2(\mathbf{W})]$, the claim of theorem follows. □

In order to explore how fast the spectral radius, $\rho(\mathbf{\Phi}_3[\alpha^\star] - \mathbf{J}) = \sqrt{-\alpha^\star\theta_1}$, goes to one as $n \to \infty$, we can take its asymptotic Taylor series expansion:

$$\rho(\mathbf{\Phi}_3[\alpha^*] - \mathbf{J}) = 1 - \sqrt{\frac{2(\theta_3 - 1) + \theta_2}{\theta_3 - 1 + \theta_2}}\sqrt{\Psi(n)} + \mathcal{O}(\Psi(n)). \quad (4.117)$$

From this expression, we see that the bound presented in Theorem 4.6 correctly captures the convergence rate of the accelerated consensus algorithm. Alternatively, leaving only two terms in the expansion above, $\rho(\mathbf{\Phi}_3[\alpha^*] - \mathbf{J}) = 1 - \Omega(\sqrt{\Psi(n)})$, we see that the bound presented is rate optimal in Landau notation.

We can also use (4.117) to provide guidelines for choosing asymptotically optimal prediction parameters θ_3 and θ_2. In particular, it is clear that the coefficient $\gamma(\theta_2, \theta_3) = \sqrt{[2(\theta_3 - 1) + \theta_2]/[\theta_3 - 1 + \theta_2]}$ should be maximized to minimize the spectral radius $\rho(\mathbf{\Phi}_3[\alpha^\star] - \mathbf{J})$. It is straightforward to verify that setting $\theta_2 = 0$ and $\theta_3 = 1 + \epsilon$ for any $\epsilon > 0$ satisfies the assumptions of Theorem 4.5 and also satisfies $\gamma(0, 1 + \epsilon) > \gamma(\theta_2, 1 + \epsilon)$ for any positive θ_2. Since $\gamma(0, 1 + \epsilon) = \sqrt{2}$ is independent of ϵ (or θ_3) we conclude that setting $(\theta_1, \theta_2, \theta_3) = (-\epsilon, 0, 1 + \epsilon)$ satisfies the assumptions of Theorem 4.5 and asymptotically yields the optimal limiting ε-averaging time for the proposed approach, as $n \to \infty$.

4.3.4 Processing Gain Analysis

Next, we investigate the gain that can be obtained by using the accelerated algorithm in the $M = 3$ case. We consider the ratio $\tau_{\text{asym}}(\mathbf{W})/\tau_{\text{asym}}(\mathbf{\Phi}_3[\alpha^*])$ of the asymptotic convergence time of the standard consensus algorithm using weight matrix $\mathbf{W}$ and the asymptotic convergence time of the proposed accelerated algorithm. This ratio shows how many times fewer iterations, asymptotically, the optimized predictor-based algorithm must perform to reduce error by a factor of e^{-1}.

If the network topology is modeled as random (e.g., a sample from the family of random geometric graphs), we adopt the expected gain $\mathcal{G}(\mathbf{W}) = \mathbb{E}\{\tau_{\text{asym}}(\mathbf{W})/\tau_{\text{asym}}(\mathbf{\Phi}_3[\alpha^\star])\}$ as a performance metric, where $\mathbf{\Phi}_3[\alpha^*]$ is implicitly constructed using the same matrix $\mathbf{W}$. The expected gain characterizes the average improvement obtained by running the algorithm over many realizations of the network topology. In this case the spectral radius, $\rho(\mathbf{W} - \mathbf{J})$, is considered to be a random variable dependent on the particular realization of the graph. Consequently, the expectations in the following theorem are taken with respect to the measure induced by the random nature of the graph.

Theorem 4.7 (Expected gain)**.** *Suppose the assumptions of Theorem 4.5 hold. Suppose further that the original matrix* $\mathbf{W}$ *satisfies* $\mathbb{E}\{\rho(\mathbf{W} - \mathbf{J})\} = 1 - \Psi(n)$ *for some function* $\Psi : \mathbb{N} \to (0, 1)$ *of the network size* n*. Then* $\mathcal{G}(\mathbf{W}) = 1/\sqrt{\Psi(n)}$*.*

Proof. First, condition on a particular realization of the graph topology, and observe from the definition of $\tau_{\text{asym}}(\cdot)$ that

$$\frac{\tau_{\text{asym}}(\mathbf{W})}{\tau_{\text{asym}}(\mathbf{\Phi}_3[\alpha^\star])} = \frac{\log \rho(\mathbf{\Phi}_3[\alpha^\star] - \mathbf{J})}{\log \rho(\mathbf{W} - \mathbf{J})}. \tag{4.118}$$

Next, fixing $\rho(\mathbf{W} - \mathbf{J}) = 1 - \psi$, where $\Psi(n) = \mathbb{E}\{\psi\}$, and using Theorem 4.6, we have

$$\frac{\tau_{\text{asym}}(\mathbf{W})}{\tau_{\text{asym}}(\mathbf{\Phi}_3[\alpha^\star])} \geq \frac{\log(1 - \sqrt{\psi})}{\log(1 - \psi)}. \tag{4.119}$$

Let $f(\psi) = \log(1 - \sqrt{\psi})/\log(1 - \psi)$. Taking the Taylor series expansion of $f(\psi)$ at $\psi = 0$, we obtain

$$f(\psi) = \frac{1}{\sqrt{\psi}} + \frac{1}{2} - \frac{1}{6}\psi^{1/2} - \frac{1}{20}\psi^{3/2} - \dots. \tag{4.120}$$

Noting that $\psi > 0$ we conclude that the following holds uniformly over $\psi \in [0,1]$: $f(\psi) \leq \frac{1}{\sqrt{\psi}} + \frac{1}{2}$. At the same time, taking the Taylor series expansions of the numerator and denominator of $f(\psi)$, we obtain

$$f(\psi) = \frac{\sqrt{\psi} + \frac{\psi}{2} + \frac{\psi^{3/2}}{3} + \frac{\psi^2}{4} + \frac{\psi^{5/2}}{5} + \dots}{\psi + \frac{\psi^2}{2} + \frac{\psi^3}{3} + \dots}. \tag{4.121}$$

Noting that $1/6 + 1/3 = 1/2$, $2/15 + 1/5 = 1/3$, we can express this as

$$f(\psi) = \frac{1}{\sqrt{\psi}} \frac{\sqrt{\psi} + \frac{\psi}{2} + \frac{\psi^{3/2}}{3} + \frac{\psi^2}{4} + \frac{\psi^{5/2}}{5} + \dots}{\sqrt{\psi} + \frac{\psi^{3/2}}{6} + \frac{\psi^{3/2}}{3} + \frac{2\psi^{5/2}}{15} + \frac{\psi^{5/2}}{5} + \dots}, \tag{4.122}$$

and using the fact that $1/2\psi \geq 1/6\psi^{3/2}$, $1/4\psi^2 \geq 2/15\psi^{5/2}$, ... uniformly over $\psi \in [0,1]$, we conclude that $f(\psi) \geq \frac{1}{\sqrt{\psi}}$. Thus, $\frac{1}{\sqrt{\psi}} \leq f(\psi) \leq \frac{1}{\sqrt{\psi}} + \frac{1}{2}$. Finally, observe that $\frac{\partial^2}{\partial \psi^2}\psi^{-1/2} = 3/4\psi^{-5/2} > 0$ if $\psi > 0$, implying that $1/\sqrt{\psi}$ is convex. To complete the proof we take the expectation with respect to graph realizations and apply Jensen's inequality to obtain

$$\mathbb{E}\left\{\frac{\tau_{\text{asym}}(\mathbf{W})}{\tau_{\text{asym}}(\mathbf{\Phi}_3[\alpha^\star])}\right\} \geq \mathbb{E}\left\{\frac{1}{\sqrt{\psi}}\right\} \geq \frac{1}{\sqrt{\mathbb{E}\{\psi\}}}. \tag{4.123}$$

□

We note that there is no loss of generality in considering the expected gain since, in the case of a deterministic network topology, instead of taking the expectation with respect to a non-trivial graph distribution, we can operate with the distribution in the form of delta function and results will still hold since they are based on the deterministic derivations in Theorems 4.5 and 4.6.

For a chain graph (path of n vertices) the eigenvalues of the normalized graph Laplacian $\mathcal{L}$ are given by $\lambda_i(\mathcal{L}) = 1 - \cos(\pi i/(n-1)), i = 0, 1, \dots, n-1$ [83]. It is straightforward to verify that for the MH weight matrix a similar expression holds: $\lambda_i(\mathbf{W}_{\text{MH}}) = 1/3 + 2/3\cos(\pi(i-1)/n), i = 1, 2, \dots, n$. Thus, in this case, $\rho(\mathbf{W}_{\text{MH}} - \mathbf{J}) = 1/3 + 2/3\cos(\pi/n)$. For large enough n this results in $\rho(\mathbf{W}_{\text{MH}} - \mathbf{J}) \approx 1 - \frac{\pi^2}{3}\frac{1}{n^2} + \mathcal{O}(1/n^4)$. Using the same sequence of steps used to prove Theorem 4.7 above without taking expectations, we see that for the chain topology, the improvement in asymptotic convergence rate is asymptotically lower bounded by n; i.e., $\mathcal{G}(\mathbf{W}) = \Omega(n)$. Similarly, for a network with two-dimensional grid

topology, taking $\mathbf{W}$ to be the transition matrix for a natural random walk on the grid (a minor perturbation of the MH weights) it is known [84] that $(1-\lambda_2(\mathbf{W}))^{-1} = \Theta(n)$. Thus, for a two-dimensional grid, the proposed algorithm leads to a gain of $\mathcal{G}(\mathbf{W}) = \Omega(n^{1/2})$.

This discussion suggests that the following result may also be useful in characterizing the improvement in asymptotic convergence rate obtained by using the proposed algorithm.

Corollary 4.1. *Suppose that assumptions of Theorem 4.7 hold and suppose in addition that* $\rho(\mathbf{W}-\mathbf{J}) = 1-\Theta(\frac{1}{n^\beta})$ *then the improvement in asymptotic convergence rate attained by the accelerated algorithm is* $\mathcal{G}(\mathbf{W}) = \Omega(n^{\beta/2})$.

4.3.5 Initialization Heuristic: Decentralized Estimation of $\lambda_2(\mathbf{W})$

Under our assumptions, the optimal value of the mixing parameter depends only on the values of predictor coefficients and the second largest eigenvalue of initial matrix $\mathbf{W}$. In this section we introduce a decentralized procedure for estimating $\lambda_2(\mathbf{W})$. Since we assume the predictor weights, $\boldsymbol{\theta}$, and weight matrix $\mathbf{W}$ are fixed and specified, this is the only parameter that remains to be identified for a fully decentralized implementation of the algorithm. Estimation of $\lambda_2(\mathbf{W})$ is a straightforward exercise if we employ the method of DOI proposed for distributed spectral analysis in [79] and refined for distributed optimization applications in [14].

Algorithm 1 presents the proposed specialized and streamlined version of DOI, which is only used to calculate the second-largest eigenvalue of the consensus update matrix $\mathbf{W}$. The main idea of DOI is to repeatedly apply $\mathbf{W}$ to a random vector $\mathbf{v}_0$, with periodic normalization and subtraction of the estimate of the mean, until $\mathbf{v}_K = \mathbf{W}^K\mathbf{v}_0$ converges to the second-largest eigenvector of $\mathbf{W}$. Then, we estimate the second-largest eigenvalue by calculating $||\mathbf{W}\mathbf{v}_K||/||\mathbf{v}_K||$ for a valid matrix norm $\|\cdot\|$. Previous algorithms for DOI [14,79] have normalized in step 6 by the ℓ_2 norm of $\mathbf{v}_k$, estimated by K iterations of consensus, and step 9 previously required an additional K iterations to calculate $\|\mathbf{W}\mathbf{V}_K\|_2$ and $\|\mathbf{v}_K\|_2$. In addition, because the initial random vectors in [14,79] are not zero-mean, these algorithms must apply additional consensus operations to eliminate the bias (otherwise $\mathbf{v}_K$ converges to $\mathbf{1}$). Previous algorithms thus have $\mathcal{O}(K^2)$ complexity, where K is the topology-dependent number of consensus iterations needed to achieve accurate convergence to the average value. For example, for a random geometric graph, one typically needs $K \propto n$.

The main innovations of Algorithm 1 are in line 2, which ensures that the initial ran-

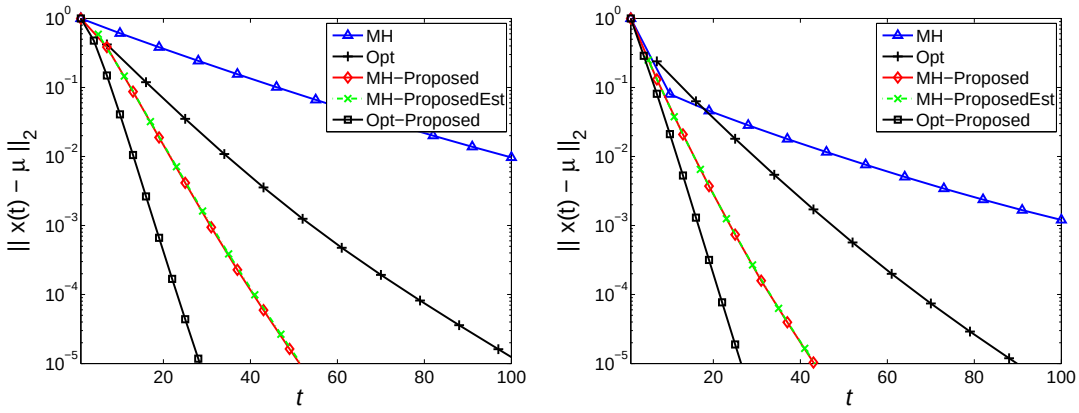
Algorithm 1: Spectral radius estimation (Input: foundational weight matrix $\mathbf{W}$)

1 Choose random vector $\mathbf{v}$;
2 Set $\mathbf{v}_0 = \mathbf{W}\mathbf{v} - \mathbf{v}$; `Generate zero-mean random vector`
3 **for** $k = 1$ *to* K **do**
4 $\quad \mathbf{v}_k = \mathbf{W}\mathbf{v}_{k-1}$; `Apply W to converge to second-largest eigenvector`
5 $\quad$ **if** $k \mod L = 0$ **then**
6 $\quad\quad \mathbf{v}_k = \mathbf{v}_k / ||\mathbf{v}_k||_\infty$; `Normalize by supremum norm every` L `iterations`
7 $\quad$ **endif**
8 **endfor**
9 Let $\widehat{\lambda}_2(\mathbf{W}) = \|\mathbf{W}\mathbf{v}_K\|_\infty / \|\mathbf{v}_K\|_\infty$;

dom vector is zero mean, in line 6, where normalization is done (after every L applications of the consensus update) using the supremum norm, and line 9, where the supremum norm is also used in lieu of the ℓ_2 norm (based on the Gelfand's formula [72] we have $\lim_{K\to\infty} \|\mathbf{W}\mathbf{v}_K\|_\infty / \|\mathbf{v}_K\|_\infty = \rho(\mathbf{W} - \mathbf{J})$). The maximum entry of the vector $\mathbf{v}_K$ can be calculated using a maximum consensus algorithm, wherein every node updates its value with the maximum of its immediate neighbors: $x_i(t) = \max_{j\in\mathcal{N}_i} x_j(t-1)$. Maximum consensus requires at most n iterations to converge for any topology; more precisely it requires a number of iterations equal to the diameter, D, of the underlying graph, which is often much less than n (and much less than K). Equally importantly, maximum consensus achieves perfect agreement. In the algorithms of [14,79] each node normalizes by a slightly different value (there are residual errors in the consensus procedure). In Algorithm 1, all nodes normalize by the same value, and this leads to much better estimation accuracy. Taken together, these innovations lead to an algorithm that is only $\mathcal{O}(K)$ (with the appropriate choice of L). In particular, the complexity of Algorithm 1 is clearly $\mathcal{O}(K + DK/L + D)$. Choosing $L \propto D$ (assuming that $\lambda_2(\mathbf{W})^D \gg \Delta$, where Δ is machine precision) we obtain an $\mathcal{O}(K)$ algorithm. The proposed initialization algorithm has significantly smaller computation/communication complexity than the initialization algorithm proposed for the distributed computation of the optimal matrix in [14].

4.3.6 Numerical Experiments and Discussion

This section presents simulation results for two scenarios. In the first simulation scenario, network topologies are drawn from the family of random geometric graphs of n nodes [13].

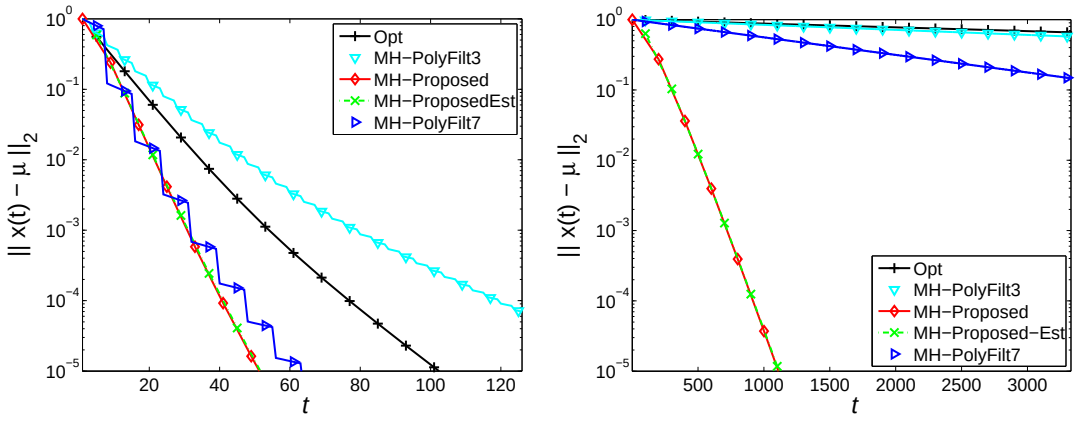


(a) Slope initialization (b) Spike initialization

Fig. 4.4 MSE vs. iterations for 200-node random geometric graphs. The algorithms compared are: optimal weights (Opt): $+$; MH weights (MH): $\triangle$; proposed method with oracle $\lambda_2(\mathbf{W})$ and MH matrix (MH-Proposed): $\diamond$; proposed with decentralized estimate of $\lambda_2(\mathbf{W})$ (MH-ProposedEst): $\times$; accelerated consensus, with oracle $\lambda_2(\mathbf{W})$ and optimal matrix (Opt-Proposed): $\square$. (a) Slope initialization. (b) Spike initialization.

In this model, n nodes are randomly assigned coordinates in the unit square, and links exist between nodes that are at most a distance $\sqrt{2\log n/n}$. (This scaling law for the connectivity radius guarantees the network is connected with high probability [13].) Two models for the initial node measurements, $\mathbf{x}(0)$, are considered. In the "Slope" model, the initial value $x_i(0)$ at node i is just the sum of its coordinates in the unit square. In the "Spike" model, all nodes are initialized to 0, except for one randomly chosen node whose initial value is set to one. All simulation results are generated based on 300 trials (a different random graph and node initialization is generated for each trial). The initial values are normalized so that the initial variance of node values is equal to 1. The second simulation scenario is for the n-node chain topology. Intuitively, this network configuration constitutes one of the most challenging topologies for distributed averaging algorithms since the chain has the longest diameter and weakest connectivity of all graphs on n nodes and information must diffuse across this distance. For this topology, we adopt analogous versions of the "Slope" and "Spike" initializations to those described above; for the "Slope", $x_i(0) = i/n$, and for the "Spike", we average over all possible locations of the one.

We run the algorithm n times with different initializations of the eigenvalue estimation algorithm to investigate the effects of initializing $\alpha^\star$ with an imperfect estimate of $\lambda_2(\mathbf{W})$.



(a) Random geometric graph (b) Chain

Fig. 4.5 MSE vs. iteration for 200-node topologies, Slope initialization. The algorithms compared are: optimal weights (Opt): $+$; polynomial filter with 3 taps (MH-PolyFilt3): $\triangledown$ and 7 taps (MH-PolyFilt7): $\triangleright$; proposed method with oracle $\lambda_2(\mathbf{W})$ and MH matrix (MH-Proposed): $\diamond$; proposed method with decentralized estimate of $\lambda_2(\mathbf{W})$ (MH-ProposedEst): $\times$.

In simulations involving the calculation of averaging time (according to the definition in (3.21)) we have fixed the required accuracy of computations, ε, at the level -100 dB (i.e., a relative error of 1×10^{-5}). For prediction parameters, we use $(\theta_1, \theta_2, \theta_3) = (-\epsilon, 0, 1+\epsilon)$, $\epsilon = 1/2$, as these were shown to be asymptotically optimal in Section 4.3.3. We compare our algorithm with two memoryless approaches, the MH weight matrix, and the optimal weight matrix [71]. MH weights are attractive because they can be calculated by each node simply using knowledge of its own degree and its neighbors' degrees. We also compare to two approaches from the literature that also make use of memory at each node to improve the rate of convergence: polynomial filtering [69], and finite-time consensus [78].

To investigate the effect of initialization on the performance of the proposed algorithm, we first plot the MSE decay curves as a function of the number of consensus iterations t for network size $n = 200$, RGG topology and different initializations. Figure 4.4 compares the performance of the proposed algorithm with the algorithms using the MH or the optimal matrix [71]. It can be seen that our decentralized initialization scheme does not have a major influence on the performance of our approach, as the method initialized using a decentralized estimate for $\lambda_2(\mathbf{W})$ (the curve labeled MH-ProposedEst) and the method initialized using precise knowledge of $\lambda_2(\mathbf{W})$ (labeled MH-Proposed) coincide nearly exactly.

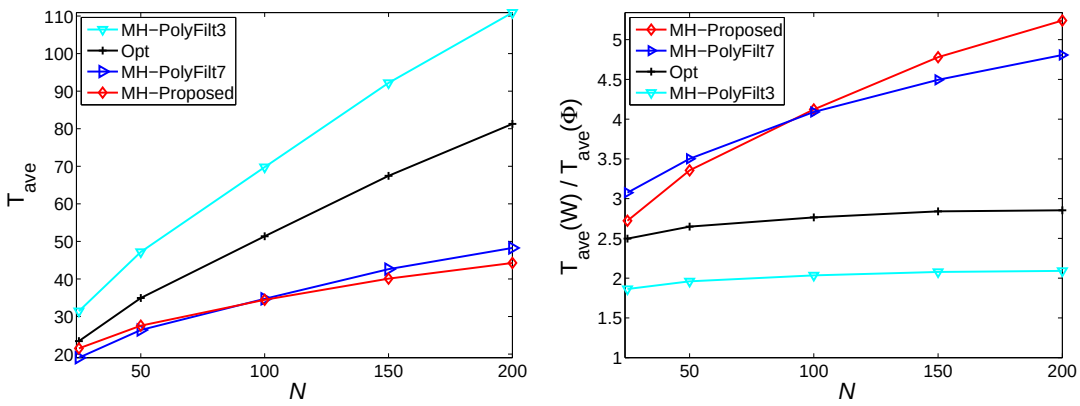


Fig. 4.6 Averaging time characterization, random geometric graph topologies. The algorithms compared are: optimal weights (Opt): $+$; polynomial filter with 3 taps (MH-PolyFilt3): $\triangledown$, and 7 taps (MH-PolyFilt7): $\triangleright$; proposed method with oracle $\lambda_2(\mathbf{W})$ and MH matrix (MH-Proposed): $\diamond$. (a) Averaging time as a function of the network size. (b) Ratio of the averaging time of the non-accelerated algorithm to that of the associated accelerated algorithm.

The procedure discussed in Section 4.3.5 provides a good estimate of $\lambda_2(\mathbf{W})$ (to within 10^{-3} maximum relative error for a 200 node RGG). It is also clear that the proposed algorithm outperforms both the memoryless MH and optimal weight matrices. In this experiment we fixed $K = 2n$ and $L = 10$. Note that the results in Figure 4.4 and all subsequent figures do not account for initialization costs. The initialization cost is relatively small. For the 200-node RGG it is equal to about $3n = 600$ consensus iterations (if we bound the diameter of the 200-node RGG by 20). If we desire a relative error of 10^{-3}, our algorithm gains approximately 70 iterations over memoryless MH consensus, based on Fig. 4.4(b). For this desired accuracy, the initialization overhead is thus recovered after less than 10 consensus operations.

Figure 4.5 compares the MSE curves for the proposed algorithm with two versions of polynomial filtering consensus [69], one using 3 taps and the other using 7 taps. We see that in the RGG scenario, our algorithm outperforms polynomial filtering with 3 memory taps and converges at a rate similar to that of the 7-tap version of polynomial filtering[2].

[2]Calculating optimal weights in the polynomial filtering framework quickly becomes ill-conditioned with increasing filter length, and we were not able to obtain stable results for more than 7 taps on random geometric graph topologies. Note that the original paper [69] also focuses on filters of length no more than 7. We conjecture that this ill-conditioning stems from the fact that the optimal solution involves pseudo-

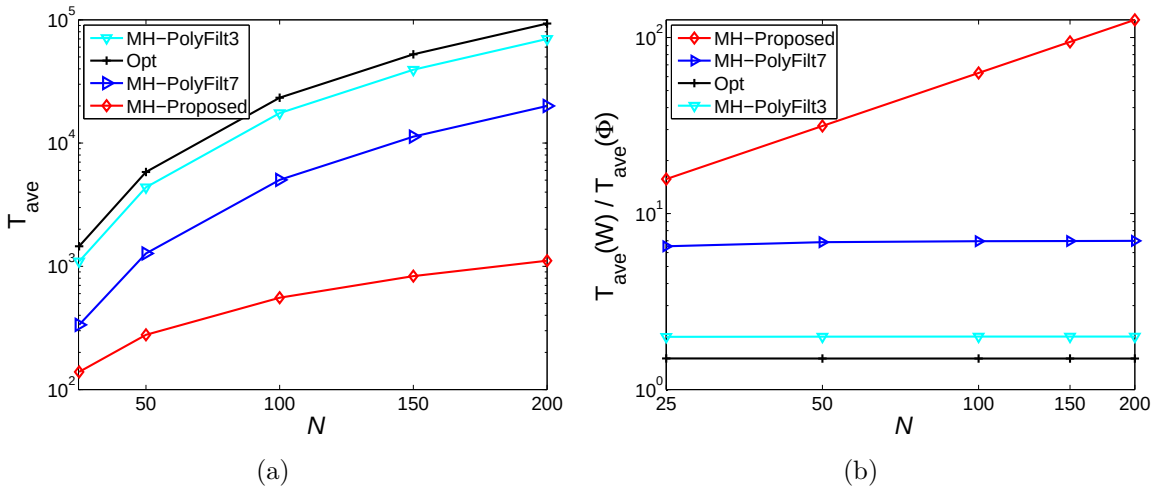
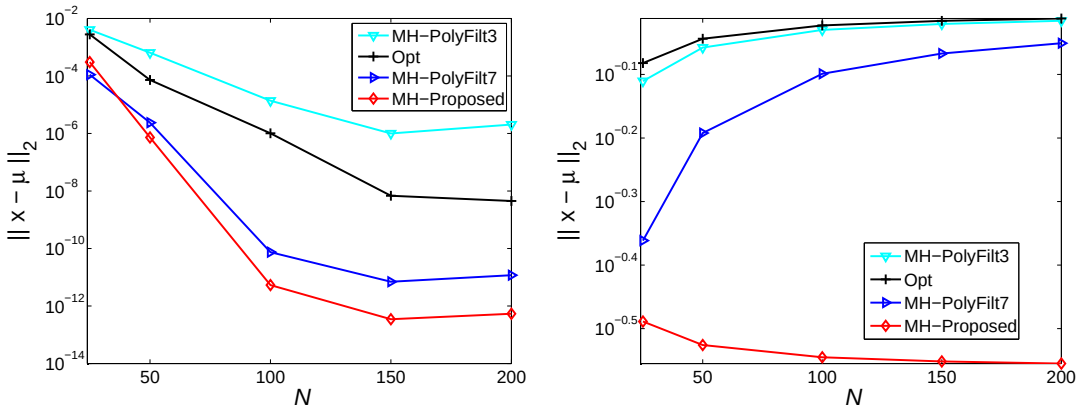


Fig. 4.7 Averaging time characterization, chain topology. The algorithms compared are: optimal weights (Opt): $+$; polynomial filter with 3 taps (MH-PolyFilt3): $\triangledown$, and 7 taps (MH-PolyFilt7): $\triangleright$; proposed method with oracle $\lambda_2(\mathbf{W})$ and MH matrix (MH-Proposed): $\diamond$. (a) Averaging time as a function of the network size. (b) Improvement due to the accelerated consensus: ratio of the averaging time of the non-accelerated algorithm to that of the associated accelerated algorithm.

Decentralized calculation of topology-adapted polynomial filter weights also remains an open problem. We conclude that for random geometric graphs, our algorithm has superior properties with respect to polynomial filtering since it has better error performance for the same computational complexity, and our approach is suitable for completely distributed implementation. Moving our attention to the chain topology only emphasizes these points, as our accelerated algorithm significantly outperforms even 7-tap polynomial filtering. Note that decentralized initialization of our algorithm also works well in the chain graph scenario. However, to obtain this result we have to increase the number of consensus iterations in the eigenvalue estimation algorithm, K, from $2n$ to n^2. This increase in the complexity of the distributed optimization of accelerated consensus algorithm is due to the properties of the power methods [85] and related eigenvalue estimation problems. The accuracy of the second largest eigenvalue computation depends on the ratio $\lambda_3(\mathbf{W})/\lambda_2(\mathbf{W})$, and this ratio

inversion of a Vandermonde matrix containing powers of the original eigenvalues. Since, for random geometric graph topologies, eigenvalues are not described by a regular function (e.g., the cosine, as for the chain graph) there is a relatively high probability (increasing with n) that the original weight matrix contains two similar-valued eigenvalues which may result in the Vandermonde matrix being ill-conditioned.



(a) Random geometric graph (b) Chain

Fig. 4.8 MSE at the point when finite time consensus of Sundaram and Hadjicostis has enough information to calculate the exact average at all nodes. The algorithms compared are: optimal weights (Opt): $+$; polynomial filter with 3 taps (MH-PolyFilt3): $\triangledown$, and 7 taps (MH-PolyFilt7): $\triangleright$; proposed method with oracle $\lambda_2(\mathbf{W})$ and MH matrix (MH-Proposed): $\diamond$. (a) Random geometric graph. (b) Chain topology.

increases much more rapidly for the chain topology as n grows than it does for random geometric graphs.

To investigate the robustness and scalability properties of the proposed algorithm, we next examine the averaging time, $T_{\text{ave}}(\mathbf{\Phi}_3[\alpha^*])$, as defined in (3.21), and the ratio $T_{\text{ave}}(\mathbf{W})/T_{\text{ave}}(\mathbf{\Phi}_3[\alpha^*])$, for random geometric graphs (Fig. 4.6) and the chain topology (Fig. 4.7). We establish through simulation that the scaling behavior of the ratio that can be measured experimentally matches very well with the asymptotic result established theoretically for the processing gain, $\tau_{\text{asym}}(\mathbf{W})/\tau_{\text{asym}}(\mathbf{\Phi}_3[\alpha^*])$. We see from Fig. 4.6 that in the random geometric graph setting, the proposed algorithm always outperforms consensus with the optimal weight matrix and polynomial filter with equal number of memory taps, and our approach scales comparably to 7-tap polynomial filtering. On the other hand, in the chain graph setting (Fig. 4.7) the proposed algorithm outperforms all the competing algorithms. Another interesting observation from Fig. 4.7 is that the gains of the polynomial filter and optimal weight matrix remain almost constant with varying network size while the gain obtained by the proposed algorithm increases significantly with n. This linear improvement with n matches well with the asymptotic behavior predicted by Theorem 4.7.

Finally, we compare the proposed algorithm with the linear observer approach of Sun-

daram and Hadjicostis [78], which works by remembering all of the consensus values, $x_i(t)$, seen at a node i (unbounded memory). After enough updates, each node is able to perfectly recover the average by locally solving a set of linear equations. To compare the method of [78] with our approach and the other asymptotic approaches described above, we determine the topology-dependent number of iterations that the linear-observer method must execute to have enough information to exactly recover the average. We then run each of the asymptotic approaches for the same number of iterations and evaluate performance based on the MSE they achieve. Figure 4.8 depicts results for both random geometric graph and chain topologies. For random geometric graphs of $n \geq 100$ nodes, we observe that the proposed algorithm achieves an error of at most 10^{-12} (roughly machine precision), by the time the linear observer approach has sufficient information to compute the average. For the chain topology the results are much more favorable for the linear-observer approach. However, the linear observer approach requires significant overhead to determine the topology-dependent coefficients that define the linear system to be solved at each node. Additionally, the linear observer approach does not scale well to large networks, since the amount of memory at each node grows with the size of the network. The approach proposed in this chapter only uses one extra memory tap per node, regardless of network size.

4.3.7 Proofs

Proof of Proposition 4.7

Proof of part (a). In Theorem 1 in [75], Johansson and Johansson show that the necessary and sufficient conditions for the consensus algorithm of the form $\mathbf{\Phi}_3[\alpha]$ to converge to the average are (JJ1) $\mathbf{\Phi}_3[\alpha]\mathbf{1} = \mathbf{1}$; (JJ2) $\mathbf{g}^\mathsf{T}\mathbf{\Phi}_3[\alpha] = \mathbf{g}^\mathsf{T}$ for vector $\mathbf{g}^\mathsf{T} = [\beta_1\mathbf{1}^\mathsf{T}\beta_2\mathbf{1}^\mathsf{T}]$ with weights satisfying $\beta_1 + \beta_2 = 1$; and (JJ3) $\rho(\mathbf{\Phi}_3[\alpha] - \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T}) < 1$. If these conditions hold then we also have $\overline{\mathbf{\Phi}}_3[\alpha] = \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T}$ [75] implying $\widetilde{\mathbf{X}}(0) = \overline{\mathbf{X}}(0)$. Condition (JJ1) is easily verified after straightforward algebraic manipulations using the definition of $\mathbf{\Phi}_3[\alpha]$ in (4.113), the assumption that $\theta_1 + \theta_2 + \theta_3 = 1$, and recalling that $\mathbf{W}$ satisfies $\mathbf{W1} = \mathbf{1}$ by design. To address condition (JJ2), we set $\beta_1 = 1/(1 + \alpha\theta_1)$ and $\beta_2 = \alpha\theta_1/(1 + \alpha\theta_1)$. Clearly, $\beta_1 + \beta_2 = 1$, and it is also easy to verify condition (JJ2) by plugging these values into the definition of $\mathbf{g}$, and using the same properties of $\mathbf{\Phi}_3[\alpha]$, the θ_i's, and $\mathbf{W}$ as above.

In order to verify that condition (JJ3) holds, we will show here that $\rho(\mathbf{\Phi}_3[\alpha] - \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T}) = \rho(\mathbf{\Phi}_3[\alpha] - \mathbf{J})$. In Section 4.3.7 we show that $\rho(\mathbf{\Phi}_3[\alpha] - \mathbf{J}) < 1$ if $\alpha \in [0, -\theta_1^{-1})$, and thus

condition (JJ3) is also satisfied under the assumptions of the proposition. To show that $\rho(\mathbf{\Phi}_3[\alpha] - \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T}) = \rho(\mathbf{\Phi}_3[\alpha] - \mathbf{J})$, we prove a stronger result, namely that $\mathbf{\Phi}_3[\alpha] - \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T}$ and $\mathbf{\Phi}_3[\alpha] - \mathbf{J}$ have the same eigenspectra. Consider the eigenvector $\mathbf{v}_i$ of $\mathbf{\Phi}_3[\alpha]$ with corresponding eigenvalue $\lambda_i(\mathbf{\Phi}_3[\alpha])$. This pair solves the eigenvalue problem, $\mathbf{\Phi}_3[\alpha]\mathbf{v}_i = \lambda_i(\mathbf{\Phi}_3[\alpha])\mathbf{v}_i$. Equivalently, expanding the definition of $\mathbf{\Phi}_3[\alpha]$, we have

$$\begin{bmatrix} \mathbf{W}_3[\alpha] & \alpha\theta_1\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{v}_i = \lambda_i(\mathbf{\Phi}_3[\alpha]) \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{v}_i. \tag{4.124}$$

We observe that (4.124) fits a modification of the first companion form of the linearization of a Quadratic Eigenvalue Problem (QEP) (see Section 3.4 in [86]). The QEP has general form $(\lambda^2\mathbf{M} + \lambda\mathbf{C} + \mathbf{K})\mathbf{u} = \mathbf{0}$, where $\mathbf{u}$ is the eigenvector associated with this QEP. The linearization of interest to us has the form:

$$\begin{bmatrix} -\mathbf{C} & -\mathbf{K} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda\mathbf{u} \\ \mathbf{u} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \lambda\mathbf{u} \\ \mathbf{u} \end{bmatrix} = \mathbf{0}. \tag{4.125}$$

The correspondence is clear if we make the associations: $\mathbf{M} = \mathbf{I}$, $\mathbf{C} = -\mathbf{W}_3[\alpha]$ and $\mathbf{K} = -\alpha\theta_1\mathbf{I}$, $\lambda = \lambda_i(\mathbf{\Phi}_3[\alpha])$ and $\mathbf{v}_i = [\lambda_i(\mathbf{\Phi}_3[\alpha])\mathbf{u}^\mathsf{T}\mathbf{u}^\mathsf{T}]^\mathsf{T}$. Eigenvectors $\mathbf{v}_i$ that solve (4.124) thus have special structure and are related to $\mathbf{u}_i$, the solution to the QEP,

$$(\lambda_i(\mathbf{\Phi}_3[\alpha])^2\mathbf{I} - \lambda_i(\mathbf{\Phi}_3[\alpha])\mathbf{W}_3[\alpha] - \alpha\theta_1\mathbf{I})\mathbf{u}_i = \mathbf{0}. \tag{4.126}$$

Because the first and third terms above are scaled identity matrices and the definition of $\mathbf{W}_3[\alpha]$ (see (4.111)) also involves scaled identity matrices, we can simplify this last equation to find that any solution $\mathbf{u}_i$ must also be an eigenvector of $\mathbf{W}$.

We have seen above, when verifying condition (JJ1), that $\mathbf{1}$ is an eigenvector of $\mathbf{\Phi}_3[\alpha]$ with corresponding eigenvalue $\lambda_i(\mathbf{\Phi}_3[\alpha]) = 1$. Likewise, we know that[3] $\mathbf{W}\mathbf{1} = \mathbf{1}$, and so this agrees with the structure of $\mathbf{v}_i$ identified above. Observe that, from the definition of $\mathbf{g}$ and because $\beta_1 + \beta_2 = 1$, we have $(\frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T})\mathbf{1} = \mathbf{1}$. Thus, $(\mathbf{\Phi}_3[\alpha] - \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T})\mathbf{1} = \mathbf{0}$. Similarly, recalling that $\mathbf{J} = \frac{1}{2n}\mathbf{1}\mathbf{1}^\mathsf{T}$, we have $\mathbf{J}\mathbf{1} = \mathbf{1}$, and thus $(\mathbf{\Phi}_3[\alpha] - \mathbf{J})\mathbf{1} = \mathbf{0}$. By design, $\mathbf{W}$ is a doubly stochastic matrix, and all eigenvectors $\mathbf{u}$ of $\mathbf{W}$ with $\mathbf{u} \neq \mathbf{1}$ are orthogonal to $\mathbf{1}$. It follows that $(\frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T})\mathbf{v}_i = \mathbf{0}$ for corresponding eigenvectors $\mathbf{v}_i = [\lambda_i(\mathbf{\Phi}_3[\alpha])\mathbf{u}^\mathsf{T}\mathbf{u}^\mathsf{T}]^\mathsf{T}$ of $\mathbf{\Phi}_3[\alpha]$,

[3] We abuse notation here, using $\mathbf{1}$ to denote the vector of all 1's, where the dimension is not explicitly indicated but should be clear from the context.

and thus $(\mathbf{\Phi}_3[\alpha] - \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T})\mathbf{v}_i = \mathbf{\Phi}_3[\alpha]\mathbf{v}_i = \lambda_i(\mathbf{\Phi}_3[\alpha])\mathbf{v}_i$. Similarly, $\mathbf{J}\mathbf{v}_i = 0$ if $\mathbf{v}_i \neq \mathbf{1}$, and $(\mathbf{\Phi}_3[\alpha] - \mathbf{J})\mathbf{v}_i = \lambda_i(\mathbf{\Phi}_3[\alpha])\mathbf{v}_i$. Therefore, we conclude that the matrices $(\mathbf{\Phi}_3[\alpha] - \overline{\mathbf{\Phi}}_3[\alpha])$ and $(\mathbf{\Phi}_3[\alpha] - \mathbf{J})$ have identical eigenspectra, and thus $\rho(\mathbf{\Phi}_3[\alpha] - \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T}) = \rho(\mathbf{\Phi}_3[\alpha] - \mathbf{J})$.

In Section 4.3.7 we show that $\rho(\mathbf{\Phi}_3[\alpha] - \mathbf{J}) < 1$ if $\alpha \in [0, -\theta_1^{-1})$, and thus the assumptions of the proposition, taken together with the analysis just conducted, verify that condition (JJ3) is also satisfied. Therefore, the limit $\lim_{t\to\infty} \mathbf{\Phi}_3[\alpha]^t = \overline{\mathbf{\Phi}}_3[\alpha] = \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T}$ exists, and $\overline{\mathbf{\Phi}}_3[\alpha]\mathbf{X}(0) = \mathbf{J}\mathbf{X}(0)$ for all $\mathbf{X}(0)$ defined in (4.112).

Proofs of parts (b) and (c). In the proof of Theorem 4.5 (see below), it is shown that $\rho(\mathbf{\Phi}_3[\alpha] - \mathbf{J}]) \geq -\alpha\theta_1$. Thus, if $\alpha > 0$ and $\theta_1 < 0$, then part (b) holds. The assumptions $\theta_1 + \theta_2 + \theta_3 = 1$, $\theta_3 \geq 1$, and $\theta_2 \geq 0$ imply that $\theta_1 \leq 0$, and by assumption, $\alpha \geq 0$. If $\alpha = 0$ or $\theta_1 = 0$, then the proposed predictive consensus scheme reduces to memoryless consensus with weight matrix $\mathbf{W}$ (and the statement follows directly from the results of [14,71]). Thus, part (b) of the proposition follows from the assumptions and the analysis in Theorem 4.5 below. By proving parts (a) and (b), we have verified the assumptions of Theorem 4.2 above. Applying the result of this Theorem, together with the equivalence of $\rho(\mathbf{\Phi}_3[\alpha] - \frac{1}{n}\mathbf{1}\mathbf{g}^\mathsf{T})$ and $\rho(\mathbf{\Phi}_3[\alpha] - \mathbf{J})$, gives the claim in part (c), thereby completing the proof. □

Proof of Theorem 4.5

In order to minimize the spectral radius of $\mathbf{\Phi}_3[\alpha]$ we need to know its eigenvalues. These can be calculated by solving the eigenvalue problem (4.124). We can multiply (4.126) by $\mathbf{u}_i^T$ on the left to obtain a quadratic equation that links the individual eigenvalues $\lambda_i(\mathbf{\Phi}_3[\alpha])$ and $\lambda_i(\mathbf{W}_3[\alpha])$:

$$\begin{aligned} \mathbf{u}_i^T(\lambda_i(\mathbf{\Phi}_3[\alpha])^2\mathbf{I} - \lambda_i(\mathbf{\Phi}_3[\alpha])\mathbf{W}_3[\alpha] - \alpha\theta_1\mathbf{I})\mathbf{u}_i = 0 \\ \lambda_i(\mathbf{\Phi}_3[\alpha])^2 - \lambda_i(\mathbf{W}_3[\alpha])\lambda_i(\mathbf{\Phi}_3[\alpha]) - \alpha\theta_1 = 0. \end{aligned} \tag{4.127}$$

Recall $\mathbf{\Phi}_3[\alpha]$ is a $2n \times 2n$ matrix, and so $\mathbf{\Phi}_3[\alpha]$ has, in general, $2n$ eigenvalues – twice as many as $\mathbf{W}_3[\alpha]$. These eigenvalues are the solutions of the quadratic (4.127), and are given by

$$\begin{aligned} \lambda_i^*(\mathbf{\Phi}_3[\alpha]) &= \frac{1}{2}\left(\lambda_i(\mathbf{W}_3[\alpha]) + \sqrt{\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1}\right) \\ \lambda_i^{**}(\mathbf{\Phi}_3[\alpha]) &= \frac{1}{2}\left(\lambda_i(\mathbf{W}_3[\alpha]) - \sqrt{\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1}\right). \end{aligned} \tag{4.128}$$

With these expressions for the eigenvalues of $\mathbf{\Phi}_3[\alpha]$, we are in a position to formulate the problem of minimizing the spectral radius of the matrix $(\mathbf{\Phi}_3[\alpha]-\mathbf{J})$, $\alpha^\star = \arg\min_\alpha \rho(\mathbf{\Phi}_3[\alpha]-\mathbf{J})$. It can be shown that this problem is equivalent to

$$\alpha^\star = \arg\min_{\alpha\geq 0} \rho(\mathbf{\Phi}_3[\alpha]-\mathbf{J}) \tag{4.129}$$

The simplest way to demonstrate this is to show that $\rho(\mathbf{\Phi}_3[\alpha]-\mathbf{J}) \geq \rho(\mathbf{\Phi}_3[0]-\mathbf{J})$ for any $\alpha < 0$. Indeed, by the definition of the spectral radius we have that $\rho(\mathbf{\Phi}_3[\alpha]-\mathbf{J}) \geq \lambda_2^*(\mathbf{\Phi}_3[\alpha])$ and $\rho(\mathbf{\Phi}_3[0]-\mathbf{J}) = \lambda_2(\mathbf{W})$ since $\mathbf{\Phi}_3[0]=\mathbf{W}$. Hence it is enough to demonstrate $\lambda_2^*(\mathbf{\Phi}_3[\alpha]) \geq \lambda_2(\mathbf{W})$. Consider the inequality $\lambda_2^*(\mathbf{\Phi}_3[\alpha]) - \lambda_2(\mathbf{W}) \geq 0$. Replacing $\lambda_i^*(\mathbf{\Phi}_3[\alpha])$ with its definition, (4.128), rearranging terms and squaring both sides gives $\alpha\theta_1 \geq \lambda_2(\mathbf{W})^2 - \lambda_2(\mathbf{W})\lambda_2(\mathbf{W}_3[\alpha])$. From the definition of $\mathbf{W}_3[\alpha]$ in (4.111), it follows that $\lambda_2(\mathbf{W}_3[\alpha]) = (1-\alpha+\alpha\theta_3)\lambda_2(\mathbf{W})+\alpha\theta_1$. Using this relation leads to the expression $\alpha(\theta_1+(\theta_3-\alpha)\lambda_2(\mathbf{W})^2 + \theta_1\lambda_2(\mathbf{W})) \geq 0$. Under our assumptions, we have $\theta_3 - 1 \geq 1$, $\theta_2 \geq 0$ and $\theta_1 \leq 0$. Thus $\theta_1 + (\theta_3-1)\lambda_2^2 + \theta_2\lambda_2 \leq \theta_1 + \theta_3 - 1 + \theta_2 = 0$. This implies that if $\alpha < 0$, the last inequality holds leading to $\lambda_2^*(\mathbf{\Phi}_3[\alpha]) \geq \lambda_2(\mathbf{W})$. Thus for any $\alpha < 0$ the spectral radius $\rho(\mathbf{\Phi}_3[\alpha]-\mathbf{J})$ cannot decrease, and so we may focus on optimizing over $\alpha \geq 0$.

Now, the proof of Theorem 4.5 boils down to examining how varying α affects the eigenvalues of $\mathbf{\Phi}_3[\alpha]$ on a case-by-case basis. We first show that the first eigenvalues, $\lambda_1^*(\mathbf{\Phi}_3[\alpha])$ and $\lambda_1^{**}(\mathbf{\Phi}_3[\alpha])$, are smaller than all the others. Then, we demonstrate that the second eigenvalues, $\lambda_2^*(\mathbf{\Phi}_3[\alpha])$ and $\lambda_2^{**}(\mathbf{\Phi}_3[\alpha])$, dominate all other pairs, $\lambda_j^*(\mathbf{\Phi}_3[\alpha])$ and $\lambda_j^{**}(\mathbf{\Phi}_3[\alpha])$, for $j > 2$, allowing us to focus on the second eigenvalues, from which the proof follows. Along the way, we establish conditions on α which guarantee stability of the proposed two-tap predictive consensus methodology.

To begin, we reformulate the optimization problem in terms of the eigenvalues of $\mathbf{\Phi}_3[\alpha]$. We first consider $\lambda_1^*(\mathbf{\Phi}_3[\alpha])$ and $\lambda_1^{**}(\mathbf{\Phi}_3[\alpha])$. Substituting $\lambda_1(\mathbf{W}_3[\alpha]) = (1-\alpha+\alpha\theta_3)+\alpha\theta_2$ we obtain the relationship $\sqrt{\lambda_1^2(\mathbf{W}_3[\alpha]) + 4\alpha\theta_1} = |1+\alpha\theta_1|$ and using the condition $\theta_1 \leq 0$, we conclude that

$$\lambda_1^*(\mathbf{\Phi}_3[\alpha]), \lambda_1^{**}(\mathbf{\Phi}_3[\alpha]) = \begin{cases} 1, -\alpha\theta_1 & \text{if } \ 1+\alpha\theta_1 \geq 0 \Rightarrow \alpha \leq -\theta_1^{-1} \\ -\alpha\theta_1, 1 & \text{if } \ 1+\alpha\theta_1 < 0 \Rightarrow \alpha > -\theta_1^{-1}. \end{cases}$$

We note that $\alpha > -\theta_1^{-1}$ implies $|\lambda_1^{**}(\mathbf{\Phi}_3[\alpha])| > 1$, leading to divergence of the linear recursion involving $\mathbf{\Phi}_3[\alpha]$, and thus conclude that the potential solution is restricted to the

range $\alpha \leq -\theta_1^{-1}$. Focusing on this setting, we write $\lambda_1^*(\mathbf{\Phi}_3[\alpha]) = 1$ and $\lambda_1^{**}(\mathbf{\Phi}_3[\alpha]) = -\alpha\theta_1$. We can now reformulate the problem (4.129) in terms of the eigenvalues of $\mathbf{\Phi}_3[\alpha]$:

$$\alpha^\star = \arg\min_{\alpha \geq 0} \max_{i=1,2,\dots n} \mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] \tag{4.130}$$

where

$$\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] = \begin{cases} |\lambda_1^{**}(\mathbf{\Phi}_3[\alpha])|, & i = 1 \\ \max(|\lambda_i^*(\mathbf{\Phi}_3[\alpha])|, |\lambda_i^{**}(\mathbf{\Phi}_3[\alpha])|) & i > 1. \end{cases}$$

We now state a lemma that characterizes the functions $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})]$.

Lemma 4.2. *Under the assumptions of Theorem 4.5,*

$$\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] = \begin{cases} \alpha^{1/2}(-\theta_1)^{1/2} & \text{if } \alpha \in [\alpha_i^*, \theta_1^{-1}] \\ \frac{1}{2}\left(|\lambda_i(\mathbf{W}_3[\alpha])| + \sqrt{\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1}\right) & \text{if } \alpha \in [0, \alpha_i^*) \end{cases} \tag{4.131}$$

where

$$\alpha_i^* = \frac{-((\theta_3 - 1)\lambda_i(\mathbf{W})^2 + \theta_2\lambda_i(\mathbf{W}) + 2\theta_1) - 2\sqrt{\theta_1^2 + \theta_1\lambda_i(\mathbf{W})\left(\theta_2 + (\theta_3 - 1)\lambda_i(\mathbf{W})\right)}}{\left(\theta_2 + (\theta_3 - 1)\lambda_i(\mathbf{W})\right)^2} \tag{4.132}$$

Over the range $\alpha \in [0, -\theta_1^{-1}]$, $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] \geq \mathcal{J}_1[\alpha, \lambda_1(\mathbf{W})]$ *for* $i = 2, 3, \dots, n$.

Proof. For $i = 2, 3, \dots n$, the eigenvalues $\lambda_i^*(\mathbf{\Phi}_3[\alpha])$ and $\lambda_i^{**}(\mathbf{\Phi}_3[\alpha])$ can admit two distinct forms; when the expression under the square root in (4.128) is less then zero, the respective eigenvalues are complex, and when this expression is positive, the eigenvalues are real. In the region where the eigenvalues are complex,

$$\begin{aligned} \max(|\lambda_i^*(\mathbf{\Phi}_3[\alpha])|, |\lambda_i^{**}(\mathbf{\Phi}_3[\alpha])|) &= \frac{1}{2}\left[\lambda_i(\mathbf{W}_3[\alpha])^2 + \imath^2\left(\sqrt{\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1}\right)^2\right]^{1/2} \\ &= \alpha^{1/2}(-\theta_1)^{1/2}. \end{aligned} \tag{4.133}$$

We note that (4.133) is a strictly increasing function of α. Recalling that $\lambda_i(\mathbf{W}_3[\alpha]) = (1 + \alpha(\theta_3 - 1))\lambda_i(\mathbf{W}) + \alpha\theta_2$ and solving the quadratic $\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1 = 0$, we can identify region, $[\alpha_i^*, \alpha_i^{**}]$, where the eigenvalues are complex. The upper boundary of this

region is

$$\alpha_i^{**} = \frac{-((\theta_3-1)\lambda_i(\mathbf{W})^2 + \theta_2\lambda_i(\mathbf{W}) + 2\theta_1) + 2\sqrt{\theta_1^2 + \theta_1\lambda_i(\mathbf{W})\left(\theta_2 + (\theta_3-1)\lambda_i(\mathbf{W})\right)}}{\left(\theta_2 + (\theta_3-1)\lambda_i(\mathbf{W})\right)^2} \tag{4.134}$$

Relatively straightforward algebraic manipulation of (4.132) and (4.134) leads to the following conclusion: if $\lambda_i(\mathbf{W}) \in [-1,1]$, $\theta_2 \geq 0$ and $\theta_3 \geq 1$, then $0 \leq \alpha_i^* \leq -\theta_1^{-1} \leq \alpha_i^{**}$. This implies that (4.133) holds in the region $[\alpha_i^*, -\theta_1^{-1}]$.

On the interval $\alpha \in [0, \alpha_i^*)$, the expression under the square root in (4.128) is positive, and the corresponding eigenvalues are real. Thus,

$$\begin{aligned}
&\max(|\lambda_i^*(\mathbf{\Phi}_3[\alpha])|, |\lambda_i^{**}(\mathbf{\Phi}_3[\alpha])|) \\
&\qquad = \frac{1}{2}\begin{cases} \left|\lambda_i(\mathbf{W}_3[\alpha]) + \sqrt{\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1}\right| & \text{if } \lambda_i(\mathbf{W}_3[\alpha]) \geq 0 \\ \left|-\lambda_i(\mathbf{W}_3[\alpha]) + \sqrt{\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1}\right| & \text{if } \lambda_i(\mathbf{W}_3[\alpha]) < 0, \end{cases}
\end{aligned}$$

or equivalently, $\max(|\lambda_i^*(\mathbf{\Phi}_3[\alpha])|, |\lambda_i^{**}(\mathbf{\Phi}_3[\alpha])|) = \frac{1}{2}\left(|\lambda_i(\mathbf{W}_3[\alpha])| + \sqrt{\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1}\right)$. These results establish the expression for $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})]$ in the lemma.

It remains to establish that $\mathcal{J}_1[\alpha, \lambda_1(\mathbf{W})]$ is less than all other $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})]$ in the region $\alpha \in [0, -\theta_1^{-1}]$. In the region $\alpha \in [\alpha_i^*, -\theta_1^{-1}]$, we have $-\alpha\theta_1^{-1} \leq 1$, implying that $\alpha^{1/2}(-\theta_1)^{1/2} \geq -\alpha\theta_1 = \mathcal{J}_1[\alpha, \lambda_1(\mathbf{W})]$. In the region $\alpha \in [0, \alpha_i^*)$, note that $\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1 > 0 \Rightarrow |\lambda_i(\mathbf{W}_3[\alpha])| \geq 2(-\alpha\theta_1)^{1/2}$, which implies that

$$\begin{aligned}
\frac{1}{2}\left(|\lambda_i(\mathbf{W}_3[\alpha])| + \sqrt{\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1}\right) &\geq \frac{1}{2}\left(2(-\alpha\theta_1)^{1/2} + 0\right) \\
&\geq (-\alpha\theta_1)^{1/2} \geq -\alpha\theta_1 = \mathcal{J}_1[\alpha, \lambda_1(\mathbf{W})], \quad (4.135)
\end{aligned}$$

thereby establishing the final claim of the lemma. □

The previous lemma indicates that we can remove $\mathcal{J}_1[\alpha, \lambda_1(\mathbf{W})]$ from (4.130), leading to a simpler optimization problem, $\alpha^\star = \arg\min_{\alpha \geq 0} \max_{i=2,3,\ldots n} \mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})]$. The following lemma establishes that we can simplify the optimization even further and focus solely on $\mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$.

Lemma 4.3. *Under the assumptions of Theorem 4.5,* $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] \leq \mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$ *and* $\alpha_i^*[\lambda_i(\mathbf{W})] \leq \alpha_2^*[\lambda_2(\mathbf{W})]$ *for* $i = 3, 4, \ldots, n$ *over the range* $\alpha \in [0, -\theta_1^{-1}]$.

Proof. Consider the derivative of $\alpha_i^*[\lambda_i(\mathbf{W})]$ in the range $\lambda_i(\mathbf{W}) \in [0,1]$:

$$\frac{\partial \alpha_i^*[\lambda_i(\mathbf{W})]}{\partial \lambda_i(\mathbf{W})} = \frac{1}{\left(\theta_2 + (\theta_3 - 1)\,\lambda_i(\mathbf{W})\right)^3} \times \Bigg[\left[4\theta_1 (\theta_3 - 1) - \theta_2 \left(\theta_2 + (\theta_3 - 1)\,\lambda_i(\mathbf{W})\right)\right] + \frac{\theta_1 \left(-\theta_2^2 + 4\theta_1 (\theta_3 - 1) + \theta_2 (\theta_3 - 1)\,\lambda_i(\mathbf{W}) + 2\,(\theta_3 - 1)^2\,\lambda_i(\mathbf{W})^2\right)}{\sqrt{\theta_1 \left(\theta_1 + \lambda_i(\mathbf{W})\left(\theta_2 + (\theta_3 - 1)\,\lambda_i(\mathbf{W})\right)\right)}} \Bigg]$$

It is clear that the multiplier outside the square brackets in the first line above is positive in the range $\lambda_i(\mathbf{W}) \in [0,1]$. Furthermore, the first summand is negative. Under the conditions $\theta_2 \geq 0$, $\theta_3 \geq 1$, it can be established that the second summand is positive and exceeds the first summand in magnitude. We conclude that the derivative is positive, and thus $\alpha_i^*[\lambda_i(\mathbf{W})]$ is an increasing function over $\lambda_i(\mathbf{W}) \in [0,1]$. This implies that $\alpha_i^*[\lambda_i(\mathbf{W})] \leq \alpha_2^*[\lambda_2(\mathbf{W})]$ for any $\lambda_i \geq 0$.

Algebraic manipulation of (4.132) leads to the conclusion that $\alpha_i^*[-\lambda_i(\mathbf{W})] \leq \alpha_i^*[\lambda_i(\mathbf{W})]$ for $\lambda_i(\mathbf{W}) \in [0,1]$. This implies that for negative λ_i, we have $\alpha_i^*[-\lambda_i(\mathbf{W})] \leq \alpha_i^*[\lambda_i(\mathbf{W})] \leq \alpha_2^*[\lambda_2(\mathbf{W})]$. We have thus shown that $\alpha_i^*[\lambda_i(\mathbf{W})] \leq \alpha_2^*[\lambda_2(\mathbf{W})]$ for any $3 \leq i \leq n$ under the assumption $|\lambda_n(\mathbf{W})| \leq \lambda_2(\mathbf{W})$.

Next we turn to proving that $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] \leq \mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$ for any $3 \leq i \leq n$. We consider this problem on three distinct intervals: $\alpha \in [0, \alpha_i^*[\lambda_i(\mathbf{W})])$, $\alpha \in [\alpha_i^*[\lambda_i(\mathbf{W})], \alpha_2^*[\lambda_2(\mathbf{W})])$ and $\alpha \in [\alpha_2^*[\lambda_2(\mathbf{W})], -\theta_1^{-1}]$. From the condition $\alpha_i^*[\lambda_i(\mathbf{W})] \leq \alpha_2^*[\lambda_2(\mathbf{W})]$ and (4.131) it is clear that on the interval $\alpha \in [\alpha_2^*[\lambda_2(\mathbf{W})], -\theta_1^{-1}]$ we have $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] = \mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})] = \alpha^{1/2}(-\theta_1)^{1/2}$. On the interval $\alpha \in [\alpha_i^*[\lambda_i(\mathbf{W})], \alpha_2^*[\lambda_2(\mathbf{W})])$ we have $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] = \alpha^{1/2}(-\theta_1)^{1/2}$ and $\mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})] = \frac{1}{2}\left(|\lambda_i(\mathbf{W}_3[\alpha])| + \sqrt{\lambda_i(\mathbf{W}_3[\alpha])^2 + 4\alpha\theta_1}\right)$. From (4.135), we see that $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] \leq \mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$.

On the first interval $\alpha \in [0, \alpha_i^*[\lambda_i(\mathbf{W})])$, we examine the derivative of $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})]$ w.r.t. $\lambda_i(\mathbf{W})$:

$$\frac{\partial \mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})]}{\partial \lambda_i(\mathbf{W})} = \frac{1 + \alpha(\theta_3 - 1)}{2} \left(\frac{\lambda_i(\mathbf{W}) + \alpha\left(\theta_2 + (\theta_3 - 1)\,\lambda_i(\mathbf{W})\right)}{\sqrt{\left(\lambda_i(\mathbf{W}) + \alpha\left(\theta_2 + (\theta_3 - 1)\,\lambda_i(\mathbf{W})\right) - 4\alpha\left(\theta_2 + \theta_3 - 1\right)\right)^2}} + \operatorname{sgn}\left[\lambda_i(\mathbf{W}) + \alpha\left(\theta_2 + (\theta_3 - 1)\,\lambda_i(\mathbf{W})\right)\right] \right) \tag{4.136}$$

We observe that the multiplier $\frac{1+\alpha(\theta_3-1)}{2}$ is positive, and the expression under the square root is positive because $\alpha \in [0, \alpha_i^*[\lambda_i(\mathbf{W})])$. Additionally, $\lambda_i(\mathbf{W}) + \alpha\left(\theta_2 + (\theta_3 - 1)\,\lambda_i(\mathbf{W})\right) \geq 0$ under the assumption $\lambda_i(\mathbf{W}) \geq 0$ and $\theta_2 \geq 0$, $\theta_3 \geq 1$. Thus $\frac{\partial}{\partial \lambda_i(\mathbf{W})}\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] \geq 0$ for any $\lambda_i(\mathbf{W}) \geq 0$ and we have $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] \leq \mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$ for any $0 \leq \lambda_i(\mathbf{W}) \leq \lambda_2(\mathbf{W})$. Finally, we note from (4.131) that $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})]$ is an increasing function of $|\lambda_i(\mathbf{W}_3[\alpha])| = |(1 + \alpha(\theta_3 - 1))\lambda_i(\mathbf{W}) + \alpha\theta_2|$. Thus, to show that $\mathcal{J}_i[\alpha, -\lambda_i(\mathbf{W})] \leq \mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})]$ for $0 \leq \lambda_i(\mathbf{W}) \leq \lambda_2(\mathbf{W})$ it is sufficient to show that $|-(1 + \alpha(\theta_3 - 1))\lambda_i(\mathbf{W}) + \alpha\theta_2| \leq |(1 + \alpha(\theta_3 - 1))\lambda_i(\mathbf{W}) + \alpha\theta_2|$. Under our assumptions, we have

$$\begin{aligned}|(1 + \alpha(\theta_3 - 1))\lambda_i(\mathbf{W}) + \alpha\theta_2|^2 - |-(1 + \alpha(\theta_3 - 1))\lambda_i(\mathbf{W}) + \alpha\theta_2|^2 \\ = 4(1 + \alpha(\theta_3 - 1))\lambda_i(\mathbf{W})\alpha\theta_2 \geq 0. \qquad (4.137)\end{aligned}$$

This implies that $\mathcal{J}_i[\alpha, \lambda_i(\mathbf{W})] \leq \mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$ on the interval $\alpha \in [0, \alpha_i^*[\lambda_i(\mathbf{W})])$, indicating that the condition applies on the entire interval $\alpha \in [0, -\theta_1^{-1}]$, which is what we wanted to show. □

The remainder of the proof of Theorem 4.5 proceeds as follows. From Lemmas 4.2 and 4.3, the optimization problem (4.114) simplifies to: $\alpha^\star = \arg\min_{\alpha \geq 0} \mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$. We shall now show that α_2^* is a global minimizer of this function. Consider the derivative of $\mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$ w.r.t. α on $[0, \alpha_2^*)$:

$$\begin{aligned}\frac{\partial}{\partial\alpha}\mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})] = \frac{2\theta_1 + \left(\theta_2 + (\theta_3 - 1)\,\lambda_2(\mathbf{W})\right)\left(\lambda_2(\mathbf{W}) + \alpha\left(\theta_2 + (\theta_3 - 1)\,\lambda_2(\mathbf{W})\right)\right)}{\sqrt{4\alpha\theta_1 + \left(\lambda_2(\mathbf{W}) + \alpha\left(\theta_2 + (\theta_3 - 1)\,\lambda_2(\mathbf{W})\right)\right)^2}} \\ + \left(\theta_2 + (\theta_3 - 1)\,\lambda_2(\mathbf{W})\right)\operatorname{sgn}\left[\lambda_2(\mathbf{W}) + \alpha\left(\theta_2 + (\theta_3 - 1)\,\lambda_2(\mathbf{W})\right)\right].\end{aligned}$$

Denote the first term in this sum by $\varphi_1(\lambda_2(\mathbf{W}), \alpha)$ and the second by $\varphi_2(\lambda_2(\mathbf{W}), \alpha)$. It can be shown that $|\varphi_1(\lambda_2(\mathbf{W}), \alpha)| \geq |\varphi_2(\lambda_2(\mathbf{W}), \alpha)|$ for any $\lambda_2(\mathbf{W}) \in [-1, 1]$ and $\alpha \in [0, \alpha_2^*)$ by directly solving the inequality. We conclude that the sign of the derivative on $\alpha \in [0, \alpha_2^*)$ is completely determined by the sign of $\varphi_1(\lambda_2(\mathbf{W}), \alpha)$ for $\lambda_2(\mathbf{W}) \in [-1, 1]$. On $\alpha \in [0, \alpha_2^*)$, the sign of $\varphi_1(\lambda_2(\mathbf{W}), \alpha)$ is determined by the sign of its numerator. The transition point for the numerator's sign occurs at:

$$\alpha^+ = -\frac{2\theta_1 + \lambda_2(\mathbf{W})(\theta_2 + (\theta_3 - 1)\lambda_2(\mathbf{W}))}{(\theta_2 + (\theta_3 - 1)\lambda_2(\mathbf{W}))^2},$$

and by showing that $\alpha^+ \geq -\theta_1^{-1}$, we can establish that this transition point is at or beyond α_2^*. This indicates that $\varphi_1(\lambda_2(\mathbf{W}), \alpha) \leq 0$ if $\alpha \in [0, \alpha_2^*)$. We observe that $\mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$ is non-increasing on $\alpha \in [0, \alpha_2^*)$ and nondecreasing on $\alpha \in [\alpha_2^*, -\theta_1^{-1})$ (as established in Lemma 4.2). We conclude that α_2^* is a global minimum of the function $\mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$, thereby proving Theorem 4.5 and establishing $\mathcal{J}_2[\alpha^\star, \lambda_2(\mathbf{W})] = |\lambda_2^*(\mathbf{\Phi}_3[\alpha^\star])| = \sqrt{-\alpha^\star\theta_1}$.

Note that the last argument also implies that $\mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})] \leq \lambda_2(\mathbf{W})$ on $\alpha \in [0, \alpha_2^*]$ and $\mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})] < 1$ on $\alpha \in (\alpha_2^*, -\theta_1^{-1})$ since $\mathcal{J}_2[\alpha, \lambda_2(\mathbf{W})]$ is non-increasing on the former interval, it is non-decreasing on the latter interval and $\mathcal{J}_2[-\theta_1^{-1}, \lambda_2(\mathbf{W})] = 1$. This fact demonstrates that the matrix $\mathbf{\Phi}_3[\alpha]$ is convergent if $\alpha \in [0, -\theta_1^{-1})$ in the sense that we have $\rho(\mathbf{\Phi}_3[\alpha] - \mathbf{J}) < 1$. $\square$

4.4 Summary

We have presented a general, predictor-based framework to improve the rate of convergence of distributed average consensus algorithm. The convergence properties of the general predictor based consensus acceleration framework were studied in the first section of this chapter and the existence of the convergent configuration of the proposed framework was established. For the convergent configurations of the proposed acceleration methodology the scaling properties of the ε-averaging time were studied and the upper bound on its asymptotic growth rate was established. For two special cases of the proposed methodology a detailed analysis was performed.

For the first case of memoryless weight matrix optimization the optimal value of the mixing parameter maximizing the asymptotic worst-case convergence rate was obtained. The associated convergence rate improvement was quantified. It turned out that in the memoryless case there is no guarantee that the improvement can be obtained. However, simulations revealed that in many scenarios the memoryless weight matrix optimization does yield a significant improvement within the proposed framework. The suboptimal choices of the value of the mixing parameter were proposed for the on-line optimization of the proposed algorithm in the memoryless case. Our simulation results indicated that with these simple suboptimal initializations the proposed algorithm provides significant convergence speed gain.

For the second, more interesting, case of the short node memory we have also derived the optimal value of mixing parameter resulting in the fastest asymptotic convergence rate. The

convergence rate analysis yielded the theoretical performance improvement guarantees. For a chain topology on n nodes, this lead to a factor of n improvement over standard consensus, and for a two-dimensional grid, our approach achieved a factor of $\sqrt{n}$ improvement, in terms of the number of iterations required to reach a prescribed level of accuracy. We believe that this result applies to the general class of distributed averaging algorithms using node state prediction, and shows that, even in its simplified form, accelerated consensus via prediction provides considerable processing gain. We concluded that this gain, measured as the ratio of the asymptotic averaging time of the non-accelerated and accelerated algorithms, grows with increasing network size. Numerical experiments confirmed our theoretical conclusions. The naive implementation of the proposed framework requires the knowledge of the network topology. To make the proposed acceleration framework more practical we proposed an on-line initialization scheme based on the distributed calculation of the spectral radius of the foundational weight matrix. Numerical experiments confirmed the feasibility of the on-line implementation of the accelerated algorithm with nearly optimal properties based on the proposed distributed on-line initialization scheme. Our numerical experiments also revealed the superior performance of the proposed approach with respect to several acceleration methodologies in the literature.

Chapter 5

Distributed Tracking with Communication Constraints

One of the major concerns in distributed WSN tracking is the maintenance of the appropriate tradeoff between tracking performance and network lifetime. If a centralized approach is used to process measurements from the sensors and in scenarios where Gaussian approximation is justifiable, one of the following well-established tracking algorithms can be used to obtain acceptable performance guarantees: the extended Kalman Filter [87,88], the Gaussian sum filter [89] or grid-based filters [90]. However, if better performance guarantees are required in the situation where the class of approximated dynamics and/or observation models is substantially non-linear and non-Gaussian, different particle filter based trackers can be used [91]. In WSN applications, there are two major disadvantages: particle filters are generally more computationally demanding [92], and communication of measurements or a particle filter representation to the fusion center can require the transfer of the large volumes of data, which is often undesirable [65]. Centralization introduces a single point of failure and can lead to high, unevenly distributed energy consumption because of the heavy communication cost involved in transmitting the data to the fusion center.

Distributed algorithms, such as the distributed particle filtering algorithms proposed in [93, 94], address the aforementioned problems. These algorithms decentralize the computation or communication so that a single fusion center is not required. Multiple particle filters run concurrently at different sensor nodes and compressed data or approximate filtering distributions are shared between them. These distributed algorithms, while mitigating

some of the inherent problems of centralization, can be computationally expensive, because multiple nodes are required to perform computation throughout the entire tracking procedure.

In the collaborative distributed scenario, the node performing the particle filtering (the *leader* node) changes over time along with the associated subset of nodes performing sensing tasks. This scheme was proposed in [6, 57, 95] and refined in [96, 97]. In attempting to alleviate the communication cost of transmitting all particle values when the leader node is exchanged (which can involve thousands of bits), the filtering distribution is often more coarsely approximated, either by transmitting only a subset of the particles or by training a parametric model. In the next chapter we perform the approximation error analysis of the leader node scheme when coarse approximations of the filtering distribution are used during leader node exchanges.

The current chapter provides an overview of the collaborative WSN based tracking scheme (leader node scheme) and introduces particle filtering concept. It also outlines important ideas related to the analysis of the particle filter performance, and presents relevant material describing sample based and mixture based random approximation analysis principles.

5.1 Sensor Collaboration and in-Network Processing for Target Tracking

The collaborative signal and information processing (CSIP) framework was discussed in a series of papers [6, 57, 95]. It is based on adaptively activating (managing) clusters in a WSN to maintain the network lifetime/tracking accuracy trade-off. Sensor management strategies within this framework typically take into account a combination of factors including sensor utility functions and activation costs. In this section WSN sensing model and optimal Bayesian estimation is first reviewed to facilitate later exposition of the CSIP methodology.

5.1.1 CSIP Sensing Model and Optimal Bayesian Estimation

An abstract graph based WSN model suitable for analyzing CSIP framework can be specified as follows. A set of leader nodes (cluster heads) is defined as a set of vertices

$\mathfrak{L} = \{1, 2, \ldots, L\}$. Similarly, a set of sensor nodes is defined as $\mathfrak{S} = \{1, 2, \ldots, S\}$. It is often assumed that leader nodes have more advanced processing and communication capabilities and are thus responsible for performing signal processing and data routing operations while sensor nodes can measure certain physical quantities and transmit measurements to the associated leader node. However, in the case of homogenous WSN any sensor can potentially be included into the set of leader nodes and then $\mathfrak{L} \subseteq \mathfrak{S}$. The set of wireless links among members of leader node set $\mathfrak{L}$ and sensor node set $\mathfrak{S}$ forms the set of edges $\mathfrak{E}$ where a pair $(\ell, s) \in \mathfrak{E}$ if and only if nodes $\ell \in \mathfrak{L}$ and $s \in \mathfrak{S}$ are adjacent (connected by a direct wireless link). The underlying graph $G(\mathfrak{L}, \mathfrak{S}, \mathfrak{E})$ is thus induced by this connectivity structure.

Within the centralized WSN tracking framework all the nodes are active during the tracking task and optimal sequential Bayesian estimation is viable. Within this framework at every time instant $t \in \mathbb{N}$ sensor nodes collect measurements $y_t^{\mathfrak{S}} = \{y_t^s\}_{s=1}^S$ and route them to the fusion center. The fusion center then performs tracking operation based on the probabilistic model induced by the general Markovian state-space representation

$$X_t = f_t(X_{t-1}, \varrho_t), \tag{5.1}$$

$$Y_t^s = g_t^s(X_t, \zeta_t^s) \quad \forall s \in \mathfrak{S}. \tag{5.2}$$

Here (5.1) is the target dynamics equation with $X_t \in \mathbb{R}^{d_x}$ being the target state vector, $f_t : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ being the nonlinear diffusion map, and ϱ_t being the system (excitation) noise. The measurement equation (5.2) describes network-wide measurement process with $Y_t^s \in \mathbb{R}^{d_y^s}$ being the sth sensor measurement vector, $g_t^s : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y^s}$ being its nonlinear measurement function, and ζ_t^s being the random measurement noise.

Within the Bayesian framework, the above state-space model is represented in terms of underlying probability density functions and the optimal Bayes recursion is employed to track the posterior distribution of the state of a target. In particular, target dynamics equation (5.1) is completely characterized by the conditional probability density function (pdf) $p_{X_t|X_{t-1}}(x_t|x_{t-1})$ and the initial target pdf $p_{X_0}(x_0)$. The measurement equation, on the other hand, generates the probabilistic measurement model $p_{Y_t^{\mathfrak{S}}|X_t}(y_t^{\mathfrak{S}}|x_t)$. The following

network-wide Bayesian recursion (see e.g. Gordon et al. [98]):

$$p_{X_{t+1}|Y_{1:t}^{\mathfrak{S}}}(x_{t+1}|y_{1:t}^{\mathfrak{S}}) = \int p_{X_{t+1}|X_t}(x_{t+1}|x_t) p_{X_t|Y_{1:t}^{\mathfrak{S}}}(x_t|y_{1:t}^{\mathfrak{S}}) \mathrm{d}x_t, \qquad \text{Predict} \quad (5.3)$$

$$p_{X_{t+1}|Y_{1:t+1}^{\mathfrak{S}}}(x_{t+1}|y_{1:t+1}^{\mathfrak{S}}) = \frac{p_{Y_{t+1}^{\mathfrak{S}}|X_{t+1}}(y_{t+1}^{\mathfrak{S}}|x_{t+1}) p_{X_{t+1}|Y_{1:t}^{\mathfrak{S}}}(x_{t+1}|y_{1:t}^{\mathfrak{S}})}{\int p_{Y_{t+1}^{\mathfrak{S}}|X_{t+1}}(y_{t+1}^{\mathfrak{S}}|x_{t+1}) p_{X_{t+1}|Y_{1:t}^{\mathfrak{S}}}(x_{t+1}|y_{1:t}^{\mathfrak{S}}) \mathrm{d}x_{t+1}}, \qquad \text{Update} \quad (5.4)$$

specifies the optimal two-step rule for updating posterior pdfs. Here $y_{1:t}^{\mathfrak{S}}$ denotes the measurement sequence acquired by the entire network during time steps $1, \ldots, t$. The two-step predict/update structure of the rule naturally fits the state-space framework description of the target evolution and measurement acquisition. The prediction step identifies current likely positions of a target given the distribution obtained through the previous measurements by marginalizing over all the movements this target can make. The update step reduces the uncertainty induced by the random dynamics of the target and previous noisy measurements by incorporating the most recent set of measurements obtained by the WSN.

Although the centralized Bayesian approach is optimal in terms of tracking performance its drawbacks are clear. The fact that the entire network is kept running during tracking exercise inevitably induces network lifetime issues. The need to route the entire set of network measurements $y_t^{\mathfrak{S}}$ (which can be large in e.g. image processing applications) to the fusion center at every time step raises concerns related to scalability, inefficient bandwidth usage, and uneven power consumption. Besides, the optimal Bayesian recursion can be computed exactly for a relatively small subset of tracking problems (e.g. linear and Gaussian). Thus the design and analysis of approximate, distributed (localized) WSN based tracking strategies employing in-network processing techniques is necessary. In the next chapter we analyze the approximation performance of one such strategy, called Collaborative Signal and Information Processing (CSIP). In the rest of this chapter this strategy and analysis framework necessary for obtaining theoretical guaranties for its performance are reviewed.

5.1.2 Collaborative Bayesian Estimation in a Wireless Sensor Network

The CSIP tracking protocol alleviates the drawbacks of the optimal Bayesian approach identified in the previous section by activating only a subset of sensor nodes at any time instant. The CSIP sensing/tracking protocol can be summarized as follows. A user (sink)

initiates the tracking exercise by querying the nearest node in the WSN (node Q). The sink can ask WSN to periodically send updates about the estimate of the state of the target [95] and supply the WSN with its prior belief regarding the current position of the target. Node Q identifies the first subset of active sensor nodes according to the prior information about the location of the target and activates this subset by sending a corresponding request to the cluster head. After the initiation of the track, at every iteration, the current cluster head activates its cluster of sensor nodes (these sensors will be called *satellite sensors*), acquires information that they sense and updates the tracking statistics (e.g. posterior pdfs) by modeling the target dynamics and incorporating the current set of measurements. Based on the current tracking statistics of the target and the appropriate quality metric, current cluster head makes an assessment as to whether it is capable of tracking the target or if another cluster head (with its associated set of satellite sensors) could perform better. The cluster head hand-off is thus performed whenever necessary. At the same time, the current cluster head may send the tracking update to the sink.

In the CSIP approach the cluster head is often called the *leader node* since it leads the associated set of satellite nodes by performing complex signal processing and sensor management tasks. Moreover, the CSIP approach is often referred to as the "leader node" approach since the main idea behind CSIP is to use the localized in-network signal processing within the cluster head (leader node). Thus the terms "CSIP" and "leader node" will be used interchangeably in the rest of the thesis.

This tracking scenario is depicted in Fig. 5.1. A leader node (depicted by big circles) is responsible for performing local tracking operations (e.g. running a particle filter) based on the data acquired by the satellite nodes (depicted by small circles). The leader node fuses the data gathered by the satellite nodes in its neighborhood; and employing a sensor management (selection) routine performs a leader node hand-off when necessary. The hand-off (decision) rule is constructed in such a manner that the position of the leader node follows (on the average) the position of the region where most informative measurements can be acquired. In many situations this results in the leader node following the position of the target (depicted by the squares in Fig. 5.1 — the colors of the squares correspond to the colors of the corresponding active nodes).

In the ideal situation, assuming that the current leader node can transmit its entire posterior pdf to the next leader node (this is only possible if the pdf has finite dimensional parametric representation), the corresponding predict/update CSIP Bayes recursion can

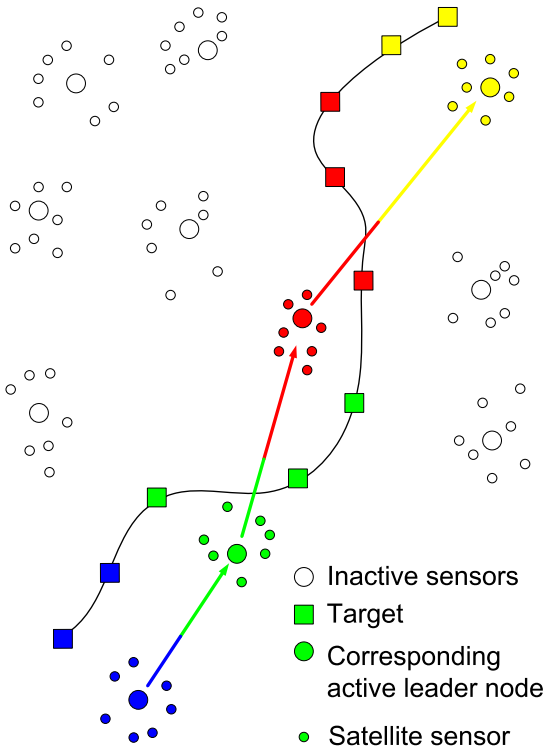


Fig. 5.1 The CSIP distributed filtering setting

be written as follows

$$p_{X_{t+1}|Y_{1:t}^{\mathfrak{S}_{1:t}}}(x_{t+1}|y_{1:t}^{\mathfrak{S}_{1:t}}) = \int p_{X_{t+1}|X_t}(x_{t+1}|x_t) p_{X_t|Y_{1:t}^{\mathfrak{S}_{1:t}}}(x_t|y_{1:t}^{\mathfrak{S}_{1:t}}) \mathrm{d}x_t, \tag{5.5}$$

$$p_{X_{t+1}|Y_{1:t+1}^{\mathfrak{S}_{1:t+1}}}(x_{t+1}|y_{1:t+1}^{\mathfrak{S}_{1:t+1}}) = \frac{p_{Y_{t+1}^{\mathfrak{S}_{t+1}}|X_{t+1}}(y_{t+1}^{\mathfrak{S}_{t+1}}|x_{t+1}) p_{X_{t+1}|Y_{1:t}^{\mathfrak{S}_{1:t}}}(x_{t+1}|y_{1:t}^{\mathfrak{S}_{1:t}})}{\int p_{Y_{t+1}^{\mathfrak{S}_{t+1}}|X_{t+1}}(y_{t+1}^{\mathfrak{S}_{t+1}}|x_{t+1}) p_{X_{t+1}|Y_{1:t}^{\mathfrak{S}_{1:t}}}(x_{t+1}|y_{1:t}^{\mathfrak{S}_{1:t}}) \mathrm{d}x_{t+1}}. \tag{5.6}$$

At every time step t every leader node $\ell_t \in \mathcal{L}$ only has access to a collection of satellite nodes in its local neighborhood, $\mathcal{N}_{[\ell_t]} = \{s \in \mathcal{S} : (s, \ell_t) \in \mathfrak{E}\}$ (cf. the definition of the neighborhood in Section 2.3). This fact is reflected in the recursion (5.5)–(5.6) through the introduction of the time-varying sequence of active sensor subsets (active neighborhoods) $\mathfrak{S}_{1:t} = \{\mathfrak{S}_i\}_{i=1}^t$ (cf. the centralized Bayes recursion (5.3)–(5.4)) such that $\mathfrak{S}_t = \mathcal{N}_{[\ell_t]}$. The set of measurements $y_{1:t}^{\mathfrak{S}_{1:t}}$ collected by the sequence of active neighborhoods is thus $y_{1:t}^{\mathfrak{S}_{1:t}} = \{y_i^{s_i} \in \mathbb{R}^{d_y^{s_i}} : 1 \leq i \leq t, s_i \in \mathfrak{S}_i\}$ a subset of the set of all the measurements $y_{1:t}^{\mathfrak{S}}$ that can potentially be acquired by the entire WSN.

Due to the local structure of the CSIP data acquisition and processing methodology, CSIP significantly alleviates the scalability, network lifetime, uneven power consumption, and inefficient bandwidth utilization issues associated with the centralized optimal Bayes approach. However, these benefits come at the cost of reduced tracking quality. The two most important factors influencing tracking quality reduction are as follows.

The first factor is the information loss: only a subset of sensors are activated at any given time-slot. Thus only a portion of the available information is acquired at any time instant potentially leading to a reduced tracking quality. In many situations these losses are small since the informativity of measurements is unevenly spread across the WSN. A small subset of sensors often provides most of the available information at a particular time instant. The influence of the first factor on the tracking performance is minimized by identifying the most informative subset of sensors $\mathfrak{S}_{t+1}$ to be activated for the next measurement [96]. The selection is generally based on an objective function that combines a utility function $\varphi_U(\cdot)$ measuring the utility of activating the subset $\mathfrak{S}_{t+1}$ and a cost function $\varphi_C(\cdot)$ measuring the cost of activating this subset [6]:

$$\Upsilon(p_{X_t|Y_{1:t}^{\mathfrak{S}_{1:t}}}(x_t|y_{1:t}^{\mathfrak{S}_{1:t}}), \mathfrak{S}_{t+1}) = \beta\varphi_U(p_{X_t|Y_{1:t}^{\mathfrak{S}_{1:t}}}(x_t|y_{1:t}^{\mathfrak{S}_{1:t}}), \mathfrak{S}_{t+1}) + (1-\beta)\varphi_C(\mathfrak{S}_{t+1}). \tag{5.7}$$

The cost function typically measures communication costs involved in activating subset

$\mathfrak{S}_{t+1}$, the utility function typically measures the information utility of incorporating measurements from sensors in the set $\mathfrak{S}_{t+1}$, and β determines the relative importance of utility and cost. Several information utility functions were proposed in [95]: Euclidean distance (nearest neighbor rule), Mahalanobis distance, and entropy based utility functions. It turns out that the performance of CSIP is best when entropy based measures of utility are used in the utility function. An example of the entropy based utility function is the mutual information $I(X_{t+1}, Y_{t+1}^{\mathfrak{S}_{t+1}}|y_{1:t}^{\mathfrak{S}_{1:t}})$. This utility function measures the amount of information that the set of measurements $Y_{t+1}^{\mathfrak{S}_{t+1}}$ (that has not yet been realized) from the subset $\mathfrak{S}_{t+1}$ can provide about the state X_{t+1} given the values $y_{1:t}^{\mathfrak{S}_{1:t}}$ of all the previous measurements. More complex utility functions incorporating multi-step selection strategies and energy constraints can be constructed [96]. The calculation of the MI utility $I(X_{t+1}, Y_{t+1}^{\mathfrak{S}_{t+1}}|y_{1:t}^{\mathfrak{S}_{1:t}})$ is often reasonably simple and is based on the knowledge of posterior $p_{X_t|Y_{1:t}^{\mathfrak{S}_{1:t}}}(x_t|y_{1:t}^{\mathfrak{S}_{1:t}})$ (or its approximation), target dynamics $p_{X_{t+1}|X_t}(x_{t+1}|x_t)$, and sensor characteristics $p_{Y_{t+1}^{\mathfrak{S}_{t+1}}|X_{t+1}}(y_{t+1}^{\mathfrak{S}_{t+1}}|x_{t+1})$. The MI based leader node selection rule can be formulated is follows:

$$\ell_{t+1} = \arg\max_{\ell \in \mathfrak{L}} I(X_{t+1}, Y_{t+1}^{\mathcal{N}_{[\ell]}}|y_{1:t}^{\mathfrak{S}_{1:t}}). \tag{5.8}$$

This criterion is known to work well when the estimation accuracy is the only objective [96].

The second factor influencing tracking quality reduction is the additional approximation errors: the hand-off of information to a new leader node is required whenever the leader changes. This involves either transmitting the tracking statistics or their approximations. Even assuming that the CSIP Bayes recursion can be approximated appropriately (e.g. using a particle filter), the volume of data to be transmitted during hand-offs can be too large. Thus additional approximations have to be made. The effect from these additional approximations has to be accounted for when the goal is to reach low bandwidth consumption (and communication cost).

In the next chapter this effect will be analyzed. We will consider two different hand-off settings. In the first setting a random subsample from the particle approximation of the filtering distribution is sent during leader node hand-off. This type of approximation will be called a *subsample approximation* or *non-parametric approximation*. In the second setting a parametric mixture estimated from the particle approximation of the filtering distribution is sent during leader node hand-off. This type of approximation will be called a *parametric approximation*.

5.2 Particle Filtering for Target Tracking

The calculation of the optimal Bayes recursion is not always possible in closed form. Therefore there is a need to approximate this calculation. One such approximate scheme is the particle filter [91]. A particle filter maintains a set of "particles" that are simply candidate state values of the system (for example, the position and velocity of the object). The filter evaluates how well individual particles correspond to the dynamic model and the set of observations, and assigns weights accordingly. The set of weighted particles provides a point-wise approximation to the filtering distribution, which represents the posterior probability of the state. This approximation allows one to form estimates of the state values and hence track the state.

The major idea behind the particle filter[1] is the following Monte Carlo (MC) approximation of an arbitrary pdf [101]:

$$\widehat{p}_N(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x-\xi^i), \tag{5.9}$$

where $\delta(\cdot)$ is the Dirac delta function and $\{\xi^i\}_{i=1}^N$ is a set of independent and identically distributed (i.i.d.) samples from $p(x)$. For a measurable function $\varphi : \mathcal{X} \to \mathbb{R}$ the empirical expectation

$$\int \varphi(x)\widehat{p}_N(x)\mathrm{d}x = \frac{1}{N}\sum_{i=1}^{N}\varphi(\xi^i), \tag{5.10}$$

converges to the true expectation almost surely as $N \to \infty$ [102]:

$$\int \varphi(x)\widehat{p}_N(x)\mathrm{d}x \to \int \varphi(x)p(x)\mathrm{d}x \quad \text{a.s.} \tag{5.11}$$

motivating the use of empirical MC sums as consistent random approximations of the true integrals in the Bayes recursion (5.5)–(5.6). The underlying assumption behind the MC approximation is that the pdf $p(x)$ is known (or can be sampled from). In the practical situation, this implies knowing the posterior distribution $p_{X_t|Y_{1:t}}(x_t|y_{1:t})$, the quantity

[1] Particle filters go by several names reflecting different aspects of the sample based sequential filtering density approximation and propagation: bootstrap filter [98], sequential Monte Carlo methods [91], condensation [99], and interacting particle systems [100].

that one has to calculate. To bypass this difficulty a statistical technique known as *importance sampling* is used [103]. The application of this technique only requires that the pdf $p_{X_t|Y_{1:t}}(x_t|y_{1:t})$ can be evaluated, point-wise, at candidate state locations (particle locations). This is a much weaker requirement than the knowledge of the pdf, or the ability to sample from it directly. Thus to be able to apply the MC approximation methodology, one has to assume that $p_{X_{0:t}|Y_{1:t}}(x_{0:t}|y_{1:t})$ can be approximated in the form:

$$p_{X_{0:t}|Y_{1:t}}(x_{0:t}|y_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta(x - \xi_{0:t}^i), \tag{5.12}$$

and importance sampling can be used to obtain the samples $\xi_{0:t}^i$. The importance sampling can be applied to approximate a target density, $p(x)$, if the target density can be evaluated point-wise and there exists a pdf, $q(x)$, known as the *proposal* density that is easy to sample from and the support of the proposal density contains that of the target density. Then the empirical approximation

$$p(x) \approx \sum_{i=1}^{N} w^i \delta(x - \xi^i), \tag{5.13}$$

can be obtained by sampling from $q(x)$ and setting the weights as $w^i = p(\xi^i)/q(\xi^i)$ (weights w^i are often referred to as *importance weights*). In the context of approximating Bayes recursion the proposal density has the general form [101] $q_{X_t|X_{0:t-1},Y_{1:t}}(x_t|x_{0:t-1}, y_{1:t})$ and the importance weights in (5.12) become:

$$w_t^i = w_{t-1}^i \frac{p_{Y_t|X_t}(y_t|\xi_t^i) p_{X_t|X_{t-1}}(\xi_t^i|\xi_{t-1}^i)}{q_{X_t|X_{0:t-1},Y_{1:t}}(\xi_t^i|\xi_{0:t-1}^i, y_{1:t})}. \tag{5.14}$$

In the important special case when only the marginal filtered density $p_{X_t|Y_{1:t}}(x_t|y_{1:t})$ is required at every time step t, the optimal importance density becomes simpler [104]:

$$q_{X_t|X_{0:t-1},Y_{1:t}}(x_t|x_{0:t-1}, y_{1:t}) = q_{X_t|X_{t-1},Y_t}(x_t|x_{t-1}, y_t).$$

The expression for importance weights changes accordingly:

$$w_t^i = w_{t-1}^i \frac{p_{Y_t|X_t}(y_t|\xi_t^i) p_{X_t|X_{t-1}}(\xi_t^i|\xi_{t-1}^i)}{q_{X_t|X_{t-1},Y_t}(\xi_t^i|\xi_{t-1}^i, y_t)}. \tag{5.15}$$

The algorithm based on the above weight update is known as the sequential importance sampling algorithm. The sequential importance sampling algorithm is known to suffer from the *degeneracy* problem: the unconditional variance of particle weights grows over time indicating that eventually only a small subset of candidate states represents the actual evolution of the observed dynamical system [105]. To overcome this issue either the optimal importance density with zero conditional weight variance [104] or the resampling approach [106] can be used.

The structure of the optimal importance density is known to be:

$$q_{X_t|X_{t-1},Y_t}(x_t|\xi_{t-1}^i, y_t) = \frac{p_{Y_t|X_t}(y_t|x_t) p_{X_t|X_{t-1}}(x_t|\xi_{t-1}^i)}{p_{Y_t|X_{t-1}}(y_t|\xi_{t-1}^i)}. \tag{5.16}$$

The expression for the associated importance weights then contains the integral of the numerator above:

$$w_t^i = w_{t-1}^i \int p_{Y_t|X_t}(y_t|x_t) p_{X_t|X_{t-1}}(x_t|\xi_{t-1}^i) \mathrm{d}x_t, \tag{5.17}$$

which is an obvious drawback since the integral can only be evaluated in special cases [101, 107]. In other cases, suboptimal sampling techniques have to be used. The examples of typical suboptimal sampling techniques are sampling from the prior [108] and the auxiliary variable sampling [109]. A common approach of avoiding degeneracy problem for suboptimal choices of the proposal density is resampling [98]. For example, in the case of sampling from prior, $q_{X_t|X_{t-1},Y_t}(x_t|\xi_{t-1}^i, y_t) = p_{X_t|X_{t-1}}(x_t|\xi_{t-1}^i)$, the expression for the unnormalized importance weights is particularly simple:

$$w_t^i = w_{t-1}^i p_{Y_t|X_t}(y_t|\xi_t^i). \tag{5.18}$$

Before resampling the importance weights are normalized, $\widetilde{w}_t^i = w_t^i / \sum_{j=1}^N w_t^j$ and the resampling is performed from this weighted sample such that the new resampled sample

$\{\widetilde{\xi}_t^i\}_{i=1}^N$ satisfies

$$\mathbb{P}\{\widetilde{\xi}_t^j = \xi_t^i\} = \widetilde{w}_t^i. \tag{5.19}$$

The particle filter that employs resampling at every step is generally referred to as the sampling importance resampling particle filter [98]. In the sampling importance resampling filter the normalized weights are simply

$$\widetilde{w}_t^i = \frac{p_{Y_t|X_t}(y_t|\xi_t^i)}{\sum_{j=1}^N p_{Y_t|X_t}(y_t|\xi_t^j)}, \tag{5.20}$$

since the resampled weights have equal weight $1/N$. Other variants of resampling based filters employ resampling step only occasionally, when the so-called *effective* sample size (the number of particles with significant weights) drops below certain threshold [106].

In the next section the important results from the particle filter stability analysis literature will be reviewed and the general framework for analyzing the performance of particle filters will be presented.

5.3 Stability Analysis of Particle Filtering Algorithms and Feynman-Kac Formulae

The analysis of approximation error propagation and stability of non-linear Markov filters has been an active research area for several decades. In [110] Kunita studied the asymptotic behavior of the error and stability of the filter that has an ergodic signal transition semigroup with respect to the initial distribution. Ocone and Pardoux [111] addressed the stability of linear filters with respect to a non-Gaussian initial condition and examined the stability of non-linear filters in the case where the signal diffusion is convergent. The important conclusion drawn by Ocone and Pardoux based on results in [110, 111] is that if the signal diffusion is stable with respect to its initial condition then the optimal filter inherits this property and it is also stable with respect to the initial condition.

Although interesting, the results in [110, 111] address the optimal filtering scenario, and more relevant to our study is the analysis of approximately optimal filters (especially particle filters). Important results concerning the stability of particle filters have been developed over the past decade [100, 112–119].

The Feynman-Kac semigroup approach to the stability analysis of particle filters has been described and developed by Del Moral, Miclo and Guionnet in [112–114]. The authors study the stability properties of general non-linear Feynman-Kac semigroups under a variety of assumptions. The Dobrushin contraction coefficient of the underlying Markov chain plays a central role in the analysis. In [113], Del Moral and Miclo formulate the conditions for the exponential asymptotic stability of the Feynman-Kac semigroup and bound the Lyapunov constant and Dobrushin coefficient. One of the applications of these results is a time-uniform upper bound on the error of interacting particle systems. In [114], Del Moral provides an extensive analysis of the properties of Feynman-Kac semigroups. His analysis forms the basis for our study in the next chapter, particularly in the case of the subsampling approximation leader node particle filter.

Stability analysis for particle filters is frequently built on relatively strong assumptions about the mixing and ergodicity properties of the underlying Markov transitions of the signal (target state). There have been some efforts to relax these types of assumptions. In [116, 117], Le Gland and Oudjane study the stability and convergence rates for particle filters using the Hilbert projective metric. In [116], they relax the signal mixing assumptions by employing a specific, "robust" particle filter architecture with truncated likelihood functions. In [117], the mixing assumption is applied not to the Markov kernel governing signal diffusion, but to the non-negative kernel that governs the evolution of the particle filter. This kernel combines the effects of the Markov transitions and the likelihood potentials, so mixing behavior can arise from either of these two components.

The papers cited thus far addressed the analysis of particle filters with fixed population size (number of particles). In the subsampling approximation leader node particle filter, the number of particles varies over time. Crisan et al. examine the stability of branching and interacting particle systems in [100]; in these systems the population size also varies, because at each time step a particle generates a random number of offspring. The population size forms a positive integer-valued martingale with respect to the filtration and the properties of the resulting particle filter depend on the initial number of particles. The variation in the number of particles is clearly very different from that of the subsampling approximation leader node particle filter, so the results are not directly applicable.

Thus far we have discussed previous work that has addressed particle filter stability when the error arises due to the sampling approximation. The sampling error is dependent on the resampling schemes, and Douc et al. have provided theoretical results that allow

various resampling schemes to be compared [119]. Other work has considered additional sources of error. Vaswani et al. analyzed the performance of a particle filter in the case of signal model mismatch (when the true underlying Markov transitions differ from the model used to update the filter) [118]. They showed, using the same assumptions as in [117], that the particle filter is stable if the mismatch persists for only a finite interval of time.

Le Gland et al. propose and analyze the kernel-based *regularized particle filter* in [115, 117], and this work is the most closely related to our study of the parametric approximation particle filter. From an algorithmic standpoint, there are also similarities with the Gaussian sum particle filter [120], but the theoretical analysis of this filter is less developed. The regularized particle filters described in [115, 117] incorporate a step in which the N-sample point-wise density approximation is replaced by a continuous density approximation, using a kernel-based density estimation approach. During resampling, N particles are generated by sampling from this continuous density. The practical benefit of this approach is the increase in the diversity of the particle system, eliminating the potential for degeneracy and improving the stability of the algorithm. Le Gland et al. provide uniform convergence results for the regularized particle filters. Although there is some similarity to the parametric approximation particle filter we analyze, the purpose of the approximation is very different. It is not performed intermittently to reduce computation or communication cost, but rather is performed every time step with a complex model (N components).

5.3.1 Feynman-Kac Formulae

In order to conduct stability analysis in the next chapter, we need to introduce slightly more rigorous mathematical notations. Let $(E_t, \mathcal{E}_t)$, $t \in \mathbb{N}$ be a sequence of measurable spaces. The target state vector evolves according to a non-homogeneous (discrete-time) Markov chain X_t with transitions M_{t+1} from E_t into E_{t+1}. We denote by $X'_t = X_{[0:t]}$ the historical path process associated with X_t, and use M'_t to denote the Markov transitions of the path process. Associated with a measurable space of the form $(E, \mathcal{E})$ is a set of probability measures $\mathcal{P}(E)$ and the Banach space of bounded functions $\mathcal{B}_b(E)$ with supremum norm:

$$||h|| = \sup_{x \in E} |h(x)|. \tag{5.21}$$

We define a convex set $\mathrm{Osc}_1(E)$ of $\mathcal{E}$-measurable test functions with finite oscillations:

$$\mathrm{osc}(h) = \sup(|h(x) - h(y)|;\ x, y \in E) \tag{5.22}$$

$$\mathrm{Osc}_1(E) = \{h : \mathrm{osc}(h) \leq 1\} \tag{5.23}$$

For any $h \in \mathcal{B}_b(E)$ it is also possible to define the following:

$$||h||_{\mathrm{osc}} = ||h|| + \mathrm{osc}(h). \tag{5.24}$$

In order to simplify the representation, we define for a measure $\mu \in \mathcal{P}(E)$,

$$\mu(h) = \int_E h(x)\mu(dx)$$

and for the Markov kernel from $(E_{i-1}, \mathcal{E}_{i-1})$ to $(E_i, \mathcal{E}_i)$:

$$(\mu_{i-1}M_i)(A_i) = \int_{E_{i-1}} \mu_{i-1}(\mathrm{d}x_{i-1})M_i(x_{i-1}, A_i).$$

Thus the composite integral operator from $(E_i, \mathcal{E}_i)$ to $(E_t, \mathcal{E}_t)$, $M_{i,t} = M_{i+1} \ldots M_t$, has the form:

$$(M_{i+1} \ldots M_t)(x_i, \mathrm{d}x_t) = \int_{E_{[i+1:t-1]}} M_{i+1}(x_i, \mathrm{d}x_{i+1}) \ldots M_t(x_{t-1}, \mathrm{d}x_t).$$

In the next chapter we adopt the methodology developed in [114] to analyze the behavior of filtering distributions arising from (5.1) and (5.2). This methodology involves representing the particle filter as an N-particle approximation of a Feynman-Kac model. In the remainder of this section, we describe how this representation is performed; for a much more detailed description and discussion, please refer to [114].

The evolution of the unconditional signal distribution in (5.1) is completely defined by the Markov transition kernel $M(\cdot, \cdot)$ and the initial signal distribution μ_0:

$$\Pr\{X_t \in \mathrm{d}x_t | X_{t-1} = x_{t-1}\} = M_t(x_{t-1}, \mathrm{d}x_t) \tag{5.25}$$

According to (5.25), the signal distribution at time t, with respect to the sequence of

random variables $X_1, \ldots, X_t$, can be written as follows

$$\mathbb{P}_{\mu,t}(\mathrm{d}(x_0, \ldots, x_t)) = \mu(\mathrm{d}x_0) M_1(x_0, \mathrm{d}x_1) \ldots M_t(x_{t-1}, \mathrm{d}x_t) \tag{5.26}$$

defining the filtered probability space

$$\left(\Omega = \prod_{i=0}^{t} E_i, \mathcal{F}_t, \mathcal{F}_\infty, \mathbb{P}_\mu \right), \tag{5.27}$$

where the family of σ-algebras has the following property: $\mathcal{F}_i \subset \mathcal{F}_j \subset \mathcal{F}_\infty$ for any $i \leq j$ and $\mathcal{F}_\infty = \sigma(\cup_{i \geq 0} \mathcal{F}_i)$.

On the other hand, bounded and non-negative potential functions $G_t : E_t \to [0, \infty)$ characterize the time-varying properties of a measurement device. This leads to the following definition of the unnormalized *prediction* Feynman-Kac model, for $h_t \in \mathcal{B}_b(E_t)$ and $t \in \mathbb{N}$.

$$\begin{aligned} \gamma_t(h_t) &= \mathbb{E}_{\mu_0} \left(h_t(X_t) \prod_{i=0}^{t-1} G_i(X_i) \right) \\ &= \int_{E_{[0:t]}} h_t(x_t) \prod_{i=0}^{t-1} G_i(x_i) \mathbb{P}_{\mu_0,t}(\mathrm{d}(x_0, \ldots, x_t)) \end{aligned} \tag{5.28}$$

where $\mathbb{E}_{\mu_0}$ denotes expectation with respect to the distribution of an E_t-valued Markov chain X_t with transitions M_t. The normalized prediction Feynman-Kac model is then:

$$\eta_t(h_t) = \frac{\gamma_t(h_t)}{\gamma_t(1)} \tag{5.29}$$

The idea behind Feynman-Kac formulae based analysis is to define an operator Φ_t acting from $\mathcal{P}(E_{t-1})$ to $\mathcal{P}(E_t)$ — the distribution update operator that maps $\eta_{t-1} \in \mathcal{P}(E_{t-1})$ to $\eta_t \in \mathcal{P}(E_t)$, and then analyze the stability of the filtering model described by the sequence of operators $\Phi_1, \ldots, \Phi_t$ by studying the properties of the semigroup formed by this sequence of operators. Intuitively, the Markov kernel describing unconditional signal evolution and the potential functions describing measurement process constitute the ingredients of the Feynman-Kac operator. The formal definition of this operator is as follows.

The Boltzmann-Gibbs transformation Ψ_t reflects the effect of the likelihood function

at time t on the normalized prediction model. The transformation Ψ_t maps the set of probability measures on E_t onto itself, i.e. $\Psi_t : \nu \in \mathcal{P}_t(E_t) \mapsto \Psi_t(\nu) \in \mathcal{P}_t(E_t)$. For a particular measure ν,

$$\Psi_t(\nu)(\mathrm{d}x_t) = \frac{1}{\nu(G_t)} G_t(x_t)\nu(\mathrm{d}x_t). \tag{5.30}$$

This transformation is used to construct the non-linear operator $\Phi_t : \mathcal{P}(E_{t-1}) \to \mathcal{P}(E_t)$, which is used to update the predictive posterior distribution from time step $t-1$ to time step t:

$$\eta_t = \Phi_t(\eta_{t-1}) \tag{5.31}$$

This operator combines the fitness assessment described by the likelihood function G_{t-1} and the diffusion step described by the Markov kernel M_t

$$\Phi_t(\eta_{t-1}) = \Psi_{t-1}(\eta_{t-1})M_t \tag{5.32}$$

The repeated application of this operator, $\Phi_t(\eta_{t-1})_{t\geq 1}$, results in the semigroups $\Phi_{i,t}$, $i \leq t$ associated with the normalized Feynman-Kac distribution flows η_t.

$$\Phi_{i,t} = \Phi_t \circ \Phi_{t-1} \circ \ldots \circ \Phi_{i+1} \tag{5.33}$$

The semigroup $\Phi_{i,t}$ describes the evolution of the normalized prediction Feynman-Kac model from time i to time t:

$$\eta_t = \Phi_{i,t}(\eta_i) \tag{5.34}$$

and can be formally defined through the Feynman-Kac formulae

$$\Phi_{i,t}(\eta_i)(h_t) = \frac{\mathbb{E}_{i,\eta_i}\{h_t(X_t)\prod_{j=i}^{t-1} G_j(X_j)\}}{\mathbb{E}_{i,\eta_i}\{\prod_{j=i}^{t-1} G_j(X_j)\}}. \tag{5.35}$$

Here $\mathbb{E}_{i,\eta_i}$ is the expectation with respect to the measure $\mathbb{P}_{i,\eta_i}$:

$$\mathbb{P}_{i,\eta_i}(\cdot) = \int_{E_i} \eta_i(\mathrm{d}x_i)\mathbb{P}_{i,x_i}(\cdot), \tag{5.36}$$

and $\mathbb{P}_{i,x_i}$ is the probability distribution of the shifted chain $(X_{i+t})_{t\geq 0}$ with the respective expectation $\mathbb{E}_{i,x_i}$ defined as follows:

$$\mathbb{E}_{i,x_i}\{h_{i,t}(X_i,\ldots,X_t)\} = \int_{E_{i+1:t}} h_{i,t}(x_i,\ldots,x_t)M_{i+1}(x_i,\mathrm{d}x_{i+1})\ldots M_t(x_{t-1},\mathrm{d}x_t). \tag{5.37}$$

The semigroup $\Phi_{i,t}$ is related to $G_{i,t} : E_i \to (0,\infty)$, potential functions on E_i, and $P_{i,t} : \mathcal{P}(E_i) \to \mathcal{P}(E_t)$, Markov kernels from E_i to E_t. In particular, $G_{i,t}$ is defined as the expectation of the composite potential constructed based on the data acquired over steps $i,\ldots,t-1$ with respect to the shifted chain $M_{i+1}\ldots M_t$:

$$G_{i,t}(x_i) = \mathbb{E}_{i,x_i}\left\{\prod_{j=i}^{t-1} G_j(X_j)\right\}, \tag{5.38}$$

and $P_{i,t}$ is defined by the Feynman-Kac formulae as follows:

$$P_{i,t}(h_t)(x_i) = \frac{\mathbb{E}_{i,x_i}\left\{h_t(X_t)\prod_{j=i}^{t-1} G_j(X_j)\right\}}{\mathbb{E}_{i,x_i}\left\{\prod_{j=i}^{t-1} G_j(X_j)\right\}}. \tag{5.39}$$

The Boltzmann-Gibbs transformation

$$\Psi_{i,t}(\eta_i)(h_i) = \frac{\eta_i(G_{i,t}h_i)}{\eta_i(G_{i,t})} \tag{5.40}$$

and the semigroup $\Phi_{i,t}$

$$\Phi_{i,t}(\eta_i) = \Psi_{i,t}(\eta_i)P_{i,t} \tag{5.41}$$

can then be represented via these two quantities.

Feynman-Kac formulae and Bayesian framework

It is convenient to draw parallels between the Feynman-Kac description of the filtering process discussed above and the Bayesian formulation of the sequential filtering process. In particular, the integral operator, $M_t(x_{t-1}, \mathrm{d}x_t)$, describing the evolution of signal diffusion is most naturally related to the state transition density (assuming one exists):

$$M_t(x_{t-1}, \mathrm{d}x_t) = p_t(x_t|x_{t-1})\mathrm{d}x_t.$$

On the other hand, the measurement equation compactly described by the potential function $G_t(x_t)$ in the Feynman-Kac framework is directly related to the likelihood function $p_t(y_t|x_t)$ in the Bayesian framework:

$$G_t(x_t) = p_t(y_t|x_t).$$

We can then see how the diffusion step within the Feynman-Kac framework is related to the prediction step within the Bayesian framework:

$$\Phi_{t+1}(\eta_t) = \int_{E_t} \Psi_t(\eta_t)(\mathrm{d}x_t) M_t(x_t, \mathrm{d}x_{t+1}) \quad \text{Feynman-Kac} \tag{5.42}$$

$$p_{t+1}(x_{t+1}|y_{1:t}) = \int_{E_t} p(x_t|y_{1:t}) p_{t+1}(x_{t+1}|x_t)\mathrm{d}x_t \quad \text{Bayes} \tag{5.43}$$

Thus the operator Φ_{t+1} generates the normalized predictive posterior distribution, $\eta_{t+1}(\mathrm{d}x_{t+1}) = \Phi_{t+1}(\eta_t)(\mathrm{d}x_{t+1}) = p_{t+1}(x_{t+1}|y_{1:t})\mathrm{d}x_{t+1}$ using the Markov diffusion $M_t(x_t, \mathrm{d}x_{t+1}) = p_{t+1}(x_{t+1}|x_t)\mathrm{d}x_{t+1}$. On the other hand, the Boltzmann-Gibbs transformation $\Psi_t(\eta_t)(\mathrm{d}x_t) = p_t(x_t|y_{1:t})\mathrm{d}x_t$ generates the normalized posterior distribution using the update step analogous to that of the Bayes model:

$$\Psi_t(\eta_t) = \frac{G_t(x_t)\eta_t(\mathrm{d}x_t)}{\int_{E_t} G_t(x_t)\eta_t(\mathrm{d}x_t)} \quad \text{Feynman-Kac} \tag{5.44}$$

$$p_t(x_t|y_{1:t}) = \frac{p_t(y_t|x_t)p_t(x_t|y_{1:t-1})}{\int_{E_t} p_t(y_t|x_t)p_t(x_t|y_{1:t-1})\mathrm{d}x_t} \quad \text{Bayes} \tag{5.45}$$

Here we recognize that the normalization constant $\eta_t(G_t)$ has the meaning of evidence at time t, $p_t(y_t|y_{1:t-1})$, within the Bayes framework. We conclude that the Feynman-Kac

formulae are directly related to the predict-update Bayesian recursion. The difference between the two formulations lies in the fact that the Feynman-Kac formulae describe the evolution of distributions, while the Bayesian framework describes the evolution of the corresponding densities (assuming that these densities exist).

N-particle approximations

A particle filter can be defined by developing an N-particle approximation to the Feynman-Kac model. This approximation consists of N path particles:

$$\xi_t'^k = (\xi_{i,t}^k)_{0 \leq i \leq t} \in E_t' = E_{[0,t]} \qquad i \in 1, \ldots, N$$

The particle approximation of the prediction Feynman-Kac model is defined as:

$$\eta_t^N = \frac{1}{N} \sum_{k=1}^{N} \delta_{\xi_{t,t}^k}$$

The N-tuple ξ_t represents the configuration at time t of N particles ξ_t^k, and resides in the product space E_t^N. The particle filter then involves a two-step updating process:

$$\xi_t \in E_t^N \overset{\text{selection}}{\longrightarrow} \widehat{\xi}_t \in E_t^N \overset{\text{mutation}}{\longrightarrow} \xi_{t+1} \in E_{t+1}^N$$

The selection stage consists of selecting randomly N particles $\widehat{\xi}_t^k$. This random selection is achieved by setting, with probability $\epsilon_t G_t(\xi_t^k)$, $\widehat{\xi}_t^k = \xi_t^k$; otherwise a random particle $\widetilde{\xi}_t^k$ is chosen with distribution $\sum_{k=1}^N \frac{G_t(\xi_t^k)}{\sum_{j=1}^N G_t(\xi_t^j)} \delta_{\xi_t^k}$, and $\widehat{\xi}_t^k$ is set $\widehat{\xi}_t^k = \widetilde{\xi}_t^k$. During the mutation phase, each particle $\widehat{\xi}_t^k$ evolves according to the Markov transition M_{t+1}.

Alternatively, a particle filter can be described by defining a random sampling operator. Let the sampling operator $S^N : \mathcal{P}(E) \to \mathcal{P}(E^N)$ be defined as:

$$S^N(\eta)(h) = \frac{1}{N} \sum_{i=1}^{N} h(\xi^i) \ . \tag{5.46}$$

where $(\xi^1, \ldots, \xi^N)$ is the i.i.d. sample from η. With this notation, the standard particle

filter can be expressed using the recursion

$$\eta_t^N = S^N(\Phi_t(\eta_{t-1}^N)). \tag{5.47}$$

5.3.2 Regularity Conditions and Particle Filter Stability

It was mentioned previously that the stability of Markov filters (and thus the associated particle approximations) is related to studying the stability of semigroups $\Phi_{i,t}$. We now cite an important result, due to Del Moral [114], describing error propagation for a general non-linear operator $\Phi_{i,t}$:

Proposition 5.1 (Del Moral [114], Proposition 4.3.7)**.** *For any $0 \leq i \leq t$, $\mu_i \in \mathcal{P}(E_i)$, and $h_t \in \mathcal{B}_b(E_i)$ with $\mathrm{osc}(h_t) \leq 1$, respectively $||h_t|| \leq 1$, there exists a function $h_{i,t}^{(\mu_i)}$ in $\mathcal{B}_b(E_i)$ with $\mathrm{osc}(h_{i,t}^{(\mu_i)}) \leq 1$, respectively $||h_{i,t}^{(\mu_i)}|| \leq 1$, such that for any $\eta_i \in \mathcal{P}(E_i)$ we have*

$$|[\Phi_{i,t}(\eta_i) - \Phi_{i,t}(\mu_i)](h_t)| \leq \beta(P_{i,t}) \frac{||G_{i,t}||_{\mathrm{osc}}}{\eta_i(G_{i,t})} |(\eta_i - \mu_i)(h_{i,t}^{(\mu_i)})| \tag{5.48}$$

and respectively

$$|[\Phi_{i,t}(\eta_i) - \Phi_{i,t}(\mu_i)](h_t)| \leq \beta(P_{i,t}) \frac{2||G_{i,t}||}{\eta_i(G_{i,t})} |(\eta_i - \mu_i)(h_{i,t}^{(\mu_i)})| \tag{5.49}$$

This result describes the propagation of one-step approximation error through the non-linear operator $\Phi_{i,t}$. It reveals the link between the initial error at time i and the propagated error at time t through the properties of potential functions $G_{i,t}$ and the Dobrushin contraction coefficient $\beta(P_{i,t}) \in [0,1]$ defined as follows:

$$\beta(P_{i,t}) = \sup\{||P_{i,t}(x_i, \cdot) - P_{i,t}(y_i, \cdot)||_{\mathrm{tv}}; x_i, y_i \in E_i\}, \tag{5.50}$$

Where the total variation norm defined for any $\mu, \nu \in \mathcal{P}(E)$ has the following form:

$$||\mu - \nu||_{\mathrm{tv}} = \sup\{|\mu(h) - \nu(h)|; h \in \mathrm{Osc}_1(E)\}. \tag{5.51}$$

The properties of the two quantities, $G_{i,t}$ and $\beta(P_{i,t})$, can be further studied through the introduction of regularity constants on M_t and G_t. To ensure that the Markov kernel M_t is sufficiently mixing the following regularity condition is introduced in [114]:

- $(M)_m$ There exists an integer $m \geq 1$ and some sequence of numbers $\epsilon_i(M) \in (0,1)$ such that for i and $x_i, y_i \in E_i$ we have

$$M_{i,i+m}(x_i, \cdot) = M_{i+1} M_{i+2} \dots M_{i+m}(x_i, \cdot) \geq \epsilon_i(M) M_{i,i+m}(y_i, \cdot)$$

The regularity condition imposed on likelihood functions G_t takes the following form:

- (G) There exists a sequence of strictly positive numbers $\epsilon_t(G) \in (0,1]$ such that for any $x_t, y_t \in E_t$

$$G_t(x_t) \geq \epsilon_t(G) G_t(y_t)$$

If assumptions (G) and $(M)_m$ hold then according to Proposition 4.3.3. [114] $\beta(P_{i,t})$ can be bounded as follows:

$$\beta(P_{i,t}) \leq \prod_{k=0}^{\lfloor (t-i)/m \rfloor - 1} \left(1 - \epsilon_{i+km}^{(m)}(G, M)\right), \tag{5.52}$$

where $\epsilon_i^{(m)}(G, M) = \epsilon_i^2(M) \prod_{k=i+1}^{i+m} \epsilon_k(G)$. Furthermore, according to the same Proposition 4.3.3. [114] and under the same assumptions, we have the following:

$$\frac{G_{i,t}(x_i)}{G_{i,t}(y_i)} \geq \epsilon_i(M) \prod_{k=i}^{m} \epsilon_k(G) \tag{5.53}$$

These expressions can be used to bound the contraction coefficients in Proposition 5.1. To see how the step-wise error bound in this proposition is related to the total N-particle approximation error it is necessary to examine the following decomposition [114]:

$$\eta_t^N - \eta_t = \sum_{i=0}^{t} \left[\Phi_{i,t}(\eta_i^N) - \Phi_{i,t}(\Phi_i(\eta_{i-1}^N))\right]. \tag{5.54}$$

Indeed, upon expanding the above expression we obtain[2]:

$$\begin{aligned}
&\sum_{i=0}^{t}\left[\Phi_{i,t}(\eta_i^N)-\Phi_{i,t}(\Phi_i(\eta_{i-1}^N))\right]\\
&\quad=\Phi_{t,t}(\eta_t^N)-\Phi_{t,t}(\Phi_t(\eta_{t-1}^N))+\Phi_{t-1,t}(\eta_{t-1}^N)-\Phi_{t-1,t}(\Phi_{t-1}(\eta_{t-2}^N))+\cdots\\
&\quad+\Phi_{1,t}(\eta_1^N)-\Phi_{1,t}(\Phi_1(\eta_0^N))+\Phi_{0,t}(\eta_0^N)-\Phi_{0,t}(\Phi_0(\eta_{-1}^N))\\
&\quad=\eta_t^N-\Phi_t(\eta_{t-1}^N)+\Phi_t(\eta_{t-1}^N)-\Phi_{t-1,t}(\eta_{t-2}^N)+\cdots\\
&\quad+\Phi_{1,t}(\eta_1^N)-\Phi_{0,t}(\eta_0^N)+\Phi_{0,t}(\eta_0^N)-\Phi_{0,t}(\eta_0)\\
&\quad=\eta_t^N-\eta_t. \qquad (5.55)
\end{aligned}$$

Thus using the triangle inequality and Proposition 5.1 the total particle filtering error can be bounded as follows

$$\begin{aligned}
\mathbb{E}|[\eta_t^N-\eta_t](h_t)| &\le \sum_{i=0}^{t}\beta(P_{i,t})\frac{||G_{i,t}||}{\eta_i(G_{i,t})}\mathbb{E}|[\eta_i^N-\Phi_i(\eta_{i-1}^N)](h_i)|\\
&\le \sum_{i=0}^{t}\prod_{k=0}^{\lfloor (t-i)/m\rfloor-1}\left(1-\epsilon_{i+km}^{(m)}(G,M)\right)\epsilon_i^{-1}(M)\prod_{k=i}^{m}\epsilon_k^{-1}(G)\mathbb{E}|[\eta_i^N-\Phi_i(\eta_{i-1}^N)](h_i)|. \qquad (5.56)
\end{aligned}$$

The analysis of the total particle filtering error can now be reduced to the analysis of the step-wise (local) N-particle approximation errors $|[\eta_i^N-\Phi_i(\eta_{i-1}^N)](h_i)|$. By the definition of the sampling operator, S^N, this local N-particle approximation error has the form

$$|[\eta_i^N-\Phi_i(\eta_{i-1}^N)](h_i)|=|[S^N(\Phi_i(\eta_{i-1}^N))-\Phi_i(\eta_{i-1}^N)](h_i)|. \qquad (5.57)$$

The errors of this type can be bounded using the following lemma related to Khinchine, Bürkholder, and other similar inequalities.

Lemma 5.1 (Del Moral [114], Lemma 7.3.3)**.** *For any $p\ge 1$ and sequence of $\mathcal{E}$-measurable functions $(h_i)_{i\ge 1}$ with finite oscillations such that $\mu_i(h_i)=0$ for all $i\ge 1$ we have*

$$\sqrt{N}\mathbb{E}\{|m(X)(h)^p|\}^{\frac{1}{p}}\le d(p)^{\frac{1}{p}}\sigma(h) \qquad (5.58)$$

[2]By convention [114], $\Phi_0(\eta_{-1}^N)=\eta_0$.

where the following definitions are used

$$m(x)(h) = \frac{1}{N}\sum_{i=1}^{N} h_i(x^i) \quad \textit{and} \quad \sigma^2(h) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{osc}^2(h_i) \tag{5.59}$$

and finite constants $d(p)$ are given by the following:

$$d(2p) = \frac{(2p)!}{p!}2^{-p}, \tag{5.60}$$

$$d(2p-1) = \frac{(2p-1)!}{(p-1)!\sqrt{p-1/2}}2^{-(p-1/2)} \tag{5.61}$$

The combination of the above results, namely, Lemma 5.1 to bound the local approximation errors, Proposition 4.3.3. [114] to bound the contraction of the operator $\Phi_{i,t}$, Proposition 5.1 to decouple the contraction of the operator and the approximation error, and decomposition (5.54) to split the total approximation error into the sum of local terms yields the following important result.

Theorem 5.1 (Del Moral [114], Theorem 7.4.4)**.** *For any $t \geq 0$, $p \geq 1$, and $h_t \in \mathrm{Osc}_1(E_t)$, we have*

$$\sqrt{N}\mathbb{E}\left\{|[\eta_t^N - \eta_t](h_t)|^p\right\}^{1/p} \leq 2d(p)^{1/p}\sum_{i=0}^{t} r_{i,t}\beta(P_{i,t}) \tag{5.62}$$

for the sequence of finite constants $d(p)$ defined in Lemma 5.1. In addition, suppose that conditions (G) and $(M)_m$ hold true for some integer $m \geq 1$ and some pair parameters $(\epsilon_t(G), \epsilon_t(M))$ such that $\epsilon(G) = \max_{i\leq t}\epsilon_i(G)$ and $\epsilon(M) = \max_{i\leq t}\epsilon_i(M)$. Then we have the uniform estimate

$$\sup_{t\geq 0}\sup_{h_t\in\mathrm{Osc}_1(E_t)} \sqrt{N}\mathbb{E}\left\{|[\eta_t^N - \eta_t](h_t)|^p\right\}^{1/p} \leq \frac{2d(p)^{1/p}m}{\epsilon^3(M)\epsilon^{2m-1}(G)} \tag{5.63}$$

Here $r_{i,t} = \sup_{x_i,y_i\in E_i}(G_{i,t}(x_i)/G_{i,t}(y_i))$ is the parameter measuring the relative oscillations of the potential functions.

5.4 Greedy Maximum Likelihood Mixture Estimation

In the next chapter the approximation error of the CSIP based WSN tracking algorithm (leader node particle filter) will be analyzed. The two settings, non-parametric and parametric hand-off (leader node exchange) will be discussed. In the parametric case, the approximation involves forming a parametric approximation to the density. The mixture model is a powerful parametric approximation that is able to represent complex general multi-modal probability density functions [121]. A parametric mixture model of a (target) probability distribution is a convex combination of probability distributions with unknown parameters that come from a certain (approximation) class. The parametric mixture model estimation problem consists of estimating the parameters and weights of probability distributions comprising the model. The estimation problem can be solved by optimizing the empirical cost function (e.g. likelihood function) based on the available data (samples from the target distribution). This, exact, empirical mixture estimation problem, however, is intractable since the empirical cost function is often non-convex and the approximation class may be too large [122]. The suboptimal, greedy, algorithms provide alternative efficient numerical solutions to the exact empirical mixture estimation problem (see [121, 123] and references therein).

To perform the estimation of the N_{p}-component mixture used during the parametric leader node hand-off operation we propose to use the greedy maximum likelihood (GML) mixture density estimation algorithm introduced by Li and Barron in [124]. The attractive features of the greedy maximum likelihood algorithm [124] discussed in the following section are threefold. First, the algorithm simplifies the maximum likelihood density estimation procedure. Instead of facing the N_{p}-mixture estimation problem we only have to solve N_{p} 2-mixture estimation problems [124]. Second, there are several papers (see e.g. [124] and [125]) that develop bounds on approximation and sampling errors of this algorithm in terms of KL-divergence. Finally, it was shown [124] that the performance of the greedy algorithm converges to that of the exact mixture estimation algorithm as N and N_{p} become large.

5.4.1 Algorithm Description

The probability density estimation problem consists of estimating an unknown probability density given the i.i.d. sample $\{x_i\}_{i=1}^{N}$ from this density [126]. The greedy procedure for

the maximum likelihood mixture density estimation with the cost induced by the Kullback-Leibler divergence was proposed by Li and Barron [124]. The analysis of this framework appears in [124] and [125]. The analysis of a more general approximation framework with an arbitrary convex cost function appears in Zhang [127].

As before, let $(E, \mathcal{E})$ be a measurable space. Denote λ a σ-finite measure on $\mathcal{E}$. Throughout this section it is assumed that the underlying distribution has a density if its Radon-Nikodym derivative with respect to λ exists.

Within the GML framework proposed by Li and Barron [124] the discrepancy between the target density f and its estimate is measured by the Kullback-Leibler (KL) divergence. For any two measures ν and μ on E KL-divergence can be defined as follows:

$$D(\nu||\mu) = \int_E \log \frac{\mathrm{d}\nu}{\mathrm{d}\mu} \mathrm{d}\nu \tag{5.64}$$

We will also abuse notation by writing KL-divergence for two arbitrary densities f and g in a similar fashion:

$$D(f||g) = \int_E \log \frac{f(x)}{g(x)} f(x) \mathrm{d}\lambda(x) \tag{5.65}$$

Consider the following class of bounded parametric densities:

$$\mathcal{H} = \left\{ \phi_\theta(x) : \theta \in \Theta \subset \mathbb{R}^d, a \leq \inf_{\theta \in \Theta, x \in E} \phi_\theta(x), \sup_{\theta \in \Theta, x \in E} \phi_\theta(x) \leq b \right\} \tag{5.66}$$

where $0 < a < b < \infty$ and Θ defines parameter space. In the setting where the leader node hand-off is accomplished using parametric approximation, we are looking for a sequence of mixture density estimators of the filtering densities. The approximation is restricted to a class of discrete N_p-component convex combinations of the form:

$$\mathcal{C}_{N_\mathrm{p}} = \mathrm{conv}_{N_\mathrm{p}}(\mathcal{H}) = \left\{ g : g(x) = \sum_{i=1}^{N_\mathrm{p}} \alpha_i \phi_{\theta_i}(x), \phi_\theta \in \mathcal{H}, \sum_{i=1}^{N_\mathrm{p}} \alpha_i = 1, \ \alpha_i \geq 0 \right\} \tag{5.67}$$

As N_p grows without bound, $\mathcal{C}_{N_\mathrm{p}}$ converges to the class of continuous convex combinations:

$$\mathcal{C} = \mathrm{conv}(\mathcal{H}) = \left\{ g : g(x) = \int_\Theta \phi_\theta(x) \mathbb{P}(\mathrm{d}\theta), \phi_\theta \in \mathcal{H} \right\} \tag{5.68}$$

The general framework for the greedy approximation of arbitrary cost functions is discussed in [127]. The particular instance of this more general framework is the GML for mixture approximation (see [125] and [124]). Algorithm 2 summarizes the computational routine used to implement the sequential greedy maximum likelihood procedure.

Algorithm 2: Greedy Maximum Likelihood (GML)

Given $g_1 \in \mathcal{H}$
for $i = 2$ *to* N_p **do**
    Find $\phi_{\theta_i} \in \mathcal{H}$ and $0 \leq \alpha_i \leq 1$ to maximize the function:
    $(\theta_i^*, \alpha_i^*) = \arg\max_{\alpha_i, \theta_i} \sum_{j=1}^{N} \log((1-\alpha_i) g_{i-1}(x_j) + \alpha_i \phi_{\theta_i}(x_j))$
    Let $g_i = (1-\alpha_i^*) g_{i-1} + \alpha_i^* \phi_{\theta_i^*}$
endfor

5.4.2 Local Error Analysis

The following notation is introduced to facilitate presentation. Assuming that f is a target density and $g \in \mathcal{C}$ we denote $D(f||\mathcal{C}) = \inf_{g \in \mathcal{C}} D(f||g)$, the least possible divergence between a target density, f, and a member g from the class of continuous convex combinations $\mathcal{C}$. Furthermore, assuming that the target density f is known, the analytical estimator $g^{N_\mathrm{p}} \in \mathcal{C}_{N_\mathrm{p}}$ can be obtained by solving the following greedy recursion for $i = 2 \ldots N_\mathrm{p}$ (see Algorithm 2):

$$(\theta_i^*, \alpha_i^*) = \arg\max_{\alpha_i, \theta_i} \int_{x \in E} \log((1-\alpha_i) g_{i-1}(x) + \alpha_i \phi_{\theta_i}(x)) f(x) \mathrm{d}x.$$

Alternatively, $\widehat{g}^{N_\mathrm{p}} \in \mathcal{C}_{N_\mathrm{p}}$ is an empirical N_p-mixture estimator constructed using Algorithm 2 based on a sample from the target density, f. The following theorem (see [124]) reveals an important general property of GML algorithm.

Theorem 5.2 (Li and Barron [124], Theorem 2)**.** *For every* $g_{\mathcal{C}}(x) \in \mathcal{C}$

$$D(f||g^{N_p}) \leq D(f||g_{\mathcal{C}}) + \frac{\gamma_{\mathcal{C}}^2}{N_p}. \tag{5.69}$$

Here,

$$c_{f,\mathcal{C}}^2 = \int \frac{\int_\Theta \phi_\theta^2(x)\mathbb{P}(\mathrm{d}\theta)}{(\int_\Theta \phi_\theta(x)\mathbb{P}(\mathrm{d}\theta))^2} f(x)\mathrm{d}x, \tag{5.70}$$

and $\gamma = 4[\log(3\sqrt{e}) + \sup_{\theta_1,\theta_2\in\Theta, x\in E} \log(\phi_{\theta_1}(x)/\phi_{\theta_2}(x))]$

One of the consequences [124] of Theorem 5.2 is the following relationship between an arbitrary $g_{\mathcal{C}}(x) \in \mathcal{C}$ and the empirical GML algorithm output $\widehat{g}^{N_\mathrm{p}} \in \mathcal{C}_{N_\mathrm{p}}$:

$$\frac{1}{N}\sum_{i=1}^{N} \log \widehat{g}^{N_\mathrm{p}}(x_i) \geq \frac{1}{N}\sum_{i=1}^{N} \log g_{\mathcal{C}}(x_i) - \frac{\gamma c_{f,\mathcal{C}}^2}{N_\mathrm{p}}. \tag{5.71}$$

Clearly, it also follows directly from Theorem 5.2 that $D(f||g^{N_\mathrm{p}}) \leq D(f||\mathcal{C}) + \frac{\gamma c_{f,\mathcal{C}}^2}{N_\mathrm{p}}$. Thus Theorem 5.2 establishes a strong formal argument that shows that the GML density estimate converges to the best possible maximum likelihood estimate as N_p grows without bound.

A stronger result for the empirical estimator $\widehat{g}^{N_\mathrm{p}}$ satisfying the following general relationship

$$\sum_{j=1}^{N} \widehat{g}_i(x_j) \geq \arg\max_{\alpha_i,\theta_i} \sum_{j=1}^{N} \log((1-\alpha_i)\widehat{g}_{i-1}(x_j) + \alpha_i \phi_{\theta_i}(x_j)), \tag{5.72}$$

appears in [125] (the GML estimator is the estimator satisfying (5.72) with equality).

Theorem 5.3 (Rakhlin et al. [125], Theorem 2.1)**.** *For any target density f such that $a \leq f(x) \leq b$ for all $x \in E$ and $\widehat{g}^{N_p}$ being either the maximizer of the likelihood over N_p-component mixtures or more generally any sequence of density estimates satisfying* (5.72)*,*

$$\mathbb{E}\{D(f\|\widehat{g}_i)\} \leq D(f\|\mathcal{C}) + \frac{c_1}{N_p} + \mathbb{E}\left\{\frac{c_2}{\sqrt{N}}\int_a^b \log^{1/2}\mathcal{N}(\epsilon,\mathcal{H},d_N)\mathrm{d}\epsilon\right\} \tag{5.73}$$

where c_1, c_2 are constants (dependent on a, b) and $\mathcal{N}(\epsilon,\mathcal{H},d_N)$ is the ϵ-covering number[3] *of $\mathcal{H}$ with respect to empirical distance $d_N^2(h_1,h_2) = \frac{1}{N}\sum_{i=1}^{N}(h_1(x_i) - h_2(x_i))^2$.*

[3]The ϵ-covering number, $\mathcal{N}(\epsilon,\mathcal{H},d_N)$, is the smallest number of sets of radius ϵ whose union contains $\mathcal{H}$.

One of the goals of the next chapter will be connecting the existing results on the performance of the GML in terms of the KL-divergence to its performance in terms of L_p error metric. For this purpose, for a collection of bounded measurable functions $\mathcal{H}$ we define the Zolotarev seminorm [114] on $\mathcal{P}(E)$

$$||\mu - \nu||_{\mathcal{H}} = \sup\{|\mu(h) - \nu(h)|; h \in \mathcal{H}\} \tag{5.74}$$

We will rely on several important results. In particular, the Rademacher sequence (ε_i) of independent random variables taking values in $\{-1, +1\}$ and $\mathbb{P}\{\varepsilon_i = 1\} = \mathbb{P}\{\varepsilon_i = -1\} = 1/2$ will be used along with the general form of the comparison inequality for Rademacher processes (see [128], p. 112) presented below.

Theorem 5.4 (Ledoux and Talagrand [128], Theorem 4.12)**.** *Let $F : \mathbb{R}_+ \to \mathbb{R}_+$ be convex and increasing. Let further $\varphi_i : \mathbb{R} \to \mathbb{R}, i \leq N$, be contractions*[4] *such that $\varphi_i(0) = 0$. Then, for any bounded subset T in $\mathbb{R}^N$*

$$\mathbb{E}F\left(\frac{1}{2}\left\|\sum_{i=1}^{N}\varepsilon_i\varphi_i(t_i)\right\|_T\right) \leq \mathbb{E}F\left(\left\|\sum_{i=1}^{N}\varepsilon_i t_i\right\|_T\right) \tag{5.75}$$

We will also use the Orlicz norm [114, 129] $\pi_{\psi_p}(Y)$ of a random variable Y defined as

$$\pi_{\psi_p}(Y) = \inf\{C > 0 : \mathbb{E}\{\psi_p(|Y|/C)\} \leq 1\} \tag{5.76}$$

and associated with a nondecreasing convex function $\psi_p(x) = e^{x^p} - 1$. The sub-Gaussian inequality (see Corollary 2.2.8 in [129]) implies for a class of nonnegative and bounded functions $\|h\| \leq b, \forall h \in \mathcal{H}$:

$$\mathbb{E}_\varepsilon \pi_{\psi_2}(\|S_\varepsilon^N(\mu)\|_{\mathcal{H}}) \leq \frac{C}{\sqrt{N}}\int_0^b \sqrt{\log(1 + \mathcal{D}(\epsilon, \mathcal{H}, d_N))}\mathrm{d}\epsilon. \tag{5.77}$$

Here C is a universal constant [129] and S_ε^N is the generator of the Rademacher process (with x_i being the i.i.d. samples from μ and ε_i being the i.i.d. Rademacher random

[4] the function $\varphi : \mathbb{R} \to \mathbb{R}$ is a contraction if we have $|\varphi(x) - \varphi(y)| \leq |x - y|, \forall x, y \in E$

variables, $\mathbb{P}\{\varepsilon_i = 1\} = \mathbb{P}\{\varepsilon_i = -1\} = 1/2$, independent of x_i):

$$S_\varepsilon^N(\mu)(h) = \frac{1}{N}\sum_{i=1}^{N} \varepsilon_i h(x_i). \tag{5.78}$$

The empirical semimetric defined for $h_1, h_2 \in \mathcal{H}$ is simply

$$d_N^2(h_1, h_2) = \frac{1}{N}\sum_{i=1}^{N} (h_1(x_i) - h_2(x_i))^2 \tag{5.79}$$

and $\mathcal{D}(\epsilon, \mathcal{H}, d_N)$ is the packing number — the maximum number of ε-separated points in $\mathcal{H}$.

Chapter 6

Analysis of the Leader Node Particle Filter

This chapter examines the impact of approximation steps that become necessary when a particle filter is implemented in the leader node resource constrained framework described in detail in the previous chapter. This particle filter performs intermittent approximation for the leader node hand-off, either by subsampling the particles or by generating a parametric approximation. For this algorithm, time uniform bounds on the weak sense L_p error and associated exponential inequalities are derived. The theoretical analysis is motivated by numerical experiments exploring the approximation performance of the leader node target tracking algorithm. The relationship of the theoretical results to the error bounds is investigated.

6.1 Leader Node Particle Filtering with Intermittent Subsampling

In this section we concentrate on analyzing the variant of the leader node particle filter that employs subsampling to approximate the particle cloud during leader node hand-off operation. We first present the algorithmic description of the leader node filtering framework and then develop a detailed signal model for this framework. Finally, we discuss how the general Feynman-Kac operators outlined in the previous chapter can be adopted for the analysis of the leader node framework and present our analysis results. The results

presented in this section serve as a basis for the analysis of the parametric approximation leader node particle filter presented in Section 6.2.

6.1.1 Algorithm Description

Suppose as before that $\mathfrak{L} = \{1, 2, \ldots, L\}$ is the set of leader node labels and every leader node with label $\ell \in \mathfrak{L}$ has a set of satellite nodes $\mathfrak{S}_\ell$ that take measurements and transmit them to the leader node. The number of such satellite nodes in the vicinity of the leader node ℓ is $|\mathfrak{S}_\ell|$. The state-space model describing the target evolution and measurement process at every leader node is then a simple modification of the general state-space model described in Section 5.1.1:

$$X_t = f_t(X_{t-1}, \varrho_t) \tag{6.1}$$

$$Y_{\ell_t}^j = g_{\ell_t}^j(X_t, \zeta_{\ell_t}^j) \quad \forall j \in \mathfrak{S}_{\ell_t} \tag{6.2}$$

Here $X_t \in \mathbb{R}^{d_x}$ is the target state vector, $f_t : \mathbb{R}^{d_x} \to \mathbb{R}^{d_x}$ is the nonlinear diffusion map, and ϱ_t is the system noise; $Y_{\ell_t}^j \in \mathbb{R}^{d_y^j}$ is the jth sensor measurement vector, $g_{\ell_t}^j : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y^j}$ is the nonlinear measurement function, and $\zeta_{\ell_t}^j$ is the measurement noise. Thus the target motion model is the same at every leader node and the measurement process may be different.

Denote by δ_t a binary variable which indicates whether a subsampling approximation is performed at time-step t. In our analysis, we will assume that this variable is the outcome of a decision function based on the set of particles $\{\xi_{t-1}^k\}_{k=1}^N$ and observations $y_t^{\mathfrak{S}_{\ell_t}}$ at the current time-step. Similar results could be obtained should the decision function be of a more general nature (for example, based on either the entire history of the particle filter, $(\xi'^k)_{k=1}^N$, and/or the entire history of measurements $y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}$). We define $\delta_0(\cdot, \cdot) = 0$, and we assume there exist probabilities:

$$\mathbb{E}\{\delta_t\} = \mathbb{P}\{\delta_t = 1\} = q_t$$

The expectation is with respect to the Monte-Carlo (MC) sampling, measurement noise and the possible target trajectories. The value of q_t characterizes the frequency of subsampling approximation at time-step t.

Recall from Chapter 5 that η_t denotes the normalized predictive posterior distribution at time instant t and η_t^N denotes its N-particle approximation. The subsample approximation

leader node particle filter can then be expressed as:

$$\Phi_{\ell_t}(\eta_{t-1}^N) \Rightarrow \eta_t^{N_\mathrm{b}} \longrightarrow \eta_t^{N_\mathrm{b}} \Rightarrow \eta_t^N \quad \text{if } \delta_t = 1, \tag{6.3}$$

$$\Phi_{\ell_t}(\eta_{t-1}^N) \Rightarrow \eta_t^N \quad \text{if } \delta_t = 0 \tag{6.4}$$

Here the implication sign $\Rightarrow$ represents a sampling operation and the right arrow $\longrightarrow$ denotes the communication process. N is the number of particles used by the leader node for the particle filter computations and N_b is the number of particles in the sub-sampled particle cloud used for the leader node hand-off. Using the sampling operator S^N defined in the previous chapter the subsample approximation particle filter can further be expressed as:

$$\begin{aligned} \eta_t^N &= S^N \circ S^{N_\mathrm{b}}(\Phi_{\ell_t}(\eta_{t-1}^N)) \quad \text{if } \delta_t = 1, \\ \eta_t^N &= S^N(\Phi_{\ell_t}(\eta_{t-1}^N)) \quad \text{if } \delta_t = 0 \end{aligned} \tag{6.5}$$

If $\delta_t = 1$, the current leader node ℓ_t determines the next leader node ℓ_{t+1} (through a sensor management algorithm, see e.g. [96]), and calculates the N_b-particle approximation to the current predictive posterior distribution, η_t^N, by sub-sampling the output of the standard N-particle propagation step. Finally, the leader node ℓ_t transmits $\eta_t^{N_\mathrm{b}}$ to the next leader node ℓ_{t+1}, which recovers the N-particle approximation by up-sampling. If $\delta_t = 0$, the current leader node performs standard particle propagation.

6.1.2 Feynman-Kac Formulae and Regularity Conditions

Assuming that in every leader node neighborhood $\mathfrak{S}_\ell$ observation noises, ζ_t^j, $j \in \mathfrak{S}_{\ell_t}$, in (6.2) are independent[1] the composite potential function at leader node ℓ_t can be written via the product of the individual potential functions of satellite nodes, $G_{\ell_t}^j$:

$$G_{\ell_t} = \prod_{j \in \mathfrak{S}_{\ell_t}} G_{\ell_t}^j.$$

[1] The assumption of independence among the sensor observations is not necessary for the error analysis performed in the chapter but is adopted because it allows for a concrete discussion and concise presentation of results.

Then the propagation of the leader node Feynman-Kac model is described by the pair of prediction-update operators. Since prediction operator is only concerned with the dynamics of the target, it coincides with the Markov kernel in (5.25). On the other hand, the Boltzmann-Gibbs transformation has the following product form

$$\Psi_{\ell_t}(\eta_t)(\mathrm{d}x_t) = \frac{\prod\limits_{j\in\mathfrak{S}_{\ell_t}} G^j_{\ell_t}(x_t)\eta_t(\mathrm{d}x_t)}{\eta_t\left(\prod\limits_{j\in\mathfrak{S}_{\ell_t}} G^j_{\ell_t}\right)}.$$

Thus the unnormalized Feynman-Kac model in the leader node setting is

$$\begin{aligned}\gamma_t(h_t) &= \mathbb{E}_{\mu_0}\left\{h_t(X_t)\prod_{i=0}^{t-1}\prod_{j\in\mathfrak{S}_{\ell_i}} G^j_{\ell_i}(X_i)\right\} \\ &= \int_{E_{[0:t]}} h_t(x_t)\prod_{i=0}^{t-1}\prod_{j\in\mathfrak{S}_{\ell_i}} G^j_{\ell_i}(x_i)\mathbb{P}_{\mu_0,t}(\mathrm{d}(x_0,\ldots,x_t)) \end{aligned} \tag{6.6}$$

Corresponding multi-step potential functions G_{ℓ_i,ℓ_t} can be written as follows:

$$G_{\ell_i,\ell_t}(x_i) = \mathbb{E}_{i,x_i}\left\{\prod_{k=i}^{t-1}\prod_{j\in\mathfrak{S}_{\ell_k}} G^j_{\ell_k}(X_k)\right\}. \tag{6.7}$$

This leads to the following expression for Markov kernels P_{ℓ_i,ℓ_t}

$$P_{\ell_i,\ell_t}(h_t)(x_i) = \frac{\mathbb{E}_{i,x_i}\left\{h_t(X_t)\prod\limits_{k=i}^{t-1}\prod\limits_{j\in\mathfrak{S}_{\ell_k}} G^j_{\ell_k}(X_k)\right\}}{\mathbb{E}_{i,x_i}\left\{\prod\limits_{k=i}^{t-1}\prod\limits_{j\in\mathfrak{S}_{\ell_k}} G^j_{\ell_k}(X_k)\right\}}. \tag{6.8}$$

The Boltzmann-Gibbs transformation can be expressed as:

$$\Psi_{\ell_i,\ell_t}(\eta_i)(h_i) = \frac{\eta_i(G_{\ell_i,\ell_t}h_i)}{\eta_i(G_{\ell_i,\ell_t})},$$

and the Feynman-Kac semigroup Φ_{ℓ_i,ℓ_t} as:

$$\Phi_{\ell_i,\ell_t}(\eta_i) = \Psi_{\ell_i,\ell_t}(\eta_i)P_{\ell_i,\ell_t}.$$

Leader Node Setting Regularity Conditions

The regularity conditions for the Feynman-Kac semigroups describing the evolution of filtering distributions in the leader node particle filter can be formulated based on the material of Section 5.3.2 and Proposition 5.1.

In particular, using the following reasoning we can obtain a result intermediate between (5.48) and (5.49). Recall that, for some positive function $\varphi : E \to \mathbb{R}^+$, the norm $||\cdot||_{\text{osc}}$ is defined as $||\varphi||_{\text{osc}} = ||\varphi|| + \text{osc}(\varphi)$, where $||\cdot||$ denotes the supremum norm and $\text{osc}(\varphi) = \sup(|h(x) - h(y)|;\ x, y \in E)$. Then:

$$\begin{aligned}
||\varphi||_{\text{osc}} = ||\varphi|| + \text{osc}(\varphi) &= ||\varphi|| + \sup_{x,y\in E} |\varphi(x) - \varphi(y)| \\
&\leq ||\varphi|| \left(1 + \frac{1}{||\varphi||} \sup_{x,y\in E} \left|1 - \frac{\varphi(y)}{\varphi(x)}\right| |\varphi(x)|\right) \\
&\leq ||\varphi|| \left(1 + \frac{||\varphi||}{||\varphi||} \sup_{x,y\in E} \left|1 - \frac{\varphi(y)}{\varphi(x)}\right|\right) \\
&\leq ||\varphi|| \left(2 - \frac{\inf_{y\in E} \varphi(y)}{\sup_{x\in E} \varphi(x)}\right)
\end{aligned}$$

The following variant of Proposition 5.1 summarizes the above conclusions.

Proposition 6.1. *For any $0 \leq i \leq t$, $\mu_i \in \mathcal{P}(E_i)$, and $h_t \in \mathcal{B}_b(E_i)$[2] with $\text{osc}(h_t) \leq 1$ there exists a function $h_{i,t}^{(\mu_i)}$ in $\mathcal{B}_b(E_i)$ with $\text{osc}(h_{i,t}^{(\mu_i)}) \leq 1$ such that for any $\eta_i \in \mathcal{P}(E_i)$ we have*

$$|[\Phi_{i,t}(\eta_i) - \Phi_{i,t}(\mu_i)](h_t)| \leq \beta(P_{i,t}) \frac{||G_{i,t}||}{\eta_i(G_{i,t})} \left[2 - \frac{\inf_{y_i\in E_i} G_{i,t}(y_i)}{||G_{i,t}||}\right] |(\eta_i - \mu_i)(h_{i,t}^{(\mu_i)})|$$

To use this proposition in the leader node approximation error analysis we need to impose regularity conditions on the components of semigroups Φ_{ℓ_i,ℓ_t}. In particular, in Section 5.3.2 we have seen that to ensure that the Markov kernel M_i is sufficiently mixing, the regularity condition [114] imposed on Markov diffusions takes the following form:

[2]Recall that $\mathcal{P}(E_i)$ is a set of probability measures and $\mathcal{B}_b(E_i)$ is the Banach space of bounded functions associated with the measurable space E_i.

- $(M)_m$ There exists an integer $m \geq 1$ and some sequence of numbers $\epsilon_i(M) \in (0,1)$ such that for i and $x_i, y_i \in E_i$ we have

$$M_{i,i+m}(x_i, \cdot) = M_{i+1}M_{i+2}\dots M_{i+m}(x_i, \cdot) \geq \epsilon_i(M)M_{i,i+m}(y_i, \cdot)$$

Since the signal model is assumed to be known at the leader nodes, the Markov diffusion kernels M_i are the same for all leader nodes. Thus the assumption $(M)_m$ does not change in the leader node framework. However, for the convenience of presentation the following *time uniform* mixing assumption will be adopted:

- $(M)_{\mathrm{u}}^{(m)}$ There exists an integer $m \geq 1$ and strictly positive number $\epsilon_{\mathrm{u}}(M) \in (0,1)$ such that for any $i \geq 0$ and $x_i, y_i \in E_i$ we have

$$M_{i,i+m}(x_i, \cdot) = M_{i+1}M_{i+2}\dots M_{i+m}(x_i, \cdot) \geq \epsilon_{\mathrm{u}}(M)M_{i,i+m}(y_i, \cdot)$$

We have also seen in Section 5.3.2 that the regularity condition imposed on likelihood functions G_t to bound their oscillations takes the following form in the case of an arbitrary Feynman-Kac semigroup $\Phi_{i,t}$:

- (G) There exists a sequence of strictly positive number $\epsilon_t(G) \in (0,1]$ such that for any $x_t, y_t \in E_t$

$$G_t(x_t) \geq \epsilon_t(G)G_t(y_t)$$

In the leader node particle filter setting the modification of this assumption imposed on the composite potentials in the vicinity of every leader node can be formulated:

- $(G)_{\ell_t}$ There exists a sequence of strictly positive numbers $\epsilon_{\ell_t}(G) \in (0,1]$ such that for any sequence of leader nodes ℓ_t and $x_t, y_t \in E_t$

$$G_{\ell_t}(x_t) \geq \epsilon_{\ell_t}^{|\mathfrak{S}_{\ell_t}|}(G)G_{\ell_t}(y_t)$$

Indeed, $(G)_{\ell_t}$ holds assuming that (G) holds for every likelihood function in the neighborhood of ℓ_t-th leader node at time step t and we have

$$\epsilon_{\ell_t}(G) = \min_{j \in \mathfrak{S}_{\ell_t}} \epsilon_{\ell_t,j}(G),$$

where $\epsilon_{\ell_t,j}(G)$ satisfies (G) for $G^j_{\ell_t}$. Finally, we formulate a uniform condition $(G)_{\mathrm{u}}$, where uniformity is over leader nodes and time:

- $(G)_{\mathrm{u}}$ There exists a strictly positive number $\epsilon_{\mathrm{u}}(G) \in (0,1]$ such that for any ℓ, t and $x_t, y_t \in E_t$
$$G_{\ell_t}(x_t) \geq \epsilon_{\mathrm{u}}^{K_{\mathrm{u}}}(G) G_{\ell_t}(y_t)$$

Indeed, $(G)_{\mathrm{u}}$ holds if $(G)_{\ell_t}$ holds uniformly over time and we take $\epsilon_{\mathrm{u}}(G) = \inf_{t\geq 0} \min_{\ell_t \in \mathfrak{L}} \epsilon_{\ell_t}(G)$ and $K_{\mathrm{u}} = \max_{\ell \in \mathfrak{L}} |\mathfrak{S}_\ell|$.

Using these assumptions we can apply Proposition 4.3.3. [114] to the leader node Feynman-Kac semigroup. Then under assumptions $(G)_{\ell_t}$ and $(M)_m$ the corresponding Dobrushin coefficient can be upper bounded as follows:

$$\beta(P_{\ell_i,\ell_t}) \leq \prod_{k=0}^{\lfloor (t-i)/m \rfloor - 1} \left(1 - \epsilon^{(m)}_{\ell_{i+km}}(G,M)\right),$$

where $\epsilon^{(m)}_{\ell_{i+km}}(G,M)$ is a sequence of constants depending on the sequence of leader nodes selected at time instances between i and t:

$$\epsilon^{(m)}_{\ell_i}(G,M) = \epsilon_i^2(M) \prod_{k=i+1}^{i+m} \epsilon_{\ell_k}^{|\mathfrak{S}_{\ell_k}|}(G).$$

If, however, assumptions $(G)_{\mathrm{u}}$ and $(M)^{(m)}_{\mathrm{u}}$ hold, then we have time uniform estimates for this sequence of constants:

$$\epsilon^{(m)}_{\ell_i}(G,M) \geq \epsilon^2_{\mathrm{u}}(M) \epsilon^{(m-1)K_{\mathrm{u}}}_{\mathrm{u}}(G), \forall \ell_i$$

and the estimate for the Dobrushin contraction coefficient changes accordingly:

$$\beta(P_{\ell_i,\ell_t}) \leq \left(1 - \epsilon^2_{\mathrm{u}}(M) \epsilon^{(m-1)K_{\mathrm{u}}}_{\mathrm{u}}(G)\right)^{\lfloor (t-i)/m \rfloor}.$$

We further note that in the general case and according to Proposition 4.3.3. [114] oscillations

of potential functions have following estimates under assumptions (G) and $(M)_m$:

$$\frac{G_{i,t}(x_i)}{G_{i,t}(y_i)} \geq \epsilon_i(M) \prod_{k=i}^{m} \epsilon_k(G)$$

Applying this estimate to the leader node case under assumptions $(G)_{\mathrm{u}}$ and $(M)_{\mathrm{u}}^{(m)}$ we obtain the following two time uniform estimates:

$$\frac{\inf_{x_i \in E_i} G_{\ell_i,\ell_t}(x_i)}{||G_{\ell_i,\ell_t}||} \geq \epsilon_{\mathrm{u}}(M)\epsilon_{\mathrm{u}}^{mK_{\mathrm{u}}}(G)$$
$$\frac{||G_{\ell_i,\ell_t}||}{\eta_i(G_{\ell_i,\ell_t})} \leq \epsilon_{\mathrm{u}}^{-1}(M)\epsilon_{\mathrm{u}}^{-mK_{\mathrm{u}}}(G)$$

Thus Proposition 6.1 implies that in the case of leader node and under assumptions $(G)_{\mathrm{u}}$ and $(M)_{\mathrm{u}}^{(m)}$ the error propagation in the sequential Feynman-Kac filter can be characterized, with the abuse of notation $h_i = h_{i,t}^{(\mu_i)}$, as follows:

$$|[\Phi_{\ell_i,\ell_t}(\eta_i) - \Phi_{\ell_i,\ell_t}(\mu_i)](h_t)| \leq \left(1 - \epsilon_{\mathrm{u}}^2(M)\epsilon_{\mathrm{u}}^{(m-1)K_{\mathrm{u}}}(G)\right)^{\lfloor (t-i)/m \rfloor} \frac{2 - \epsilon_{\mathrm{u}}(M)\epsilon_{\mathrm{u}}^{mK_{\mathrm{u}}}(G)}{\epsilon_{\mathrm{u}}(M)\epsilon_{\mathrm{u}}^{mK_{\mathrm{u}}}(G)} |(\eta_i - \mu_i)(h_i)| \tag{6.9}$$

These results will be applied to the analysis of the leader node approximation error propagation. The next section presents the analysis of *local* errors arising during particle (sub)sampling.

6.1.3 Local Approximation Error Analysis

We have previously seen that the true filtering distribution and its N-particle approximation at time t can be related as follows [114]:

$$\eta_t^N - \eta_t = \sum_{i=0}^{t} \left[\Phi_{\ell_i,\ell_t}(\eta_i^N) - \Phi_{\ell_i,\ell_t}(\Phi_{\ell_i}(\eta_{i-1}^N))\right]. \tag{6.10}$$

This demonstrates that by combining (6.10) and (6.9) the global approximation error, $\eta_t^N - \eta_t$, can be related to the sequence of local approximation errors $\eta_i^N - \Phi_{\ell_i}(\eta_{i-1}^N), i = 0, \ldots, t$. Recall that $\eta_i^N = S^N(\Phi_{\ell_i}(\eta_{i-1}^N))$. This implies that $\mathbb{E}\{\eta_i^N | \mathcal{F}_{i-1}\} = \Phi_{\ell_i}(\eta_{i-1}^N)$

and we are therefore interested in the analysis of the errors of the form $[S^N(P) - P](h)$ for some $P \in \mathcal{P}(E)$ and $h \in \mathcal{B}_b(E)$. Thus we proceed with the analysis of these local approximation errors that will later be related to the global approximation error using the above decomposition.

Local L_p error bounds

Lemma 5.1 provides bounds for the weak L_p error of the approximations of the form $S^N(P) - P$. This result can be further tightened and generalized for an arbitrary non-integer $p \geq 0$ with a sequence of constants $c(p)$ that are smaller in magnitude than $d(p)$. The following lemma provides such a generalization.

Lemma 6.1. *Suppose $P \in \mathcal{P}(E)$, then for any $p \geq 1$ and a $\mathcal{E}$-measurable function h with finite oscillations we have*

$$\mathbb{E}\{|[P - S^N(P)](h)|^p\}^{\frac{1}{p}} \leq c(p)^{\frac{1}{p}} \frac{\sigma(h)}{\sqrt{N}},$$

where $c(p)$ is[3]

$$c(p) = \begin{cases} 1 & \text{if } 1/2 \leq p \leq 1 \\ 2^{-p/2} p \Gamma[p/2] & \text{if } p > 1 \end{cases}$$

and $\Gamma[\cdot]$ is the Gamma function.

Proof. Since $\mathbb{E}\{[P - S^N(P)](h)\} = P(h) - P(h) = 0$, we have from Chernov-Hoeffding inequality:

$$\mathbb{P}\{|[P - S^N(P)](h)| \geq \epsilon\} \leq 2e^{-\frac{2N\epsilon^2}{\sigma^2(h)}}$$

We note that

$$\mathbb{P}\{|[P - S^N(P)](h)|^p \geq \epsilon\} = \mathbb{P}\{|[P - S^N(P)](h)| \geq \epsilon^{1/p}\}$$

[3] Although Lemma 6.1 only applies for $p \geq 1$, we also define $c(p)$ over the range $1/2 < p < 1$, because of its use in conjunction with Theorem 6.4 where p is allowed to take values in this extended range.

and we have from the Chernov-Hoeffding inequality:

$$\mathbb{P}\{|[P - S^N(P)](h)| \geq \epsilon^{1/p}\} \leq 2e^{-2N\epsilon^{2/p}/\sigma^2(h)}$$

Next we recall the following property:

$$\mathbb{E}\{|[P - S^N(P)](h)|\} = \int_0^\infty \mathbb{P}\{|[P - S^N(P)](h)| \geq \epsilon\}\mathrm{d}\epsilon$$

And finally we obtain:

$$\begin{aligned}
\mathbb{E}\{|[P - S^N(P)](h)|^p\}^{\frac{1}{p}} &= \left[2\int_0^\infty \mathbb{P}\{|[P - S^N(P)](h)| \geq \epsilon^{1/p}\}\mathrm{d}\epsilon\right]^{\frac{1}{p}} \\
&\leq \left[2\int_0^\infty e^{-2N\epsilon^{2/p}/\sigma^2(h)}\mathrm{d}\epsilon\right]^{\frac{1}{p}} \\
&= \left[\sigma^p(h)p(2N)^{-\frac{p}{2}}\Gamma\left[\frac{p}{2}\right]\right]^{\frac{1}{p}}
\end{aligned}$$

Noting that since according to Lemma 5.1, $d(1) = 1$, we can use $c(1) = 1$ instead of $c(1) = 2^{-1/2}\Gamma[1/2] = \sqrt{\pi/2}$, completes the proof. □

We note that the sequence of constants $c(p)$ provides tighter error bounds than the sequence $d(p)$: $c(p) \leq d(p)$. The comparison of the bounds presented in Lemma 6.1 with the bounds in Lemma 5.1 is outlined in the Appendix B.1.

Local moment generating function

The following theorem provides a bound on the moment generating function of the empirical measure $m(X)$. It is used in Section 6.1.4 to obtain exponential inequalities for the subsample approximation leader node particle filter. The result employs Lemma 6.1 to tighten Theorem 7.3.1 of [114] that is based on Lemma 5.1. For a comparison of the two bounds see Appendix B.2.

Theorem 6.1. *For any sequence of $\mathcal{E}$–measurable functions $(h_k)_{k\geq 1}$ such that $\mu_k(h_k) = 0$ for all $k \geq 1$ and $\sigma(h) < \infty$, we have for any ε*

$$\mathbb{E}\left\{e^{\varepsilon\sqrt{N}|m(X)(h)|}\right\} \leq 1 + \varepsilon\sigma(h)\left(1 - \sqrt{\frac{\pi}{2}} + \sqrt{\frac{\pi}{2}}e^{\frac{\varepsilon^2}{8}\sigma^2(h)}\left[1 + \mathrm{Erf}\left[\frac{\varepsilon\sigma(h)}{\sqrt{8}}\right]\right]\right)$$

Proof. We first utilize the power series representation of the exponential:

$$\begin{aligned}\mathbb{E}\left\{e^{\varepsilon|m(X)(h)|}\right\} &= \sum_{n\geq 0}\frac{\varepsilon^n}{n!}\mathbb{E}\left\{|m(X)(h)|^n\right\} \\ &= \varepsilon^0\mathbb{E}\left\{|m(X)(h)|^0\right\} + \varepsilon\mathbb{E}\left\{|m(X)(h)|\right\} + \sum_{n\geq 2}\frac{\varepsilon^n}{n!}\mathbb{E}\left\{|m(X)(h)|^n\right\}\end{aligned}$$

Utilizing Lemma 6.1 we have:

$$\begin{aligned}&\mathbb{E}\left\{e^{\varepsilon|m(X)(h)|}\right\} \\ &\leq 1 + \frac{\varepsilon\sigma(h)}{\sqrt{N}} + \sum_{n\geq 2}\frac{\varepsilon^n}{n!}\sigma^n(h)n(2N)^{-n/2}\Gamma\left[n/2\right] \\ &= 1 + \frac{\varepsilon\sigma(h)}{\sqrt{N}} + \sum_{n\geq 2}\left[\frac{\varepsilon\sigma(h)}{(2N)^{1/2}}\right]^n\frac{\Gamma\left[n/2\right]}{(n-1)!} \\ &= 1 + \frac{\varepsilon\sigma(h)}{\sqrt{N}} - \frac{\varepsilon\sigma(h)\sqrt{\pi}}{\sqrt{2N}} + \frac{\varepsilon\sigma(h)\sqrt{\pi}}{\sqrt{2N}}e^{\frac{\varepsilon^2\sigma^2(h)}{8N}}\left[1 + \mathrm{Erf}\left[\frac{\varepsilon\sigma(h)}{\sqrt{8N}}\right]\right]\end{aligned}$$

Choosing $\varepsilon = \varepsilon\sqrt{N}$ and rearranging terms completes the proof. □

The following corollary containing a more tractable variation of the previous theorem can be useful for deriving the exponential inequalities for the particle approximations of the Feynman-Kac models.

Corollary 6.1. *For any sequence of $\mathcal{E}$–measurable functions $(h_k)_{k\geq 1}$ such that $\mu_k(h_k) = 0$ for all $k \geq 1$ we have for any ε*

$$\sigma(h) < \infty \Longrightarrow \mathbb{E}\left\{e^{\varepsilon\sqrt{N}|m(X)(h)|}\right\} \leq \left(1 + \sqrt{2\pi}\varepsilon\sigma(h)\right)e^{\frac{\varepsilon^2}{8}\sigma^2(h)}$$

Proof. The proof is straightforward since $\sup_x \mathrm{Erf}(x) = 1$, $1 - \sqrt{\pi/2} < 0$ and $e^{\frac{\varepsilon^2}{8}\sigma^2(h)} \geq 1$. □

We note that the simplified estimate of the moment-generating function in Corollary 6.1 is much tighter than the bound in Theorem 7.3.1 of [114] for asymptotically large deviations ε while the more complex bound in Theorem 6.1 is uniformly tighter over the range of ε.

6.1.4 Time Uniform Error Bounds and Exponential Inequalities

We now analyze the global approximation error for the leader node particle filtering with intermittent subsampling. We first present a theorem that specifies a time uniform bound on the weak-sense L_p error.

Theorem 6.2. *Suppose assumptions* $(G)_u$ *and* $(M)_u^{(m)}$ *hold. Suppose further that* $\mathbb{P}\{\delta_i = 1\} \leq q_u$ *for any* $i \geq 0$ *and* $0 \leq q_u \leq 2/3$. *Then for a positive integer* χ *such that* $N = \chi N_b$, $t \geq 0$, $p \geq 1$ *and* $h_t \in \mathrm{Osc}_1(E_t)$ *we have the time uniform estimate*

$$\sup_{t\geq 0} \mathbb{E}\left\{|[\eta_t^N - \eta_t](h_t)|^p\right\}^{1/p} \leq \frac{\epsilon_{u,m} c^{1/p}(p)}{\sqrt{N}} \left(q_u^{1/p}\sqrt{\chi} + (1-q_u)^{1/p}\right)$$

where the constant $\epsilon_{u,m}$ *is:*

$$\epsilon_{u,m} = \frac{m(2 - \epsilon_u(M)\epsilon_u^{mK_u}(G))}{\epsilon_u^3(M)\epsilon_u^{(2m-1)K_u}(G)}. \tag{6.11}$$

Proof. We begin by applying Minkowski's inequality to (5.54)

$$\mathbb{E}\left\{\left|[\eta_t^N - \eta_t](h_t)\right|^p\right\}^{\frac{1}{p}} \leq \sum_{i=0}^{t} \mathbb{E}\left\{\left|\left[\Phi_{\ell_i,\ell_t}(\eta_i^N) - \Phi_{\ell_i,\ell_t}(\Phi_{\ell_i}(\eta_{i-1}^N))\right](h_i)\right|^p\right\}^{\frac{1}{p}}$$

and then applying Proposition 6.1:

$$\begin{aligned}
&\sum_{i=0}^{t} \mathbb{E}\left\{\left|\left[\Phi_{\ell_i,\ell_t}(\eta_i^N) - \Phi_{\ell_i,\ell_t}(\Phi_{\ell_i}(\eta_{i-1}^N))\right](h_i)\right|^p\right\}^{\frac{1}{p}} \\
&\leq \sum_{i=0}^{t} \mathbb{E}\left\{\left|\beta(P_{\ell_i,\ell_t})\frac{||G_{\ell_i,\ell_t}||}{\eta_i(G_{\ell_i,\ell_t})}\left[2 - \frac{\inf_{y_i\in E_i} G_{\ell_i,\ell_t}(y_i)}{||G_{\ell_i,\ell_t}||}\right]\right|^p \left|\left[\eta_i^N - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{1/p}.
\end{aligned}$$

Furthermore, applying (6.9) we have:

$$\begin{aligned}
&\sum_{i=0}^{t} \mathbb{E}\left\{\left|\left[\Phi_{\ell_i,\ell_t}(\eta_i^N) - \Phi_{\ell_i,\ell_t}(\Phi_{\ell_i}(\eta_{i-1}^N))\right](h_i)\right|^p\right\}^{\frac{1}{p}} \\
&\leq \frac{2-\epsilon_{\mathrm{u}}(M)\epsilon_{\mathrm{u}}^{mK_{\mathrm{u}}}(G)}{\epsilon_{\mathrm{u}}(M)\epsilon_{\mathrm{u}}^{mK_{\mathrm{u}}}(G)} \\
&\qquad \times \sum_{i=0}^{t} \left(1-\epsilon_{\mathrm{u}}^2(M)\epsilon_{\mathrm{u}}^{(m-1)K_{\mathrm{u}}}(G)\right)^{\lfloor (t-i)/m \rfloor} \mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{1/p}.
\end{aligned}$$

Next we analyze each individual expectation comprising the sum above. In particular, using the structure of the algorithm defined in (6.5) and the definition of sampling operator introduced in (5.46) we can rewrite the terms under the above sum in the following explicit way:

$$\begin{aligned}
&\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}} \\
&\quad = \mathbb{E}\left\{\left|\left[\delta_i S^N \circ S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N)) + (1-\delta_i)S^N(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}}
\end{aligned} \tag{6.12}$$

Grouping the terms and using Minkowski's inequality again we conclude the following:

$$\begin{aligned}
\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}} \leq\ & \mathbb{E}\left\{\left|\delta_i\left[S^N \circ S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}} \\
&+ \mathbb{E}\left\{\left|(1-\delta_i)\left[S^N(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}}.
\end{aligned}$$

Adding and subtracting $\delta_i S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N))$ in the first term on the right and applying Minkowski's inequality again we have:

$$\begin{aligned}
&\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}} \\
&\qquad \leq \mathbb{E}\left\{\left|\delta_i\left[S^N \circ S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N)) - S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N))\right](h_i)\right|^p\right\}^{\frac{1}{p}} \\
&\qquad\quad + \mathbb{E}\left\{\left|\delta_i\left[S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}} \\
&\qquad\quad + \mathbb{E}\left\{\left|(1-\delta_i)\left[S^N(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}}
\end{aligned} \tag{6.13}$$

We see that each error term comprising the sum splits into three individual terms, describing the approximation paths the leader node algorithm can follow at time i. If $N = \chi N_{\mathrm{b}}$ then the N-particle approximation after subsampling can be recovered from the N_{b}-particle

approximation without error by replicating the N_{b}-particle approximation χ times. Thus the first term in (6.13) is zero.

The analysis of the remaining two terms is similar. We first concentrate on the second term. Recall that $\delta_i = \delta_i(\{\xi_{i-1}^j\}_{j=1}^N, Y_i^{\mathfrak{S}_{\ell_i}})$. Thus given the σ-algebra $\mathcal{F}_{i-1}$ introduced in (5.27) and the realization of the current measurement, $Y_i^{\mathfrak{S}_{\ell_i}} = y_i^{\mathfrak{S}_{\ell_i}}$, the output of the decision rule is independent of the sampling error, $[S^N(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)](h_i)$. We exploit this Markovian nature of the decision rule and apply Lemma 6.1 to the conditional expectation rendering the following bound:

$$\begin{aligned}
&\mathbb{E}\left\{\left|\delta_i\left[S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{1/p} \\
&= \mathbb{E}\left\{\delta_i \mathbb{E}\left\{\left|\left[S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p \middle| \mathcal{F}_{i-1}, Y_i^{\mathfrak{S}_{\ell_i}} = y_i^{\mathfrak{S}_{\ell_i}}\right\}\right\}^{1/p} \\
&\leq \frac{c^{1/p}(p)}{\sqrt{N_{\mathrm{b}}}} q_i^{1/p}
\end{aligned} \tag{6.14}$$

Combining the analysis results for all three terms we obtain:

$$\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_t)\right|^p\right\}^{1/p} \leq c^{1/p}(p)\left(q_i^{1/p}\frac{1}{\sqrt{N_{\mathrm{b}}}} + (1-q_i)^{1/p}\frac{1}{\sqrt{N}}\right)$$

We note that the expression in brackets has the form $\varphi(q_i) = q_i^{1/p}(\alpha+\beta) + (1-q_i)^{1/p}\alpha$ for some $\beta > \alpha \geq 0$. For $p \geq 1$, $\varphi(q_i)$ has maximum at $q_i = q_{\max}$ with

$$q_{\max} = \frac{1}{1 + \left[\frac{\alpha+\beta}{\alpha}\right]^{p/(1-p)}}.$$

We have that $\varphi(q_i)$ is non-decreasing on $q_i \in [0, q_{\max}]$ and non-increasing on $q_i \in (q_{\max}, 1]$. Noting that $[(\alpha+\beta)/\alpha]^{p/(1-p)}$ is increasing in p we obtain:

$$q_{\max} \geq \frac{1}{1 + \left[\frac{\alpha}{\alpha+\beta}\right]} \geq \inf_{\beta:\beta>\alpha} \frac{1}{1 + \left[\frac{\alpha}{\alpha+\beta}\right]} = 2/3.$$

Thus if $q_{\mathrm{u}} \leq 2/3 \leq q_{\max}$ then for any $i \geq 0$ we have the time uniform estimate:

$$\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_t)\right|^p\right\}^{1/p} \leq c^{1/p}(p)\left(q_{\mathrm{u}}^{1/p}\frac{1}{\sqrt{N_{\mathrm{b}}}} + (1-q_{\mathrm{u}})^{1/p}\frac{1}{\sqrt{N}}\right)$$

Finally, noting [114] that:

$$\sum_{i=0}^{t} \left(1 - \epsilon_{\mathrm{u}}^2(M)\epsilon_{\mathrm{u}}^{(m-1)K_{\mathrm{u}}}(G)\right)^{\lfloor (t-i)/m \rfloor} \leq \frac{m}{\epsilon_{\mathrm{u}}^2(M)\epsilon_{\mathrm{u}}^{(m-1)K_{\mathrm{u}}}(G)} \tag{6.15}$$

we complete the proof of theorem. □

The result can be generalized to cases where N is not an integer multiple of N_b, at the expense of a slight loosening of the bound.

Corollary 6.2. *Suppose the assumptions of Theorem 6.2 apply, except we allow any integer $N_b < N$. Then for any $t \geq 0$, $p \geq 1$ and $h_t \in \mathrm{Osc}_1(E_t)$ we have the time uniform estimate*

$$\sup_{t\geq 0} \mathbb{E}\left\{|[\eta_t^N - \eta_t](h_t)|^p\right\}^{1/p} \leq \epsilon_{u,m} c^{1/p}(p) \left(q_u^{1/p} \left[\frac{1}{\sqrt{N}} + \frac{1}{\sqrt{N_b}} \right] + (1-q_u)^{1/p} \frac{1}{\sqrt{N}} \right)$$

where the constant $\epsilon_{u,m}$ is defined as in (6.11).

The corollary follows by allowing for sampling error to arise in the first term in (6.13):

$$\mathbb{E}\left\{ \left| \delta_i \left[S^N \circ S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N)) - S^{N_{\mathrm{b}}}(\Phi_{\ell_i}(\eta_{i-1}^N)) \right](h_i) \right|^p \right\}^{\frac{1}{p}} \leq \frac{c^{1/p}(p)}{\sqrt{N}} q_i^{1/p}.$$

and incorporating this error bound throughout the rest of the proof of Theorem 6.2.

Corollary 6.3. *Under the same assumptions as Theorem 6.2, we have for any $p \in \mathbb{N}$, $0 \leq q_u \leq 1$ and $h_t \in \mathrm{Osc}_1(E_t)$ the time uniform estimate*

$$\sup_{t\geq 0} \mathbb{E}\left\{|[\eta_t^N - \eta_t](h_t)|^p\right\}^{1/p} \leq \frac{\epsilon_{u,m} c^{1/p}(p)}{\sqrt{N}} \left(q_u \chi^{p/2} + (1-q_u) \right)^{1/p} \tag{6.16}$$

Proof. Starting with (6.12), we perform a different error decomposition:

$$
\begin{aligned}
&\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}} \\
&= \mathbb{E}\Big\{\Big|\delta_i\left[S^N \circ S^{N_\mathrm{b}}(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i) \\
&+ (1-\delta_i)\left[S^N(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\Big|^p\Big\}^{\frac{1}{p}} \\
&\leq \mathbb{E}\left\{\delta_i^p\left|\left[S^N \circ S^{N_\mathrm{b}}(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right. \\
&+ \sum_{k=1}^{p-1}\binom{p}{k}\delta_i^k(1-\delta_i)^{p-k}\left|\left[S^N \circ S^{N_\mathrm{b}}(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^k \\
&\times \left|\left[S^N(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^{p-k} \\
&+ \left.(1-\delta_i)^p\left|\left[S^N(\Phi_{\ell_i}(\eta_{i-1}^N)) - \Phi_{\ell_i}(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{1/p}.
\end{aligned} \tag{6.17}
$$

We observe that $\delta_i(1-\delta_i) = 0$ and that if $N = \chi N_\mathrm{b}$ for integer χ, we can reconstruct an N-sample representation from the N_b sample with no additional error. Thus:

$$
\begin{aligned}
\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}} \leq \mathbb{E}\Big\{&\delta_i\left|\left[S^{N_\mathrm{b}}(\Phi_i(\eta_{i-1}^N)) - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p \\
&+ (1-\delta_i)\left|\left[S^N(\Phi_i(\eta_{i-1}^N)) - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\Big\}^{1/p}.
\end{aligned}
$$

Applying the same conditioning as in (6.14) and utilizing Lemma 6.1

$$
\begin{aligned}
\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}} &\leq \left(\frac{q_i c(p)}{N_\mathrm{b}^{p/2}} + \frac{(1-q_i)c(p)}{N^{p/2}}\right)^{1/p} \\
&= \frac{c(p)^{1/p}}{\sqrt{N}}\left(q_i\chi^{p/2} + (1-q_i)\right)^{1/p}
\end{aligned} \tag{6.18}
$$

We note that $\chi \geq 1$ and $q_i\chi + (1-q_i) \leq q_\mathrm{u}\chi + (1-q_\mathrm{u})$ under the assumption $q_i \leq q_\mathrm{u}$. The final step in the proof involves applying (6.15) as in the proof of Theorem 6.2. □

The intuitive implication of Theorem 6.2 and Corollary 6.3 is that rare approximation events have limited effect on the average error performance of the subsample approximation particle filter. The L_2 error bound for the standard particle filter is the same as (6.16) of Corollary 6.3 taken with $p = 2$, except for the term $(q_\mathrm{u}\chi + (1-q_\mathrm{u}))^{1/2}$. This expression thus quantifies the performance deterioration, in terms of L_2 error bounds, due to the subsample approximation step. If the compression factor, χ, is $\chi = 10$, and subsample approximations

occur with probability 0.1, then the deterioration of the root mean-square performance captured, in terms of bounds, by the factor $(0.1 \times 10 + (1 - 0.1))^{1/2}$ is around 40%. The communication overhead, on the other hand, represented by the total number of particles transmitted during leader node hand-off, is reduced by a factor of 10. The compressed particle cloud exchanges are most efficient in scenarios where the targets being tracked have slow dynamics and the density of leader nodes is relatively low (both implying rare hand-off events), but the tracking accuracy requirements and leader-to-leader communication costs are high.

Theorem 6.3 below provides the exponential estimate for the probability of large deviations of the approximate Feynman-Kac flows associated with the subsample approximation particle filter. Before proceeding to Theorem 6.3 we state a technical lemma.

Lemma 6.2. *Let X and Y be real random variables taking values in $\mathcal{X} \subseteq \mathbb{R}$ and $\mathcal{Y} \subseteq \mathbb{R}$ and let the joint distribution of these variables be $P_{X,Y}(\mathrm{d}(x, y))$. Then for any $\varepsilon \in \mathbb{R}$ we have:*

$$\mathbb{P}\{X + Y \geq \varepsilon\} \leq \mathbb{P}\{X \geq \varepsilon/2\} + \mathbb{P}\{Y \geq \varepsilon/2\}$$

Proof. Let us define subsets $\mathcal{X}_{x \geq y} \subseteq \mathcal{X}$, $\mathcal{X}_{x \geq y} = \{x \in \mathcal{X} : x \geq y, y \in \mathcal{Y}\}$ and $\mathcal{X}_{x<y} = \mathcal{X}^c_{x \geq y}$, $\mathcal{X}_{x<y} = \{x \in \mathcal{X} : x < y, y \in \mathcal{Y}\}$. Denote by $\mathbf{1}_{\mathrm{cond}}$ the indicator function, taking value 1 where the condition cond holds and 0 elsewhere. Now write the explicit expression for $\mathbb{P}\{X + Y \geq \varepsilon\}$:

$$\begin{aligned}\mathbb{P}\{X + Y \geq \varepsilon\} &= \int_{\mathcal{Y}} \int_{\mathcal{X}} \mathbf{1}_{x+y \geq \varepsilon} P_{X,Y}(\mathrm{d}(x, y)) \\ &= \int_{\mathcal{Y}} \int_{\mathcal{X}_{x \geq y}} \mathbf{1}_{x+y \geq \varepsilon} P_{X,Y}(\mathrm{d}(x, y)) + \int_{\mathcal{Y}} \int_{\mathcal{X}_{x<y}} \mathbf{1}_{x+y \geq \varepsilon} P_{X,Y}(\mathrm{d}(x, y))\end{aligned}$$

Since $\mathbf{1}_{x+y \geq \varepsilon} \leq \mathbf{1}_{2x \geq \varepsilon}$ on $\mathcal{X}_{x \geq y}$ and $\mathbf{1}_{x+y \geq \varepsilon} \leq \mathbf{1}_{2y \geq \varepsilon}$ on $\mathcal{X}_{x<y}$ we have

$$\begin{aligned}\mathbb{P}\{X + Y \geq \varepsilon\} &\leq \int_{\mathcal{Y}} \int_{\mathcal{X}_{x \geq y}} \mathbf{1}_{x \geq \varepsilon/2} P_{X,Y}(\mathrm{d}(x, y)) + \int_{\mathcal{Y}} \int_{\mathcal{X}_{x<y}} \mathbf{1}_{y \geq \varepsilon/2} P_{X,Y}(\mathrm{d}(x, y)) \\ &\leq \int_{\mathcal{Y}} \int_{\mathcal{X}} \mathbf{1}_{x \geq \varepsilon/2} P_{X,Y}(\mathrm{d}(x, y)) + \int_{\mathcal{Y}} \int_{\mathcal{X}} \mathbf{1}_{y \geq \varepsilon/2} P_{X,Y}(\mathrm{d}(x, y)),\end{aligned}$$

and the claim of lemma follows. □

Theorem 6.3. *Suppose assumptions* $(G)_u$ *and* $(M)_u^{(m)}$ *hold. Suppose further that* $\mathbb{P}\{\delta_i = 1\} \leq q_u$ *for* $i \geq 0$ *and* $0 \leq q_u \leq 1$. *Then for any* $N_b < N$, $t \geq 0$ *and* $h_t \in \mathrm{Osc}_1(E_t)$ *we have*

$$\sup_{t\geq 0} \mathbb{P}\left\{|[\eta_t^N - \eta_t](h_t)| \geq \epsilon\right\} \leq \left(1 + 4\sqrt{2\pi}\frac{\varepsilon\sqrt{N}}{\epsilon_{u,m}}\right) e^{-\frac{N\varepsilon^2}{2\epsilon_{u,m}^2}} + q_u \left(1 + 4\sqrt{2\pi}\frac{\varepsilon\sqrt{N_b}}{\epsilon_{u,m}}\right) e^{-\frac{N_b\varepsilon^2}{2\epsilon_{u,m}^2}}$$

Proof. Using the triangle inequality in (5.54) we have

$$\left|[\eta_t^N - \eta_t](h_t)\right| \leq \sum_{i=0}^{t} \left|\left[\Phi_{i,t}(\eta_i^N) - \Phi_{i,t}(\Phi_i(\eta_{i-1}^N))\right](h_i)\right|$$

Following the methodology presented in Theorem 6.2 and denoting $\omega_i = \left(1 - \epsilon_{\mathrm{u}}^2(M)\epsilon_{\mathrm{u}}^{(m-1)}(G)\right)^{\lfloor (t-i)/m \rfloor}$ and $a = \frac{2-\epsilon_{\mathrm{u}}(M)\epsilon_{\mathrm{u}}^{mK_{\mathrm{u}}}(G)}{\epsilon_{\mathrm{u}}(M)\epsilon_{\mathrm{u}}^{mK_{\mathrm{u}}}(G)}$ we have:

$$\left|[\eta_t^N - \eta_t](h_t)\right| \leq a \sum_{i=0}^{t} \omega_i \left|\left[\eta_i^N - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|.$$

Using the structure of the algorithm defined in (6.5) and the definition of sampling operator introduced in (5.46) we obtain the following (similarly to Theorem 6.2):

$$\begin{aligned}
\left|[\eta_t^N - \eta_t](h_t)\right| &\leq a \sum_{i=0}^{t} \omega_i \delta_i \left|\left[S^N \circ S^{N_{\mathrm{b}}}(\Phi_i(\eta_{i-1}^N)) - S^{N_{\mathrm{b}}}(\Phi_i(\eta_{i-1}^N))\right](h_i)\right| \\
&\quad + a \sum_{i=0}^{t} \omega_i (1-\delta_i) \left|\left[S^N(\Phi_i(\eta_{i-1}^N)) - \Phi_i(\eta_{i-1}^N)\right](h_i)\right| \\
&\quad + a \sum_{i=0}^{t} \omega_i \delta_i \left|\left[S^{N_{\mathrm{b}}}(\Phi_i(\eta_{i-1}^N)) - \Phi_i(\eta_{i-1}^N)\right](h_i)\right| \\
&= Z_1 + Z_2,
\end{aligned}$$

where

$$Z_1 = a\sum_{i=0}^{t}\omega_i\delta_i\left|\left[S^N\circ S^{N_\mathrm{b}}(\Phi_i(\eta_{i-1}^N)) - S^{N_\mathrm{b}}(\Phi_i(\eta_{i-1}^N))\right](h_i)\right| + a\sum_{i=0}^{t}\omega_i(1-\delta_i)\left|\left[S^N(\Phi_i(\eta_{i-1}^N)) - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|$$

$$Z_2 = a\sum_{i=0}^{t}\omega_i\delta_i\left|\left[S^{N_\mathrm{b}}(\Phi_i(\eta_{i-1}^N)) - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|$$

Noting that

$$\sup_{t\geq 0}\ \mathbb{P}\left\{|[\eta_t^N - \eta_t](h_t)| \geq \epsilon\right\} \leq \sup_{t\geq 0}\ \mathbb{P}\left\{Z_1 + Z_2 \geq \epsilon\right\}$$

and applying Lemma 6.2 we have:

$$\sup_{t\geq 0}\ \mathbb{P}\left\{|[\eta_t^N - \eta_t](h_t)| \geq \epsilon\right\} \leq \sup_{t\geq 0}\ \mathbb{P}\left\{Z_1 \geq \epsilon/2\right\} + \sup_{t\geq 0}\ \mathbb{P}\left\{Z_2 \geq \epsilon/2\right\}.$$

Now applying Markov inequality we conclude:

$$\begin{aligned}\sup_{t\geq 0}\ \mathbb{P}\left\{|[\eta_t^N - \eta_t](h_t)| \geq \epsilon\right\} &\leq \sup_{t\geq 0}\ \mathbb{P}\left\{e^{\tau_1 Z_1} \geq e^{\tau_1\epsilon/2}\right\} + \sup_{t\geq 0}\ \mathbb{P}\left\{e^{\tau_2 Z_2} \geq e^{\tau_2\epsilon/2}\right\}\\ &\leq \sup_{t\geq 0}\ e^{-\tau_1\epsilon/2}\mathbb{E}\left\{e^{\tau_1 Z_1}\right\} + \sup_{t\geq 0} e^{-\tau_2\epsilon/2}\mathbb{E}\left\{e^{\tau_2 Z_2}\right\}\end{aligned}$$

Next we apply the exponential series expansion

$$\mathbb{E}\left\{e^{\tau_1 Z_1}\right\} = \sum_{n\geq 0}\frac{\tau_1^n}{n!}\mathbb{E}\{Z_1^n\} \tag{6.19}$$

and use the fact that according to the following conditioning argument and Lemma 6.1 we

have

$$\begin{aligned}
\mathbb{E}\{Z_1^n\}^{1/n} &= \left(\mathbb{E}\{Z_1^n|\delta_i=1\}\mathbb{P}\{\delta_i=1\} + \mathbb{E}\{Z_1^n|\delta_i=0\}\mathbb{P}\{\delta_i=0\}\right)^{1/n} \\
&\leq a\sum_{i=0}^{t}\omega_i\mathbb{E}\left\{\left(\delta_i\left|\left[S^N\circ S^{N_\mathrm{b}}(\Phi_i(\eta_{i-1}^N)) - S^{N_\mathrm{b}}(\Phi_i(\eta_{i-1}^N))\right](h_i)\right|\right.\right. \\
&+ \left.\left.(1-\delta_i)\left|\left[S^N(\Phi_i(\eta_{i-1}^N)) - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|\right)^n\right\}^{1/n} \\
&= a\sum_{i=0}^{t}\omega_i\left(\mathbb{P}\{\delta_i=1\}\mathbb{E}\left\{\left|\left[S^N\circ S^{N_\mathrm{b}}(\Phi_i(\eta_{i-1}^N)) - S^{N_\mathrm{b}}(\Phi_i(\eta_{i-1}^N))\right](h_i)\right|^n|\delta_i=1\right\}\right. \\
&+ \left.\mathbb{P}\{\delta_i=0\}\mathbb{E}\left\{\left|\left[S^N(\Phi_i(\eta_{i-1}^N)) - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^n|\delta_i=0\right\}\right)^{1/n} \\
&\leq a\sum_{i=0}^{t}\omega_i(q_ic(n)N^{-n/2} + (1-q_i)c(n)N^{-n/2})^{1/n} \\
&= \frac{c^{1/n}(n)}{\sqrt{N}}a\sum_{i=0}^{t}\omega_i.
\end{aligned}$$

Noting that $a\sum_{i=0}^{t}\omega_i \leq \epsilon_{\mathrm{u},m}$ we have:

$$\mathbb{E}\{Z_1^n\} \leq \epsilon_{\mathrm{u},m}^n c(n)N^{-n/2}$$

Substituting this into (6.19) and employing the same simplifications as in the proofs of Theorem 6.1 and Corollary 6.1 we obtain:

$$\begin{aligned}
e^{-\varepsilon\tau_1/2}\mathbb{E}\left\{e^{\tau_1 Z_1}\right\} &\leq \sum_{n\geq 0}\left(\frac{\tau_1\epsilon_{\mathrm{u},m}}{\sqrt{N}}\right)^n\frac{c(n)}{n!}e^{-\varepsilon\tau_1/2} \\
&\leq \left[1 + \frac{\tau_1\epsilon_{\mathrm{u},m}}{\sqrt{N}} + \sum_{n\geq 2}\left(\frac{\tau_1\epsilon_{\mathrm{u},m}}{\sqrt{2N}}\right)^n\frac{\Gamma(n/2)}{(n-1)!}\right]e^{-\varepsilon\tau_1/2} \\
&\leq \left(1 + \sqrt{2\pi}\frac{\tau_1\epsilon_{\mathrm{u},m}}{\sqrt{N}}\right)e^{\frac{\tau_1^2\epsilon_{\mathrm{u},m}^2}{8N} - \varepsilon\tau_1/2}
\end{aligned}$$

Choosing $\tau_1 = \frac{2\varepsilon N}{\epsilon_{\mathrm{u},m}^2}$ we have

$$e^{-\varepsilon\tau_1/2}\mathbb{E}\left\{e^{\tau_1 Z_1}\right\} \leq \left(1 + 4\sqrt{2\pi}\frac{\varepsilon\sqrt{N}}{\epsilon_{\mathrm{u},m}}\right) e^{-\frac{N\varepsilon^2}{2\epsilon_{\mathrm{u},m}^2}}$$

Similar analysis yields

$$e^{-\varepsilon\tau_2/2}\mathbb{E}\left\{e^{\tau_2 Z_2}\right\} \leq q_{\mathrm{u}}\left(1 + \sqrt{2\pi}\frac{\tau_2\epsilon_{\mathrm{u},m}}{\sqrt{N_{\mathrm{b}}}}\right) e^{\frac{\tau_2^2\epsilon_{\mathrm{u},m}^2}{8N_{\mathrm{b}}} - \varepsilon\tau_2/2},$$

which after choosing $\tau_2 = \frac{2N_{\mathrm{b}}\varepsilon}{\epsilon_{\mathrm{u},m}^2}$ results in:

$$e^{-\varepsilon\tau_2/2}\mathbb{E}\left\{e^{\tau_2 Z_2}\right\} \leq q_{\mathrm{u}}\left(1 + 4\sqrt{2\pi}\frac{\varepsilon\sqrt{N_{\mathrm{b}}}}{\epsilon_{\mathrm{u},m}}\right) e^{-\frac{N_{\mathrm{b}}\varepsilon^2}{2\epsilon_{\mathrm{u},m}^2}}.$$

This completes the proof. □

6.2 Leader Node Particle Filtering with Intermittent Parametric Approximations

In this section we analyze the error behavior of a particle filter that incorporates intermittent parametric mixture estimation of the filtering density. Recall that $(E, \mathcal{E})$ is a measurable space and λ is a σ-finite measure on $\mathcal{E}$. Throughout this section it is assumed that the underlying distribution has a density if its Radon-Nikodym derivative with respect to λ exists.

It is assumed that with the sequence of the approximate filtering distributions, $\Phi_i(\eta_{i-1}^N)(\mathrm{d}x_i)$, there exists an associated and well-behaved sequence of approximate filtering densities $\frac{1}{\mathrm{d}x_i}\Phi_i(\eta_{i-1}^N)(\mathrm{d}x_i)$ so that the mixture density estimation problem is well-defined. The main result of the section, constituted in Theorem 6.5, is a time uniform, weak-sense L_p error bound characterizing the expected behavior of the parametric approximation leader node particle filter.

6.2.1 Parametric Approximation Leader Node Particle Filter Algorithm

For this algorithm, the binary variable δ_t now indicates whether a parametric approximation is performed at time-step t. Again we assume that it is the outcome of a decision function based on the set of particles $\{\xi_{t-1}^k\}_{k=1}^N$ and observations $Y_t^{\mathfrak{S}_{\ell_t}}$ at the current time-step. When it employs parametric approximation, the leader node particle filter can be represented as follows.

$$\begin{aligned}\Phi_{\ell_t}(\eta_{t-1}^N) \Rightarrow \eta_t^N \Rrightarrow \widehat{\eta}_t^{N_\mathrm{p}} \longrightarrow \widehat{\eta}_t^{N_\mathrm{p}} \Rightarrow \eta_t^N \quad &\text{if } \delta_t = 1,\\ \Phi_{\ell_t}(\eta_{t-1}^N) \Rightarrow \eta_t^N \quad &\text{if } \delta_t = 0\end{aligned}$$

Here the $\Rrightarrow$ represents the local distribution parametric approximation process and N_p is the number of the components in the mixture. As before, $\Rightarrow$ represents an N-particle sampling operation and $\longrightarrow$ represents communication between leader nodes. Thus the second particle filter we define relies upon a parametric approximation of the distribution $\Phi_i(\eta_{i-1}^N)$ based on a particle set (a sample from this distribution).

Denote by $\mathbb{W}_{N_\mathrm{p}} : \mathcal{P}(E) \to \mathcal{P}(E^{N_\mathrm{p}})$ an operator that represents a parametric mixture approximation procedure that involves N_p mixture components (we will provide a concrete example below). The parametric approximation particle filter can then be expressed in a compact form:

$$\begin{aligned}\eta_t^N = S^N \circ \mathbb{W}_{N_\mathrm{p}}(\Phi_{\ell_t}(\eta_{t-1}^N)) \quad &\text{if } \delta_t = 1,\\ \eta_t^N = S^N(\Phi_{\ell_t}(\eta_{t-1}^N)) \quad &\text{if } \delta_t = 0\end{aligned} \tag{6.20}$$

Consider the following class of bounded parametric densities:

$$\mathcal{H}_i = \left\{\phi_{\theta_i}(x) : \theta_i \in \Theta_i, a_i \leq \inf_{\theta_i, x_i} \phi_{\theta_i}(x_i), \sup_{\theta_i, x_i} \phi_{\theta_i}(x_i) \leq b_i\right\}$$

where $0 < a_i < b_i < \infty$ and $\Theta_i \subset \mathbb{R}^{d_i}$ defines the parameter space, and inf and sup are taken over Θ_i and E_i. In the setting where the intermittent approximation during leader node hand-off is accomplished using parametric approximation, we are looking for a sequence of mixture density estimators of the filtering densities. We thus define the class of bounded parametric densities, $\phi_{\theta_i}(x)$, indexing it by time-step i to emphasize that the parameterization can be time-varying. The approximation is restricted to a class of discrete

N_{p}-component convex combinations of the form:

$$\mathcal{C}_{N_{\mathrm{p}},i} = \mathrm{conv}_{N_{\mathrm{p}}}(\mathcal{H}_i) = \left\{ g : g(x) = \sum_{j=1}^{N_{\mathrm{p}}} \alpha_{i,j}\phi_{\theta_{i,j}}(x), \phi_{\theta_{i,j}} \in \mathcal{H}_i, \sum_{j=1}^{N_{\mathrm{p}}} \alpha_{i,j} = 1,\ \alpha_{i,j} \geq 0 \right\}$$

As N_{p} grows without bound, $\mathcal{C}_{N_{\mathrm{p}},i}$ converges to the class of continuous convex combinations:

$$\mathcal{C}_i = \mathrm{conv}(\mathcal{H}_i) = \left\{ g : g(x) = \int_{\Theta} \phi_\theta(x)\mathbb{P}_i(\mathrm{d}\theta), \phi_\theta \in \mathcal{H}_i \right\}$$

The general framework for the parametric greedy approximation of arbitrary cost functions is discussed in [6]. The GML Algorithm 2 is a particular instance of this more general framework proposed by Li and Barron [124]. The GML algorithm for mixture approximation is based on the greedy KL-divergence cost function minimization over classes of the type $\mathcal{C}_{N_{\mathrm{p}},i}$.

To link the Li and Barron's [124] GML maximization framework in Algorithm 2 to the minimization of KL-divergence we recall that if ν is a known distribution and μ is the KL-based fit to this distribution, the KL-divergence minimization problem can be written as follows (assuming that the corresponding densities exist):

$$\begin{aligned}
\min_{\mu\in\mathcal{P}(E)} D(\nu||\mu) &= \min_{\mu\in\mathcal{P}(E)} \int_E \log \frac{\mathrm{d}\nu}{\mathrm{d}\mu}\mathrm{d}\nu \\
&= \min_{\mu\in\mathcal{P}(E)} \left[\int_E \log \frac{\mathrm{d}\nu}{\mathrm{d}\lambda(x)}\mathrm{d}\nu - \int_E \log \frac{\mathrm{d}\mu}{\mathrm{d}\lambda(x)}\mathrm{d}\nu \right] \\
&= \max_{\mu\in\mathcal{P}(E)} \int_E \log \frac{\mathrm{d}\mu}{\mathrm{d}\lambda(x)}\mathrm{d}\nu
\end{aligned}$$

In practice ν itself is unknown, but a sample from this distribution may be available. The approximation of the true expectation with respect to ν above by the expectation with respect to its empirical counterpart, $S^N(\nu)$, leads to the maximum likelihood density

estimator:

$$\begin{aligned}\min_{\mu\in\mathcal{P}(E)} D(\nu||\mu) &= \max_{\mu\in\mathcal{P}(E)} \mathbb{E}_\nu \log \frac{\mathrm{d}\mu}{\mathrm{d}\lambda(x)} \\ &\approx \max_{\mu\in\mathcal{P}(E)} \mathbb{E}_{S^N(\nu)} \log \frac{\mathrm{d}\mu}{\mathrm{d}\lambda(x)} \\ &= \max_{\mu\in\mathcal{P}(E)} \sum_{i=1}^{N} \log \frac{\mathrm{d}\mu}{\mathrm{d}\lambda(x)}(x_i)\end{aligned}$$

Thus the error committed by resorting to the suboptimal GML Algorithm 2 consists of three contributions.

First, there is the error associated with the limitations of the approximation class $\mathcal{C}$: even the best possible $\mu \in \mathcal{C}$ will have non-zero $D(\nu||\mu)$ if $\nu \notin \mathcal{C}$. We will call this the approximation bias. Second, there is the error associated with the greedy optimization of the KL-cost function. We will call it the approximation error. Third, the error caused by approximating the true expectation by its empirical counterpart will be called the estimation error. In the following we analyze these errors for the one-step approximation and then link the results of the analysis to the overall error of the parametric approximation particle filter.

6.2.2 Local Approximation Error Analysis

The attractive features of Algorithm 2 are threefold. First, the algorithm simplifies the maximum likelihood density estimation procedure. Instead of facing the N_p-mixture estimation problem we only have to solve N_p 2-mixture estimation problems [124]. Second, there are several bounds on approximation and sampling errors of Algorithm 2 in terms of KL-divergence (see [124] and [125]). Third, it was shown [124, 125] that the performance of the greedy algorithm converges to the performance of the optimal mixture estimation algorithm as N and N_p become large. Thus if these conditions hold, the results obtained for the parametric approximation particle filter that uses GML are also applicable if other density estimators are employed.

The goal of this section is to extend the existing results and perform the L_p error analysis of the GML algorithm. The next result reveals the L_p error bound characterizing the average performance of the GML algorithm. One of the components of the bound is the packing number $\mathcal{D}(\varepsilon, \mathcal{H}, d_N)$, which is the the maximum number of ε-separated points

in $\mathcal{H}$ (the class of parametric density functions) under the empirical semimetric d_N.

Theorem 6.4. *Suppose $\widehat{g}^{N_p} \in \mathcal{C}_{N_p}$ is constructed using Algorithm 2 and $\widehat{\mathcal{G}}^{N_p} \in \mathcal{P}(E)$ is the distribution associated with $\widehat{g}^{N_p}$. Suppose further that there exists density f associated with the target distribution $F \in \mathcal{P}(E)$. Then for any $h \in \mathcal{B}_b(E)$ with $||h||_{\text{osc}} \leq 1$, $N, N_p \in \mathbb{N}$, and $p \geq 1$ we have:*

$$\mathbb{E}\left\{|[\widehat{\mathcal{G}}^{N_p} - F](h)|^p\right\}^{1/p} \leq \sqrt{2}\left[\frac{8}{a\sqrt{N}}\Big(2c^{2/p}(p/2) + (p/4)!C\mathbb{E}\int_0^b \sqrt{\log\left(1+\mathcal{D}(\varepsilon,\mathcal{H},d_N)\right)}\mathrm{d}\varepsilon\Big) + \frac{\gamma c_{f,\mathcal{C}}^2}{N_p} + D(f||\mathcal{C})\right]^{1/2}$$

where C is a universal constant[4]*.*

Proof. Using Pinsker's inequality, $\int |f-g| \leq \sqrt{2D(f||g)}$, [130] we have

$$\begin{aligned}
\mathbb{E}\left\{|[\widehat{\mathcal{G}}^{N_\mathrm{P}} - F](h)|^p\right\}^{1/p} &= \mathbb{E}\left\{\left(\int_E [\widehat{g}^{N_\mathrm{P}}(x) - f(x)]h(x)\mathrm{d}x\right)^p\right\}^{1/p} \\
&\leq ||h||\mathbb{E}\left\{\left(\int_E |\widehat{g}^{N_\mathrm{P}}(x) - f(x)|\mathrm{d}x\right)^p\right\}^{1/p} \\
&\leq \mathbb{E}\left\{\left(\sqrt{2D(f||\widehat{g}^{N_\mathrm{P}})}\right)^p\right\}^{1/p} \\
&= \sqrt{2}\left[\mathbb{E}\left\{D(f||\widehat{g}^{N_\mathrm{P}})^{p/2}\right\}^{2/p}\right]^{1/2}
\end{aligned}$$

Now, suppose $p \geq 2$. The following decomposition can be used to analyze the previous expression:

$$D(f||\widehat{g}^{N_\mathrm{P}}) = D(f||\widehat{g}^{N_\mathrm{P}}) - D(f||\mathcal{C}) + D(f||\mathcal{C})$$

Denoting $g^* = \arg\min_{g\in\mathcal{C}} D(f||g)$ we have the following modification of the decomposition

[4] See [129] for details.

proposed by Rakhlin et al. in [125]:

$$\begin{aligned} D(f||\widehat{g}^{N_\mathrm{p}}) - D(f||\mathcal{C}) &= -\int \log \widehat{g}^{N_\mathrm{p}}(x) F(\mathrm{d}x) + \int \log g^*(x) F(\mathrm{d}x) \\ &= -\int \log \widehat{g}^{N_\mathrm{p}}(x) F(\mathrm{d}x) + \frac{1}{N}\sum_{i=1}^{N} \log \widehat{g}^{N_\mathrm{p}}(x_i) \\ &+ \frac{1}{N}\sum_{i=1}^{N} \log g^*(x_i) - \frac{1}{N}\sum_{i=1}^{N} \log \widehat{g}^{N_\mathrm{p}}(x_i) \\ &+ \int \log g^*(x) F(\mathrm{d}x) - \frac{1}{N}\sum_{i=1}^{N} \log g^*(x_i) \end{aligned}$$

Applying (5.71) to the middle term we see:

$$D(f||\widehat{g}^{N_\mathrm{p}}) - D(f||\mathcal{C}) \leq |[F - S^N(F)](\log \widehat{g}^{N_\mathrm{p}})| + |[F - S^N(F)](\log g^*)| + \frac{\gamma c_{f,\mathcal{C}}^2}{N_\mathrm{p}}$$

By the definition of $D(f||\mathcal{C})$ it follows that $D(f||\widehat{g}^{N_\mathrm{p}}) - D(f||\mathcal{C}) \geq 0$ and thus we conclude:

$$\left|D(f||\widehat{g}^{N_\mathrm{p}}) - D(f||\mathcal{C})\right| \leq 2 \sup_{g\in\mathcal{C}} \left|[F - S^N(F)]\,(\log g)\right| + \frac{\gamma c_{f,\mathcal{C}}^2}{N_\mathrm{p}}$$

This allows splitting the effect of approximation and estimation errors by applying Minkowski's inequality (since $p \geq 2$):

$$\begin{aligned} \mathbb{E}\left\{D(f||\widehat{g}^{N_\mathrm{p}})^{p/2}\right\}^{2/p} &= \mathbb{E}\left\{\left|D(f||\widehat{g}^{N_\mathrm{p}}) - D(f||\mathcal{C}) + D(f||\mathcal{C})\right|^{p/2}\right\}^{2/p} \\ &\leq 2\mathbb{E}\left\{\left[\sup_{g\in\mathcal{C}}\left|[F - S^N(F)]\,(\log g)\right|\right]^{p/2}\right\}^{2/p} + \frac{\gamma c_{f,\mathcal{C}}^2}{N_\mathrm{p}} + D(f||\mathcal{C}). \end{aligned}$$

The next step of the proof makes use of a symmetrization argument. Recall that S_ε^N is the generator of the signed Rademacher measure. Using the symmetrization lemma (see e.g. Lemma 2.3.1 in [129] or Lemma 6.3 in [128]) we deduce:

$$\mathbb{E}\left\{\left[\sup_{g\in\mathcal{C}}\left|[F - S^N(F)]\,(\log g)\right|\right]^{p/2}\right\}^{2/p} \leq 2\mathbb{E}\left\{\left[\sup_{g\in\mathcal{C}}\left|S_\varepsilon^N(F)\,(\log g)\right|\right]^{p/2}\right\}^{2/p}$$

Denoting $\kappa = g - 1$ and using the fact [122] that $\varphi(\kappa) = a\log(\kappa+1)$ is a contraction and $\varphi(0) = 0$, we apply the comparison inequality (Theorem 5.4), observing that $[\cdot]^{p/2}$ is convex and increasing for $p \geq 2$ and κ is a bounded function:

$$\begin{aligned}
\mathbb{E}\left\{\left[\sup_{g\in\mathcal{C}}\left|S_\varepsilon^N(F)\left(\log g\right)\right|\right]^{p/2}\right\}^{2/p} &= \mathbb{E}\left\{\left[\frac{1}{2}\frac{2}{a}\sup_{g\in\mathcal{C}}\left|S_\varepsilon^N(F)\left(a\log(\kappa+1)\right)\right|\right]^{p/2}\right\}^{2/p} \\
&\leq \frac{2}{a}\mathbb{E}\left\{\left[\sup_{g\in\mathcal{C}}\left|S_\varepsilon^N(F)\left(g-1\right)\right|\right]^{p/2}\right\}^{2/p} \\
&\leq \frac{2}{a}\mathbb{E}\left\{\left[\sup_{g\in\mathcal{C}}\left|S_\varepsilon^N(F)(g)\right|\right]^{p/2}\right\}^{2/p} + \frac{2}{a}\mathbb{E}\left\{|S_\varepsilon^N(F)(1)|^{p/2}\right\}^{2/p}
\end{aligned}$$

Using Lemma 6.1 we have:

$$\mathbb{E}\left\{|S_\varepsilon^N(F)(1)|^{p/2}\right\}^{2/p} = \mathbb{E}\left\{\left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i\right|^{p/2}\right\}^{2/p} \leq \frac{2c^{2/p}(p/2)}{\sqrt{N}}$$

On the other hand, we have for any $g \in \mathcal{C}$ and corresponding $\phi_\theta \in \mathcal{H}$:

$$\begin{aligned}
|S_\varepsilon^N(F)(g)| &= \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i g(x_i)\right| \\
&= \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i \int_{\theta\in\Theta}\phi_\theta(x_i)\mathbb{P}(\mathrm{d}\theta)\right| \\
&= \left|\int_{\theta\in\Theta}\frac{1}{N}\sum_{i=1}^N \varepsilon_i\phi_\theta(x_i)\mathbb{P}(\mathrm{d}\theta)\right| \\
&\leq \int_{\theta\in\Theta}\left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i\phi_\theta(x_i)\right|\mathbb{P}(\mathrm{d}\theta) \\
&\leq \sup_{\theta\in\Theta}\left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i\phi_\theta(x_i)\right|
\end{aligned}$$

Thus we have

$$\mathbb{E}\left\{\left[\sup_{g\in\mathcal{C}}|S_{\varepsilon}^{N}(F)(g)|\right]^{p/2}\right\}^{2/p}\leq\mathbb{E}\left\{\left[\sup_{g\in\mathcal{H}}|S_{\varepsilon}^{N}(F)(g)|\right]^{p/2}\right\}^{2/p}$$

We next relate the L_p norm of the empirical process above with the associated Orlicz norm [114, 129] $\pi_{\psi_p}(\cdot)$. In particular, by Hoeffding's inequality the Rademacher process $S_{\varepsilon}^{N}(F)(g)$ is sub-Gaussian for the semimetric d_N [129]. Using the fact that $\mathbb{E}\{X^p\}^{1/p}\leq(p/2)!\pi_{\psi_2}(X)$ (see e.g. Lemma 7.3.5 in [114] or [129], p. 105, Problem 4) we deduce:

$$\mathbb{E}\mathbb{E}_{\varepsilon}\left\{\left[\sup_{g\in\mathcal{H}}|S_{\varepsilon}^{N}(F)(g)|\right]^{p/2}\right\}^{2/p}\leq(p/4)!\mathbb{E}\pi_{\psi_2}(\sup_{g\in\mathcal{H}}|S_{\varepsilon}^{N}(F)(g)|)$$

In addition, since $S_{\varepsilon}^{N}(F)(g)$ is sub-Gaussian, we have for some universal constant C (see Proof of Corollary 2.2.8 in [129]):

$$\mathbb{E}\pi_{\psi_2}(\sup_{g\in\mathcal{H}}|S_{\varepsilon}^{N}(F)(g)|)\leq\frac{C}{\sqrt{N}}\mathbb{E}\int_{0}^{b}\sqrt{\log\left(1+\mathcal{D}(\varepsilon,\mathcal{H},d_N)\right)}\mathrm{d}\varepsilon$$

Combining the above we have:

$$\begin{aligned}\mathbb{E}\left\{|[\widehat{\mathcal{G}}^{N_{\mathrm{P}}}-F](h)|^{p}\right\}^{1/p}\leq\sqrt{2}\Bigg[\frac{8}{a\sqrt{N}}\Big(2c^{2/p}(p/2)\\ +(p/4)!C\mathbb{E}\int_{0}^{b}\sqrt{\log\left(1+\mathcal{D}(\varepsilon,\mathcal{H},d_N)\right)}\mathrm{d}\varepsilon\Big)+\frac{\gamma c_{f,\mathcal{C}}^{2}}{N_{\mathrm{P}}}+D(f||\mathcal{C})\Bigg]^{1/2}\end{aligned}$$

Finally, suppose $1\leq p<2$. In this case using Jensen's inequality we have:

$$\mathbb{E}\left\{D(f||\widehat{g}^{N_{\mathrm{P}}})^{p/2}\right\}^{2/p}\leq\mathbb{E}\left\{D(f||\widehat{g}^{N_{\mathrm{P}}})\right\}$$

Thus the above analysis applies if we choose $p=2$ and the proof is now complete. □

Corollary 6.4. *Suppose that the assumptions of Theorem 6.4 hold. Suppose in addition*

that $f \in \mathcal{C}$ then we have for any $p \geq 1$:

$$\mathbb{E}\left\{|[\widehat{\mathcal{G}}^{N_p} - F](h)|^p\right\}^{1/p} \leq \sqrt{2}\left[\frac{8}{a\sqrt{N}}\left(2c^{2/p}(p/2)\right.\right.$$
$$\left.\left.+(p/4)!C\mathbb{E}\int_0^b \sqrt{\log\left(1+\mathcal{D}(\varepsilon,\mathcal{H},d_N)\right)}\mathrm{d}\varepsilon\right) + 4\log(3\sqrt{e}(b/a))\frac{(b/a)^2}{N_p}\right]^{1/2}$$

Proof. The proof follows from the fact that under the additional assumption we have $D(f||\mathcal{C}) = 0$. Furthermore, we note that under this assumption $c_{f,\mathcal{C}}^2 \leq (b/a)^2$ and $\gamma = 4\log(3\sqrt{e}(b/a))$ □

6.2.3 Time Uniform Error Bounds

In this section we present a result specifying time uniform error bounds for the leader node particle filter performing intermittent parametric approximation. The result links the properties of Markov transitions $M_i(x_{i-1}, \mathrm{d}x_i)$ and error bounds for the parametric GML approximation (Theorem 6.4) with the propagation of approximation errors through the Feynman-Kac operators. It is based on the following observations. For an absolutely continuous Markov kernel with the associated density $p_i(x_i|x_{i-1})$, we can write [114]:

$$M_i(x_{i-1}, \mathrm{d}x_i) = \Pr\{X_i \in \mathrm{d}x_i | X_{i-1} = x_{i-1}\} = p_i(x_i|x_{i-1})\mathrm{d}x_i = p_{\vartheta_i}(x_i)\mathrm{d}x_i,$$

where we explicitly assume that the structure of the kernel M_i can be captured by a set of parameters $\vartheta_i \in \Theta_i \subset \mathbb{R}^{d_i}$ (these parameters include the state-value x_{i-1}). We can further define a class $\mathcal{M}_i$ of such densities:

$$\mathcal{M}_i = \left\{p_{\vartheta_i}(x_i) : \vartheta_i \in \Theta_i \subset \mathbb{R}^{d_i}\right\}.$$

Thus if M_i is such that $p_{\vartheta_i}(x_i) \in \mathcal{M}_i$ and $\mathcal{M}_i \subseteq \mathcal{H}_i$ then the assumption $(M)_{\mathrm{u}}^{(m)}$ is satisfied with $m = 1$ and $\epsilon_{\mathrm{u}}(M) = a/b$, yielding for any $x_{i-1}, y_{i-1} \in E_{i-1}$:

$$M_i(x_{i-1}, \cdot) \geq \frac{a}{b} M_i(y_{i-1}, \cdot)$$

Furthermore, using the definitions of the one-step Boltzmann-Gibbs transformation and the associated Feynman-Kac operator we see that the distribution at time i is related to

the distribution at time $i-1$ as follows:

$$\begin{aligned}\eta_i &= \Phi_i(\eta_{i-1}) = \Psi_{i-1}(\eta_{i-1})M_i \\ &= \int_{E_{i-1}} M_i(x_{i-1}, \mathrm{d}x_i)\Psi_{i-1}(\eta_{i-1})(\mathrm{d}x_{i-1}) \\ &= \int_{E_{i-1}} M_i(x_{i-1}, \mathrm{d}x_i)\frac{G_{i-1}(x_{i-1})\eta_{i-1}(\mathrm{d}x_{i-1})}{\eta_{i-1}(G_{i-1})}.\end{aligned}$$

Thus for an absolutely continuous Markov kernel with $p_{\vartheta_i}(x_i) \in \mathcal{M}_i$ we can rewrite the previous equation with a suitable change of measure:

$$\frac{\eta_i(\mathrm{d}x_i)}{\mathrm{d}x_i} = \int_{\Theta_i} p_{\vartheta_i}(x_i)\mathbb{P}(\mathrm{d}\vartheta_i).$$

This implies that for an N-particle approximation η_{i-1}^N we have that $\frac{\eta_i(\mathrm{d}x_i)}{\mathrm{d}x_i} \in \mathrm{conv}_N(\mathcal{M}_i)$ and, as N grows without bound, we have $\frac{\eta_i(\mathrm{d}x_i)}{\mathrm{d}x_i} \in \mathrm{conv}(\mathcal{M}_i)$. Thus the performance of the GML approximation algorithm is determined by the properties of Markov transition kernel $M_i(x_{i-1}, \mathrm{d}x_i)$ and the class of approximating densities $\mathcal{H}_i$. In particular, for an absolutely continuous Markov kernel with $p_{\vartheta_i}(x_i) \in \mathcal{M}_i$ and a sufficiently rich class $\mathcal{H}_i$, such that $\mathcal{M}_i \subseteq \mathcal{H}_i$ we have asymptotically unbiased approximation:

$$D\left(\frac{\eta_i(\mathrm{d}x_i)}{\mathrm{d}x_i}\middle|\middle|\mathcal{C}\right) = 0.$$

The preceding discussion can be summarized in the form of a concise assumption:

- $(\mathcal{H})_{\mathrm{u}}$: The Markov kernels associated with the target dynamics are absolutely continuous and can be expressed in the form $M_i(x_{i-1}, \mathrm{d}x_i) = p_{\vartheta_i}(x_i)\mathrm{d}x_i$. The class of densities associated with M_i is defined as $\mathcal{M}_i = \left\{p_{\vartheta_i}(x_i) : \vartheta_i \in \Theta_i \subset \mathbb{R}^{d_i}\right\}$. For each $\mathcal{M}_i$ there exists an approximation class $\mathcal{H}_i$ and strictly positive numbers $a_{\mathrm{u}} = \inf_{i\geq 0} a_i$, $b_{\mathrm{u}} = \sup_{i\geq 0} b_i$ satisfying $0 < a_{\mathrm{u}} < b_{\mathrm{u}} < \infty$ such that for any $i \geq 0$ we have

$$\mathcal{M}_i \subseteq \mathcal{H}_i \qquad \text{and hence} \qquad M_i(x_{i-1}, \cdot) \geq \frac{a_{\mathrm{u}}}{b_{\mathrm{u}}} M_i(y_{i-1}, \cdot)$$

The following result describes the analog of Theorem 6.2 for the case of a parametric approximation particle filter using the GML algorithm.

Theorem 6.5. *Suppose assumptions* $(G)_u$ *and* $(\mathcal{H})_u$ *hold. Suppose further that* $\mathbb{P}\{\delta_i = 1\} \leq q_u$ *for any* $i \geq 0$ *and* $0 \leq q_u \leq 1$. *Then for any* $N_p, N \geq 1$, $t \geq 0$, $p \geq 1$ *and* $h_t \in \mathrm{Osc}_1(E_t)$ *we have the time uniform bound*

$$\sup_{t\geq 0}\mathbb{E}\left\{|[\eta_t^N - \eta_t](h_t)|^p\right\}^{1/p} \leq \epsilon_u \left[\frac{c^{1/p}(p)}{\sqrt{N}} + q_u^{1/p}\left[\frac{16}{a\sqrt{N}}\left(2c^{2/p}(p/2)\right.\right.\right.$$
$$\left.\left.\left. + C(p/4)!\sup_{i\geq 0}\mathbb{E}\int_0^{b_i}\sqrt{\log\left(1+\mathcal{D}(\varepsilon,\mathcal{H}_i,d_N)\right)}\mathrm{d}\varepsilon\right) + 8\log(3\sqrt{e}(b/a))\frac{(b/a)^2}{N_p}\right]^{1/2}\right]$$

where the constant ϵ_u *is:*

$$\epsilon_u = \frac{2-(a_u/b_u)\epsilon_u^{K_u}(G)}{(a_u/b_u)^3\epsilon_u^{K_u}(G)}.$$

Proof. Using the same argument as in Theorem 6.2 we have

$$\mathbb{E}\left\{|[\eta_t^N - \eta_t](f_t)|^p\right\}^{1/p}$$
$$\leq \frac{2-\epsilon_{\mathrm{u}}(M)\epsilon_{\mathrm{u}}^{K_{\mathrm{u}}}(G)}{\epsilon_{\mathrm{u}}(M)\epsilon_{\mathrm{u}}^{K_{\mathrm{u}}}(G)}\sum_{i=0}^{t}\left(1-\epsilon_{\mathrm{u}}^2(M)\right)^{(t-i)}\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{1/p}.$$

Based on the Minkowski inequality we have the decomposition for each individual expectation comprising the sum above:

$$\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}}$$
$$\leq \mathbb{E}\left\{\left|\delta_i\left[S^N(\widehat{\mathcal{G}}^{N_{\mathrm{p}}}) - \widehat{\mathcal{G}}^{N_{\mathrm{p}}}\right](h_i) + (1-\delta_i)\left[S^N(\Phi_i(\eta_{i-1}^N)) - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}}$$
$$+ \mathbb{E}\left\{\left|\delta_i\left[\widehat{\mathcal{G}}^{N_{\mathrm{p}}} - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{\frac{1}{p}}$$

Using the same conditioning argument as in Theorem 6.2 and applying Corollary 6.4 based

on the assumption $(\mathcal{H})_{\mathrm{u}}$ to the second term we have:

$$\begin{aligned}
&\mathbb{E}\left\{\left|\delta_i\left[\widehat{\mathcal{G}}^{N_{\mathrm{p}}} - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{1/p} \\
&= \mathbb{E}\left\{\delta_i \mathbb{E}\left\{\left|\left[\widehat{\mathcal{G}}^{N_{\mathrm{p}}} - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p \middle| \mathcal{F}_{i-1}, Y_t^{\mathfrak{S}_{\ell_t}}\right\}\right\}^{1/p} \\
&\leq q_i^{1/p}\sqrt{2}\left[\frac{8}{a_i\sqrt{N}}\left(2c^{2/p}(p/2) + (p/4)!C\mathbb{E}\int_0^{b_i}\sqrt{\log\left(1+\mathcal{D}(\varepsilon,\mathcal{H}_i,d_N)\right)}\mathrm{d}\varepsilon\right)\right. \\
&\quad \left. + 4\log(3\sqrt{e}(b_i/a_i))\frac{(b_i/a_i)^2}{N_{\mathrm{p}}}\right]^{1/2}
\end{aligned}$$

Applying Lemma 6.1 and the same conditioning argument as in Theorem 6.3 to the remaining term we have:

$$\begin{aligned}
&\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{1/p} \leq \frac{c^{1/p}(p)}{\sqrt{N}} \\
&+ q_i^{1/p}\left(\sqrt{2}\left[\frac{8}{a_i\sqrt{N}}\left(2c^{2/p}(p/2) + (p/4)!C\mathbb{E}\int_0^{b_i}\sqrt{\log\left(1+\mathcal{D}(\varepsilon,\mathcal{H}_i,d_N)\right)}\mathrm{d}\varepsilon\right)\right.\right. \\
&\left.\left. + 4\log(3\sqrt{e}(b_i/a_i))\frac{(b_i/a_i)^2}{N_{\mathrm{p}}}\right]^{1/2}\right)
\end{aligned}$$

We conclude that since $q_i \leq q_{\mathrm{u}}$ then for any $i \geq 0$ we have the time uniform estimate:

$$\begin{aligned}
&\mathbb{E}\left\{\left|\left[\eta_i^N - \Phi_i(\eta_{i-1}^N)\right](h_i)\right|^p\right\}^{1/p} \leq \frac{c^{1/p}(p)}{\sqrt{N}} \\
&+ q_{\mathrm{u}}^{1/p}\sqrt{2}\left[\frac{8}{a_{\mathrm{u}}\sqrt{N}}\left(2c^{2/p}(p/2) + (p/4)!C\sup_{i\geq 0}\mathbb{E}\int_0^{b_i}\sqrt{\log\left(1+\mathcal{D}(\varepsilon,\mathcal{H}_i,d_N)\right)}\mathrm{d}\varepsilon\right)\right. \\
&\left. + 4\log(3\sqrt{e}(b_{\mathrm{u}}/a_{\mathrm{u}}))\frac{(b_{\mathrm{u}}/a_{\mathrm{u}})^2}{N_{\mathrm{p}}}\right]^{1/2}
\end{aligned}$$

This along with a variation of (6.15) with $m = 1$ and the fact that according to assumption $(\mathcal{H})_{\mathrm{u}}$, $\epsilon_{\mathrm{u}}(M) \geq a_{\mathrm{u}}/b_{\mathrm{u}}$, completes the proof of theorem. □

The above theorem provides an error bound for the parametric approximation particle filter (using the GML algorithm to perform approximation) that is similar in structure to that specified for the subsampling approximation particle filter. The error bound consists of two distinct contributions, one corresponding to the normal operation of the filter and

the other capturing the impact of the parametric approximation operation. The theorem establishes a sufficiency requirement on the sequence of approximating classes $\mathcal{H}_i$ leading to the asymptotically unbiased approximation of distribution flows: as both the number of particles N and the number of terms N_{p} in the mixture approximation increases, the approximate particle filter distribution converges, weakly, to the exact distribution. The requirement is that the Markov transition kernel must have an associated bounded density and this density must be a member of the class $\mathcal{H}_i$. This implies that the class $\mathcal{H}_i$ should be sufficiently reach to represent the properties of the Markov kernel M_i. This condition is reminiscent of the modeling assumptions that underpin Gaussian sum particle filtering (see e.g. [120]), where the premise is that the filtering density can asymptotically be represented as an infinite sum of Gaussians.

6.3 Numerical Experiments

In this section we present the results of numerical experiments exploring the performance of the leader node particle filter. The experiments provide an example of how the subsampling and parametric approximation particle filters can be applied in a practical tracking problem. They provide an opportunity to compare the performance of the two algorithms and to examine whether practical behavior is similar to that predicted by the theoretical analysis.

The Simulation Set Up

We adopt the following information acquisition and target movement models. The state of the target is two-dimensional with dynamics [97]

$$X_t = X_{t-1} + r_0([\cos\varphi_t; \sin\varphi_t]) + u_t.$$

Here r_0 is a constant (we set $r_0 = 0.02$) and φ_t, u_t are independent and uniformly distributed $u_t \sim U[0,1]$, $\varphi_t \sim U[-\pi, \pi]$. $K_l = 20$ leader nodes and $K_s = 200$ satellite nodes are distributed uniformly in the unit square. A satellite sensor node j with coordinates $s_j = [s_{1,j}, s_{2,j}]$ can transmit its measurement to any active leader node within the connectivity radius r_c. The connectivity radius is set to $r_c = \sqrt{2\log(K_s)/K_s}$ (note that if every node can be a leader node, $K_l = K_s$, the resultant network topology is a random geometric graph). We assume that any active leader node can transmit an approximation of its posterior

representation to any other potential leader node.

The measurement equation of every satellite sensor is the binary detector [131] capable of detecting a target within radius r_d with probability p_d and false alarm rate p_f:

$$\mathbb{P}\{Y_t^j = 1|X_t\} = \begin{cases} p_d & \text{if} \quad X_t \in \mathcal{X}_d^j \\ p_f & \text{if} \quad X_t \notin \mathcal{X}_d^j \end{cases},$$

where the detection region $\mathcal{X}_d^j$ of satellite sensor j is defined as $\mathcal{X}_d^j = \{x : \|x - s_j\|_2 \leq r_d\}$. To perform sensor selection step we use the mutual information (MI) criterion [6]:

$$\ell_{t+1} = \arg\max_{\ell_{t+1} \in \mathcal{L}} I(X_{t+1}, Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}} | y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \tag{6.21}$$

Here $y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}$ denotes the entire history of measurements, and the random variable $Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}$ denotes the (potential) set of measurements at time $t+1$ by the set of satellite sensor nodes ($\mathfrak{S}_{\ell_{t+1}}$) of a candidate leader node ℓ_{t+1}. The calculation of the mutual information in the multiple sensor framework is generally a computationally demanding exercise. In the binary sensor framework, the calculations can be simplified using an efficient approximation (see Appendix B.4 for details).

Williams et al. pointed out in [96] that the application of the one-step mutual information criterion for sensor selection can result in undesirable leader node bouncing (frequent, unnecessary hand-off). To prevent this, Williams et al. proposed a computationally demanding finite-time horizon dynamic program [96]. In our simulations we use a simpler randomized algorithm to control the leader node exchange rate. In this algorithm the current leader node flips a biased coin with the probability of the flip outcome being 1 equal to λ. If the outcome is 1 then the current leader node calculates the mutual information criterion. It then determines if the current particle representation should be transferred to a new leader node that is more likely to make informative measurements. If the outcome is 0, then no calculations are performed. With this approach, the computational load for each leader node is significantly reduced and the communication overhead can be regulated by the choice of λ. However, the value of λ should be tailored depending on the application (as the mobility of the target increases, leader node hand-offs must be considered more frequently). In our experiments we fix $\lambda = 1/5$.

We consider two leader node particle filtering algorithms, with one employing non-

parametric approximation (subsampling) and the other using parametric approximation. To create a subsample for transmission in the non-parametric framework we use the general residual resampling scheme [119]. The parametric leader node particle filter is implemented using the GML algorithm with N_{p} components. Each component consists of a two-dimensional Gaussian density with diagonal covariance matrix. The mean vector and covariance matrix are estimated using the particle representation available at the current leader node. To implement the GML algorithm we used the standard MATLAB nonlinear optimization routine `fmincon` (see Appendix B.3 for details of the implementation).

Leader Node Communication Costs Calculation

In the following we make a number of non-restrictive simplifying assumptions. First, data routing in the WSN is implemented using the greedy geographic routing algorithm described by Dimakis et al. in [132]. Second, only the satellite nodes participate in the data routing during data exchanges. Third, the dimensionality of measurements acquired by the satellite nodes is the same for all the nodes comprising the network.

The communication costs of the leader node particle filter algorithm at every iteration consist of three contributions. First, the cost of communicating the raw data from the satellite nodes to the corresponding active leader node. Second, the cost of routing the tracking update from the current active leader node to the sink. Third, the cost of the leader node hand-off. This can be summarized in the following formula:

$$C_{\mathrm{tot}} = C_{\mathrm{raw}} + C_{\mathrm{upd}} + C_{\mathrm{ho}}. \tag{6.22}$$

The cost of routing the raw data, $C_{\mathrm{raw}} = D_{\mathrm{raw}} d_\ell$, is equal to the product of the dimensionality of the measurement, D_{raw}, and the number of active satellite nodes (the degree of the active leader node, d_ℓ). In our setting, the satellite nodes and the current active leader node form a random geometric graph with connectivity radius $r_c = \sqrt{2\log(K_s)/K_s}$. For sufficiently large K_s we have, with high probability, that a random geometric graph is regular (see Boyd et al. [14], Lemma 10). Thus the degree of the active leader node, according to Lemma 10 in [14], is at most $d_\ell = \mathcal{O}(\log K_s)$, with high probability. This implies that

$$C_{\mathrm{raw}} = D_{\mathrm{raw}} \mathcal{O}(\log K_s). \tag{6.23}$$

The cost of sending the tracking update, $C_{\text{raw}} = D_{\text{upd}} R_{\text{upd}}$ is equal to the product of the dimensionality of the update, D_{upd}, and the number of hops, R_{upd}, necessary to route the update to the sink. It was shown by Dimakis et al. in [132] that the hop-wise communication cost of the greedy geographic routing, in the random geometric graph setting, is at most $\mathcal{O}(\sqrt{K_s / \log K_s})$. This results in

$$C_{\text{upd}} = D_{\text{upd}} \mathcal{O}\left(\sqrt{\frac{K_s}{\log K_s}}\right). \tag{6.24}$$

The average cost of the leader node hand-off is $C_{\text{raw}} \leq q_{\text{u}} D_{\text{ho}} R_{\text{ho}}$, where D_{ho} is the dimensionality of hand-off; R_{ho} is the hop-wise communication cost of the hand-off; and, as before, q_{u} is the upper bound on the probability of hand-off. For the greedy geographic routing of the leader node hand-off we have, as previously, $R_{\text{ho}} = \mathcal{O}(\sqrt{K_s / \log K_s})$ and thus

$$C_{\text{upd}} = q_{\text{u}} D_{\text{ho}} \mathcal{O}\left(\sqrt{\frac{K_s}{\log K_s}}\right). \tag{6.25}$$

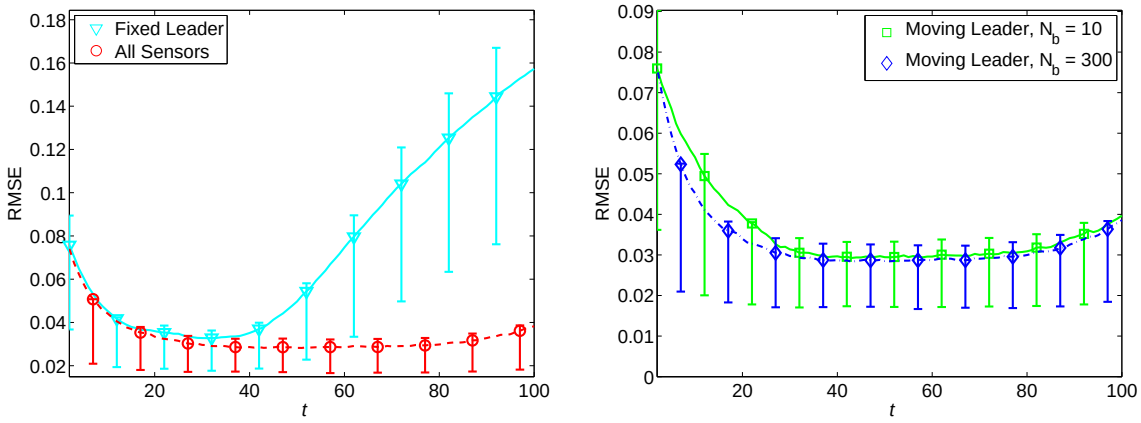
The total communication cost of every iteration of the leader node particle filter is thus

$$C_{\text{tot}} = D_{\text{raw}} \mathcal{O}(\log K_s) + \mathcal{O}\left(\sqrt{\frac{K_s}{\log K_s}}\right)(D_{\text{upd}} + q_{\text{u}} D_{\text{ho}}). \tag{6.26}$$

In the centralized scenario, every satellite node has to route its raw data to the fusion center an thus the total communication cost of the centralized tracking assuming the greedy routing strategy is

$$C_{\text{tot}} = D_{\text{raw}} \mathcal{O}\left(\frac{K_s^{3/2}}{\log^{1/2} K_s}\right). \tag{6.27}$$

Thus in the random geometric graph scenario and for large networks, the implementation of the leader node tracking protocol results in the order of K_s improvement in terms of the communication costs involved in transmitting the data to the sink.



(a) (b)

Fig. 6.1 Performance (RMSE) of different fusion schemes versus time. Error bars show lower and upper quartiles. (a) $\bigtriangledown$ denotes the scheme with fixed leader node selected at initialization, $\circ$ denotes the centralized scheme using the entire set of measurements from all sensors at every step. (b) $\square$ denotes the scheme with leader node selected using approximate Mutual Information (MI) criterion and non-parametric (subsampling) approximation with $N_{\mathrm{b}} = 10$; $\diamond$ denotes the scheme with leader node selected using approximate MI criterion but no subsampling approximation ($N_{\mathrm{b}} = 300$)

Results

In the following we report the simulation results obtained using the set-up discussed above. All results are achieved using 5000 Monte Carlo trials, and in each trial a new trajectory of the target is generated.

We first demonstrate that the discussed sensor selection procedure (leader node exchange rule) has good information fusion properties. Fig. 6.1 depicts the performance in terms of Root Mean Squared Error (RMSE) between the true position of the target and its estimate using different information diffusion schemes. The first scheme denoted by $\bigtriangledown$ corresponds to the situation when the leader node is selected at the initialization and is fixed throughout the tracking exercise. The second and third schemes denoted by $\square$ and $\diamond$ respectively correspond to non-parametric leader node algorithms using $N_{\mathrm{b}} = 10$ and $N_{\mathrm{b}} = 300$ particles for communications respectively. The fourth scheme denoted by $\circ$ corresponds to the centralized scenario when all the measurements available from every sensor at every time step t are used to track the target. Note that the baseline particle filter

uses $N = 300$ particles[5] in all scenarios (so the $N_\mathrm{b} = 300$ case corresponds to no subsampling). We can see from Fig. 6.1(a) that the centralized scheme has the best performance in terms of RMSE. However, it is only marginally better (cf. Fig. 6.1(a) and Fig. 6.1(b)) than the leader node scenario without compression ($N = N_\mathrm{b} = 300$). This highlights the effectiveness of the leader node particle filtering method and confirms that leader node selection based on the approximate mutual information is a valid approach. Compared to the centralized scheme, the communication and power consumption costs are significantly decreased since only a small subset of nodes is activated at any particular time step.

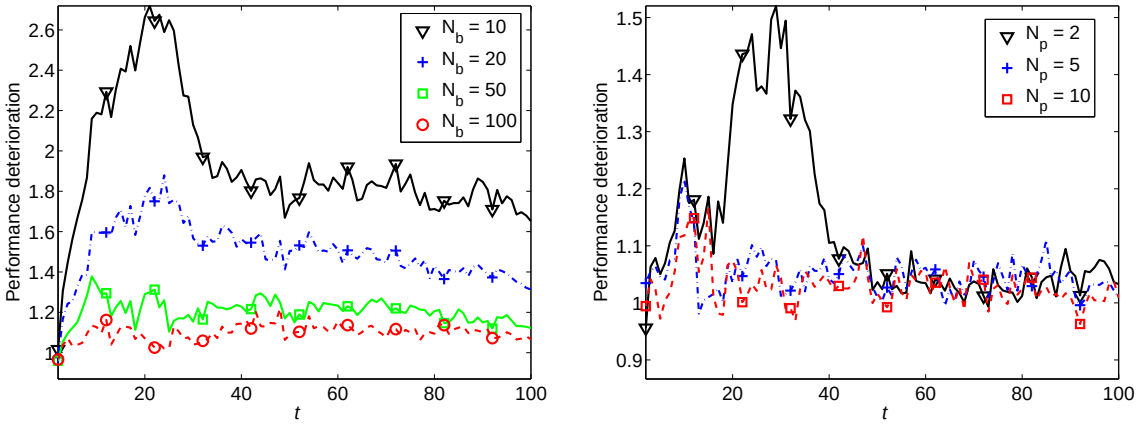
The leader node particle filter that uses a very small number of transmitted particles ($N_\mathrm{b} = 10$) performs comparably well. This suggests that there are practical scenarios where a particle filter can incorporate aggressive approximation to reduce communication overhead without incurring a significant penalty in tracking accuracy. The fixed leader node approach performs poorly, because the activated sensors only provide useful information when the target is nearby. As the target moves further away, the particle cloud approximating the filtering distribution becomes very diffuse, and tracking accuracy is 4 times worse than that of any of the other schemes.

In the next set of results, we explore the approximation error, i.e. the error induced by both sampling and the additional parametric/subsampling approximations. The RMSE combines both approximation error and estimation error resulting from the inaccuracy and/or ambiguity of the measurement information. We can estimate a *Root Mean Squared Approximation Error* (RMSAE) by calculating the error between a candidate particle filter and an "ideal" reference particle filter. As our reference filter, we employ a particle filter that uses $N = 3000$ particles, with no approximation during hand-off. For each of the 5000 Monte Carlo trials, we apply this reference filter to the generation of location estimates. The approximation error for our test filters is measured relative to these estimates rather than the true locations.

Figure 6.2[6] depicts the deterioration of the approximation performance as a function of (a) varying number of transmitted particles for the subsampling approximation leader node

[5]This value was selected after experimentation with multiple values of N because it provides sufficient accuracy without inducing unnecessary computational overhead. The primary purpose of the simulations is to examine the impact of the approximation steps.

[6]The additional random variation in Fig. 6.2 compared to Fig. 6.1 is due to the subtraction of the reference filter estimates and the division by the approximation error of the leader node without compression ($N = N_\mathrm{b} = 300$).

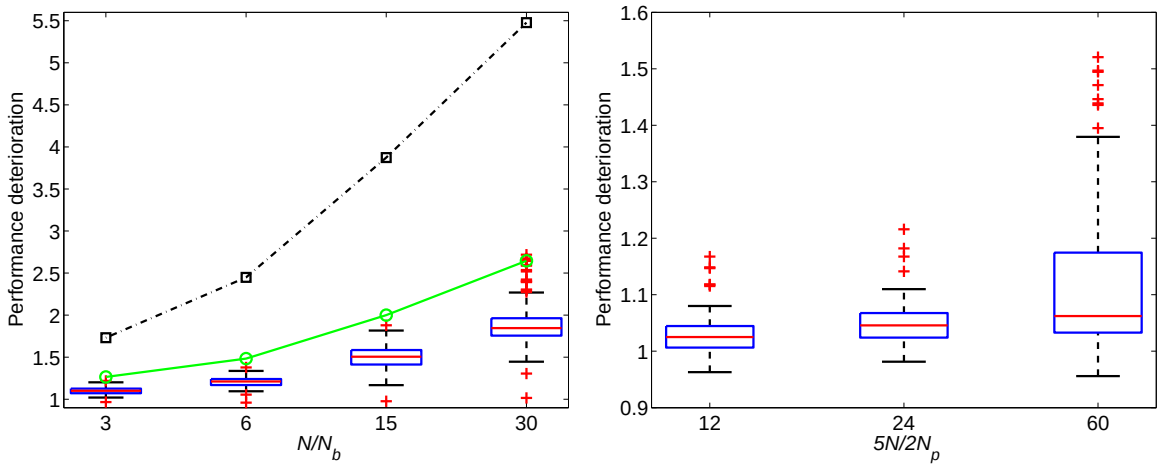


(a) Non-parametric leader node particle filter (b) Parametric leader node particle filter

Fig. 6.2 Deterioration of performance as a function of (a) varying number of transmitted particles for the subsampling approximation leader node particle filter; and (b) varying number of transmitted mixture components for the parametric approximation leader node particle filter. The performance deterioration is measured as the ratio of the Root Mean Squared Approximation Error (RMSAE) averaged over 5000 Monte Carlo trials of the candidate particle filtering algorithm with intermittent approximation (subsampling or parametric) to that of a leader node particle filter that performs no approximation ($N_{\mathrm{b}} = 300$).

particle filter; and (b) varying number of transmitted mixture components for the parametric approximation leader node particle filter. The performance deterioration is measured as the ratio of the RMSAE averaged over 5000 Monte Carlo trials of the candidate leader node particle filtering algorithm with intermittent approximation (subsampling or parametric) to that of a leader node particle filter that performs no approximation ($N_{\mathrm{b}} = 300$), i.e. uses $N_{\mathrm{b}} = N = 300$ particles during hand-off. Figure 6.2(a) shows how the approximation performance is affected as the number of particles in the subsampling step (N_{b}) changes; Figure 6.2(b) provides similar results for the parametric approximation method as the number of components in the mixture model (N_p) is varied.

Fig. 6.2 indicates that the performance of the leader node particle filter has interesting dynamic structure. In particular, in the time period $t \in [1, 50]$ we can see an articulated transient behavior (see Fig. 6.2(a), $N_{\mathrm{b}} = 10$ in particular). The transient in these curves arises because the particle representation of the target location density is initially highly dispersed and multi-modal, making it relatively difficult to approximate using either



(a) Non-parametric leader node particle filter. □ denotes the naive performance deterioration characterization, $\sqrt{N/N_{\mathrm{b}}}$. ∘ denotes the proposed characterization captured by the factor $(q_{\mathrm{u}}\chi + (1-q_{\mathrm{u}}))^{1/2}$ in Corollary 6.3.

(b) Parametric leader node particle filter

Fig. 6.3 Box-plots showing the relationship between deterioration of approximation performance and compression factor. The performance deterioration is measured as the ratio of the Root Mean Squared Approximation Error (RMSAE) of the candidate particle filtering algorithm with intermittent approximation (subsampling or parametric) to that of a leader node particle filter that performs no approximation ($N_{\mathrm{b}} = 300$). The compression factor, defined in Section 6.3, is the ratio of N to the number of values transmitted during leader node exchange (N_b or $2.5N_p$). The boxes show lower quartile, median and upper quartile of the 5000 Monte Carlo trials. Whiskers depict 1.5 times the interquartile range and capture most of the extreme values, and the + values denote outliers extending beyond the whiskers.

a subsampling or parametric method with a small number of particles/mixture components. However, as time progresses ($t \in [51, 100]$) the particle representation of the target becomes more localized and closer to unimodal, so approximation performance improves significantly. Qualitatively, the performance deteriorates gracefully with respect to the extent of the compression during hand-off (reduction in number of particles or components), as theoretically predicted in the previous sections.

For the final performance analysis, we define a *compression factor* as the ratio of the number of particles used during regular particle filter computations to the number of *values* transmitted during the hand-off. For the subsample approximation case, this is simply

N/N_{b}. In our case of a Gaussian mixture, variance information is transmitted in addition to the locations of the Gaussians and the mixture weights, so the factor is $2N/5N_{\mathrm{p}}$. Figure 6.3 presents a box-plot depicting performance deterioration (ratio of approximation error of the leader node with $N_b < N$ and the leader node with $N_b = N$) versus the compression factor. Both the median and the maximal deviations of the performance deterioration scale smoothly with changing compression factor. Parametric approximation clearly outperforms subsampling.

For the subsampling case, Theorem 6.2 and Corollary 6.3 provide an analytical bound on the expected approximation error. The curve based on these results is depicted in Figure 6.3(a) and provides a meaningful characterization of the expected performance deterioration. Indeed, the theoretical prediction based on the factor $(q_{\mathrm{u}}\chi + (1 - q_{\mathrm{u}}))^{1/2}$ from Corollary 6.3 closely coincides with the maximal performance deterioration observed for each compression factor. For comparison purposes, we include a similar characterization derived based on a simple worst-case assumption that the subsample approximation particle filter performs only as well as a particle filter that uses N_{b} particles at all times. The characterization based on the bounds developed in this chapter clearly provides a better indication of the performance deterioration.

6.4 Applicability of Results

Throughout the chapter, we motivated the theoretical analysis by considering the concrete example of the "leader node" particle filter [57], an algorithm that has been proposed for collaborative distributed tracking in sensor networks. Below we outline two other examples to illustrate that the analyzed problem arises in several practical settings and hence the analysis results presented in the this chapter can be easily generalized and applied in other contexts.

Example 1: Tracking with delayed measurements

In wireless sensor networks, packet losses can lead to measurements arriving out-of-order to a node performing tracking. Incorporating delayed measurements into a particle filter is important, because they can be highly informative and improve tracking performance. One of the simplest, and most effective, strategies is to run the particle filter again from the time-step corresponding to the delayed measurement. This strategy can be hampered by

the limited memory of most sensor network devices, which means it is impossible to store full particle representations for multiple time-steps. The alternative is to store an approximation, either a subsampled set of particles, or a parametric representation, for previous time-steps. When the particle filter is run again, it is initialized by sampling from the approximated distribution. The effect is equivalent to injecting intermittent approximations (subsampling or parametric) into the particle filter.

Example 2: Real-time tracking with computational constraints

When real-time tracking is performed on an embedded processor with computational limitations, it can be important to adjust the time devoted to particle filter computation. For example, consider a mobile robot that employs a particle filter to track its position and at the same time conducts iterative strategic planning of its motion in order to reach a target location. The goal can be achieved more efficiently (in less time and with less energy expenditure) if there is an adjustment of the computational time devoted to each of these two tasks. The adaptive particle filter proposed in [133] and the real-time particle filter of [134] adjust the number of particles at each time-step based on an estimate of the complexity of the filtering distribution (assigning fewer particles for simple distributions). Through these schemes, the accuracy of the position estimation can be preserved, but more time can be devoted to motion planning. The adaptation of the number of particles is an example of the subsampling approximation that we analyze in this chapter.

6.5 Summary

This chapter presented the analysis of the leader node particle filter that performs intermittent approximation whenever hand-off occurs. Such approximation steps become necessary when particle filters are deployed on resource-constrained WSN platforms, where the resource can be energy, memory or computational power. The main results have the form of upper bounds on the expected L_p error of the leader node particle filter that occasionally employs either subsampling or parametric approximations in order to execute sensor management tasks (leader node hand-offs). The important conclusion of our analysis is that these approximation steps do not induce instability, and moreover, the frequency of the approximation steps significantly affects the extent of performance degradation. If the approximation steps are rare, then the compression can be very high (very few subsamples

or very few mixture components) and the average approximation error remains reasonable. Numerical experiments indicate that the bound for the subsample approximation particle filter provides a meaningful characterization of practical performance.

Chapter 7

Conclusions

This thesis explores the potential of in-network signal processing techniques applied to building efficient distributed information fusion and aggregation protocols for application in wireless sensor networks (WSNs). The performance of information fusion and aggregation protocols is measured in terms of WSN performance metrics such as network lifetime, communication and bandwidth load requirements, and estimation (tracking) accuracy. The centralized approaches to information fusion in WSNs provide the best possible estimation accuracy, however, they often suffer from the need to transfer undesirably large quantities of data through the entire network and uneven power consumption in sensor nodes. These factors lead to the reduced network lifetime under most network lifetime metrics and inefficient use of communication channels, leading to reduced network capacity. Distributed algorithms alleviate these drawbacks using smart in-network processing and data aggregation protocols. However, for such algorithms the reduction in estimation accuracy and/or the increase in time required to complete a particular fusion task may be significant. Therefore, the design and analysis of fast and accurate distributed information fusion algorithms with guaranteed performance characteristics is important. In this thesis, we have addressed two specific problems associated with the analysis and design of such distributed algorithms.

Predictor Based Accelerated Distributed Average Consensus

The distributed average consensus algorithm solves the problem of finding the arithmetic mean of the values captured by the sensors comprising a WSN using local pairwise mes-

sage exchanges. The memoryless distributed average consensus is known to suffer from the poor scalability of the averaging time required to compute the arithmetic mean within a prescribed level of accuracy. In Chapter 4, we have proposed and analyzed the local, memory based acceleration methodology for the distributed average consensus algorithm. In particular, we proposed a simple general predictive methodology for the acceleration of average distributed consensus algorithms. The methodology consists of mixing the prediction of the local state value at every node with the outcome of the conventional consensus iteration. The key parameter of the methodology is the mixing weight, which determines the influence of the prediction component. We have studied the convergence properties of the proposed framework and identified the existence of parameter configurations that ensure convergence. For these convergent configurations of the algorithm we have quantified the limiting averaging time growth rate for the asymptotically small ℓ_2 deviation of the distributed computation result from the result obtained via centralized computation. For two important cases of the proposed algorithm we have identified the optimal value of the key mixing parameter and studied the improvement achieved in the convergence rate of the accelerated algorithm compared to its baseline, non-accelerated, variant.

In the first, memoryless, acceleration setting we observed significant performance improvement and studied the extent of this improvement via simulation. However, our theoretical results imply that this simplest configuration of the proposed acceleration methodology is not guaranteed to provide improvement. There exist conventional distributed averaging consensus algorithms for which no further improvement in the memoryless accelerated framework is possible (e.g. the fastest consensus averaging matrix). We proposed a number of suboptimal distributed initialization schemes for our proposed memoryless acceleration framework. These initializations can be useful due to their simplicity in situations when other distributed average consensus solutions (e.g. the fastest consensus averaging matrix) may not be available due to the complexity associated with their on-line initialization.

Observing the drawbacks of the memoryless acceleration methodology, we moved one step further and analyzed a more complex instance of the proposed framework involving the predictor based on one tap of memory. We have performed thorough analysis of this second setting and identified the optimal value of the mixing parameter resulting in the fastest worst-case asymptotic convergence rate. The analysis of the convergence rate provided theoretical guarantees for the amount of the improvement. The results indicate that the proposed one-tap memory predictive consensus acceleration procedure is guaranteed to

reduce the convergence time of any conventional distributed averaging consensus algorithm. Another important conclusion was that the amount of the improvement grows with the size of the network implying good scalability properties of the proposed algorithm. We have quantified the improvement for a number of important network topologies.

Observing that the optimal setting of the mixing parameter requires knowledge of the second largest eigenvalue of the foundational weight matrix, we have devised a simple distributed scheme for its on-line initialization. We have conducted simulations that confirmed our theoretical derivations and revealed that the on-line initialization of the mixing parameter results in almost optimal performance of the proposed accelerated methodology. We have compared the performance of our proposed algorithm with the performance of several other accelerated algorithms in the literature and concluded that the proposed algorithm has superior performance characteristics with respect to these algorithms. We have thus accomplished the important task of designing an accelerated distributed average consensus algorithm with significantly improved convergence time and guaranteed performance characteristics.

Analysis of the Leader Node Particle Filter

The second half of the thesis explores the approximation performance of the leader node particle filter tracking architecture. The leader node particle filter prolongs the network lifetime by activating only a subset of nodes at any particular time instant. The activated nodes transmit their measurements to the cluster head, the leader node, and the leader node performs tracking operations based on the local measurements. The best subset of active nodes are determined, at every time step, using mutual information based criterion. The current filtering distribution is sent to the new leader node whenever a hand-off occurs. Periodically, an update is sent to the requestor of the tracking information. This methodology improves scalability and wireless communication channel usage efficiency by performing computations and measurement transmissions in the localized fashion. To further reduce the communication costs, a compressed (approximate) representation of the current filtering distribution is sent during leader node hand-offs. This approximation exercise inevitably induces additional errors in the particle filtering recursion. In Chapter 6, we have examined these additional approximation errors.

We have considered two types of errors associated with different filtering distribution

approximation techniques. The first approximation method is subsampling, when only a subset of particles is sent during the leader node hand-off. The second approximation method is parametric mixture approximation when a mixture model estimated from the set of particles is sent during the leader node hand-off. To analyze the approximation error propagation in the leader node particle filter, we utilized the Feynman-Kac distribution flow modeling approach. We have modified the existing models applicable to conventional particle filters to fit the leader node framework and used suitable regularity conditions to obtain time-uniform error bounds for the weak-sense L_p errors of the leader node particle filter.

In the first scenario of subsample approximation leader node particle filter, we have extended currently available local L_p sampling error analysis results by finding tighter bounding constants. We then linked these results to the contractions of Feynman-Kac semigroups and obtained the time-uniform L_p error bounds and exponential inequalities for probability of large deviations characterizing the deterioration of approximation performance resulting from the additional approximations during leader node hand-offs. Our results reveal that the additional approximation error is characterized by the probability of the leader node hand-off. Thus in the scenarios when leader node hand-offs do not happen often, considerable compression can be applied without significantly affecting overall approximation performance. The additional subsample approximation can thus be efficiently used to reduce communication costs of leader node hand-offs.

In the second scenario of the parametric mixture approximation leader node hand-off, we have characterized the time-uniform L_p error bounds and obtained similar results. We have used the greedy maximum likelihood mixture estimation framework to obtain the mixture representation used during hand-off. We have extended the existing results by obtaining the local L_p error bounds for the error of the greedy maximum likelihood mixture estimation algorithm applied during the leader node hand-off. We then used Feynman-Kac framework to study the propagation of these local errors. We have formulated the requirements on the components of the mixture leading to the asymptotically unbiased approximation of filtering distribution during leader node hand-off and identified the upper bound on the rate, at which the approximate leader node particle filter distribution converges to the true leader node Feynman-Kac flow. Our numerical experiments revealed that the parametric approximation leader node particle filter performs better than the subsample approximation leader node particle filter. The numerical experiments also demonstrated that the

subsample approximation leader node particle filter error bounds provide useful characterization of the average performance deterioration observed during experiments. We have thus solved an important problem of establishing theoretical guarantees for the approximation performance of the leader node particle filter, an efficient distributed tracking algorithm implemented within the collaborative sensing framework.

Future Work

In Chapter 4, we have designed and analyzed an accelerated distributed average consensus algorithm with improved convergence time. However, we have only derived the optimal solution for two configurations of the proposed predictor based methodology. A promising research direction is the analysis of the more general configurations of the predictor including arbitrary predictor lengths. An intriguing question is the analysis of the predictor based consensus with quantization noise. In this setting, nodes exchange quantized data and thus an extra noise term is added to the consensus state vector at every iteration. The important questions are how the introduction of memory affects noise propagation in the consensus framework and whether it is possible to minimize the undesirable noise accumulation by varying the parameters of the proposed predictor based framework.

In Chapter 6, we have analyzed the additional approximation errors of the leader node particle filter. Our results can be extended in several ways. First, we believe that for the parametric approximation scenario a tighter bound characterizing the additional approximation error could be obtained. Our current results are based on the relationship between the Kullback-Leibler divergence and the L_p error and the analysis of the expectation of the powers of the Kullback-Leibler divergence. Perhaps, the adaptation of a more direct analysis of the L_p error would yield a better bound. The challenge here is the design of a suitable error decomposition. Second, a promising research direction is the analysis of the additional approximation errors of the parametric approximation leader node particle filter for more general mixture approximation schemes. Our current results are based on the analysis of the greedy maximum likelihood mixture estimation algorithm. Third, our analysis only explores the behavior of the additional *approximation* errors induced by the information compression during hand-off that occurs in the leader node protocol. The total error induced by the leader node protocol, when compared to a centralized particle filter collecting measurements from all available sensors, also includes the additional *estimation*

error associated with activating only a subset of nodes in a WSN. The important extension of our current work is thus the exploration of the behavior of this additional estimation error for reasonable choices of the underlying network topology (e.g. a random geometric graph or a grid).

Appendix A

A.1 General Expressions for Predictor Weights for Arbitrary M and k

In this appendix we present the expressions for the predictor weights $\boldsymbol{\theta}$ in (4.9) as a function of algorithm parameters and previous states for the case of arbitrary M and k.

First, we present the rationale behind the design of weights $\boldsymbol{\theta}$. As shown in Fig. A.1, given the set of previous values at some time instant t_0 and node i, $x_i(t_0 - M + 1 : t_0) = [x_i(t_0 - M + 1), x_i(t_0 - M + 2), \ldots, x_i(t_0 - 1), x_i^{\mathrm{W}}(t_0)]^{\mathsf{T}}$ we would like to design the best linear least squares approximation to the model generating the available data. Then using the approximate model we would like to extrapolate the current state k time steps forward. The easiest way to do this is to note that the approximate model of the form $\widehat{x}_i(t) = at + b$ with a and b being the parameters of the linear model can be rewritten in the matrix form for the set of available data:

$$\widehat{x}_i(t_0 - M + 1 : t_0) = \mathbf{B}_{t_0 - M + 1 : t_0} \psi. \tag{A.1}$$

Here

$$\mathbf{B}_{t_0 - M + 1 : t_0} = \begin{bmatrix} t_0 - M + 1 & 1 \\ t_0 - M + 2 & 1 \\ \ldots & \ldots \\ t_0 - 1 & 1 \\ t_0 & 1 \end{bmatrix} \qquad \text{and} \qquad \psi = [a, b]^{\mathsf{T}} \tag{A.2}$$

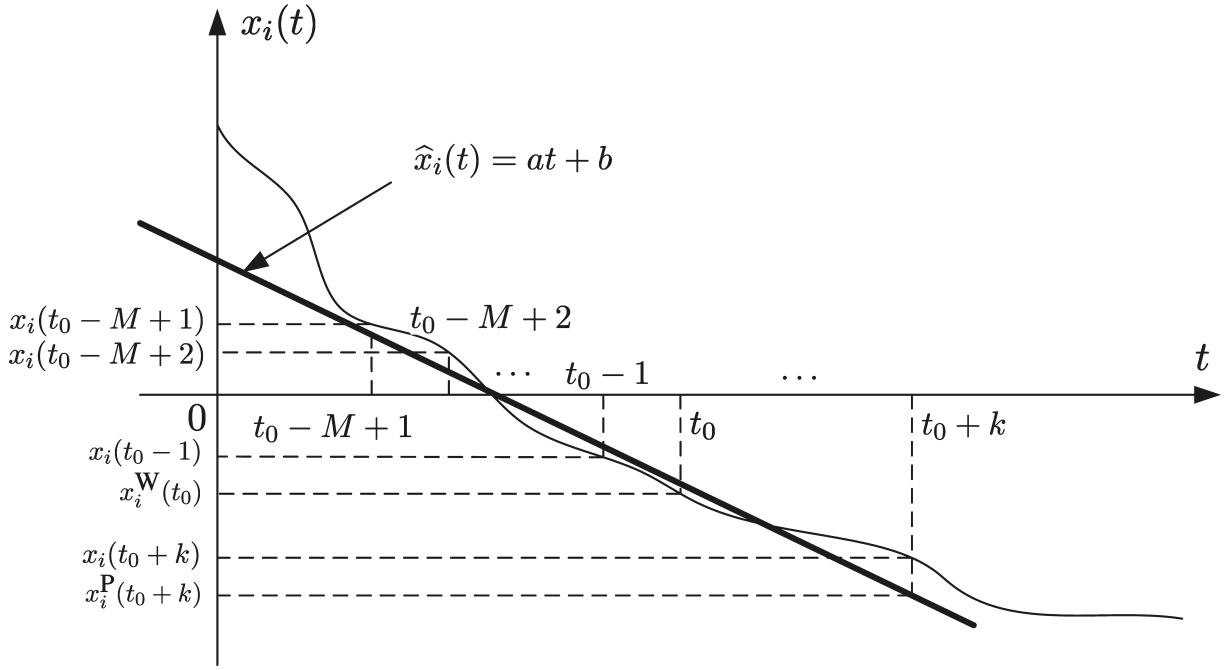


Fig. A.1 Linear approximation to the model generating available data comprising linear predictor

Using the standard least squares technique we define the cost function

$$\mathcal{I}(\psi) = (x_i(t_0 - M + 1 : t_0) - \widehat{x}_i(t_0 - M + 1 : t_0))^{\mathsf{T}}(x_i(t_0 - M + 1 : t_0) - \widehat{x}_i(t_0 - M + 1 : t_0)) \tag{A.3}$$

$$= (x_i(t_0 - M + 1 : t_0) - \mathbf{B}_{t_0 - M + 1 : t_0}\psi)^{\mathsf{T}}(x_i(t_0 - M + 1 : t_0) - \mathbf{B}_{t_0 - M + 1 : t_0}\psi) \tag{A.4}$$

and find the optimal approximate linear model $\widehat{\psi}$ as the global minimizer of the cost function:

$$\widehat{\psi} = \arg\min_{\psi} \mathcal{I}(\psi) \tag{A.5}$$

Taking into account the convexity of the cost function and equating the derivative of $\mathcal{I}(\psi)$ with respect to ψ to zero we get the solution:

$$\widehat{\psi} = (\mathbf{B}_{t_0 - M + 1 : t_0}^{\mathsf{T}} \mathbf{B}_{t_0 - M + 1 : t_0})^{-1} \mathbf{B}_{t_0 - M + 1 : t_0}^{\mathsf{T}} x_i(t_0 - M + 1 : t_0). \tag{A.6}$$

Now, given the linear approximation of the model generating current data, we extrapolate

the current state k steps forward using $\mathbf{B}_{t_0+k} = [t_0 + k, 1]$:

$$x_i^{\mathrm{P}}(t_0+k) = \mathbf{B}_{t_0+k}\widehat{\psi} = \underbrace{\mathbf{B}_{t_0+k}(\mathbf{B}_{t_0-M+1:t_0}^{\mathsf{T}}\mathbf{B}_{t_0-M+1:t_0})^{-1}\mathbf{B}_{t_0-M+1:t_0}^{\mathsf{T}}}_{\boldsymbol{\theta}^{\mathsf{T}}(t_0)} x_i(t_0 - M + 1 : t_0). \quad \text{(A.7)}$$

Finally, noting the time invariance of predictor weights $\boldsymbol{\theta}(t_0)$, that is $\boldsymbol{\theta}(t_0) = \boldsymbol{\theta}(0), \forall t_0$, we substitute $\mathbf{B}_{t_0-M+1:t_0}$ and $\mathbf{B}_{t_0+k}$ by their time-invariant analogs $\mathbf{B}$ and $\mathbf{c}$ as defined in (4.9).

Second, we need an expression for the pseudoinverse $\mathbf{B}^{\dagger}$. From the definition of $\mathbf{B}$ in (4.10) we can derive the inverse of $\mathbf{B}^{\mathsf{T}}\mathbf{B}$ in closed form:

$$(\mathbf{B}^{\mathsf{T}}\mathbf{B})^{-1} = \frac{2}{M(M+1)} \begin{bmatrix} \frac{6}{M-1} & 3 \\ 3 & 2M-1 \end{bmatrix} \quad \text{(A.8)}$$

The expression for the pseudoinverse $\mathbf{B}^{\dagger}$ follows immediately:

$$\mathbf{B}^{\dagger} = (\mathbf{B}^{\mathsf{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathsf{T}} = \frac{2}{M(M+1)} \begin{bmatrix} \frac{-6(M-1)}{M-1} + 3 & -\frac{-6(M-2)}{M-1} + 3 & \cdots & 3 \\ -M+3-1 & -M+6-1 & \cdots & 2M-1 \end{bmatrix} \quad \text{(A.9)}$$

This results in the following expression for predictor weights:

$$\boldsymbol{\theta} = \mathbf{B}^{\dagger\mathsf{T}}\mathbf{c} = \frac{2}{M(M+1)} \begin{bmatrix} \left(-\frac{6(M-1)}{M-1} + 3\right) k - M + 3 - 1 \\ \left(-\frac{6(M-2)}{M-1} + 3\right) k - M + 6 - 1 \\ \vdots \\ 3k + 2M - 1 \end{bmatrix} \quad \text{(A.10)}$$

A.2 Probability That Two Arbitrary Nodes Are Connected

In this section we present the calculation of the probability that two randomly selected nodes in a sensor network with connectivity radius r_c and sensors uniformly distributed $p_{x_i,y_i}(x_i, y_i) = 1, \quad x_i, y_i \in [0, 1]$ in a normalized square area $\mathcal{D}$ such that $\mathcal{D} = \{x, y | x, y \in [0, 1]\}$ on the plane are connected. As was mentioned before, this probability can be evaluated using integral of the form

$$p = \int_{\mathcal{S}} p_{x_i,y_i}(x_i, y_i) p_{x_j,y_j}(x_j, y_j) \mathrm{d}x_i \mathrm{d}y_i \mathrm{d}x_j \mathrm{d}y_j \quad \text{(A.11)}$$

where the set $\mathcal{S}$ is defined as follows

$$\mathcal{S} = \{(x_i, y_i, x_j, y_j) |\ (x_i - x_j)^2 + (y_i - y_j)^2 \leq r_c^2;\ x_i, y_i, x_j, y_j \in [0, 1]\} \tag{A.12}$$

To facilitate calculation of the integral (A.11) given the set of integration limits (A.12) we can divide this problem into two parts: $r_c \leq 1$ and $1 < r_c \leq \sqrt{2}$. Note also that random variables x and y can be introduced:

$$x = x_i - x_j, \quad y = y_i - y_j. \tag{A.13}$$

It is obvious that due to the fact that $p_{x_i,y_i}(x_i, y_i)$ is uniform, the joint distribution of x, y is triangular:

$$f_{x,y}(x, y) = \begin{cases} \frac{1}{4}(1 - |x|)(1 - |y|) & \text{if } (x, y) \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}. \tag{A.14}$$

Hence the integral in (A.11) can be reformulated into double integral:

$$p = \int\limits_{\sqrt{x^2+y^2}<r_c,\quad (x,y)\in[-1,1]} f_{x,y}(x, y)\mathrm{d}x\mathrm{d}y. \tag{A.15}$$

Due to the symmetry of the problem we can consider only a positive quadrant during the calculation of (A.15). In the case $r_c \leq 1$ it reduces to the following:

$$p = 4 \int\limits_{x=0}^{r_c} (1 - x) \int\limits_{y=0}^{\sqrt{r_c^2 - x^2}} (1 - y)\mathrm{d}y\mathrm{d}x = \frac{1}{2} r_c^4 - \frac{8}{3} r_c^3 + \pi r_c^2. \tag{A.16}$$

On the other hand, when $1 < r_c \leq \sqrt{2}$ we can reformulate (A.15) as follows:

$$\begin{aligned} p &= 1 - 4 \int\limits_{y=\sqrt{r_c^2-1}}^{1} (1-y) \int\limits_{x=\sqrt{r_c^2-y^2}}^{1} (1-x)\mathrm{d}x\mathrm{d}y \\ &= -\frac{1}{2}r_c^4 + \frac{8}{3}r_c^2\sqrt{r_c^2-1} - 2r_c^2 + \frac{4}{3}\sqrt{r_c^2-1} \\ &\quad - 2r_c^2 \arcsin\left(\sqrt{1-\frac{1}{r_c^2}}\right) + 2r_c^2 \arcsin\left(\frac{1}{r_c}\right) + \frac{1}{3}. \end{aligned} \tag{A.17}$$

Appendix B

B.1 The Comparison of Local Approximation Error Bounds

It is relatively straightforward to see why the sequence of constants $c(p)$ provides tighter bounds in Lemma 6.1 than the sequence $d(p)$ in Lemma 5.1. For example, for the even $p = 2n$ the ratio of the two sequences is

$$\begin{aligned}\frac{d(2n)}{c(2n)} &= \frac{(2n)!2^{-n}}{n!(2n)\Gamma(n)2^{-n}} \\ &= \frac{(2n-1)!}{n(n-1)!\Gamma(n)} = \frac{\Gamma(2n)}{n\Gamma(n)\Gamma(n)} \\ &= \frac{1}{n\mathrm{B}(n,n)}. \end{aligned} \tag{B.1}$$

Here B is the beta function. $\mathrm{B}(n,n)$ is a quickly decaying function. In particular, for large n Stirling's approximation gives a simple expression for beta function, $\mathrm{B}(n,n) \sim \sqrt{2\pi}n^{-1/2}2^{-2n+1/2}$, yielding the large n Stirling's approximation for (B.1):

$$\frac{d(2n)}{c(2n)} \sim \frac{1}{\sqrt{2\pi n}}2^{2n-1/2}.$$

This shows that $c(p)$ grows much slower with p than $d(p)$. This improved nature of constants $c(p)$ results in better estimates of moment generating function in Theorem 6.1 and Corollary 6.1, and better exponential inequality in Theorem 6.3. The comparison of the bound in Lemma 5.1 with the bound in Lemma 6.1 is provided in Fig. B.1 for the L_p error of the N-sample mean estimator in the case of uniform random variable distributed over

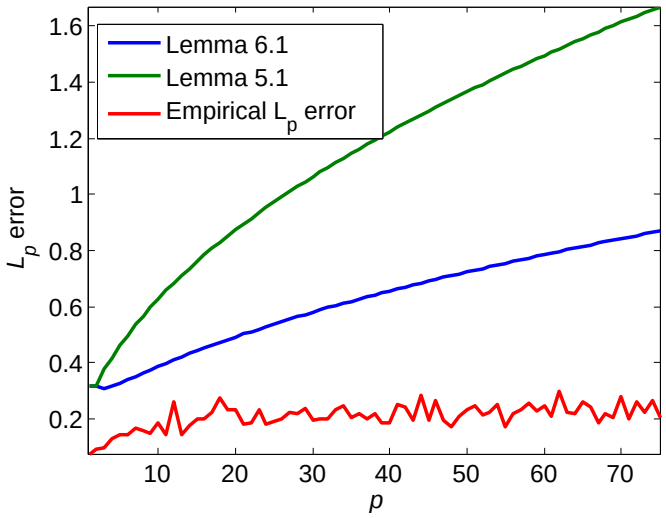


Fig. B.1 Comparison of bounds in Lemma 5.1 and Lemma 6.1 for the L_p error of the N-sample mean estimator of the uniform random variable distributed over the interval $[0, 10]$. Sample size, $N = 1000$.

the interval $[0, s]$. We verify that in this case the function that we are analyzing is:

$$\begin{aligned}\mathbb{E}\{|[P - S^N(P)](h)|^p\}^{\frac{1}{p}} &= \mathbb{E}\left\{\left|\frac{1}{N}\sum_{i=1}^{N} h(x_i) - \int h(x)P(\mathrm{d}x)\right|^p\right\}^{\frac{1}{p}} \\ &= \mathbb{E}\left\{\left|\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)\right|^p\right\}^{\frac{1}{p}}\end{aligned}$$

We conclude that our test function has the following form in this setting: $h(x_i) = x_i$ and oscillations of this function can be estimated straightforwardly:

$$\begin{aligned}\sigma^2(h) &= \frac{1}{N}\sum_{i=1}^{N} \operatorname{osc}^2(h_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} \sup\{|h_i(x_i) - h_i(y_i)|; x_i, y_i \in E_i\}^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} \sup\{|x_i - y_i|; x_i, y_i \in [0, s]\}^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} s^2 = s^2\end{aligned}$$

Hence we have in this case $\sigma(h) = s$ and the bound from Lemma 6.1 takes the form: $\mathbb{E}\{|[P - S^N(P)](h)|^p\}^{\frac{1}{p}} \leq c(p)^{1/p}\frac{s}{\sqrt{N}}$. On the other hand, applying Lemma 5.1 gives: $\mathbb{E}\{|[P - S^N(P)](h)|^p\}^{\frac{1}{p}} \leq d(p)^{1/p}\frac{s}{\sqrt{N}}$. Fig. B.1 depicts the two bounds plotted for different values of p and compares it with the actual errors observed during simulations. We used the following settings to obtain this plot: $N = 1000$, $s = 10$.

B.2 The Estimates of the Moment Generating Function

In this appendix we show how the impact of improved constants in Lemma 6.1 can be used to improve the estimate of the moment generating function in Theorem 7.3.1 [114]. We now state the Theorem 7.3.1.

Theorem B.1 (Del Moral [114], Theorem 7.3.1)**.** *For any sequence of $\mathcal{E}$–measurable functions $(h_i)_{i\geq 1}$ such that $\mu_i(h_i) = 0$ for all $i \geq 1$ we have for any ε*

$$\sigma(h) < \infty \Longrightarrow \mathbb{E}\left\{e^{\varepsilon\sqrt{N}|m(X)(h)|}\right\} \leq (1 + \varepsilon\sigma(h))\, e^{\frac{\varepsilon^2}{2}\sigma^2(h)}$$

We note that the simplified estimate of the moment generating function in Corollary 6.1 is much tighter than the bound in Theorem B.1 for asymptotically large deviations ε while the more complex bound in Theorem 6.1 outperforms the one in the Theorem B.1 uniformly over the range of ε. The comparison of the bounds obtained in Theorem B.1 and Theorem 6.1 with the empirical estimate is provided in Fig. B.2. The test setup is the

same as in Appendix B.1 The parameters of the simulation can be summarized as follows: scale parameter, $s = 10$, number of i.i.d. samples, $N = 100$, averaging is performed over $M = 10000$ trials, ε ranges from 0 to 1. Similar results are obtained in other settings.

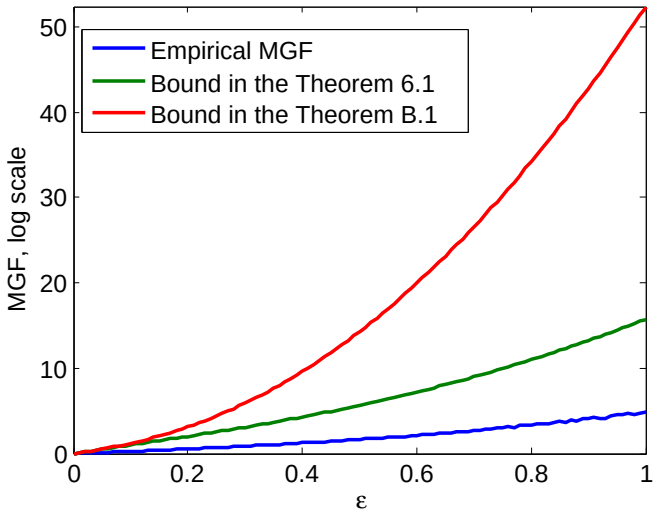


Fig. B.2 Comparison of bounds in Theorem B.1 and Theorem 6.1 for the moment generating function of N-sample mean estimator of the uniform random variable distributed over the interval $[0, 10]$. Sample size, $N = 100$.

B.3 GML Implementation Details (Objective Function and Its Derivatives)

In this section we present the derivatives of the objective function of the GML algorithm. As was mentioned earlier, the injection of this information into the numerical optimization routine results in a significant (two times) acceleration of the GML speed. Assuming that $\{\xi^{(j)}\}_{j=1}^{N}$ is the current particle set, ϕ_{θ_i} comes from the class of two-dimensional Gaussian densities with diagonal covariance matrix

$$\phi_{\theta_i}(\xi^{(j)}) = \frac{1}{2\pi\sigma_{2,i}\sigma_{1,i}} e^{-\frac{\left(\xi_1^{(j)}-\mu_{1,i}\right)^2}{2\sigma_{1,i}^2}-\frac{\left(\xi_2^{(j)}-\mu_{2,i}\right)^2}{2\sigma_{2,i}^2}}$$

and ith-step GML objective is written as follows (according to Algorithm 2)

$$\mathcal{J}_i = -\sum_{j=1}^{N} \log[\alpha_i \phi_{\theta_i}(\xi^{(j)}) + (1-\alpha_i) g_{i-1}(\xi^{(j)})]$$

we can calculate the following set of the first- and second-order derivatives necessary to construct gradient and Hessian for the non-linear optimization routine at iteration i. The acceleration is achieved by evaluating the expensive exponential terms

$$V_{i,j} = e^{\frac{\left(\xi_1^{(j)} - \mu_{1,i}\right)^2}{2\sigma_{1,i}^2} + \frac{\left(\xi_2^{(j)} - \mu_{2,i}\right)^2}{2\sigma_{2,i}^2}}, \quad 1 \le j \le N$$

only once per GML iteration and vectorizing the code with respect to the terms of the type $(\xi_1^{(j)} - \mu_{1,i})$ and $\left(\left(\xi_1^{(j)} - \mu_{1,i}\right)^2 - \sigma_{1,i}^2\right)$.

$$\frac{\partial \mathcal{J}_i}{\partial \mu_{1,i}} = \sum_{j=1}^{N} \frac{-\alpha_i \left(\xi_1^{(j)} - \mu_{1,i}\right)}{\sigma_{1,i}^2 \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i} \sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)})\right)}$$

$$\frac{\partial \mathcal{J}_i}{\partial \mu_{2,i}} = \sum_{j=1}^{N} \frac{-\alpha_i \left(\xi_2^{(j)} - \mu_{2,i}\right)}{\sigma_{2,i}^2 \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i} \sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)})\right)}$$

$$\frac{\partial \mathcal{J}_i}{\partial \sigma_{1,i}} = \sum_{j=1}^{N} \frac{-\alpha_i \left(\left(\xi_1^{(j)} - \mu_{1,i}\right)^2 - \sigma_{1,i}^2\right)}{\sigma_{1,i}^3 \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i} \sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)})\right)}$$

$$\frac{\partial \mathcal{J}_i}{\partial \sigma_{2,i}} = \sum_{j=1}^{N} \frac{-\alpha_i \left(\left(\xi_2^{(j)} - \mu_{2,i}\right)^2 - \sigma_{2,i}^2\right)}{\sigma_{2,i}^3 \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i} \sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)})\right)}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \mu_{1,i}^2} = \sum_{j=1}^{N} \frac{\alpha_i \left(\alpha_i \sigma_{1,i} + 2V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i) \left(\left(\xi_1^{(j)} - \mu_{1,i} \right)^2 - \sigma_{1,i}^2 \right) \sigma_{2,i} \right)}{\sigma_{1,i}^3 \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i}\sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)}) \right)^2}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \mu_{1,i} \partial \mu_{2,i}} = \sum_{j=1}^{N} \frac{2V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i)\alpha_i \left(\xi_1^{(j)} - \mu_{1,i} \right) \left(\xi_2^{(j)} - \mu_{2,i} \right)}{\sigma_{1,i}\sigma_{2,i} \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i}\sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)}) \right)^2}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \mu_{1,i} \partial \sigma_{1,i}} = \sum_{j=1}^{N} \frac{2\alpha_i \left(\xi_1^{(j)} - \mu_{1,i} \right) \left(\alpha_i + \frac{V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i) \left(\left(\xi_1^{(j)} - \mu_{1,i} \right)^2 - 3\sigma_{1,i}^2 \right) \sigma_{2,i}}{\sigma_{1,i}} \right)}{\sigma_{1,i}^3 \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i}\sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)}) \right)^2}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \mu_{1,i} \partial \sigma_{2,i}} = \sum_{j=1}^{N} \frac{2V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i)\alpha_i \left(\xi_1^{(j)} - \mu_{1,i} \right) \left(\left(\xi_2^{(j)} - \mu_{2,i} \right)^2 - \sigma_{2,i}^2 \right)}{\sigma_{1,i}\sigma_{2,i}^2 \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i}\sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)}) \right)^2}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \mu_{2,i}^2} = \sum_{j=1}^{N} \frac{\alpha_i \left(\alpha_i \sigma_{2,i} + 2V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i)\sigma_{1,i} \left(\left(\xi_2^{(j)} - \mu_{2,i} \right)^2 - \sigma_{2,i}^2 \right) \right)}{\sigma_{2,i}^3 \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i}\sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)}) \right)^2}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \mu_{2,i} \partial \sigma_{1,i}} = \sum_{j=1}^{N} \frac{2V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i)\alpha_i \left(\xi_2^{(j)} - \mu_{2,i} \right) \left(\left(\xi_1^{(j)} - \mu_{1,i} \right)^2 - \sigma_{1,i}^2 \right)}{\sigma_{1,i}^2\sigma_{2,i} \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i}\sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)}) \right)^2}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \mu_{2,i} \partial \sigma_{2,i}} = \sum_{j=1}^{N} \frac{2\alpha_i \left(\xi_2^{(j)} - \mu_{2,i} \right) \left(\alpha_i + \frac{V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i)\sigma_{1,i} \left(\left(\xi_2^{(j)} - \mu_{2,i} \right)^2 - 3\sigma_{2,i}^2 \right)}{\sigma_{2,i}} \right)}{\sigma_{2,i}^3 \left(\alpha_i + (1-\alpha_i) 2\pi \sigma_{1,i}\sigma_{2,i} V_{i,j} g_{i-1}(\xi^{(j)}) \right)^2}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \sigma_{1,i}^2} = \sum_{j=1}^{N} \frac{2V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i)\alpha_i \left(\left(\xi_1^{(j)} - \mu_{1,i}\right)^2 - \sigma_{1,i}^2\right)\left(\left(\xi_2^{(j)} - \mu_{2,i}\right)^2 - \sigma_{2,i}^2\right)}{\sigma_{1,i}^2\sigma_{2,i}^2\left(\alpha_i + (1-\alpha_i)2\pi\sigma_{1,i}\sigma_{2,i}V_{i,j}g_{i-1}(\xi^{(j)})\right)^2}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \sigma_{1,i}\partial \sigma_{2,i}} = \sum_{j=1}^{N} \frac{2\alpha_i\left(\xi_2^{(j)} - \mu_{2,i}\right)\left(\alpha_i\sigma_{2,i} + V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i)\sigma_{1,i}\left(\left(\xi_2^{(j)} - \mu_{2,i}\right)^2 - 3\sigma_{2,i}^2\right)\right)}{\sigma_{2,i}^4\left(\alpha_i + (1-\alpha_i)2\pi\sigma_{1,i}\sigma_{2,i}V_{i,j}g_{i-1}(\xi^{(j)})\right)^2}$$

$$\frac{\partial^2 \mathcal{J}_i}{\partial \sigma_{2,i}^2} = \sum_{j=1}^{N} \frac{2V_{i,j}\pi g_{i-1}(\xi^{(j)})(1-\alpha_i)\alpha_i \left(\xi_1^{(j)} - \mu_{1,i}\right)\left(\left(\xi_2^{(j)} - \mu_{2,i}\right)^2 - \sigma_{2,i}^2\right)}{\sigma_{1,i}\sigma_{2,i}^2\left(\alpha_i + (1-\alpha_i)2\pi\sigma_{1,i}\sigma_{2,i}V_{i,j}g_{i-1}(\xi^{(j)})\right)^2}$$

Gradient and Hessian calculated using the above formulae can be inserted into any standard non-linear optimization routine to boost its performance. Note that as we mentioned above, for more efficient operation, the exponential terms should be evaluated only once for every iteration of the non-linear optimization routine — during the evaluation of the objective function.

B.4 Approximate Calculation of the Leader Node Selection Criterion

In this section we show how to efficiently calculate the approximate information based leader-node selection criterion based on the definition (6.21). We first note that from the relationship between the mutual information and conditional entropy, $I(X, Y|Z = z) = H(Y|Z = z) - H(Y|X, Z = z)$ we have

$$I(X_{t+1}, Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) = H(Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) - H(Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|X_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \tag{B.2}$$

Second, recalling our assumption of the conditional independence of the measurements we can see

$$H(Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|X_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) = \sum_{j\in\mathfrak{S}_{\ell_{t+1}}} H(Y_t^j|X_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \tag{B.3}$$

Using the definition of the conditional entropy (where $y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}$ is the sequence of measurements that has already been realized [96])

$$\begin{aligned} H(Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|X_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) &= -\sum_{j\in\mathfrak{S}_{\ell_{t+1}}} \int \log p(y_{t+1}^j|x_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) p(x_{t+1}, y_{t+1}^j|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \mathrm{d}x_{t+1}\mathrm{d}y_{t+1}^j \\ &= -\sum_{j\in\mathfrak{S}_{\ell_{t+1}}} \int \log p(y_{t+1}^j|x_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) p(y_{t+1}^j|x_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \mathrm{d}x_{t+1}\mathrm{d}y_{t+1}^j \end{aligned}$$

Since the true predictive density $p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})$ is unknown we have to use its Monte-Carlo approximation consisting of the set of diffused (predictive) particles $\{\xi_{t+1}^{(i)}\}_{i=1}^N$. This results in the following efficient approximation of the above integral:

$$H(Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|X_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) = -\sum_{j\in\mathfrak{S}_{\ell_{t+1}}} \frac{1}{N}\sum_{i=1}^N \int \log p(y_{t+1}^j|\xi_{t+1}^{(i)}) p(y_{t+1}^j|\xi_{t+1}^{(i)}) \mathrm{d}y_{t+1}^j$$

According to our sensor model, the likelihood function can be represented as follows:

$$p(y_{t+1}^j|\xi_{t+1}^{(i)}) = p_d^{y_{t+1}^j\Delta_i^j}(1-p_d)^{(1-y_{t+1}^j)\Delta_i^j} p_f^{y_{t+1}^j(1-\Delta_i^j)}(1-p_f)^{(1-y_{t+1}^j)(1-\Delta_i^j)}, \tag{B.4}$$

where $\Delta_i^j = \mathbf{1}_{\xi_{t+1}^{(i)}\in\mathcal{X}_d^j}$. Straightforward calculation gives

$$\begin{aligned} H(Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|X_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) = -\sum_{j\in\mathfrak{S}_{\ell_{t+1}}} & q^j(p_d\log p_d + (1-p_d)\log(1-p_d)) \\ & + (1-q^j)(p_f\log p_f + (1-p_f)\log(1-p_f)). \end{aligned}$$

Here $q^j = \frac{1}{N}\sum_{i=1}^N \Delta_i^j$ is the average number of particles in the detection region of sensor j. Thus we have constructed an efficient Monte-Carlo approximation to the second summand in the expression for the mutual information between the predicted state X_{t+1}

and the measurements arising in the neighborhood of the leader node ℓ_{t+1}. The first summand, $H(Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})$ is much more difficult to approximate directly using Monte-Carlo technique. One form of decomposing this term [96]

$$H(Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) = \sum_{j=1}^{|\mathfrak{S}_{\ell_{t+1}}|} H(Y_{t+1}^{j}|Y_{t+1}^{1:j-1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \tag{B.5}$$

implies that for a general measurement model the evaluation complexity grows exponentially in the size of the neighborhood $|\mathfrak{S}_{\ell_{t+1}}|$. This is because the sequence of measurements $Y_{t+1}^{1:j-1}$ is unknown and averaging over all possible cases is required. Our experiments revealed that approximating this term using Monte-Carlo sampling of possible measurements is also inefficient. However, for sensors with uninformative (noisy) measurements the following approximation can be used

$$H(Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) = \sum_{j=1}^{|\mathfrak{S}_{\ell_{t+1}}|} H(Y_{t+1}^{j}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})$$

The intuition behind this approximation can be explained as follows. We can represent each term in the decomposition in the following way:

$$H(Y_{t+1}^{j}|Y_{t+1}^{1:j-1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) = -\int \log\left[\int p(y_{t+1}^{j}|x_{t+1}) p(x_{t+1}|y_{t+1}^{1:j-1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \mathrm{d}x_{t+1}\right] \left(\int p(y_{t+1}^{1:j}|x_{t+1}) p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \mathrm{d}x_{t+1}\right) \mathrm{d}y_{t+1}^{1:j}$$

If measurements $Y_{t+1}^{1:j-1}$ are uninformative with respect to the current predictive density $p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})$ their incorporation will not significantly affect the density and the following will hold $p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \approx p(x_{t+1}|y_{t+1}^{1:j-1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})$. In our particular setting a measurement Y_{t+1}^{j} is uninformative (according to the likelihood model (B.4)) if all the particles are either simultaneously inside or outside the detection region $\mathcal{X}_d^j$ of sensor j. This implies that Δ_i^j is same for all i and, consequently, the updated weight of every particle does not depend on the realization of measurement Y_{t+1}^{j}. Such a measurement can be excluded from mutual information calculation without affecting the accuracy of calculation. We observed in our simulations that when the particle representation of predictive density becomes localized

most of the sensors in the neighborhoods of leader-nodes become uninformative and we introduced the approximation $p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \approx p(x_{t+1}|y_{t+1}^{1:j-1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})$ into the calculation of the mutual information. This resulted in a significant simplification (dimensionality reduction) of calculations:

$$\begin{aligned} H(Y_{t+1}^j|Y_{t+1}^{1:j-1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \approx & -\int \log\left[\int p(y_{t+1}^j|x_{t+1})p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})\mathrm{d}x_{t+1}\right] \\ & \left(\int p(y_{t+1}^{1:j}|x_{t+1})p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})\mathrm{d}x_{t+1}\right)\mathrm{d}y_{t+1}^{1:j} \\ = & -\int \log\left[\int p(y_{t+1}^j|x_{t+1})p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})\mathrm{d}x_{t+1}\right] \\ & \left(\int p(y_{t+1}^j|x_{t+1})p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})\mathrm{d}x_{t+1}\right)\mathrm{d}y_{t+1}^j. \end{aligned}$$

Furthermore, using Monte-Carlo representation of predictive density we can approximate the inner integral:

$$\int p(y_{t+1}^j|x_{t+1})p(x_{t+1}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}})\mathrm{d}x_{t+1} \approx q^j p_d^{y_{t+1}^j}(1-p_d)^{1-y_{t+1}^j} + (1-q^j)p_f^{y_{t+1}^j}(1-p_f)^{1-y_{t+1}^j}$$

Finally, calculating the outer integral we obtain:

$$\begin{aligned} H(Y_{t+1}^j|Y_{t+1}^{1:j-1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \approx & -(q^j p_d + (1-q^j)p_f)\log(q^j p_d + (1-q^j)p_f) \\ & -(q^j(1-p_d) + (1-q^j)(1-p_f))\log(q^j(1-p_d) + (1-q^j)(1-p_f)) \end{aligned}$$

Thus according to (B.2), (B.3) and (B.5) we have the following approximate expression for the calculation of mutual information:

$$\begin{aligned} I(X_{t+1}, Y_{t+1}^{\mathfrak{S}_{\ell_{t+1}}}|y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) = & \sum_{j=1}^{|\mathfrak{S}_{\ell_{t+1}}|} H(Y_{t+1}^j|Y_{t+1}^{1:j-1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) - H(Y_{t+1}^j|X_{t+1}, y_{1:t}^{\mathfrak{S}_{\ell_{1:t}}}) \\ \approx & \sum_{j\in\mathfrak{S}_{\ell_{t+1}}} -(q^j p_d + (1-q^j)p_f)\log(q^j p_d + (1-q^j)p_f) \\ & -(q^j(1-p_d) + (1-q^j)(1-p_f))\log(q^j(1-p_d) + (1-q^j)(1-p_f)) \\ & + q^j(p_d\log p_d + (1-p_d)\log(1-p_d)) \\ & + (1-q^j)(p_f\log p_f + (1-p_f)\log(1-p_f)) \end{aligned}$$

Note that this is an extremely fast approximation since its complexity is proportional to $N|\mathfrak{S}_{\ell_{t+1}}|$ operations as opposed to the exponential complexity of exact calculation, which is proportional to $N2^{|\mathfrak{S}_{\ell_{t+1}}|}$.

References

[1] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005.

[2] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design*. Academic Press, Jan. 2008.

[3] P. De and S. K. Das, "Epidemic models, algorithms, and protocols in wireless sensor and ad hoc networks," in *Algorithms and Protocols for Wireless Sensor Networks* (A. Boukerche, ed.), ch. 3, New York, NY: John Wiley & Sons, Inc., Oct. 2008.

[4] W. Morris, *The American Heritage dictionary of the English language*. American Heritage Pub. Co., 1969.

[5] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[6] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, "Collaborative signal and information processing: an information-directed approach," *Proc. IEEE*, vol. 91, pp. 1199–1209, Aug. 2003.

[7] F. Zhao, "Challenges in programming sensor networks," in *Proc. DCOSS*, (Marina del Rey, CA), p. 3, Jul. 2005.

[8] J. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, and S. Roundy, "PicoRadio supports ad hoc ultra-low power wireless networking," *IEEE Computer*, vol. 33, pp. 42–48, Jul. 2000.

[9] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[10] N. Nasser and L. M. Arboleda, "Clustering in wireless sensor networks: a graph theory perspective," in *Algorithms, Protocols Wireless Sensor Networks* (A. Boukerche, ed.), ch. 7, New York, NY: John Wiley & Sons, Inc., Oct. 2008.

[11] S. Schmid and R. Wattenhofer, "Modeling sensor networks," in *Algorithms and Protocols for Wireless Sensor Networks* (A. Boukerche, ed.), ch. 4, New York, NY: John Wiley & Sons, Inc., Oct. 2008.

[12] D. M. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Applications, 3rd Revised and Enlarged Edition*. Vch Verlagsgesellschaft Mbh, Dec. 1998.

[13] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Info. Theory*, vol. 46, pp. 388–404, Mar. 2000.

[14] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Info. Theory*, vol. 52, pp. 2508–2530, Jun. 2006.

[15] M. Penrose, *Random Geometric Graphs*. Oxford University Press, USA, Jul. 2003.

[16] Y. Li, M. T. Thai, and W. Wu, *Wireless Sensor Networks and Applications*. Signals and Communication Technology, Secaucus, NJ: Springer-Verlag New York, Inc., 2007.

[17] A. Mielke, S. Brennan, M. Smith, D. Torney, A. Maccabe, and J. F. Karlin, "Independent sensor networks," *IEEE Instrumentation & Measurement Magazine*, vol. 8, pp. 33–37, Jun. 2005.

[18] S. Brennan, A. Mielke, and D. Torney, "Radioactive source detection by sensor networks," *IEEE Trans. Nuclear Science*, vol. 52, pp. 813–819, Jun. 2005.

[19] S. Li, S. Son, and J. Stankovic, "Event detection services using data service middleware in distributed sensor networks," *Telecomm. Syst.*, vol. 26, pp. 351–368, Jun. 2004.

[20] N. Dziengel, G. Wittenburg, and J. Schiller., "Towards distributed event detection in wireless sensor networks," in *Proc. DCOSS*, (Santorini, Greece), Jun. 2008.

[21] G. Wittenburg, N. Dziengel, and J. Schiller, "In-network training and distributed event detection in wireless sensor networks," in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, (Raleigh, NC), pp. 387–388, Nov. 2008.

[22] I. Krontiris, Z. Benenson, T. Giannetsos, F. C. Freiling, and T. Dimitriou, "Cooperative intrusion detection in wireless sensor networks," in *Proc. EWSN*, (Cork, Ireland), pp. 263–278, Feb. 2009.

[23] E. A. Basha, S. Ravela, and D. Rus, "Model-based monitoring for early warning flood detection," in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, (Raleigh, NC), pp. 295–308, Nov. 2008.

[24] J. Ko, R. Musăloiu-Elefteri, J. H. Lim, Y. Chen, A. Terzis, T. Gao, W. Destler, and L. Selavo, "MEDiSN: medical emergency detection in sensor networks," in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, (Raleigh, NC), pp. 361–362, Nov. 2008.

[25] J. Yuan, X. Liu, and G. H. Chen, "An efficient event detection scheme for wireless sensor networks," in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, (Raleigh, NC), pp. 377–378, Nov. 2008.

[26] M. Arattano and L. Marchi, "Systems and sensors for Debris-flow monitoring and warning," *Sensors*, vol. 8, no. 4, pp. 2436–2452, 2008.

[27] N. Ahmed, Y. Dong, T. Bokareva, S. Kanhere, S. Jha, T. Bessell, M. Rutten, B. Ristic, and N. Gordon, "Detection and tracking using wireless sensor networks," in *Proc. 5th ACM Conf. Embedded Netw. Sens. Syst.*, (Sydney, Australia), pp. 425–426, 2007.

[28] B. Kusy, A. Ledeczi, and X. Koutsoukos, "Tracking mobile nodes using RF doppler shifts," in *Proc. 5th ACM Conf. Embedded Netw. Sens. Syst.*, (Sydney, Australia), pp. 29–42, 2007.

[29] K. Kim, J. Jun, S. Kim, and B. Y. Sung, "Medical asset tracking application with wireless sensor networks," in *Proc. 2nd Int. Conf. Sens. Tech. and Appl.*, (Cap Esterel, France), pp. 531–536, Aug. 2008.

[30] H.-C. Lee, C.-J. Liu, J. Yang, J.-T. Huang, Y.-M. Fang, B.-J. Lee, and C.-T. King, "Using mobile wireless sensors for in-situ tracking of debris flows," in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, (Raleigh, NC), pp. 407–408, 2008.

[31] D. Aghajarian and R. Berangi, "A fast distributed target tracking algorithm for low density binary sensor networks," in *Proc. 2nd Int. Conf. Sens. Tech. and Appl.*, pp. 1–6, Aug. 2008.

[32] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. A. Stankovic, and T. Abdelzaher, "Achieving real-time target tracking using wireless sensor networks," in *Proc. 12th IEEE Real-Time Embedded Tech. Appl. Symp.*, (San Jose, CA), pp. 37–48, Apr. 2006.

[33] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Trans. Sens. Netw.*, vol. 2, no. 1, pp. 1–38, 2006.

[34] T. Onel, E. Onur, C. Ersoy, and H. Delic, "Wireless sensor networks for security: issues and challenges," in *Adv. Sensing Security Appl.* (J. Byrnes and G. Ostheimer, eds.), vol. 2, ch. 5, pp. 95–119, Netherlands: Springer, 2006.

[35] S. Duttagupta, K. Ramamritham, P. Kulkarni, and K. M. Moudgalya, "Tracking dynamic boundary fronts using range sensors," in *Proc. EWSN*, (Bologna, Italy), pp. 125–140, Jan. 2008.

[36] T. Zhao and A. Nehorai, "Detecting and estimating biochemical dispersion of a moving source in a semi-infinite medium," *IEEE Trans. Signal Process.*, vol. 54, pp. 2213–2225, Jun. 2006.

[37] T. Zhao and A. Nehorai, "Localization of diffusive sources using distributed sequential Bayesian methods in wireless sensor networks," in *Proc. ICASSP*, vol. 4, pp. 985–988, May 2006.

[38] P. Beyens, A. Nowe, and K. Steenhaut, "High-density wireless sensor networks: a new clustering approach for prediction-based monitoring," in *Proc. EWSN*, (Istanbul, Turkey), pp. 188–196, Jan. 2005.

[39] E.-O. Blass, J. Horneber, and M. Zitterbart, "Analyzing data prediction in wireless sensor networks," in *IEEE Vehicular Tech. Conf.*, (Marina Bay, Singapore), pp. 86–87, May 2008.

[40] G. Mao, B. D. Anderson, and B. Fidan, "Path loss exponent estimation for wireless sensor network localization," *Comp. Netw.*, vol. 51, no. 10, pp. 2467–2483, 2007.

[41] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Gamt, "A system for LEASE: location estimation assisted by stationary emitters for indoor RF wireless networks," in *Proc. INFOCOM*, vol. 2, (Hong Kong), pp. 1001–1011, Mar. 2004.

[42] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, pp. 2137–2148, Aug. 2003.

[43] A.-K. Chandra-Sekaran, G. Flaig, C. Kunze, W. Stork, and K. D. Mueller-Glaser, "Efficient resource estimation during mass casualty emergency response based on a location aware disaster aid network," in *Proc. EWSN*, (Bologna, Italy), pp. 205–220, Jan. 2008.

[44] M. Arabaci and R. N. Strickland, "Direction of arrival estimation in reverberant rooms using a resource-constrained wireless sensor network," in *Proc. IEEE Conf. Pervasive Services*, (Istanbul, Turkey), pp. 29–38, Jul. 2007.

[45] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, pp. 215–233, Jan. 2007.

[46] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filter," in *Proc. 44th IEEE Conf. Decision Control*, (Seville, Spain), pp. 8179–8184, Dec. 2005.

[47] D. V. Dimarogonas and K. J. Kyriakopoulos, "Formation control and collision avoidance for multi-agent systems and a connection between formation infeasibility and flocking behavior," in *Proc. 44th IEEE Conf. Decision Control*, (Seville, Spain), pp. 84–89, Dec. 2005.

[48] E. Di Nitto, D. J. Dubois, R. Mirandola, F. Saffre, and R. Tateson, "Applying self-aggregation to load balancing: experimental results," in *Proc. 3rd Int. Conf. Bio-Inspired Models of Netw. Info. and Comp. Syst.*, (Hyogo, Japan), pp. 1–8, 2008.

[49] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline, "PIPENET: a wireless sensor network for pipeline monitoring," in *Proc. IPSN*, (Cambridge, MA), pp. 264–273, ACM, Apr. 2007.

[50] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proc. EWSN*, (Istanbul, Turkey), pp. 108–120, Jan. 2005.

[51] Y. K. Keewook Na and H. Cha, "Acoustic sensor network-based parking lot surveillance system," in *Proc. EWSN*, (Cork, Ireland), pp. 247–262, Feb. 2009.

[52] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Comm. Letters*, vol. 9, pp. 976–978, Nov. 2005.

[53] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. Sens. Netw.*, vol. 5, no. 1, pp. 1–39, 2009.

[54] V. R. Syrotiuk and B. Li, "Heterogeneous wireless sensor networks," in *Algorithms and Protocols for Wireless Sensor Networks* (A. Boukerche, ed.), ch. 7, New York, NY: John Wiley & Sons, Inc., Oct. 2008.

[55] Crossbow, "MICA2 datasheet." available at `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf`, Last accessed July 2009.

[56] S. Kumar, *Encyclopaedia of Operating System.* Anmol Publications Pvt Ltd., 2005.

[57] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP J. Appl. Signal Process.*, vol. 2003, pp. 378–391, 2003.

[58] N. A. Lynch, *Distributed Algorithms.* Morgan Kaufmann, 1996.

[59] V. V. Veeravalli and J.-F. Chamberland, "Detection in sensor networks," in *Signal Processing for Sensor Networks* (A. Swami, Q. Zhao, Y.-W. Hong, and L. Tong, eds.), ch. 6, New York, NY: John Wiley & Sons, Inc., 2007.

[60] A. Ribeiro, I. D. Schizas, J.-J. Xiao, G. B. Giannakis, and Z.-Q. Luo, "Distributed estimation under bandwidth and energy constraints," in *Signal Process. for Sensor Networks* (A. Swami, Q. Zhao, Y.-W. Hong, and L. Tong, eds.), ch. 7, New York, NY: John Wiley & Sons, Inc., 2007.

[61] M. Çetin, L. Chen, J. W. Fisher, A. T. Ihler, O. P. Kreidl, R. L. Moses, M. J. Wainwright, J. L. Williams, and A. S. Willsky, "Graphical models and fusion in sensor networks," in *Signal Processing for Sensor Networks* (A. Swami, Q. Zhao, Y.-W. Hong, and L. Tong, eds.), ch. 9, New York, NY: John Wiley & Sons, Inc., 2007.

[62] C. Moallemi and B. Van Roy, "Consensus propagation," *IEEE Trans. Info. Theory*, vol. 52, pp. 4753–4766, Jun. 2006.

[63] M. H. De Groot, "Reaching a consensus," *J. Am. Stat. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.

[64] V. Borkar and P. Varaiya, "Asymptotic agreement in distributed estimation," *IEEE Trans. Automatic Control*, vol. 27, pp. 650–655, Jun. 1982.

[65] J. N. Tsitsiklis, *Problems in Decentralized Decision Making and Computation*. PhD thesis, Department of EECS, MIT, Nov. 1984.

[66] D. Bauso, L. Giarré, and R. Pesenti, "Non-linear protocols for optimal distributed consensus in networks of dynamic agents," *Syst. Control Letters*, vol. 55, no. 11, pp. 918–928, 2006.

[67] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Rev.*, vol. 46, no. 4, pp. 667–689, 2004.

[68] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. IEEE/ACM Int. Symp. Info. Process. Sens. Netw.*, (Los Angeles, CA), Apr 2005.

[69] E. Kokiopoulou and P. Frossard, "Polynomial filtering for fast convergence in distributed consensus," *IEEE Trans. Signal Process.*, vol. 57, pp. 342–354, Jan. 2009.

[70] P. Denantes, F. Benezit, P. Thiran, and M. Vetterli, "Which distributed averaging algorithm should I choose for my sensor network?," in *Proc. IEEE INFOCOM*, (Phoenix, AZ), pp. 986–994, Apr. 2008.

[71] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging.," *Syst. Control Letters*, vol. 53, pp. 65–78, Sep. 2004.

[72] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, 1985.

[73] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automatic Control*, vol. 49, pp. 1520–1533, Sep. 2004.

[74] T. Aysal, B. Oreshkin, and M. Coates, "Accelerated distributed average consensus via localized node state prediction," *IEEE Trans. Signal Process.*, vol. 57, pp. 1563–76, Apr. 2009.

[75] B. Johansson and M. Johansson, "Faster linear iterations for distributed averaging," in *Proc. IFAC World Congress*, (Seoul, South Korea), Jul. 2008.

[76] M. Cao, D. A. Spielman, and E. M. Yeh, "Accelerated gossip algorithms for distributed computation," in *Proc. 44th Allerton Conf. Comm. Control Comp.*, (Monticello, IL, USA), Sep. 2006.

[77] D. Scherber and H. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proc. ACM/IEEE Int. Symp. Info. Process. Sens. Netw.*, (Berkeley, CA, USA), Apr. 2004.

[78] S. Sundaram and C. Hadjicostis, "Distributed consensus and linear function calculation in networks: an observability perspective," in *Proc. IEEE/ACM Int. Symp. Info. Process. Sens. Netw.*, (Cambridge, MA, USA), Apr. 2007.

[79] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. ACM Symp. Theory Comp.*, (Chicago, IL, USA), Jun. 2004.

[80] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distributed Comp.*, vol. 67, pp. 33–46, Jan. 2007.

[81] C. D. Meyer and R. J. Plemmons, "Convergent powers of a matrix with applications to iterative methods for singular linear systems," *SIAM J. Num. Analysis*, vol. 14, no. 4, pp. 699–705, 1977.

[82] A. Poznyak, "A new version of the strong law of large numbers for dependent vector processes with decreasing correlation," in *Proc. IEEE Conf. Decision and Control*, vol. 3, (Sydney, NSW, Australia), pp. 2881–2882, Dec. 2000.

[83] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, Feb. 1997.

[84] D. Aldous and J. Fill, "Reversible Markov chains and random walks on graphs." Manuscript in preparation; available at `http://www.stat.berkeley.edu/~aldous/RWG/book.html`, Last accessed March, 2009.

[85] G. H. Golub and C. F. Van Loan, *Matrix computations*. Baltimore, MD: Johns Hopkins Press, 3 ed., 1996.

[86] F. Tisseur and K. Meerbergen, "The quadratic eigenvalue problem," *SIAM Rev.*, vol. 43, no. 2, pp. 235–286, 2001.

[87] A. Jazwinski, *Stochastic processes and filtering theory*. New York: Academic Press, 1970.

[88] R. R. Brooks, P. Ramanathan, and A. Sayeed, "Distributed target classification and tracking in sensor networks," *Proc. IEEE*, vol. 91, pp. 1163–1171, Aug. 2003.

[89] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Automatic Control*, vol. 17, pp. 439–448, Aug. 1972.

[90] F. Martinerie, "Data fusion and tracking using HMMs in a distributed sensor network," *IEEE Trans. Aerospace and Electronic Syst.*, vol. 33, pp. 11–28, Jan. 1997.

[91] A. Doucet, N. de Freitas, and N. Gordon, eds., *Sequential Monte Carlo Methods in Practice*. Berlin: Springer–Verlag, 2001.

[92] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 174–188, Feb. 2002.

[93] M. Coates, "Distributed particle filtering for sensor networks," in *Proc. IEEE/ACM Int. Symp. Info. Process. Sens. Netw.*, (Berkeley, CA), Apr. 2004.

[94] X. Sheng and Y.-H. Hu, "Distributed particle filter with GMM approximation for multiple target localization and tracking in wireless sensor network," in *Proc. IEEE/ACM Int. Symp. Info. Process. Sens. Netw.*, (Los Angeles, CA), Apr. 2005.

[95] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Process. Magazine*, vol. 19, pp. 61–72, Mar. 2002.

[96] J. L. Williams, J. W. Fisher, and A. S. Willsky, "Approximate dynamic programming for communication-constrained sensor network management," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4300–4311, 2007.

[97] A. Ihler, J. Fisher, and A. Willsky, "Particle filtering under communication constraints," in *Proc. IEEE Workshop Stat. Signal Process.*, (Bordeaux, France), May 2005.

[98] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F Radar and Signal Process.*, vol. 140, pp. 107–113, Apr. 1993.

[99] M. Isard and A. Blake, "CONDENSATION — conditional density propagation for visual tracking," *Int. J. Comp. Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[100] D. Crisan, P. Del Moral, and T. Lyons, "Discrete filtering using branching and interacting particle systems," *Markov Processes and Related Fields*, vol. 5, no. 3, pp. 293–318, 1999.

[101] S. Maskell and N. Gordon, "A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking," in *Proc. IEE Workshop Target Tracking: Algorithms Appl.*, vol. 2, pp. 2/1–2/15, Oct. 2001.

[102] D. Pollard, *Convergence of Stochastic Processes*. Springer-Verlag, 1984.

[103] J. Geweke, "Bayesian inference in econometric models using Monte Carlo integration," *Econometrica*, vol. 57, pp. 1317–1340, 1989.

[104] V. S. Zaritskii, V. B. Svetnik, and L. I. Shimelevich, "Monte Carlo technique in problems of optimal data processing," *Automation and Remote Control*, vol. 12, pp. 95–103, 1975.

[105] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *J. Am. Statist. Assoc.*, vol. 89, pp. 278–288, 1994.

[106] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. Am. Statist. Assoc.*, vol. 93, pp. 1032–1044, 1998.

[107] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for Bayesian filtering," *Stat. Comp.*, vol. 10, no. 3, pp. 197–208, 2000.

[108] J. E. Handschin and D. Q. Mayne, "Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering," *Int. J. Control*, vol. 9, pp. 547–559, 1969.

[109] M. Pitt and N. Shepard, "Filtering via simulation: auxiliary particle filters," *J. Am. Statist. Assoc.*, vol. 94, pp. 590–599, Jun. 1999.

[110] H. Kunita, "Asymptotic behavior of the nonlinear filtering errors of markov processes," *J. Multivariate Analysis*, vol. 1, pp. 365–393, Dec. 1971.

[111] D. Ocone and E. Pardoux, "Asymptotic stability of the optimal filter with respect to its initial condition," *SIAM J. Control Optimization*, vol. 34, no. 1, pp. 226–243, 1996.

[112] P. Del Moral and A. Guionnet, "On the stability of interacting processes with applications to filtering and genetic algorithms," *Ann. de l'Inst. H. Poincaré*, vol. 37, no. 2, pp. 155–194, 2001.

[113] P. Del Moral and L. Miclo, "On the stability of nonlinear Feynman-Kac semigroups," *Ann. de la faculté des sciences de Toulouse*, vol. (6)11, no. 2, pp. 135–175, 2002.

[114] P. Del Moral, *Feynman–Kac formulae. Genealogical and interacting particle approximations*. New York: Springer, 2004.

[115] F. Le Gland, C. Musso, and N. Oudjane, "An analysis of regularized interacting particle methods for nonlinear filtering," in *Proc. IEEE European Workshop Comp.-Intensive Methods in Control and Data Process.*, pp. 167–174, Sep. 1998.

[116] F. Le Gland and N. Oudjane, "A robustification approach to stability and to uniform particle approximation of nonlinear filters: the example of pseudo-mixing signals," *Stochastic Processes and their Appl.*, vol. 38, pp. 279–316, Aug. 2003.

[117] F. Le Gland and N. Oudjane, "Stability and uniform approximation of nonlinear filters using the Hilbert metric, and application to particle filters," *Ann. Appl. Probab.*, vol. 14, pp. 144–187, Feb. 2004.

[118] N. Vaswani, "Change detection in partially observed nonlinear dynamic systems with unknown change parameters," in *Proc. Am. Control Conf.*, vol. 6, pp. 5387–5393, Jul. 2004.

[119] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering," in *Proc. 4th Int. Symp. Image and Signal Process. and Analysis*, pp. 64–69, Sep. 2005.

[120] J. Kotecha and P. Djuric, "Gaussian sum particle filtering," *IEEE Trans. Signal Process.*, vol. 51, pp. 2602–2612, Oct. 2003.

[121] M. A. F. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 381–396, Mar. 2002.

[122] A. Rakhlin, *Applications of Empirical Processes in Learning Theory: Algorithmic Stability and Generalization Bounds*. PhD thesis, MIT, Jun. 2006.

[123] A. J. Zeevi and R. Meir, "Density estimation through convex combinations of densities: approximation and estimation bounds," *Neural Netw.*, vol. 10, pp. 99–109, Jan. 1997.

[124] J. Li and A. Barron, "Mixture density estimation," in *Adv. Neural Info. Process. Syst. 12* (S. A. Solla, T. K. Leen, and K.-R. Muller, eds.), San Mateo, CA.: Morgan Kaufmann Publishers, 1999.

[125] A. Rakhlin, D. Panchenko, and S. Mukherjee, "Risk bounds for mixture density estimation," *ESAIM: Probab. and Statist.*, vol. 9, pp. 220–229, 2005.

[126] B. W. Silverman, *Density Estimation*. London: Chapman and Hall, 1986.

[127] T. Zhang, "Sequential greedy approximation for certain convex optimization problems," *IEEE Trans. Info. Theory*, vol. 49, no. 3, pp. 682–691, 2003.

[128] M. Ledoux and M. Talagrand, *Probability in Banach Spaces*. Springer, May 1991.

[129] A. van der Vaart and J. Wellner, *Weak Convergence and Empirical Processes*. New York: Springer, 1996.

[130] L. Devroye and G. Lugosi, *Combinatorial Methods in Density Estimation*. New York: Springer-Verlag, 2001.

[131] M. J. Coates and G. Ing, "Sensor network particle filters: Motes as particles," in *Proc. IEEE Workshop Stat. Signal Process.*, (Bordeaux, France), Jul. 2005.

[132] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: efficient averaging for sensor networks," *IEEE Trans. Signal Process.*, vol. 56, pp. 1205–1216, Mar. 2008.

[133] D. Fox, "KLD-sampling: Adaptive particle filters," in *Adv. in Neural Info. Process. Syst. 14*, pp. 713–720, MIT Press, 2001.

[134] C. Kwok, D. Fox, and M. Meila, "Real-time particle filters," *Proc. IEEE*, vol. 92, pp. 469–484, Mar. 2004.