OPTIMIZATION AND RESOURCE MANAGEMENT IN WIRELESS SENSOR NETWORKS

by

NICHOLAS ROSEVEARE

B. S., Colorado State University, 2005
M. S., Colorado State University, 2007

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2013

Abstract

In recent years, there has been a rapid expansion in the development and use of low-power, low-cost wireless modules with sensing, computing, and communication functionality. A wireless sensor network (WSN) is a group of these devices networked together wirelessly. Wireless sensor networks have found widespread application in infrastructure, environmental, and human health monitoring, surveillance, and disaster management.

While there are many interesting problems within the WSN framework, we address the challenge of energy availability in a WSN tasked with a cooperative objective. We develop approximation algorithms and execute an analysis of concave utility maximization in resource constrained systems. Our analysis motivates a unique algorithm which we apply to resource management in WSNs. We also investigate energy harvesting as a way of improving system lifetime. We then analyze the effect of using these limited and stochastically available communication resources on the convergence of decentralized optimization techniques.

The main contributions of this research are: (1) new optimization formulations which explicitly consider the energy states of a WSN executing a cooperative task; (2) several analytical insights regarding the distributed optimization of resource constrained systems; (3) a varied set of algorithmic solutions, some novel to this work and others based on extensions of existing techniques; and (4) an analysis of the effect of using stochastic resources (e.g., energy harvesting) on the performance of decentralized optimization methods.

Throughout this work, we apply our developments to distribution estimation and rate maximization. The simulation results obtained help to provide verification of algorithm performance. This research provides valuable intuition concerning the trade-offs between energy-conservation and system performance in WSNs.

OPTIMIZATION AND RESOURCE MANAGEMENT IN WIRELESS SENSOR NETWORKS

by

NICHOLAS ROSEVEARE

B. S., Colorado State University, 2005
M. S., Colorado State University, 2007

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2013

Approved by:

Major Professor
Balasubramaniam Natarajan

Copyright

Abstract

In recent years, there has been a rapid expansion in the development and use of low-power, low-cost wireless modules with sensing, computing, and communication functionality. A wireless sensor network (WSN) is a group of these devices networked together wirelessly. Wireless sensor networks have found widespread application in infrastructure, environmental, and human health monitoring, surveillance, and disaster management.

While there are many interesting problems within the WSN framework, we address the challenge of energy availability in a WSN tasked with a cooperative objective. We develop approximation algorithms and execute an analysis of concave utility maximization in resource constrained systems. Our analysis motivates a unique algorithm which we apply to resource management in WSNs. We also investigate energy harvesting as a way of improving system lifetime. We then analyze the effect of using these limited and stochastically available communication resources on the convergence of decentralized optimization techniques.

The main contributions of this research are: (1) new optimization formulations which explicitly consider the energy states of a WSN executing a cooperative task; (2) several analytical insights regarding the distributed optimization of resource constrained systems; (3) a varied set of algorithmic solutions, some novel to this work and others based on extensions of existing techniques; and (4) an analysis of the effect of using stochastic resources (e.g., energy harvesting) on the performance of decentralized optimization methods.

Throughout this work, we apply our developments to distribution estimation and rate maximization. The simulation results obtained help to provide verification of algorithm performance. This research provides valuable intuition concerning the trade-offs between energy-conservation and system performance in WSNs.

Table of Contents

List of Figures

List of Tables

Acknowledgments

It is difficult to express the immense gratitude I have for the opportunity to work with my advisor, Bala Natarajan. Your persistent patience, motivating guidance, and enthusiastic stream of new ideas has been incalculably valuable. Not only in academics, but through many difficult and trying life events you have been a constant source of encouragement and direction. I much admire your ability to maintain the high level view of problems and not get lost in the details, and I now constantly aspire to find and hold on to this perspective for life and research.

I would like to thank M2 Technologies for partially funding this work through a contract from the Marine Corps Systems Command. Additionally, I am extremely grateful to Prof. Bala and the Department of Electrical and Computer Engineering at Kansas state university for funding the final years of this research.

I am grateful to my committee for their input, whether in teaching classes from which I benefited, or in the discussions we have had on various academic or applied topics. Prof.s Starrett, Schinstock, Burckel, and Robby, thank you. I am grateful to Prof. Burckel for putting up with my periodic visits and my varied, immaterial, and often ill-posed mathematical questions. Thank you also for the chance to practice my halting German.

I am beholden for the chance to be a part of such a unique group of students, faculty, and staff in the department of electrical engineering at Kansas state university. The members of the wireless communication laboratory in particular have made my experience memorable. My lab mates Siddharth, Shafiul, Mohammad, Mohammed, Michal, Ahmad, Lutfa, Chang, Allison, Sayak, Boran, Brian, and Dalin, have all provided elements of an experience that will shape the rest of my life. Their input and friendship during this time will forever be a part of my model of earnest academic personal development, creative thinking, and sincere conversation. I am especially thankful for you, Sidd, for being a comrade-in-arms and fellow 'sufferer' as we persevered to the end. The professors I have worked under in fulfilling my

teaching responsibilities, Prof.s Natarjan, Devore, and Soldan, have been most gracious and with whom working has most enjoyable. They have encouraged me to try new teaching tactics and have often provided technical aid when necessary.

I am indebted to the many friends who have supported me during my time in Manhattan, particularly when I was less available. To my many and varied roommates, David, Daniel, and John, I am grateful for your continued input in my life. In particular, Rick and Neva have exhorted me through many life decisions and struggles. While circumstances give us learning opportunities, we need wisdom to understand and live rightly in them. Thank you.

To my family, who, even a state away, has been a constant source of encouragement and has reminded me of the things that matter. My brothers Nate and John, thank you for the conversations we have had with increasing frequency, I am proud of you and so glad of our brotherhood and friendship. My sisters, Kirstin and Kelsey, always bringing light and joy to others wherever you go, thank you for waking me up to the realities of relationship when my world seemed at times to exclude it. My girl, Teresa, it has been incredible being with you, thank you for receiving my little joys and great weaknesses, and for your patience with my limited availability during our time in Manhattan. My parents, Jim and Susan, your perseverance in life and dedication to one another despite trials and hardship, is a standard up to which I will hold the direction of the rest of my life, thank you for giving.

A significant aspect of all these relationships and the encouragement and direction they have offered has helped me to keep a tension, that tension between envisioning the future and exerting in the present. Both are necessary in life.

The discerning sets his face toward wisdom,
but the eyes of a fool are on the ends of the earth.
Proverbs 17:24 (ESV)

Where there is no prophetic vision the people cast off restraint,
but blessed is he who keeps the law.
Proverbs 29:18 (ESV)

Dedication

zu denen, die ihr Leben widmen, damit andere die Wahrheit wissen können

soli Deo gloria

Notation and Abbreviations

ADMM	alternating direction multiplier method
BLU(E)	best linear unbiased (estimator)
BPSK	binary phase shift keying
BW	bandwidth
CDF	cumulative density function
CLT	central limit theorem (page 173)
CTG	(optimal) cost-to-go
CVX	convex (approximation)
DC	difference of convex functions program
DP	dynamic programming
du	distance units
EH	energy harvesting
eu	energy units
FB	feedback
KF	Kalman filter
KKT	Karush-Kuhn-Tucker (optimality condition) (page 29)
LCVX	linear constraint convex (approximation)
LQG	linear quadratic Gaussian (regulator)
LS	least squares
LTI	linear time-invariant
MAP	maximum *a priori* (estimation)
MIMM MIMUM	minimum increment matched marginal utility method (page 151)
MINLP	mixed-integer non-linear program
ML	maximum likelihood
MMSE	minimum mean-squared error (estimator)
MPC	model predictive control
MSE	mean-squared error
NC	non-causal
NCV	nearly constant velocity (state-space movement model)
NUM	network utility maximizaion
PDF	probability density function
PSK	phase shift keying
pu	power units
PV	position-velocity
QAM	quadrature amplitude modulation
rem	remaining (battery)
RFA	random feasible allocation (method)
RV	random variable

SCP	sequential convex programming
SINR	signal-to-interference-plus-noise ratio
SQP	sequential quadratic programming
SVD	singular value decomposition
tu	time units
UPBD	upper bound
WC	worst case
WSN	wireless sensor network

a	scalar
$\mathbf{a}$	vector (bold lower case) (e.g., $\mathbf{a} \in \mathbb{R}^d$: real, of dimension $d \times 1$)
$\mathbf{A}$	matrix (bold upper case)
$[\mathbf{A}]_{(i,j)}$	i, j element of matrix
$(\cdot)^T$	matrix/vector transpose
$\det(\mathbf{A})$ or $\|\mathbf{A}\|$	determinant of a matrix
$\mathbf{tr}(\mathbf{A})$	trace of a matrix
$\mathrm{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_n)$	matrix composed of diagonal blocks of arguments
$\|\|\mathbf{a}\|\|$, $\|\|\mathbf{A}\|\|$	vector and matrix norms (usually Euclidean)
$\mathrm{vec}(\mathbf{A})$	column-wise vectorization of matrix $\mathbf{A}$
$\mathbf{I}_d$	identity matrix size $d \times d$
$\mathbf{0}_d$ or $\mathbf{0}$	zero vector (or zero matrix) size $d \times 1$ (or $d \times d$)
δ_{jk}	Kronecker delta function (:= 1 iff $j = k$, and 0 otherwise)
$\mathbb{R}, \mathbb{N}, \mathbb{Z}$	real numbers, natural numbers, and integers, resp.
$\mathbf{x}_{nk}$	subscripts indicate sensor index n and time index k
$\lfloor \cdot \rfloor$	nearest integer rounded down
$\lceil \cdot \rceil$	nearest integer rounded up
$[\mathbf{x}]^+$	positive orthant projection, (element-wise) $\max\{\mathbf{x}, \mathbf{0}\}$
$\mathcal{N}_n$	neighborhood of nodes to node n
$f(\cdot)$, $\mathbf{f}(\cdot)$	generic scalar/vector function
$D(\cdot)$	distortion metric/objective
$\hat{x}$, $\hat{\mathbf{x}}$	estimated vectors or scalars
$\mathbf{x}_{k\|j}$	prediction or update at time k based on time j
$\mathbf{E}[\cdot]$	expectation operator
$\mathrm{var}(x)$ or σ_x^2	variance of a random quantity
$\mathrm{cov}(\cdot, \cdot)$	covariance between a pair of random quantities
$\mathbf{v} \sim \mathcal{N}(\mu, \mathbf{C})$	multi-variate Gaussian random vector of dimension $k \times 1$ with mean μ, covariance matrix $\mathbf{C}$, and density $f(\mathbf{v}) := (2\pi)^{-k/2}\|\mathbf{C}\|^{-1/2} \exp(-(\mathbf{v}-\mu)^T\mathbf{C}^{-1}(\mathbf{v}-\mu)/2)$, (note duplicate use of $\mathcal{N}$)

"Mathematics is the art of giving the same name to different things." $\sim$ Henri Poincaré

Chapter 1

Introduction

1.1 Why Wireless Sensor Networks?

Recent advances in fabrication technology have enabled the design of low-power, low-cost sensors equipped with self-contained sensing, computing, and communications functions. Such devices networked wirelessly have been referred to as wireless sensor networks (WSNs) and have been hailed as one of the most influential technologies of this century [1]. Wireless sensor networks have the potential to dramatically reshape a broad spectrum of applications, including infrastructure monitoring and surveillance, disaster management, monitoring of the environment, the human health [2], industrial equipment, workplaces, *et cetera*.

Our work focuses on the management of resources with respect to the completion of cooperative tasks within the WSN framework. We specifically focus on the optimal (or sometimes heuristically 'smart') management of energy. In many WSNs, energy is limited to whatever the onboard battery or storage device can hold. We implement and analyze a common solution, which is to devise energy harvesting and management techniques to aid in intelligent allocation of power to sensor nodes at run-time [3, 4].

Challenges

Wireless sensor networks can be thought of as a special, cooperative, case of wireless ad hoc networks. WSNs usually assume a multi-hop communication framework with no centralized infrastructure. Instead, individual sensors coordinate by forwarding packets from other

nodes for delivery from a source to a destination node. Although some kind of centralization of the data is necessary for an end-user, the multi-hop nature of sensor networks is a practical requirement imposed by energy-consumption. This is motivated by the superlinear power loss of wireless transmissions with respect to the propagation distance. More generally, the coordination of wireless sensor networks poses a number of interesting challenges: minimal energy availability and low power operation in each sensor node, use of power-saving sleep/wake-up schemes, scalability in the presence of a large number of sensors, possibility of delayed data, frequent node failures and network topology changes, collaborative signal processing and data aggregation techniques to cope with the large number of sensors (the information from which may congest the network), and smart communication protocols to deal with high collision rates common when so many sensors are broadcasting data [5]. Challenges also surface with regard to the implementation of energy harvesting devices. Since harvestable energy in the environment is often limited and stochastic, the goal of a WSN with an energy harvesting device is to communicate efficiently while relying on unpredictable energy resources.

This discussion concerning the unique features and challenges of wireless sensor networks (WSNs) can be summarized in the following points:

- Objective oriented: Sensor networks are focused on processing and moving information around the network, with the objective of delivering measured data to appropriate destinations in a timely manner. Since the system inherently deals with mother nature in its measurements, the quality of the data represents level of value to the destinations. Therefore, the overall objective is no longer to maximize the raw data throughput, but instead to maximize the useful information delivered to destinations.

- Application based: Various metrics have different value depending on the application. The choice of metrics and data/resource management methods used in sensor networks thus depend on the characteristics and needs of the application. As an example, consider mission-critical applications, the end-to-end latency is critical, and should be

kept below certain threshold. Which added constraints the system expresses is also dependent on the applications.

- Collaborative: How the networked nodes work together to realize the global system objective usually supersedes the objective of achieving fairness of individual connections. This opposes the convention in typical networks (e.g., non-cooperative multi-user throughput optimization) where fairness to users is a chief design criterion.

- Energy-constrained: Most of the low-power devices in sensor networks have limited energy storage and replacing batteries on thousands of these devices is infeasible. Even in systems with energy harvesting capabilities, the energy must be carefully managed to prevent a critical node from becoming inoperable. This becomes a preeminent consideration in the design of algorithms for processing measurements and control commands.

These systems are characterized by common goals molded to specific problems and an apathy to individual fairness with respect to individual nodes (unless they are directed to conserve energy).

In this dissertation, we have asked the following questions concerning these challenges, (which more or less follow the order in which our findings are presented):

- What kinds of metrics are suitable for the representing cooperative behavior of wireless sensor networks?

- How can we represent resource constrained tasks in a way that useful optimization techniques apply?

- Are there any useful analytical insights concerning the optimization of constrained communication in WSNs? How can these insights be leveraged algorithmically?

- Are there any practical decentralized optimization algorithms that can be applied to the WSN system?

- What is the effect of intermittent resource availability via energy harvesting on the performance of decentralized optimization methods in WSNs?

1.2 Prior Work

The pertinent prior work can be broken down into categories based on the previously mentioned challenges and questions of interest. The question of metrics and energy-constrained problem formulations are addressed by a plethora of work on distributed estimation and tracking (related and unrelated to WSNs). The desire for further analytical insights concerns previous efforts on the analysis and design of utility maximization techniques, in both the centralized and decentralized architectures. Lastly, we highlight the relevant prior work which has addressed the analysis of energy harvesting systems.

Distributed Estimation and Tracking

Related to the challenges of the distributed nature of WSNs and the formulation of metrics and optimal solutions, distributed estimation incorporates the effects of quantization and communication has received significant interest in the recent past. Early works [6, 7] typically consider scenarios with spatially distributed processors utilizing linear measurements with knowledge of the joint distribution of the measurement noise. The authors in [8] generalize distributed estimation to nonlinear observations with the similar assumption of partially known statistics. Early work on quantization [9–11] uses joint distributions of the measurement noise for efficient estimation in distributed systems while considering noiseless communication. The work in [12] explores sequential signal encoding with power and delay constraints on the distributed estimation framework. The authors in [13] achieve the optimal quantization for distributed estimation based on a training sample in unknown noise statistics. The spatial correlation among sensor measurements is accounted for in the design of quantizers for distributed estimation in [14]. A class of maximum likelihood (ML) estimators of a parameter are proposed in [15] which achieve the estimation performance

of the sample mean merely from observations which are quantized to a single bit. Similarly, in [16], ML estimates of a variance parameter from single bit quantizations utilize a sequentially updated adaptive quantization threshold.

A universal decentralized estimator is designed in [17] which utilizes Best Linear Unbiased Estimation (BLUE) without knowledge of the measurement noise statistics. In [18], a BLU estimator is used while considering of the effect of channel noise and measurement noise on the variance of the estimator. Here, an upper bound for the variance is derived which produces a power and rate efficient estimator. Using the result in [18], the authors in [19] find an upper bound on the variance of the BLUE which is used to design an efficient estimator. A rate-constrained distributed estimation scheme is designed in [20] which trades off the total rate used by the WSN with the number of active sensors. Similarly, [21] investigates the trade-off between number of active sensors and the energy used by each sensor. The same authors (in [22]) introduce function-based network lifetime and optimize it to produce a specified estimation accuracy at the fusion node. The authors in [20, 22] both assume distortion free communication. The joint optimal energy allocation and quantization level to minimize error in a binary symmetric channel with non-zero cross over probabilities is analyzed in [23]. Distributed BLU estimators that are utilized in previous work either consider only measurement noise variance or measurement and quantization noise variances. The only previous works that incorporate measurement, quantization, and channel noise variance for use with a BLU estimator are [24] and [18]. The work in [18] considers *scheduling* of sensor energy transmission and quantization levels for local estimation at the sensors from which other control actions are taken. Distributed estimation is not implemented. This estimator follows from a model [24] used to investigate the effect of channel fading on the accuracy of a sensor node. Prior efforts in distributed tracking are primarily concerned with sensor scheduling and selection algorithms. In [25], the lifetime of the network is optimized by determining how many sensors to keep active. A detailed energy function is used to construct an energy-usage based cost function which is optimized. The authors in [26]

formulate the sensor scheduling problem in terms of disjoint set covers of the observation space.

Distributed tracking using WSNs is done in [27] where quantization is accomplished by reducing the dimension of the state variable such that the transmit power budget will be met. Channel-aware distributed tracking is accomplished in [28] by performing particle filtering at the fusion node, however, only centralized tracking and transmission power levels are considered. A unique alternative to the above is found in [29] where sensor scheduling is formulated as approximate dynamic programming problem which chooses a leader node and a subset of observation nodes. The approach integrates the value of measurement information and the cost of transmitting the data over a finite time horizon. The distributed tracking method in [30] relaxes the discreteness of sensor positions to find approximate scheduling and sensor policies which use the ℓ_1-norm of the distance to the tracked object to approximate the multi-hop communication cost. This cost is used to threshold the change of the leader node across a finite-horizon. Decentralized consensus methods for distributed estimation and tracking are developed in [31–34]. These methods rely on a network of sensor nodes connected (in a graph-theoretic sense) by single-hop communication. The nodes update their local estimates with their neighbors and convergence of the estimates has been shown under the right conditions. Methods in [33, 34] specifically focus on the exchange of locally updated filter state estimates. However, while these methods consider power constraints and estimation performance, the *quantization and energy state of the system are not considered.* These works on consider many of the possible sources of estimate and tracking error and possible, as well as trying to take into account various system levels details like decentralization or fusion node selection. *However, while these efforts cite energy consumption as motivation for many of the intelligent methods, they usually leave off the explicit consideration of the energy state of the system. We attempt to incorporate this distinction throughout our work. Thus, distinct from prior work, our work uses the explicit consideration of the energy states of the system.* We consider both battery only and periodic

availability of energy from harvesting systems.

Utility Maximization Techniques

Significant work has be done to address the challenges of optimal resource allocation in distributed systems and provided analytical insights the behavior of these optimization problems. General convex optimization techniques, notable for rapid numerical solutions, are provably optimal and immense research has resulted in its wide applicability, especially in communications [35, 36], particularly systems modeled where utility dictates the value of expended resources. In this work and many others, (non-decreasing) concave utility functions are usually assumed. The impetus promoting this assumption is the conforming of many practical utilities to a kind of 'law of diminishing returns' modeled by concavity [37]. That is, with more resource allocated, more utility is realized; however, for increasing resource usage decreasing additional utility is returned. The assumption of strictly concave functions also allows the use of the dual formulation. In this case, the optimal primal and dual solutions are equivalent (zero duality gap) [38]. Constrained utility maximizations formulations have been pursued with significant depth in the allocation of bandwidth in networks aiming for congestion control and queue management [36, 39–43]. These approaches, while offering an acceptable framework and proven performance, do not provide particular insights into the structure of the underlying problems, namely in the pursuit of optimality with respect to the constraints in each iteration. We attempt to provide some insights into the structure of this class of problem.

Where the NUM approaches focus on iterative solutions for resource management in WSNs, there are significant works which obtain precise analytic solutions for specific goal-oriented optimization problems in WSNs. Distributed estimation and tracking applications represent another class of network problems (where the exchange of information (bandwidth) or energy is the resource of interest [18, 19, 37, 44]) which benefit from constrained utility maximization problems and solutions. Distributed estimation in energy sensitive networks, from both an error minimization and energy conservation perspective have been analyzed

in [18] and [19], respectively. However, the analytic solutions in these works do not scale well to multiple dimensions or deal with systems involving many varied constraints. Other approaches are sought in [44], which solve the distributed estimation problem (sharing communication resources, but with estimation information as the commodity) with an iterative primal-dual algorithm, which scales well across flow constraints but does not explicitly control the resource allocation based on predictions of future resource availability. Typical of some distributed approaches to utility maximization, the method proposed in [44] does not have an analogous intuitive analytical solution as do those in [18, 19].

Distributed decision making in multi-agent systems has gained significant attention in recent research [45–49], and has been applied to wireless networks without sensing capabilities in [50]. Distributed optimization problems have a wide range of applications, for example, in the competition amongst nodes/users for a shared network resource, where each node/user possesses a different individual objective [39, 40]. The goal of the users is to cooperatively assign themselves resources such that the total utility is maximized. The nodes in the system make local decisions, while trying to coordinate an overall solution with other nodes in the system. Other applications beside network resource allocation which have received attention include distributed cooperative control of unmanned vehicles [31, 51], motion planning of distributed vehicles providing roving surveillance [52], consensus in estimation across a network [34], and network control of distributed power delivery [53]. Common to all these applications is the need for the optimal control algorithms to be completely distributed and rely only on local information. Two approaches exist to decompose the underlying optimization problem: primal and dual decomposition. The work in [48] has provided a decentralized algorithm based on primal decomposition. The work in [49] has provided a solution based on dual decomposition along with the analysis of the effects of delayed communication on the resulting optimization solution. Additionally, the decentralized consensus methods considered in [31–34] are a specific subclass of primal decomposition optimization algorithms. In this case, the primal variable on which consensus is obtained is the public (and only)

variable with the objective being agreement between the variables. The authors in [54] use a smoothed version of the Lagrangian to accelerate the dual-decomposition optimization of a graph-based MAP inference problem for Markov Random Fields. Similarly, in [55], the use of dual-decomposition with alternating direction method of multipliers [56] is modified with a quadratic regularization to speed up convergence for solving relaxed binary problems on factor graphs.

Optimization and Energy Harvesting

The effectiveness of energy harvesting for improving WSNs resource allocation performance has been pursued in notable recent works. These have produce some useful analysis, techniques, and algorithms for managing resources with uncertain availability. The authors of [57] consider the maximum rate energy-neutral problem for a multiple epoch fading channel of a single transmitter and develop "directional water-filling" heuristics for the stochastic planning problem. Previous work on rate allocation for wireless sensor networks has been done in [58–61]. The approach of utility maximization for the wireless sensor network is taken for linear utility functions in [58, 60]. More general assumptions of concave differential utility functions are used in [59], and non-differentiable concave utility functions in [61]. Techniques for optimizing the flow of data in a network also applies to this research. But in EH-WSNs a resources of interest are energy or data. The Network Utility Maximization (NUM) formulation for Internet congestion control in [36, 41] is adapted in [61] for WSN applications. Work in [62, 63] address utility maximization for energy-constrained systems that execute periodic real time assignments for solar-powered systems in which a highly dynamic energy harvesting model is assumed. However, [62] assumes a limited set of epochs (only two). The harvested resources at a node executing a given task may frequently run out of energy at times. If the goal is predictable, continuous operation of a node, energy management techniques have to be adapted. In addition to classical energy conservation techniques, the WSN nodes must consider the stochastic availability of the environmental energy and maximize the utility of the task in a long-term perspective. The authors

in [3] call this energy-neutral operation: the utilization of resources is adjusted optimally during runtime and dictated by the available energy. Energy storage devices like batteries are used as "energy buffers" to smooth the variations of available energy. A difficult task for a single node, this challenge is magnified when attempting to optimize with respect to an entire network of energy harvesting nodes. For handling such distributed allocation problems there is substantial body of recent work on Network Utility Maximization (NUM) which uses standard dual decomposition-based distributed algorithms [36, 39–43]. Simplistic dual decomposition forms tend to be negatively affected by the inflexibility of the specific application or non-concave utilities. Alternative decomposition forms of the mathematical optimization problem can indicate new possibilities for network architecture. Therefore, alternative forms must be investigated to understand architectural possibilities [36, 56, 64], which we attempt as a part of our contribution.

1.3 Contribution

The contribution of this dissertation consists of four parts which more or less correspond to the categories delineated at the beginning of the review of prior work. As the next section will provide a more thorough understanding of the approach and details, we do not describe each contribution fully, but detail the important novelties.

Chapter 3 begins by providing a motivating model for the different aspects of the WSN and concludes with a justification for distortion as a metric for the cooperative estimation tasks and algorithms presented in Chapter 4. While many previous efforts consider power constraints and estimation performance, the *quantization and energy state of the system are not jointly considered*. Prior work considers many of the possible sources of estimation and tracking error, in addition to various system levels details like decentralization or fusion node selection. *However, while previous research cites energy consumption as motivation for many of the intelligent methods, the explicit consideration of the energy state of the system is not addressed. We attempt to incorporate this distinction in Chapter 4 and Chapter 5.*

Others have addressed the issue of energy management and harvesting, while leaving behind the purposed implementation of the WSN for cooperative tasks, (in our case, distributed tracking and estimation). In Chapter 4 we consider the joint optimal allocation of bandwidth and power with respect to remaining node resources, which is based on the following works.

[65] N. J. Roseveare and B. Natarajan, "Distributed Tracking with Energy Management in Wireless Sensor Networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3494-3511, 2012.

[66] N. J. Roseveare and B. Natarajan, "Optimizing Network Lifetime for Distributed Tracking with Wireless Sensor Networks," in *ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, pp. 41-48, 2011.[1]

[67] N. J. Roseveare and B. Natarajan, "Energy-Aware Distributed Tracking in Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference, WCNC 2011*, pp. 363-368, 2011.[2]

In Chapter 5 we attempt to shed light on analytical solutions to concave maximization problems for WSNs with complex sets of constraints. We develop a simplified communication model and employ energy harvesting as a part of the application. Other similar works do not analytically consider large number of highly varied constraints and objective functions. For instance, some solutions assume uniform utility functions across all agents/users [59, 61]. A similar method for harvested energy allocation in [63] involves extending the horizon as energy is utilized, whereas *our method considers the entire action space simultaneously*. Work in [60] simultaneously considers epochs and sensors, and while the authors utilize an allocation scheme based on predictions of the harvested energy, the objective is assumed to be linear and the method is not suited for the more appropriate class of concave

[1] 36% acceptance rate
[2] 48% acceptance rate

utility functions [61]. Formulations for utility maximization which do consider additional dimensions (and constraints) of interest [41, 44] are based on primal-dual iterative algorithms *which do not exploit the underlying structure of the constrained problems, as our approach attempts*. A Markov decision process framework for time-varying utility functions is considered in [37], whereas *our analysis considers the case of time-varying utility functions and provides an direct solution approximation achieved using model predictive control (MPC)* [68, 69]. This chapter is based on the following works.

[68] N. J. Roseveare and B. Natarajan, "A Structured Approach to Optimization of Energy Harvesting Wireless Sensor Networks," in *IEEE Consumer Communications and Networking Conference, CCNC 2012 - Wireless Communications Track*, pp. 420-425, 2013.[3]

[70] N. J. Roseveare and B. Natarajan, "An Alternative Perspective on Utility Maximization in Energy Harvesting Wireless Sensor Networks," under review for *IEEE Transactions on Vehicular Technology*, 2013.

Up to this point, we have discussed our answers to the questions concerning WSN tasks, optimization formulations, and algorithms. However, while these works consider distributed nodes, they require a centralized controller to inform the nodes of their allocations, *etc.* Chapter 6 seeks to address the decentralized optimization of WSN systems. This effort expands the initial definition of the decentralized optimization problem in [49] to include a more general constraint set, for which we produce an augmented dual decomposition method [56, 71]. A modification is presented for our decomposition method which provides a 'primed' initial solution. Other works [48, 49, 54, 55] modify the Lagrangian or utilize additional penalties, *while our approach solves the problem with a two-stage algorithm. Unique to this work is the first stage method which applies an optimality condition-based optimization solution.* The second stage is a decentralized version of the method of multipliers using

[3] 30% acceptance rate

subgradients. The final work in Chapter 6 and in this dissertation is *the analysis of the convergence bound for the aforementioned dual-decomposition-based decentralized optimization algorithm. We have not found any work which considers this convergence variability with respect to the energy harvesting process.* The presentation of this decentralized algorithmic work and probabilistic convergence analysis is from the following.

[72] N. J. Roseveare and B. Natarajan, "A Modified Distributed Dual Algorithm for Decentralized Optimization in Multi-Agent Systems," to be submitted *in IEEE Global Communications Conference*, 2013.

[73] N. J. Roseveare and B. Natarajan, "Probabilistic Analysis of Communication Strategies for Decentralized Signal Processing in Energy Harvesting Systems," to be submitted *IEEE Transactions on Signal Processing*, 2013.

1.4 Overview

Throughout our work there are three main goals continuously motivating the progression of thought.

- Models - useful but complex models which usually produce 'difficult' problems, and models where 'solution aware' simplifications have been made
- Analytical insights - from appropriately simplified versions of the model we attempt to uncover the behavior of the optimization in our resource-constrained problems
- Solutions - we investigate standard applied numerical approaches, as well as algorithmic solutions based on problem behavior discovered via analysis

This perspective applies to generic cooperative tasks for WSNs, but in this dissertation focuses on the estimation tasks within the WSN framework, and specifically on the optimal allocation of energy to accomplish said task.

In Chapter 2, we introduce the reader to the basic state space dynamical system model and show how optimization analysis and theory can be applied to estimation and control in such systems. Here we provide the necessary conceptual understanding and briefly review standard techniques which will aid in understanding this research.

Chapter 3 presents the models for the various aspects of the wireless sensor network (WSN) including data acquisition, communication, energy storage and harvesting, and sensing. Appropriate metrics for WSNs are also discussed. Chapter 4 details novel centralized optimization formulations for distributed estimation and tracking with consideration of the energy state of the sensor nodes. This first set of formulations rely only on fixed amount of stored energy (a battery or similar energy storage device). Relying on an estimation-distortion metric, this method dictates with what transmission power and rate levels the nodes in the network should communicate. The problem formulation results in a Mixed-Integer Nonlinear Program. The standard method of finding optimal solutions for this kind of problem relies on exhaustive search based approaches. We explore the reduction in problem complexity from NP-Hard to polynomial time by a relaxed integer convex formulation of the original non-convex problem. The effectiveness of the proposed algorithms are evaluated using a one-shot (single time instance) solution with and without energy state awareness (battery only). We also benchmark these algorithms against fixed strategies and offer some performance bounds as well as simulation results.

Chapter 5 begins with a simplification of the communication model which allows us to assume a concave utility function as the distortion metric of interest. We then analyze the solutions to a class of "water-filling" problems, of which the WSN estimate-distortion-minimizing problem is a specific example. Based on a few principles obtained from this analysis, we provide a matched marginal utility algorithm for general problems within our utility function calculus. We then apply this algorithm to the estimation problem by designing and analyzing a WSN-specific, energy-harvesting/aware, distortion minimization. Simulation results demonstrate the performance of the new algorithm.

In Chapter 6, we finally address the preference of multi-hop communication exhibited by low-power WSNs. We evaluate different decentralized optimization strategies which which are limited by the energy-constrained communication. Within this category, we explore the use of optimization decomposition techniques and propose a algorithmic modification. The first stage method of the two-stage algorithm applies a optimality condition-based optimization solution. The second stage is a decentralized version of the method of multipliers using subgradients. The 'priming' algorithm allows the initial decision variables with the 'most importance' in the final solution to receive initial allocations. This provides a better initial 'guess' in the case that communication is limited in the network of agents or if the network experiences communication failures. Finally, we present an analysis of the convergence bound for the aforementioned dual-decomposition-based decentralized optimization algorithm. Specifically, we investigate how the convergence bound varies probabilistically with respect to the random resources available through energy harvesting. The result is a telling summary of the trade-offs of energy-conservation and system performance in EH-WSNs, as well as demonstrating the effectiveness and limitations of the algorithms presented herein.

We conclude in Chapter 7 with a summary of the important details and findings of this research, as well as suggestions for possible future research.

Chapter 2

Optimization for Estimation and Control

Amongst the techniques available to engineers and mathematicians, optimization theory provides a vast and incredibly useful toolbox for analyzing and solving various modeling and decision problems. Optimization can be applied to various fields, including estimation and control. When the objectives and constraints are modeled aptly, most estimation and control solutions can be found either through analysis of the optimization formulation, or through numerical techniques afforded by modern computational systems, which boast guaranteed convergence. This convergence, however, is dependent on a specific assumption about the class of problem. Wireless sensor networks modeled using the details in this chapter allow the aforementioned framework of analyses and techniques to be applied. This application provides intuition and informative results on the utilization of wireless sensor networks.

This chapter provides an introduction to the basic state space model much of this work utilizes. General optimization and specifically convex optimization are then given some treatment. The topics of estimation and control, and how they fit into the optimization discussion are then covered. This overview instantiates our discussion, not because the unique contribution of this work is proliferated with its details, rather, because of how the the varied topics of our work fit into these models and methodologies.

2.1 Dynamical Systems and State Space Models

If we are to optimize the behavior (or state) of a system, we first need some model in which we can reasonably represent this behavior. Dynamical models allow a formal functional description of the evolution of the state of a system. A typical model of the dynamics of a physical system is given in terms a differential equation

$$\frac{d\mathbf{x}_t}{dt} = \dot{\mathbf{x}}_t = \mathbf{f}_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t, \mathbf{p}), \tag{2.1}$$

where $\mathbf{x}_t \in \mathbb{R}^d$ represents the state of the system, the t subscript denotes the time dependence, with

$$\mathbf{f}_t : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x},$$

and $\mathbf{u}_t \in \mathbb{R}^{n_u}$, $\mathbf{w}_t \in \mathbb{R}^{n_w}$, and $\mathbf{p} \in \mathbb{R}^{n_p}$ are the state propagation equation, controllable inputs, random disturbances, and parameters of the system, respectively, where all of these later quantities are usually vectors with n_x, n_u, n_w, and n_p number of elements, respectively. While in some systems the state, $\mathbf{x}_t$, is perfectly known, in others a measurement is made available which is a function of the state as

$$\mathbf{z}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t), \tag{2.2}$$

where $\mathbf{h}_t : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_n} \to \mathbb{R}^{n_x}$ is the observation function, $\mathbf{z}_t \in \mathbb{R}^{n_z}$ and $\mathbf{v}_t \in \mathbb{R}^{n_z}$ are the measurement and measurement noise, respectively. As an example of an indirectly measured state, take the speed of a moving vehicle. The speed can be obtained by knowing the radius of a wheel and measuring how many rotations it makes per unit time.

In this research, whether or not explicitly represented in state space form with a proper transition matrix, the states of the system consist of the energy currently stored in the battery, the channel state, and the physical state of the process which is tracked. See more in Chapters 3, 4, and 5.

The above functional descriptions are completely general, and in many practical cases extremely difficult to analyze. There are a plethora of properties such a system can have which enable a framework fitted for finding solutions in a reasonable manner.

"Essentially, all models are wrong, some are useful" $\sim$ Box

2.1.1 Assumptions for the Basic System Model

There are ways to execute analysis on the above general problems, and there are closed form solutions for many simple problems. However, most tractable way to extract information about a particular dynamical model is to make some careful simplifying assumptions. In many cases these assumptions allow the model to closely follow the actual system...models are idealizations of reality providing a usable mathematical description of how a process works.

The first and most essential assumption is that of **linearity**. This assumption is, for dynamical system analysis, on the same order of usefulness as Weierstrass grants us in the approximation of continuous functions by polynomials. While the assumption of linearity does not always hold exactly, the linearization of the non-linear system at the equilibrium points (around the roots of the autonomous state equation) provides insight into the behavior of the system. Additionally, linearization about the current state (usually by finding the Jacobian, $\frac{\partial f_t}{\partial \mathbf{x}_t}$, $\frac{\partial f_t}{\partial \mathbf{u}_t}$, $\frac{\partial h_t}{\partial \mathbf{x}_t}$, *etc.* [74]) can provide an approximate model for the propagation of the system. This process can even be performed as the system evolves so that

$$\begin{aligned} \dot{\mathbf{x}}_t &= \mathbf{A}_t\mathbf{x}_t + \mathbf{B}_t\mathbf{u}_t + \mathbf{G}_t\mathbf{w}_t \\ \mathbf{z}_t &= \mathbf{C}_t\mathbf{x}_t + \mathbf{D}_t\mathbf{u}_t + \mathbf{F}_t\mathbf{v}_t, \end{aligned} \tag{2.3}$$

where we have allowed the definition of the parameter vector to be implicit, and notationally $\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$ are all matrices. Sometimes the measurement matrix $\mathbf{C}$ is represented as $\mathbf{H}$. A further assumption which holds for some systems is that of **time-invariance**, i.e., the original state and observation equation do not depend on time as $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$. Together this type of system is called LTI (linear time-invariant). It could be that only some parts of the system description have time-dependent components. Lastly, it is useful for computation and for other reasons if the state can accurately modeled as **discrete** with respect to time. We switch to using k as the time index, where $k \in \mathbb{Z}$. The discretized linearized version of

the time-invariant state update and observation equations become

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_{k+1} + \mathbf{G}\mathbf{w}_{k+1} \\ \mathbf{z}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{F}\mathbf{v}_k. \end{aligned} \tag{2.4}$$

It is especially important when attempting to use an time-varying linearized representation of a non-linear model, but for any discretization in general, to be aware of the effect of sampling the dynamical system. Aliasing is caused by not observing the requirement of Shannon-Nyquist on the sampling frequency, f_s, with respect to the highest frequency component, f_h, i.e., $f_s \geq 2f_h$. However, aliasing effects are only the beginnings of trouble for a haphazard selection of the discretization. Mildly stable systems can have higher frequency components which, improperly sampled, cause instabilities [75, 76].

2.1.2 Modeling Stochasticity

In the above, we briefly mentioned the disturbance effects and measurement noise, $\mathbf{w}_k$ and $\mathbf{v}_k$. These vectors are random quantities which are modeled as having specific traits. But we must first define random variables.

A random variable (RV) is defined on a set of possible outcomes (the sample space Ω) and a probability distribution which associates the events in the sample space with a probability. Informally, a real valued random variable takes events from the outcome space and assigns a real number representing the fraction of "measurable mass" the event occupies in the sample space. There are several conditions on the definition of the sample space and probability space, e.g., existence of σ-algebra $\mathcal{F}$ on Ω, measurability of P the probability distribution, *etc.* [77]. Without these significant tangents we define the cumulative distribution (CDF) of the probability of an event, $\{\omega \in \Omega : X(\omega) \leq x\}$, for a continuous random variable as

$$F_X(x) := P(X \leq x)$$

with the probability density $f_X(x) = \frac{d}{dx}F_X(x)$ wherever the derivative is defined. The CDF for a set of discrete events has a stair-step-like behavior at the points where probability

mass is non-zero. We write that the RV X is drawn according to a particular distribution $\mathcal{D}$ as $X \sim \mathcal{D}$. We will assume for this discussion that these random variables are real-valued, although complex random variables will be utilized later, this will simplify notation.

Amongst the attributes of random variables, we primarily concern ourself with particular moments. Assuming the probability density is defined, the centralized moment (centered about the mean, $m_1 = \mu_x$) is

$$m_n := \int_B (x - \mu_X)^n f_X(x) dx, \tag{2.5}$$

where B is the range of the probability distribution [78]. The primary practical moments used to define randomness in most dynamical systems are $m_1 = \mu_X$ and $m_2 = \sigma_X^2$, the mean and variance. Finding the first moment is also referred to as the expectation of a RV X, $\mu_X = \mathbf{E}[X]$. The linearity of integration endows the same to the expectation operation. We can find estimates of these statistics empirically from data $\{x_1, \ldots, x_K\}$ using

$$\hat{\mu}_X = \frac{1}{K} \sum_{k=1}^{K} x_k \tag{2.6}$$

$$\hat{\sigma}_X^2 = \frac{1}{K} \sum_{k=1}^{K} (x_k - \hat{\mu}_X). \tag{2.7}$$

The distribution of a pair of interrelated random variables is represented as a joint probability

$$F_{XY}(x, y) = F_{X|Y}(x|y) F_Y(y) = F_{Y|X}(y|x) F_X(x),$$

where $F_{X|Y}(x|y)$ and $F_{Y|X}(y|x)$ are the conditional distributions and $F_X(x)$ and $F_Y(y)$ are the marginal distributions of the RVs. We say that a pair of random variables are independent if $F_{XY}(x, y) = F_X(x) F_Y(y)$. The covariance of a pair of random variables is defined as

$$\mathbf{E}[(X - \mu_X)(Y - \mu_Y)] = \mathrm{cov}(X, Y) = \mathbf{E}[XY] - \mu_X \mu_Y,$$

and the RVs are defined as uncorrelated if $\mathbf{E}[XY] = 0$.

We can extend these definitions to larger vectors of random variables. Without loss of generality, if we consider a sequence of random variables drawn at discrete time instances,

$\{X_i\}$, which are drawn according to a joint distribution $F_{\ldots,X_i,X_{i+1},\ldots}(\ldots,X_i,X_{i+1},\ldots)$. The distribution is said to have the quality of strong stationarity if

$$F_{\ldots,X_i,X_{i+1},\ldots}(\ldots,X_i,X_{i+1},\ldots) = F_{\ldots,X_{i+k},X_{i+k+1},\ldots}(\ldots,X_i,X_{i+1},\ldots) \;\forall\; k. \tag{2.8}$$

Strong stationarity is automatically achieved if the RVs are independent and identically distributed (i.i.d.). Similarly, weak stationarity is achieved if the first and second moments maintain time-invariance, i.e.,

$$\mu_{X_i} = \mu_{X_{i+k}} \text{ and } \sigma^2_{X_i} = \sigma^2_{X_{i+k}} \;\forall\; k.$$

Yet another useful demarcation of processes is second-order stationarity in which

$$F_{X_i,X_j}(X_i,X_{i+1}) = F_{X_{i+k},X_{j+k}}(X_{i+k},X_{j+k}) \;\forall\; k,$$

that is, the distributions are the same for any pair of random variables in the process. The cross-correlation of two processes $\{X_i\}$ and $\{Y_i\}$ (in the discrete time case) is

$$R_{XY}(i,k) = \mathbf{E}[X_i Y_{i+k}]. \tag{2.9}$$

The auto-correlation is defined similarly as $R_X(i,k) = \mathbf{E}[X_i X_{i+k}]$.

Lemma 1. *Second order stationarity for a process implies weak stationarity when the mean and correlation functions are integrable. That is, the auto-correlation depends only on the offset between the draws in the process,*

$$R_X(i,k) = \mathbf{E}[X_i X_{i+k}] = \mathbf{E}[X_0 X_k] = R_X(k). \tag{2.10}$$

A tool of analysis related to the correlation, is that of the power spectral density (PSD) [78], which describes what frequency components are extant in a random process. The PSD for a given sampling of a continuous (or discrete) random process, $x_t|_{t\in[0,T]}$ (or $\{x_1,\ldots,x_K\}$), is given by

$$\begin{aligned}\mathbf{E}[|\mathcal{F}_x(\omega)|^2] :=& \mathbf{E}\left[\frac{1}{T}\int_0^T x_t e^{j\omega t}\int_0^T x_\tau^* e^{-j\omega\tau}dtd\tau\right] && \text{(Continuous)}\\ =& \frac{1}{T}\int_0^T\int_0^T \mathbf{E}[x_t x_\tau *]e^{j\omega(t-\tau)}dtd\tau && (2.11)\end{aligned}$$

$$\mathbf{E}[|\mathcal{F}_x(\omega)|^2] := \mathbf{E}\left[\frac{1}{K}\sum_{k=1}^{K} x_k e^{j\omega k}\sum_{i=1}^{K} x_i^* e^{-j\omega i}\right] \quad \text{(Discrete)}$$

$$= \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{K}\mathbf{E}[x_k x_i^*]e^{j\omega(k-i)} \tag{2.12}$$

where $\mathcal{F}_x(\omega) = \frac{1}{T}\int_0^T x_t e^{-j\omega t}dt$ (discrete: $\mathcal{F}_x(\omega) = \frac{1}{K}\sum_{k=1}^{K} x_k e^{-j\omega k}$). We have included the complex conjugate, $\cdot^*$, in the definition above for completeness of the definition, even though we have stated that our discussion is for real valued RVs. The PSD, in the case of at least second-order stationarity, is merely the Fourier transform [77] of the auto-correlation.

The first and second central moments of the fixed dimension random vector, $\mathbf{X} = [X_1, \ldots, X_n]^T$, result in the definition of the vector mean, matrix covariance and correlation notations

$$\mathbf{E}[\mathbf{X}] = \mu_{\mathbf{X}} \tag{2.13}$$

$$\mathbf{E}[(\mathbf{X} - \mu_{\mathbf{X}})(\mathbf{X} - \mu_{\mathbf{X}})^T] = \mathbf{C}_{\mathbf{X}}, \tag{2.14}$$

allowing distributions on random vectors to have their distributions characterized by these vector and matrix quantities, e.g., for Gaussian, $\mathbf{X} \sim \mathcal{N}(\mu_{\mathbf{X}}, \mathbf{C}_{\mathbf{X}})$. A characterization of the cross correlation (and therefore covariance) for random vectors $\mathbf{X}$ and $\mathbf{Y}$ can be obtained when samples of the ergodic [78] second-order stationary random vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ and $\{\mathbf{y}_1, \ldots, \mathbf{y}_K\}$ as

$$\hat{\mathbf{R}}_{\mathbf{XY}}(k) = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k\mathbf{y}_k^T. \tag{2.15}$$

The sample mean and covariance can be defined in a similar way to find their estimated values from sample vectors of an ergodic process. These and the previous above estimates of the attributes of the random quantities are shown to have various valuable qualities such as unbiasedness and efficiency [78].

Common assumptions made in dealing with dynamical systems is that the random vectors representing the disturbance and measurement errors are distributed Gaussian with zero

mean and uncorrelated in time (and between processes), but with some spatial correlation. That is, $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_\mathbf{v})$ and $\mathbf{W}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_\mathbf{w})$ so that $E[\mathbf{v}_k \mathbf{v}_j^T] = \delta_{kj}\mathbf{C}_\mathbf{v}$, $E[\mathbf{w}_k \mathbf{w}_j^T] = \delta_{kj}\mathbf{C}_\mathbf{w}$, and $E[\mathbf{v}_k \mathbf{w}_j^T] = \mathbf{0}$, where δ_{kj} is the Kronecker delta function. This assumption is weak stationarity on the random vectors and is valid in many practical settings. Additionally, when non-zero mean or time-correlation exists, these aspects can usually be absorbed into the dynamical model for $\mathbf{f}_{\text{new}}(\cdot)$ and $\mathbf{h}_{\text{new}}(\cdot)$ which are driven by zero-mean uncorrelated random vectors $\mathbf{w}'_k$ and $\mathbf{v}'_k$.

2.2 Optimization

Optimization in the traditional sense can involve some extremely difficult problems, with some so complex, that given a solution we cannot even determine whether it is optimal. Convex optimization, a subset of these problems, can be thought of as a fusion of three disciplines: optimization [79], convex analysis [80], and numerical computation [81, 82]. Recently, it has become a tool of significance in engineering, permitting solutions of very large, practical engineering problems reliably and efficiently. In some sense, convex optimization is provides a natural extension of our ability to solve problems such as least squares and linear programming to a much larger and more complex class of problems.

The section deals with some of the concepts and principles of standard optimization techniques. The ordering of this section is purposed so that subsequent sections can be thought of in light of the topics reviewed here. Both of the subjects reviewed in the following rely on many of the analysis techniques in optimization and many of the problems in these fields of estimation and control can be solved using these techniques. In the subsequent chapters we rely heavily on techniques optimization theory and analysis. In particular, Chapters 4 and 5 apply standard and sequential convex optimization, while Chapter 6 utilizes the ideas of Lagrange dual decomposition.

The solutions we can attempt to find must be considered concurrently with the selection and simplification of a model. Optimization solutions are only as good as the problem posed.

Generally, these problems come in a few flavors

- General global solution: exhaustive, exact, expensive (most difficult for problems with integer decision variables, non-linear objective or constraints, *etc.*)
- Convex programs: guarantees local solutions are global solutions, two approaches,

  1) Test for convexity (concavity for maximization),

  2) Construct problem to be convex (from a calculus of convex functions [38]).

  Usually problems like these have no analytical solution. For example, a convex constraint

  $$f(x) = \max\{x, 1/x\} \leq 0$$

  can be optimized numerically, but is not continuously differentiable (see next item).

- Continuously differentiable convex/linear programs: we can use Lagrange analysis to explore the problem for simple cases, and use gained intuition to develop optimal algorithmic solutions (e.g., Least-squares), but method can be difficult to apply to complex systems of constraints.

2.2.1 Basic Problem and Challenges

The problem formulation is to find the extremal value of some function while satisfying some set of constraints

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & h_i(x) = 0, i = 1, \ldots, I \\ & f_j(x) \leq 0, j = 1, \ldots, J \end{array} \tag{2.16}$$

where $x \in \mathcal{D} \subset \mathbb{R}^n$ is the vector of decision variables with $\mathcal{D}$ the common domain of the functions, f_0, h_j and f_i, which are the objective function, equality constraint functions, and inequality constraint functions. These problems are generally difficult to solve, particularly for a large number of decision variables. These difficulties stem primarily from 1) the presence of local optima for various choices of x or from 2) the difficulty of finding a feasible

point (one which satisfies all of the equality and inequality constraints) [35]. The feasible space, $\mathcal{D}$, may not be fully connected, or could even be empty. The list of difficulties goes on with 3) the lack of concrete stopping criteria, 4) poor convergence of algorithms, and 5) numerical stability issues. These issues can be mediated by the assumption of convexity.

2.2.2 Implications of Convexity

The designation of convex obviously applies to convex sets, regions, *etc.*, the term convex is also used in reference to a function. Namely, we say that for $\mathcal{D}$ a convex domain, a function $f : \mathcal{D} \to \mathbb{R}$ is convex if and only if $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$, for $\alpha \in [0,1],\ x, y \in \mathcal{D}$. The same holds for a concave function with the opposite inequality.

The first three issues are immediately addressed if it is assumed that f_i are all convex and h_j is affine: any local optimum is a global optimum. The feasibility of optimization problems can be determined straightforwardly, in addition to the fact that the formulation of the dual allows for very precise stopping conditions [79, 80]. The issues related to numerical instability and convergence can be mediated when, in addition to convexity, the f_i's satisfy a property known as self-concordance. Satisfying these two properties, problems can be solved using interior-point methods [83, 84]. Self-concordance is a property held by many important functions in science and engineering, thus, it is possible to solve problems utilizing such functions effectively and efficiently.

Apart from the excellent numerical techniques provided in convex optimization methods, we also need to understand optimization analysis techniques. These analysis techniques allow us to partition and rewrite the problem in ways that provide appropriate solutions for estimation and control problems in wireless sensor networks.

Optimality Conditions

Given an convex program, we can easily determine the conditions which guarantee an optimal solution. Differential calculus provides us with the ability to find and tell us about the extremal points of an objective function f_0 [38].

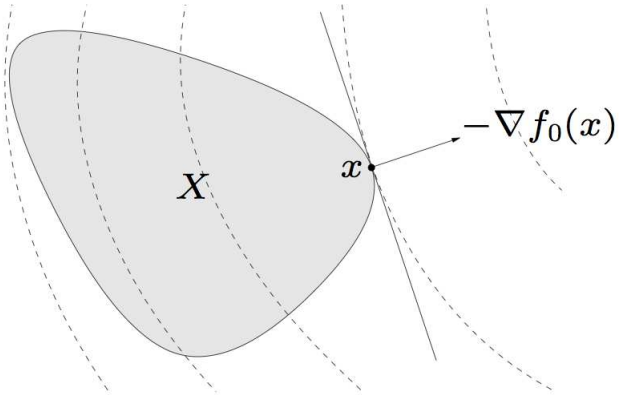


Figure 2.1: *Example of supporting hyperplane at x defined by $\nabla f_0(x)$ on the edge of the constraint space. $\mathbf{X} = relint(\mathcal{D})$ is the relative interior of the feasible domain $\mathcal{D}$. Dashed lines indicate the level curves of the objective.*

Theorem 1. *First Order Optimality Condition*

If the gradient of the differentiable objective function, $\nabla f_0(x) = \left(\frac{\partial f_0}{\partial x_1}, \ldots, \frac{\partial f_0}{\partial x_1}\right)^T$, at the decision variable point x, then x is optimal if and only it it is feasible and

$$\nabla f_0(x)^T(y - x) \geq 0, \tag{2.17}$$

for all feasible y, (i.e., $y \in \mathcal{D}$).

Intuitively, if $\nabla f_0(x) = \mathbf{0}$ then we have achieved an extremal point, and given convexity, the minimum. If nonzero, $\nabla f_0(x)$ defines a supporting hyperplane to feasible set $\mathcal{D}$ at x, i.e., x is on the boundary and all feasible points on the other side of the hyperplane less optimal. This is illustrated in Figure 2.1.

Similarly, the convexity of the program would imply that further specifications on the optimality can be made with respect to the curvature of the objective function.

Theorem 2. *Second Order Optimality Condition*

If given the gradient and Hessian of the twice differentiable objective function, $\nabla f_0(x)$ and $\nabla^2 f_0(x)_{ij} = \frac{\partial \partial f_0}{\partial x_i \partial x_j}$, at the decision variable point x, then x is optimal if and only it it is feasible and

$$\nabla f_0(x) = \mathbf{0} \text{ and } \nabla^2 f_0(x) \succeq 0. \tag{2.18}$$

We understand this result to mean, intuitively, that the minimum is "at the bottom" of the convex objective, where the shape of the function is "bowl-like" and upward.

The purpose of clarifying the basic conditions for optimality is that we intend to use a similar set of conditions in our analysis later in development a convex programming algorithm.

Another useful property (already mentioned) is that local solutions to convex programs are also global solutions, we show this with the following.

We say that a decision x is locally optimal if there is an $R > 0$ such that x is optimal for

$$\begin{array}{ll} \text{minimize} & f_0(z) \\ \text{subject to} & h_i(z) \leq 0, i = 1, \ldots, I \\ & f_j(z) \leq 0, j = 1, \ldots, J \\ & ||z - x||_2 \leq R \end{array} \tag{2.19}$$

An example of this would be $f_0(x) = 1/x$ with $\mathcal{D} = \mathbb{R}_{++}$ and optimal value $f_0^* = 0$ ($\cdot^*$ here denotes optimal). Within the feasible domain only locally optimal values can be found within distance R.

Theorem 3. *Any locally optimal point of a convex program is globally optimal.*

Proof. Suppose x is locally optimal and y is optimal with $f_0(y) < f_0(x)$. The local optimality of x means there is an $R > 0$ such that

$$z \text{ feasible}, \quad ||z - x||_2 \leq R \quad \Rightarrow \quad f_0(z) \geq f_0(x)$$

Consider $z = \theta y + (1-\theta)x$ with $\theta = R/(2||y-x||_2)$, thus $||y-x||_2 > R$, so $\theta \in (0, 1/2)$. Since z is a convex combination of two feasible points, it is also feasible. We have $||z-x||_2 = R/2$ and

$$f_0(z) \leq \theta f_0(x) + (1-\theta) f_0(y) \leq f_0(x)$$

which produces a contradiction. □

2.2.3 Lagrange Duality

As mentioned, the duality afforded by convex optimization problems allows the solutions of the problems to be guaranteed with a certificate of optimality (or a bound on the accuracy of

the solution). We can define the Lagrangian for any optimization problem (not necessarily convex) as

The Lagrangian $\mathcal{L} : \mathcal{D} \times \mathbb{R}^J \times \mathbb{R}^I \to \mathbb{R}$ is defined as

$$\mathcal{L}(x, \mu, \lambda) := f_0(x) + \sum_{j=1}^{J} \mu_j f_j(x) + \sum_{i=1}^{I} \lambda_i h_i(x), \tag{2.20}$$

The dual variables $\lambda_i,\ i = 1, \ldots, I$ and $\mu_j,\ j = 1, \ldots, J$, are associated with each of the equality and inequality constraints, respectively. Thus, the Lagrangian provides us with an unconstrained version of the original problem.

The corresponding Lagrange dual function $q : \mathbb{R}^J \times \mathbb{R}^I \to \mathbb{R}$ is the infimum with respect to the primal variable x,

$$\begin{aligned} q(\mu, \lambda) := & \inf_{x \in \mathcal{D}} \mathcal{L}(x, \mu, \lambda) \\ & \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{j=1}^{J} \mu_j f_j(x) + \sum_{i=1}^{I} \lambda_i h_i(x) \right). \end{aligned} \tag{2.21}$$

The function q is concave, and can be $-\infty$ for some $\mu,\ \lambda$.

Lemma 2. *If $\mu \succeq 0$, then $g(\mu, \lambda) \leq f_0^*$.*

Proof. If $\bar{x}$ is feasible and $\mu \succeq 0$, then

$$f_0(\bar{x}) \geq \mathcal{L}(x, \mu, \lambda) \geq \inf_{x \in \mathcal{D}} \mathcal{L}(x, \mu, \lambda) = q(\mu, \lambda). \tag{2.22}$$

Minimizing over all feasible $\bar{x}$ produces $f^* \geq q(\mu, \lambda)$. □

The Lagrange dual problem is the maximization of the dual function with the Lagrange multipliers as primary variables,

$$\begin{array}{ll} \text{minimize} & q(\mu, \lambda) \\ \text{subject to} & \lambda \succeq 0. \end{array} \tag{2.23}$$

The solution to this problem finds the best lower bound on f^*. The dual problem is convex, its optimal value is denoted q^* with μ, λ dual feasible if $\mu \succeq 0$ and $(\mu, \lambda) \in \mathcal{D}(q)$.

As useful as having a lower bound on the optimal value of the primal problem, the dual function and problem can sometimes provide even more information. We say that weak duality holds if $f^* \geq q^*$, and $|f^* - q^*|$ is called the duality gap. This relationship holds for all convex and non-convex problems. Weak duality is useful for finding non-trivial (not $-\infty$) lower bounds on the optimal of difficult primal problems. Strong duality is achieved when $f^* = q^*$, which does not hold in general. Strong duality does, however, hold for the majority of convex problems. This is incredibly useful relationship that allows use to solve either the primal or dual problem. We also have conditions which can be checked to ensure strong duality in convex problems, these are called constraint qualifications. An example follows.

Lemma 3. *Slater's Constraint Qualification*

Strong duality holds for a convex program of the form in (2.16) if it is strictly feasible, i.e.,

$$\begin{aligned} \exists\ x\ \in \text{relint}(\mathcal{D}): \quad & f_j(x) < 0,\ \forall\ j \in \{1, \ldots, J\} \backslash \mathcal{A} \\ & f_j(x) \leq 0,\ \forall\ j \in \mathcal{A}, \\ & h_i = 0\ \forall\ i, \end{aligned}$$

where relint$(\cdot)$ *is the relative interior of a domain and* $\mathcal{A}$ *is the set indices representing the linear inequality constraints.*

As there are several conditions for identifying strong duality [38], there are also conditions extracted from the dual form by which we can verify optimality.

Theorem 4. *Karush-Kuhn-Tucker Optimality Conditions*

Given f_0, f_j, h_i *differentiable for all* i, j*, let* x^* *and* (μ^*, λ^*) *are any pair of primal and dual optimal points with zero duality gap. Since* x^* *minimizes* $\mathcal{L}(x, \mu^*, \lambda^*)$ *over* x*, it follows that the gradient of the Lagrangian must vanish at* x^**,*

$$\nabla f_0(x^*) + \sum_{j=1}^{J} \mu_j^* \nabla f_j(x^*) + \sum_{i=1}^{I} \lambda_i^* \nabla h_i(x^*) = 0. \tag{2.24}$$

Thus we have

$$f_j(x^*) \leq 0, \ \forall \ j \tag{2.25}$$

$$h_i(x^*) = 0, \ \forall \ i \tag{2.26}$$

$$\mu_j^* \geq 0, \ \forall \ j \tag{2.27}$$

$$\mu_j^* f_j(x^*) = 0, \ \forall \ j \tag{2.28}$$

$$f_0(x^*) + \sum_{j=1}^{J} \mu_j^* \nabla f_j(x^*) + \sum_{i=1}^{I} \lambda_i^* \nabla h_i(x^*) = 0. \tag{2.29}$$

Summarized, we need primary constraint feasibility, dual multipliers feasibility ($\mu > 0$), complementary slackness (Eq. (2.28), as well as the gradient of the Lagrangian vanishing with respect to x.

This condition is necessary for any problem, it is additionally sufficient for any differentiable convex problem. We illustrate this optimality condition as a far reaching optimality condition, of which our later developments using optimality conditions are a sub class.

2.2.4 Standard Methods

It will help our future intuition in examining and analyzing problems to have some notion of the types of problems we can solve using convex optimization. There are some simple constrained problems which have analytical solutions, but which involve the solution to matrix problems such as $Ax = b$ for ungainly problem sizes. Other problems tell us what the problem looks like, but are in general unrealistic, as most problems of interest involves many more constraints than considered here.

Example Consider the problem of finding the minimum norm solution to a set of linear equations, where the problem can be formulated as

$$\begin{array}{ll} \text{minimize} & ||x||_2^2 = x^T x \\ \text{subject to} & Ax = b \end{array} \tag{2.30}$$

The Lagrangian is $\mathcal{L}(x, \lambda) = x^T x + \lambda(Ax - b)$. Minimizing $\mathcal{L}$ over x by setting the gradient

to zero,

$$\nabla_x \mathcal{L}(x, \lambda) = 2x + A^T\lambda = 0 \;\Rightarrow\; x = -(1/2)A^T\lambda,$$

and substitute this value into $\mathcal{L}$ to obtain q

$$q(\lambda) = \mathcal{L}((-1/2)A^T\lambda, \lambda) = -\frac{1}{4}\lambda^T AA^T\lambda - b^T\lambda,$$

which is a concave function of λ. The lower bound is $f^* \geq -\frac{1}{4}\lambda^T AA^T\lambda - b^T\lambda$.

Example: Least squares. We want to minimize the "distance" of the fit of x through the range of a matrix relative to a desired vector. Formulating the problem as

$$\text{minimize} \quad ||Ax - b||_2^2. \tag{2.31}$$

This problem has an analytical solution which is just the pseudo-inverse of A applied to b, $x^* = A^\dagger b$. A solution is easily obtained through numerical techniques efficiently nonetheless. We mention the problem so that it may be clear that optimization problems have analytical as well as numerical solutions. We pursue some of both in the forthcoming chapters.

When starting with a specific application, it is often difficult to find a form of the problem which will be solvable by convex methods. There are number of reformulations which aid in determining a problem convex, or finding a form in which it is more suitably analyzed. We now given some examples of equivalent problems which produce the same solution. Transformations between problems and example instances include the following [38].

- Eliminate equality constraints: [min. $f_0(x)$ s.t. $Ax = b$] $\Leftrightarrow$ [min. $f_0(Fz + x_0)$], where F and x_0 allow $Ax = b \;\Leftrightarrow\; x = Fz + x_0$ for some z.

- Introduce equality constraints, the reverse transformation of the previous bullet.

- Introduce slack variables so that only equality constraints are present with respect to x:

$$[\text{min. } f_0(x) \text{ s.t. } Ax \leq b] \;\Leftrightarrow\; [\text{min. } f_0(x) \text{ s.t. } Ax + s = b,\; s \succeq 0]$$

- Use the epigraph form of the problem:

$$[\text{min. } f_0(x) \text{ s.t. } f_i(x) \leq 0] \;\Leftrightarrow\; [\text{min. } t \text{ s.t.} f_0(x) - t \leq 0,\; f_i(x) \leq 0]\,.$$

- Minimize over a subset of the variables:

$$[\text{min. } f_0(x_1, x_2) \text{ s.t. } f_i(x_1) \leq 0] \;\Leftrightarrow\; \left[\text{min. } \bar{f}_0(x_1) \text{ s.t. } f_i(x_1) \leq 0\right],$$

  where $\bar{f}_0(x_1) = \inf_{x_2} \; f_0(x_1, x_2)$.

There are a plethora of other functions which are sums, products, and compositions of other convex functions, which are convex under the correct conditions on the composing functions [38].

Example: Quasi-Convex Problem. In the case that the objective, $f_0(x)$ is a quasi-convex problem, there exists function $\phi_t(x)$ such that

$$f_0(x) \leq t \;\Leftrightarrow\; \phi_t(x) \leq 0.$$

If we have $f_0(x) = p(x)/q(x)$ with p convex and q concave with $p(x) \geq 0$ and $q(x) > 0$ on the domain of f_0, then we can take

$$\phi_t(x) = p(x) - tq(x).$$

For $t \geq 0$, $\phi_t(x)$ is convex in x and $p(x)/q(x) \leq t$ if and only if $\phi_t(x) \leq 0$. The problems can be solved using bisection, where at each iteration a convex feasibility problem is solved which updates either an upper or lower bound at each iteration. As the upper a lower bounds of the feasibility testing converges a solution to the quasi-convex problem can be obtained.

2.3 Estimation

Our discussion in the next section concerns the estimation underlying states from noisy samples. There are a variety of techniques and metrics which are used to determine what

qualifies as a "good" estimate. Typically, this involves (at least asymptotic) efficiency and unbiasedness of an estimator [85]. We will see that estimation is a readily applicable domain for optimization, which can aid us in the analysis and development of optimal estimators. Estimation is introduced in Chapter 3 as an example of a cooperative task in a WSN. Further application of estimation theory to optimization objectives is also presented in Chapter 4.

The problem is to find a function $\phi : \mathbb{R}^m \to \mathbb{R}^n$, given random vector $\mathbf{y} \in \mathbb{R}^m$, such that $\hat{\mathbf{x}} = \phi(\mathbf{y})$ is near $\mathbf{x} \in \mathbb{R}^n$, the vector being estimated. This nearness can, be as mentioned, be quantified in different ways. So there is no single optimality metric, but based on which metric of nearness is employed, different estimators result. The standard conditions for a "good" estimator are the following.

- Unbiasedness requires: $\mathbf{E}[\phi(\mathbf{y})] = \mathbf{E}[\mathbf{x}]$.
- The efficiency of an estimator is

$$e(\phi(\mathbf{y})) := \frac{\mathcal{I}^{-1}(\mathbf{y})}{\text{var}(\phi)} \leq 1, \tag{2.32}$$

  where $\mathcal{I}(\mathbf{y})$ is the Fisher information of the sample [85]. Thus, when the variance of the estimator achieves the inverse Fisher information then it is minimum variance, and therefore efficient.

An estimator achieving efficient is required to be a minimum variance unbiased estimator (MVUE). However, in general the MVUE is not necessarily efficient, if it even exists.

2.3.1 Optimal Estimates

One of the common measures of "nearness" is the mean squared error,

$$MSE_{\mathbf{XY}}(\phi) = \mathbf{E}[||\phi(\mathbf{y}) - \mathbf{x}||^2]. \tag{2.33}$$

An optimization can be posed, $\min_{\phi(\cdot)} \; MSE_{XY}(\phi)$, which produces the minimum mean square error estimator and results in the general solution

$$\phi_{\text{MMSE}}(\mathbf{y}) = \mathbf{E}[\mathbf{x}|\mathbf{y}] = \arg\min_{\phi(\cdot)} \mathbf{E}[||\phi(\mathbf{y}) - \mathbf{x}||^2], \tag{2.34}$$

which is just the conditional expectation of x given y. Thus, this estimator requires a conditional, and therefore a joint, density on x and y, $p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) = p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y})p_X(\mathbf{x})$. We can also imagine the simultaneous estimation of multiple quantities from multiple other known quantities, for example, using the conditional density $p_{\mathbf{X}_1,\ldots,\mathbf{X}_k|\mathbf{Y}_1,\ldots,\mathbf{Y}_j}(\mathbf{x}_1,\ldots,\mathbf{x}_k|\mathbf{y}_1,\ldots,\mathbf{y}_j)$ to estimate $\mathbf{x}_1,\ldots,\mathbf{x}_k$ from $\mathbf{y}_1,\ldots,\mathbf{y}_j$.

Example Suppose that $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ are jointly Gaussian with

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{\mathbf{X}} \\ \mu_{\mathbf{Y}} \end{bmatrix}, \begin{bmatrix} \mathbf{C}_{\mathbf{X}} & \mathbf{C}_{\mathbf{XY}} \\ \mathbf{C}_{\mathbf{YX}} & \mathbf{C}_{\mathbf{Y}} \end{bmatrix} \right)$$

Using some algebra and the Schur complement we write the conditional density as

$$p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x} - (\mu_{\mathbf{X}} + \mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{Y}}^{-1}(\mathbf{y} - \mu_{\mathbf{Y}})))^T \mathbf{C}_{\mathbf{X}|\mathbf{Y}}^{-1}(\mathbf{x} - (\mu_{\mathbf{X}} + \mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{Y}}^{-1}(\mathbf{y} - \mu_{\mathbf{Y}})))\right)}{\sqrt{\det(\mathbf{C}_{\mathbf{X}|\mathbf{Y}})(2\pi)^n}}$$

where $\mathbf{C}_{\mathbf{X}|\mathbf{Y}} = \mathbf{C}_{\mathbf{X}} - \mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{Y}}^{-1}\mathbf{C}_{\mathbf{YX}}$ is the covariance of $\mathbf{x}$ conditioned on $\mathbf{y}$. Thus, the conditional expectation

$$\hat{\mathbf{x}} = \phi_{\text{MMSE}}(\mathbf{y}) = \mathbf{E}[\mathbf{x}|\mathbf{y}] = \mu_{\mathbf{X}} + \mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{Y}}^{-1}(\mathbf{y} - \mu_{\mathbf{Y}}), \tag{2.35}$$

results in the minimum mean-squared error estimator for the jointly Gaussian random vectors. The estimation error for the MMSE estimator is also a Gaussian random vector

$$\hat{\mathbf{x}} - \mathbf{x} \sim \mathcal{N}(0, \mathbf{C}_{\mathbf{X}} - \mathbf{C}_{\mathbf{XY}}\mathbf{C}_{\mathbf{Y}}^{-1}\mathbf{C}_{\mathbf{YX}}),$$

with covariance $\mathbf{C}_{\mathbf{X}|\mathbf{Y}} \preceq \mathbf{C}_{\mathbf{X}}$, i.e., estimation error is less than the prior covariance of $\mathbf{x}$.

If we consider the case of finding a optimal affine estimator of $\mathbf{x}$ given $\mathbf{y}$, where only first and second order moments are known. That is, we assume that $\mathbf{y}$ is linearly related to $\mathbf{x}$ as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v},$$

where $\mathbf{H} \in \mathbb{R}^{n \times m}$ is a LTI observation matrix with no assumption of a jointly Gaussian relationship, then the optimization problem is

$$\text{minimize}_{\mathbf{W},\mathbf{b}}\ \mathbf{E}[||(\mathbf{W}\mathbf{y} + \mathbf{b}) - \mathbf{x}||^2] \tag{2.36}$$

$$\text{subject to } \mathbf{E}[\mathbf{W}\mathbf{y} + \mathbf{b}] = \mu_{\mathbf{X}}, \tag{2.37}$$

where $\mathbf{W}, \mathbf{b}$ are a matrix weight and vector offset of the appropriate size to produce a vector of size $\dim(\mathbf{x}) = n$. The constraint (which is for unbiasedness) can be rewritten as $\mathbf{W}\mu_{\mathbf{Y}}+\mathbf{b} = \mathbf{WH}\mu_{\mathbf{X}}+\mathbf{W}\mu_{\mathbf{V}}+\mathbf{b} = \mu_{\mathbf{X}}$, which requires $\mathbf{b} = (\mathbf{I}-\mathbf{WH})\mu_{\mathbf{X}}-\mathbf{W}\mu_{\mathbf{V}} = \mu_{\mathbf{X}}-\mathbf{W}\mu_{\mathbf{Y}}$ (with $\mathbf{I}$ the identity matrix). Then the optimization involves setting the derivative of the $MSE(\mathbf{W})$ (which equals $\mathbf{W}[\mathbf{HC_X H}^T + \mathbf{C_V}]\mathbf{W}^T - 2\mathbf{C_X H}^T\mathbf{W}^T + \mathbf{C_X}$) with respect to $\mathbf{W}$ to zero. An initial solution reveals itself to be

$$\mathbf{W} = \mathbf{C_X H}^T[\mathbf{HC_X H}^T + \mathbf{C_V}]^{-1} = \mathbf{C_{XY}C_Y^{-1}}.$$

This produces the best linear unbiased (BLU) estimator,

$$\begin{aligned}
\hat{\mathbf{x}} &= \phi_{\mathrm{BLU}}(\mathbf{y}) = \mathbf{Wy} + \mathbf{b} \\
&= \mathbf{C_{XY}C_Y^{-1}y} + (\mu_{\mathbf{X}} - \mathbf{C_{XY}C_Y^{-1}}\mu_{\mathbf{Y}}) \\
&= \mu_{\mathbf{X}} + \mathbf{C_{XY}C_Y^{-1}}(\mathbf{y} - \mathbf{H}\mu_{\mathbf{X}} - \mu_{\mathbf{V}}) \\
&= \mu_{\mathbf{X}} + \mathbf{C_X H}^T[\mathbf{HC_X H}^T + \mathbf{C_V}]^{-1}(\mathbf{y} - \mathbf{H}\mu_{\mathbf{X}} - \mu_{\mathbf{V}}). \qquad (2.38)
\end{aligned}$$

Remark 1. *If we instead of minimizing mean squared error, we minimize the trace of the covariance of the estimate. This is an equivalent minimization, since for scalar* $\mathbf{tr}\{\mathbf{E}[||\mathbf{e}||^2]\} = \mathbf{E}[\mathbf{tr}\{\mathbf{e}^T\mathbf{e}\}] = \mathbf{E}[\mathbf{tr}\{\mathbf{ee}^T\}] = \mathbf{tr}\{\mathbf{E}[\mathbf{ee}^T]\}$*, from the linearity of the trace and expectation, with* $\mathbf{e}$ *the error vector. Thus, minimizing*

$$\begin{aligned}
\mathbf{tr}\{\mathbf{E}[(\mathbf{Wy} + \mathbf{b} - \mu_{\mathbf{X}})(\mathbf{Wy} + \mathbf{b} - \mu_{\mathbf{X}})^T]\} &= \mathbf{tr}\{\mathrm{cov}(\phi_{\mathrm{BLU}}(\mathbf{y}))\} \\
&= \mathbf{tr}\{\mathbf{W}\mathrm{cov}(\mathbf{HC_X H}^T + \mathbf{C_V})\mathbf{W}^T\} \\
&= \mathbf{tr}\{[\mathbf{H}^T(\mathbf{HC_X H}^T + \mathbf{C_V})^{-1}\mathbf{H}]^{-1}\},
\end{aligned}$$

subject to the unbiasedness constraint $\mathbf{WH} = \mathbf{I}$*, results in the weighting term becomes*

$$\mathbf{W} = \left[\mathbf{H}^T(\mathbf{HC_X H}^T + \mathbf{C_V})^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^T(\mathbf{HC_X H}^T + \mathbf{C_V})^T$$

and the affine offset simplifies to

$$\mathbf{b} = -\mathbf{W}\mu_{\mathbf{V}}.$$

The BLU estimator is unbiased and has the minimum mean squared error among all affine estimators (hence, minimum variance among affine estimators). The BLU estimator is useful for estimating parameters, for instance when the vector $\mathbf{x}$ is deterministic and $\mathbf{y}$ is a set of noisy affine measurements of $\mathbf{x}$. In the case of the measurements of a deterministic value, $\mathbf{C_X} = \mathbf{0}$ and the above equations simplify greatly.

2.3.2 The Kalman Filter

In the case that we want to obtain sequential estimates of the state $\{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_k\}$, given some corresponding sequence of measurements, $\{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$, according to the model

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_{k+1} + \mathbf{G}\mathbf{w}_{k+1} \\ \mathbf{z}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k + \mathbf{F}\mathbf{v}_k. \end{aligned} \tag{2.39}$$

where $\hat{\cdot}$ represents the estimate of the state. We have the general joint distribution determining the relationship between the random quantities

$$p_{\mathbf{X}_k|\mathbf{X}_{k-1},\ldots,\mathbf{X}_0,\mathbf{Z}_k,\ldots,\mathbf{Z}_1}(\mathbf{x}_k|\mathbf{x}_{k-1},\ldots,\mathbf{x}_0,\mathbf{z}_k,\ldots,\mathbf{z}_1),$$

from which to extract an estimate. A simplification can be made, however, as is evident from the model above. This is that of the above sequences of random vectors being Markov. That is, knowledge of the previous vector renders the current vector independent of the past,

$$p_{\mathbf{X}_k|\mathbf{X}_{k-1},\ldots,\mathbf{X}_0}(\mathbf{x}_k|\mathbf{x}_{k-1},\ldots,\mathbf{x}_0) = p_{\mathbf{X}_k|\mathbf{X}_{k-1}}(\mathbf{x}_k|\mathbf{x}_{k-1}),$$

similarly for the measurement model,

$$p_{\mathbf{Z}_k|\mathbf{X}_k,\ldots,\mathbf{X}_0,\mathbf{Z}_{k-1},\ldots,\mathbf{Z}_1}(\mathbf{z}_k|\mathbf{x}_k,\ldots,\mathbf{x}_0,\mathbf{z}_{k-1},\ldots,\mathbf{z}_1) = p_{\mathbf{Z}_k|\mathbf{X}_k}(\mathbf{z}_k|\mathbf{x}_k).$$

While the Markov assumption obviously holds from the above discrete linear time-invariant dynamical model, this assumption is valid for less simplified system or is at least a good approximation in many cases [86]. If we additionally assume that the disturbance and measurement noise are distributed Gaussian, then the above is an example of a Gauss-Markov

system where the states and measurements are pair-wise jointly Gaussian distributed. The Kalman filter is an excellent tool for estimating the state of a dynamical system, particularly when the system has a well-behaved linearization for the selected sample rate and the disturbances can be modeled as Gaussian noise.

The idea behind the Kalman filter is to iteratively perform a one-step update which augments the estimate with a new measurement when one is available. Of particular importance to Kalman filtering is the optimal gain with which the new measurement is weighted before being used to update the system. The first step in the algorithm is to predict the state of the system in the current time instant. This is done for the above linear system as,

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_{k-1} \tag{2.40}$$

$$\mathbf{P}_{k-1|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{G}\mathbf{C}_\mathbf{W}\mathbf{G}^T \tag{2.41}$$

where $\mathbf{A}$ is the state propagation, $\mathbf{B}$ and $\mathbf{G}$ are the control-input and disturbance-input models, and $\mathbf{C}_\mathbf{W}$ is the stationary covariance of the disturbance $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_\mathbf{W})$. The covariance of the estimate, $\mathbf{P}_{k|k}$ is the primary metric by which the accuracy of a filtering process can be monitored. As the Kalman filter is based on the assumption of a Gauss-Markov process (or at least approximated Gauss-Markov), it should be noted that technique keeps a record of only the first two moments of the stochastic system, since a Gaussian distributed is completely characterized by a mean and covariance.

An optimization problem can be formulated to minimize the MSE as

$$\text{minimize}_\mathbf{K}\mathbf{E}[||\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) - \mathbf{x}_k||^2], \tag{2.42}$$

where $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state of the system at the current time instant based on the previous state and $\mathbf{K}$ is called the Kalman gain. The optimal Kalman gain can intuitively be understood as containing information concerning whether the measurement or dynamic model is more accurate, and weights the measurement/prediction accordingly. The update

step requires the following,

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k-1|k-1}\mathbf{H}^T + \mathbf{F}\mathbf{C}_\mathbf{V}\mathbf{F}^T \tag{2.43}$$

$$\mathbf{K}_k = \mathbf{P}_{k-1|k-1}\mathbf{H}^T\mathbf{S}_k^{-1} \tag{2.44}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \tag{2.45}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_{k-1|k-1}. \tag{2.46}$$

The matrix $\mathbf{S}_k$ is called the innovation covariance, $\mathbf{C}_\mathbf{V}$ is the covariance of the measurement $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_\mathbf{V})$, and $\mathbf{F}$ is the measurement disturbance model.

Remark 2. *It makes sense to talk about whether the state of a process is observable within the Kalman filter estimation paradigm. For discrete time-invariant linear systems and ignoring disturbances and control inputs, we have the observation*

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k = \mathbf{C}\mathbf{A}\mathbf{x}_{k-1} = \mathbf{C}\mathbf{A}^k\mathbf{x}_0 = \mathbf{C}\Phi_{k:0}\mathbf{x}_0.$$

Constructing a vector of observations for $k = 0, \ldots, n_x - 1$ results in

$$\begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_{n_x-1} \end{bmatrix} = \begin{bmatrix} \mathbf{C}\Phi_{0:0} \\ \mathbf{C}\Phi_{1:0} \\ \vdots \\ \mathbf{C}\Phi_{n_x-1:0} \end{bmatrix} \mathbf{x}_0 = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{n_x-1} \end{bmatrix} \mathbf{x}_0 = \mathbb{O}\mathbf{x}_0. \tag{2.47}$$

Even in the case of a strictly non-observable system, a "matched" or observer system can be evaluated in order to estimate the state via measurements. While the accuracy of the observer system utilizing the Kalman filter will have accuracy limit by the degrees of freedom in which it has measurements.

Lemma 4. *The LTI discrete system $(\mathbf{A}, \mathbf{C})$ is observable if and only if $\mathbb{O}$ in (2.47) is non-singular. Observability implies that the state in any configuration can be discovered from the measurements.*

As mentioned, the Kalman filter is an powerful tool for sequentially estimating the state of a dynamical system as new measurements become available. It is also optimal in the sense

that it is a Bayesian filter, that is, it computes the posterior probability distribution over the hidden state [87]. The update step does not actually required well-behaved linearizations, as it is only the prediction part of the algorithm is what relies on the nonlinear propagation of the state and covariance. To date, there are many methods of obtaining approximations of the predicted state and covariance, including the extended and unscented Kalman filter, as well as various other Monte Carlo methods [75]. It is important that the disturbances can be modeled as Gaussian noise, otherwise there may be higher order moments of which the filter is unable to keep track. Some types of correlation or non-gaussian noise effects can be modeled in ways that suitably allow the use of the Kalman filter.

The Kalman filter represents one half of optimization necessary in most dynamical systems. While in some cases the system is uncontrollable but is only to be tracked by the filtering algorithm, other times the estimate from the Kalman filter is used to generate control inputs for the system. The estimation and control of the dynamical system have a unique duality in the expression of their solutions. The grand view of the continuous control and estimation of the dynamical system is illustrated in Figure 2.2. While the continuous version of the estimation and control problem is shown, the discrete version is simply the $\int(\cdot)dt$ operation replaced with a delay, $\frac{1}{z}$.

2.4 Optimal Control

In the previous, we demonstrated how analytical optimization applied to the estimation of the state of a dynamical system. We turn our attention now to dual problem of estimation, the control of the system. As in the case of estimation, we must decide what objective is to be achieved by our solution. If we again select the mean-squared error, $E[||\mathbf{x} - \mathbf{x}_{\text{desired}}||^2]$ where $\mathbf{x}_{\text{desired}}$, then for the linear system our solution will have a similar form to the Kalman filter, but with the gain, $\mathbf{K}_{\text{FB}}$, applied to the inputs, $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$, as in Figure 2.2.

The most general form of optimal control has the objective of selecting the correct control inputs to force a system to a final desired state or region over a particular time

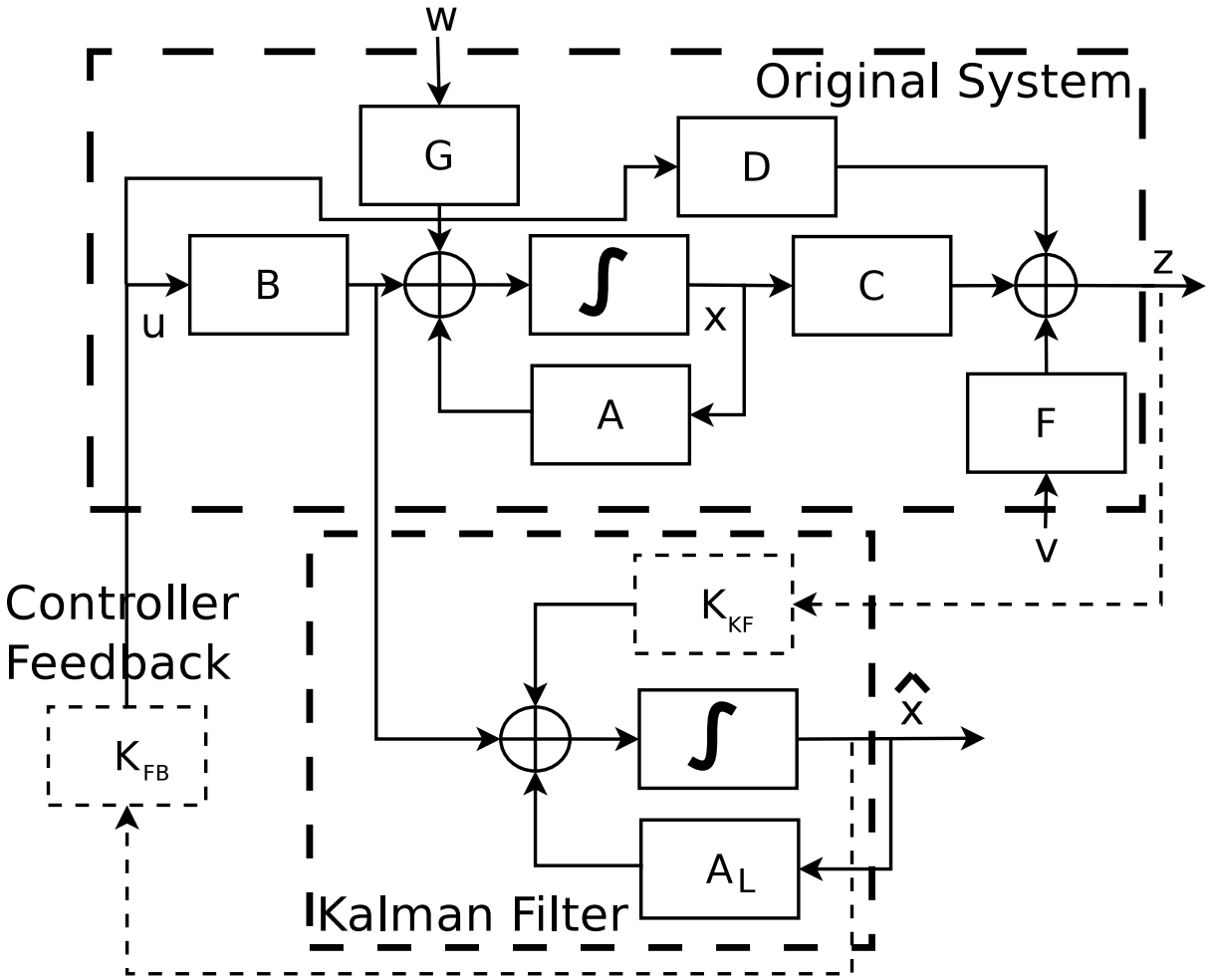


Figure 2.2: *Overview of the continuous linear dynamical system showing the interaction of the estimation and control efforts.* $\mathbf{A_L}$ *is the local approximation of the state transition,* $\mathbf{K}_{\mathrm{KF}}$ *is the Kalman filter gain, and all other matrices are as defined before. The feedback control gain,* $\mathbf{K}_{\mathrm{FB}}$*, is often replaced by a more elaborate control response. The discrete version of the above just replaces the* $\int \cdot$ *operation with a delay,* $\frac{1}{z}$*. Note the synonymous representation of the observation matrix by* $\mathbf{H}$ *and* $\mathbf{C}$*.*

horizon, sometimes in the presence of disturbances [87]. As well as the constraint of system behavior according to the state equation, the control inputs are usually constrained to a set (or range) of allowable actions. The range of time over which the controls are computed, called the horizon, can be infinite or finite. Obviously, most practical problems deal with finite time horizons, and good approximations of infinite horizon problems can be made with finite-horizon solutions given the right set of assumptions.

In the case of a linear (or linearizable) system, it is often the case that the open loop response (control input and advancement of the system state without observing the state or output) is unstable. This corresponds to the composite transfer function of the system having poles in right-half plane of the complex plane [74, 75]. A feedback loop is employed to inform the system of instability and provide a corrective input which is based on the current state (or an estimate of it) and a usually linear gain, which we have denoted $\mathbf{K}_{\text{FB}}$ in Figure 2.2. Analytically, the linear system stabilized in this way can be shown to have its unstable poles shifted into the left-half (left of the y-axis) of the complex plane.

Generalizations of this feedback technique include not only non-linear dynamics, consideration of disturbances and the allowance for a time-varying feedback gain, but also different objectives and methods of decision for the control process altogether.

The optimality of the control solution is determined with respect metric or "cost" function. These costs are almost always considered to be nonnegative. As mentioned, many motivating practical scenarios would set mean-squared error or the minimality of the control action (perhaps in some norm-sense) as the metric of fitness for a solution to the problem. The optimization problem can be written as

$$\begin{aligned} &\text{minimize}_{\mathbf{u}_t} \ J \\ &\text{subject to} \ \ \mathbf{x}_t \in \mathcal{X}, \ \mathbf{u}_t, \in \mathcal{U}, \ t = [0, T_0) \end{aligned} \tag{2.48}$$

where $J = \int_0^{T_0} c(\mathbf{x}_t, \mathbf{u}_t)dt$ is a cost function which takes into account all of the states and control decisions provided, with $c(\mathbf{x}_t, \mathbf{u}_t)$ the incremental decision cost. The space of feasible states (determined by the state dynamics) and feasible controls are represented by $\mathcal{X}$ and

$\mathcal{U}$, respectively. Thus, the solution would be a set of $\mathbf{u}(t)$ over $[0, T_0)$ that minimizes the above.

Typically, *optimal control* defines the problem of determining all of the control inputs prior to running the system. This is often referred to as an open-loop control policy. Whereas *dynamic programming*, an alternate formulation of the problem, would be considered a closed-loop control policy. In the following, we will consider a stochastic discrete state-action sequence and the associated minimum cost problem,

$$\begin{aligned} &\text{minimize}_{\mathbf{u}_k} \; J \\ &\text{subject to} \;\; \mathbf{x}_k \in \mathcal{X}, \;\; \mathbf{u}_k, \in \mathcal{U}, \;\; k = 0, \ldots, K, \end{aligned} \tag{2.49}$$

where the cost function, $J = c_K(\mathbf{x}_k) + \sum_{k=0}^{K-1} \mathbf{E}[c(\mathbf{x}_k, \mathbf{u}_k)]$, considers the expectation of any random quantities. The cost, $c_K(\mathbf{x}_K)$, is the final cost-to-go.

In the remainder of this work, optimal control techniques are not expressed nearly as formally as they are in this section. In terms of the subsequent discussion, the feedback control solution in Chapter 4 utilizes a unit-horizon (no look-ahead) policy dependent on channel and sensor uncertainty feedback. A model predictive control policy (MPC, discussed subsequently in Section 2.4.3) is used in Chapter 5 with a multi-step horizon for adequately considering future actions (note that the channel and battery state are never explicitly written in state space form, as we do not advance these states using a transition matrix, rather simple predictions using Wiener filters are used to estimate future nominal conditions of the battery and channel).

2.4.1 Dynamic Programming and Continuous Space Markov Decision Processes

The dynamic approach observes the state and selects a control input iteratively as it is realized, i.e., it produces a closed-loop control law. Dynamic programming is particularly helpful when there are unknown disturbances (e.g., $\mathbf{w}$ in our model) inputted to the system while the trajectory is being directed by the control inputs. In the deterministic case (i.e.,

no uncertainty), optimal control and dynamic programming achieve the same set of control inputs, as the state of the system is transitioned forward perfectly by the control inputs.

The calculation of the dynamic programming solution relies on the fact that, for a given state-action sequence, if we were to remove the first state-action, the resultant sequence is also optimal with respect to the remaining steps. This is the Bellman optimality principle [87]. This resembles the Markov property of stochastic processes (a property utilized in the development of the Kalman filter), that is, future optimal decisions are independent of past decisions, and the determination of the optimal control sequence involves starting at the final state and extending backwards.

Dynamic programming in the case of a stochastic dynamical system is called a Markov decision process. The problem utilizes what is called an "optimal cost-to-go", or value function, $V(\mathbf{x})$. Essentially, this cost function represents the minimum cost to reach the desired state from the current state. We represent this as an iterative optimization problem. The value function is

$$V_k(\mathbf{x}_k) = \min_{\mathbf{u}_k \in \mathcal{U}(\mathbf{x}_k)} \{c_k(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{E}[V_{k+1}(\mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k))])\}. \tag{2.50}$$

This equation along with the optimal control law

$$\mathbf{u}_k^*(\mathbf{x}_k) = \arg \min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \{c_k(\mathbf{x}, \mathbf{u}) + \mathbf{E}[V_{k+1}(\mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k))]\} \tag{2.51}$$

make up the *Bellman equations* [88], where $c_k(\mathbf{x}_k, \mathbf{u}_k)$ represents the cost of the state transition when control input $\mathbf{u}_k$ is applied. For a final time instant K, the solution to the Bellman equations for the deterministic case can be obtained from Lagrange analysis. The Lagrangian is

$$\mathcal{L}(\mathbf{X}, \mathbf{U}, \lambda) = c_K(\mathbf{x}_K) + \sum_{k=0}^{K-1} \left(c_k(\mathbf{x}_k, \mathbf{u}_k) + \lambda_k^T \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \right), \tag{2.52}$$

where $c_K(\mathbf{x}_K)$ is the optimal cost-to-go and λ_k is the Lagrange multiplier, and $\mathbf{X} = [\mathbf{x}_0, \ldots, \mathbf{x}_K]$ and $\mathbf{X} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$. Then the conditions for optimality resulting the discrete time solu-

tion,

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, 0) \\ \lambda_k &= c_k(\mathbf{x}_k, \mathbf{u}_k) + \lambda_{k+1}^T \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, 0) \\ \mathbf{u}_k &= \arg\min_{\mathbf{u}' \in \mathcal{U}} c_k(\mathbf{x}_k, \mathbf{u}') + \lambda_{k+1}^T \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}', 0). \end{aligned} \tag{2.53}$$

with $\lambda_K = c_K(\mathbf{x}_K)$, and $\mathbf{x}_0$ given. The vector of λ's together represent the gradient of the optimal value function.

If the state space were to be discretized, not only in time, but in the number of possible states and actions. As the variables being minimized over, $\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}$ are functions, the problem is infinite dimensional. The required number of discrete states needed is exponential in the dimension of the state for the discretized value function to converge to the optimal value function. This is called Bellman's curse of dimensionality, thus we can in practice only approximate the true value function [87]. In the case that the state must be estimated before being controlled, then such control problems are often termed partially observable Markov decision processes (POMDP), see for example [29, 89].

When an exact solution is desirable and mean-squared error is the cost to be minimized, then the following technique is applicable when the system is modeled as linear.

2.4.2 Linear Quadratic Regulator for Stochastic Control

As mentioned, there are several cost metrics we can optimize over in determine the control inputs. For the LTI discrete system, if we take the cost of the control strategy to be the sum of the weighted minimum mean-squared error and the weighted least norm-squared control action, then the optimization problem for stochastic control becomes

$$\begin{aligned} &\text{minimize}_{\mathbf{u}_k} \mathbf{E}\left[\mathbf{x}_K^T \mathbf{Q}_f \mathbf{x}_K + \sum_{k=1}^{K-1} (\mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}^T \mathbf{R} \mathbf{u})\right] \\ &\text{subject to } \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_{k+1} + \mathbf{w}_{k+1}, \ \mathbf{u}_k = \phi_k(\mathbf{x}_k) \end{aligned} \tag{2.54}$$

where we assume that $\mathbf{x}_{\text{desired}} = 0$, the positive semi-definite $\mathbf{Q}$ and $\mathbf{R}$ are quadratic weighting terms for the error and control input, respectively, and $\mathbf{x}_K^T \mathbf{Q}_f \mathbf{x}_K$ is the estimated cost-

to-go. Following the format of the Bellman equations we have

$$V_j(\mathbf{s}) = \min_{\mathbf{u}_j,\ldots,\mathbf{u}_{K-1}} \mathbf{s}^T\mathbf{Q}\mathbf{s} + \mathbf{u}_j^T\mathbf{R}\mathbf{u}_j + \mathbf{E}\left[\mathbf{x}_K^T\mathbf{Q}_f\mathbf{x}_K \sum_{k=j+1}^{K-1}(\mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^T\mathbf{R}\mathbf{u}_k),\right]$$
$$\text{subject to } \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_{k+1} + \mathbf{w}_{k+1},\ \mathbf{x}_j = \mathbf{s},\ \mathbf{u}_k = \phi_k(\mathbf{x}_k) \tag{2.55}$$

where the control law, $\phi_k(\mathbf{x}_k)$ is selected to minimize J. Notice that we have $V_K(\mathbf{s}) = \mathbf{s}^T\mathbf{Q}_f\mathbf{s}$ and $J^* = \mathbf{E}V_0(\mathbf{x}_0)$. We can find V_k by backward recursion,

$$V_k(\mathbf{s}) = \min_{\mathbf{u}} \left\{\mathbf{u}^T\mathbf{R}\mathbf{u} + \mathbf{E}[V_{k+1}(\mathbf{A}\mathbf{s} + \mathbf{B}\mathbf{u} + \mathbf{w}_k)]\right\},\ k = K-1,\ldots,0$$

where the expectation is taken over $\mathbf{w}_k$, optimal policies have the form

$$\phi_k^*(\mathbf{x}_t) = \arg\min_{\mathbf{u}} \left\{\mathbf{u}^T\mathbf{R}\mathbf{u} + \mathbf{E}[\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u} + \mathbf{w}_k]\right\}.$$

We assume that the V_k's have a recursive form that is quadratic with some form

$$V_k(\mathbf{x}_k) = \mathbf{x}_k^T\mathbf{P}_k\mathbf{x}_k + q_k,\ k = 0,\ldots,K,$$

with $\mathbf{P}_k \succeq 0$. We let $\mathbf{P}_K = \mathbf{Q}_f$ and $q_N = 0$. Writing $V_{k+1}(\mathbf{s}) = \mathbf{s}^T\mathbf{P}_{k+1}\mathbf{s} + q_{k+1}$ we see that the Bellman recursion for the value function is

$$\begin{aligned} V_k(\mathbf{s}) &= \mathbf{s}^T\mathbf{Q}\mathbf{s} + \min_{\mathbf{u}} \left\{\mathbf{u}^T\mathbf{R}\mathbf{u} + \mathbf{E}[(\mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u} + \mathbf{w}_k)^T\mathbf{P}_{k+1}(\mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u} + \mathbf{w}_k) + q_{k+1}]\right\} \\ &= \mathbf{s}^T\mathbf{Q}\mathbf{s} + \mathbf{tr}\{\mathbf{C}_\mathbf{W}\mathbf{P}_{k+1}\} + q_{k+1} + \min_{\mathbf{u}} \left\{\mathbf{u}^T\mathbf{R}\mathbf{u} + (\mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u})^T\mathbf{P}_{k+1}(\mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u})\right\}, \end{aligned} \tag{2.56}$$

where $\mathbf{tr}\{\cdot\}$ is the sum of the diagonals value of the matrix and $\mathbf{C}_{\mathbf{W}_k}$ is the covariance of $\mathbf{w}_k$. The minimization in the value function results in the optimal linear state feedback law

$$\phi_k^*(\mathbf{x}_k) = \mathbf{K}_k\mathbf{x}_k = -(\mathbf{B}^T\mathbf{P}_{k+1}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^T\mathbf{P}_{k+1}\mathbf{A}\ \mathbf{x}_k. \tag{2.57}$$

The recursive value of $\mathbf{P}_k$ and $\mathbf{q}_k$ are determined to be

$$\mathbf{P}_k = \mathbf{A}^T\mathbf{P}_{k+1}\mathbf{A} + \mathbf{Q} - \mathbf{A}^T\mathbf{P}_{k+1}\mathbf{B}(\mathbf{B}^T\mathbf{P}_{k+1}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^T\mathbf{P}_{k+1}\mathbf{A} \tag{2.58}$$

$$q_k = q_{k+1} + \mathbf{tr}\{\mathbf{C}_{\mathbf{W}_k}\mathbf{P}_{k+1}\}. \tag{2.59}$$

It is quite easy to see the similarity between the recursive form of the value function and respective control gain, with the Kalman filter update. Both the $\mathbf{P}_k$ value iteration and the covariance of the Kalman filter estimate have Riccati updating equations. The difference is that the Kalman filter runs forward in time, while the linear quadratic Gaussian regulator runs backward in time. When the horizon K is taken to infinity, the Riccati equation defining $\mathbf{P}_k$ can be solved to determine the steady state control gain. The optimal value of the finite horizon LQG problem is $\mathbf{V}_0 = \mathbf{tr}\{\mathbf{C}_{\mathbf{w}_k}\mathbf{P}_0\} + \sum_{k=1}^{K} \mathbf{tr}\{\mathbf{C}_{\mathbf{w}_k}\mathbf{P}_k\}$.

Remark 3. *Similar to the observability of the estimation process for an LTI discrete system, it makes sense to talk about whether the state of a process is controllable within the LQG estimation paradigm. For discrete time-invariant linear systems and ignoring disturbances we have the state equation*

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k = \mathbf{A}(\mathbf{x}_{k-2} + \mathbf{B}\mathbf{u}_{k-1}) + \mathbf{B}\mathbf{u}_k = \cdots = \mathbf{A}^k\mathbf{x}_0 + \sum_{j=0}^{k} \mathbf{A}^k\mathbf{B}\mathbf{u}_k.$$

Noting that our original state vector dimension was n_x, we construct a vector of observations for $k = 0, \ldots, n_x - 1$. What we want is to see how well the vector $\mathbf{u}$ is able to reach all the basis of $\mathbf{x}$. Our text matrix is

$$\begin{aligned}
\mathbf{x}_{n_x-1} &= \mathbf{A}^{n_x-1}\mathbf{x}_0 + \begin{bmatrix}\mathbf{B}\mathbf{u}_{n_x} & \mathbf{A}\mathbf{B}\mathbf{u}_{n_x-1} & \cdots & \mathbf{A}^{n_x-1}\mathbf{B}\mathbf{u}_1\end{bmatrix} \\
&= \mathbf{A}^{n_x-1}\mathbf{x}_0 + \begin{bmatrix}\mathbf{B} & \mathbf{A}\mathbf{B} & \cdots & \mathbf{A}^{n_x-1}\mathbf{B}\end{bmatrix} \begin{bmatrix} \mathbf{u}_{n_x} \\ \mathbf{u}_{n_x-1} \\ \vdots \\ \mathbf{u}_1 \end{bmatrix} \\
&= \mathbf{A}^{n_x-1}\mathbf{x}_0 + \mathbb{C}\mathbf{U}, .
\end{aligned} \tag{2.60}$$

Lemma 5. *The LTI discrete system $(\mathbf{A}, \mathbf{B})$ is controllable if and only if $\mathbb{C}$ in (2.60) is non-singular, that is, having linearly independent columns. Controllability implies that $\mathbf{x}$ may be controlled to reach any desired state.*

Remark 4. *An interesting note on the LQG solution is that the equation for the control gain, $\mathbf{K}_k$, is the same with respect to the recursive $\mathbf{P}_k$: with or without disturbance $\mathbf{w}_k$.*

The $\mathbf{P}_k$'s are modified when they are updated by q_k, but otherwise the solution is the same. Intuitively, the LQG is selecting the mean control decision, while keeping track of the inflated cost due to the uncertainties caused by the disturbances, $\mathbf{w}_k,\ k = 1, \ldots, K$.

The indifference of the solution toward stochastic disturbances can be noted in the following LS formulation of of the linear quadratic regulator. The state evolution of the system can be written as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_K \end{bmatrix} = \begin{bmatrix} 0 & 0 & & \ldots & 0 \\ \mathbf{B} & 0 & & & \vdots \\ \mathbf{AB} & \mathbf{B} & 0 & \ldots & \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{A}^{K-2}\mathbf{B} & \mathbf{A}^{K-3}\mathbf{B} & \ldots & \mathbf{B} & 0 \\ \mathbf{A}^{K-1}\mathbf{B} & \mathbf{A}^{K-2}\mathbf{B} & \ldots & \mathbf{AB} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_K \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \\ \vdots \\ \mathbf{A}^K \end{bmatrix} \mathbf{x}_0$$
$$+ \begin{bmatrix} 0 & 0 & & \ldots & 0 \\ \mathbf{G} & 0 & & & \vdots \\ \mathbf{A} & \mathbf{G} & 0 & \ldots & \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{A}^{K-2}\mathbf{G} & \mathbf{A}^{K-3}\mathbf{G} & \ldots & \mathbf{G} & 0 \\ \mathbf{A}^{K-1}\mathbf{G} & \mathbf{A}^{K-2}\mathbf{G} & \ldots & \mathbf{AG} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_K \end{bmatrix} = \mathbb{B}\mathbf{U} + \mathbb{A}\mathbf{x}_0 + \mathbb{G}\mathbf{W}. \tag{2.61}$$

The expected least squares minimizes $J_S = \mathbf{E}[||\mathbf{X} - [\mathbb{B}\ \mathbb{A}][U^T\ \mathbf{x}_0^T]^T||^2] + \mathbf{E}[||\mathbb{G}\mathbf{W}]||^2]$, assuming $\mathbf{x}_0$ and $\mathbf{w}_k\ \forall\ k$ are uncorrelated. Since $\mathbf{W}$ is not a variable over which minimization can be done, the same control inputs are selected for the deterministic and stochastic state equations. The only difference in the result is that the value of the minimization of the stochastic objective is offset from the minimized deterministic cost function, J_D, by precisely the expected uncertainty associated with the disturbances

$$J_S = J_D + \mathbf{tr}\{\mathbb{G}\mathbf{E}[\mathbf{W}\mathbf{W}^T]\mathbb{G}\}.$$

2.4.3 Suboptimal Control Strategies

The difficulty of discovering the cost-to-go function in most practical problems is prohibitively computationally expensive. Finding the cost-to-go must be done off-line, or be

approximated for on-line. Additionally, the final desired state may be require a prohibitive number of steps, K, to find the optimal control solution. Practical problems must have a reasonable finite time horizon, so instead of starting at the final desired state, we look at the state of the system at the end of a horizon, which we call $H \in \mathbb{N},\ H < K$. This type of approximation utilizes an *estimated* cost-to-go function which helps the naïve single-step control law maintain concern for future states and costs, even if the formulation cannot "see" all the way to the final desired state.

A broad class of suboptimal control methods, which are sometimes referred to approximate dynamic programming (ADP), are based on replacing the cost-to-go (value) function with an approximation $\tilde{V}(\mathbf{x})$. The two main types of approximations fit under the categories of off-line and on-line computation of the cost-to-go [69].

- Off-line computation of $\tilde{V}(\mathbf{x})$ (explicit cost-to-go calculation).

    1. **Derive the cost-to-go** from an optimal cost-to-go of a simpler related problem, obtaining $V'(\mathbf{x})$ through data aggregation or other type of problem simplification. The $\tilde{V}(\mathbf{x})$ are derived from $V'(\mathbf{x})$ for the simpler problem. [90]

    2. Use a **parametric approximation**, such as a neural network or weighted sum of basis functions to "learn" $\tilde{V}(\mathbf{x})$. The cost-to-go is approximated by a $\tilde{V}(\mathbf{x})$ which is tune according to the parameters of the neural network or basis function weights. Methods such as these include Q-learning, temporal difference, actor-critic-based methods, and other reinforcement learning methodologies [91].

- On-line computation of $\tilde{V}(\mathbf{x})$ (implicit cost-to-go calculation). Here the cost-to-go is computed as needed, perhaps using some truncated set of future cost-to-go values [69]

    1. Apply the **rollout** method, which computes $\tilde{V}(\mathbf{x})$ through Monte Carlo or other simulation-based methods, including use of on-line results and updates to the policy using a single step lookahead.

2. Use **open-loop feedback control** to compute the the future actions from the state $\mathbf{x}_k$, possibly based on the conditional probability distribution of the state (one step lookahead).

3. Find the **model predictive control** (MPC) solution where an optimal control solution is combined with a rolling horizon. There are variants of this methods, and other problems can be reduced to it.

There are some important modifications which can be made to some of the above algorithms which grant simplicity to the methods and allow for faster computation [69].

1. Multi-step lookahead: improved decision control performance at the cost of greater on-line computation.

2. Certainty equivalence: simplifies off-line and on-line computations by assuming that the current and future disturbances, $\mathbf{w}_k, \ldots, \mathbf{w}_K$, can be replaced by their expectations (or some nominal values).

3. Imperfectly known states: the states of the system are known imperfectly and the above schemes chooses the control based on an estimate of the state, $\hat{\mathbf{x}}_k$.

In this work, we don't have access to the aggregation of data for a similar problem and want to find an on-line policy for suboptimal control and resource allocation. We are also interested in solutions that provide us with the ability to apply convex optimization and gain intuition from analytical results without needing Monte Carlo simulation to obtain the cost-to-go function. Therefore, we focus on the result of applying model predictive control.

Model Predictive Control

The model predictive control (MPC) solution uses optimal control over a sliding window of future states. Referencing the above, we use the certainty equivalence modification which allows us to use the predicted values of random quantities in future steps. For some solutions

in this work, we additionally use imperfect estimates to propagate the state, on which we base the control decisions. The optimal control problem at time j with horizon H is

$$\begin{aligned} &\text{minimize}_{\mathbf{u}_j,\ldots,\mathbf{u}_{j+H}} \; c_K(\hat{\mathbf{x}}_K) + \sum_{k=0}^{K-1} c(\hat{\mathbf{x}}_k, \mathbf{u}_k)] \\ &\text{subject to} \quad \hat{\mathbf{x}}_k \in \mathcal{X}, \; \mathbf{u}_k, \in \mathcal{U}, \; k = 0, \ldots, K, \end{aligned} \tag{2.62}$$

and the approximate control response for time j becomes

$$\begin{aligned} \mathbf{u}_j^{\text{MPC}} = \arg_{k=0} \min_{\mathbf{u}_j,\ldots,\mathbf{u}_{j+H}} \; & c_K(\hat{\mathbf{x}}_K) + \sum_{k=j}^{j+K} c(\hat{\mathbf{x}}_k, \mathbf{u}_k) \\ \text{subject to} \quad & \hat{\mathbf{x}}_k \in \mathcal{X}, \; \mathbf{u}_k, \in \mathcal{U}, \; k = 0, \ldots, K, \end{aligned} \tag{2.63}$$

where we have used $\hat{\mathbf{x}}_k$ to represent the predicted values of the future states. These predictions are based-on the assumption of $\mathbf{w}_k$' being replaced with $\mu_{\mathbf{w}_k}$'s.

The MPC technique considers the stochastic system over which we are unable to adequately minimize the cost of controls for its entire lifetime. Instead of finding only the single-step best control inputs or intermittently at the end of each horizon, we can optimize the current control input based on looking at all control inputs up to time step H and successively optimizing over a moving horizon of length H. That is, we have a close-feedback loop, where in each time step a new decision can be made based on the new data and predictions of the future states.

An analysis similar to the LQG solution in Section 2.4.2 can be made for costs which are quadratic in their dependence on the state and control input. However, general analysis can be made for problems with convex costs and a problem where the linear system assumption is valid. Such a problem is posed as the following.

$$\begin{aligned} &\text{minimize}_{\mathbf{u}} \quad \sum_{k=j}^{j+H} f_k(\hat{\mathbf{x}}_k, \mathbf{u}_k) \\ &\text{subject to} \; \hat{\mathbf{x}}_{k+1} = \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_k + \hat{\mathbf{w}}_k \\ &\qquad\qquad \mathbf{g}(\mathbf{u}_k) \preceq 0, \end{aligned} \tag{2.64}$$

with Lagrangian

$$\mathcal{L}(\hat{\mathbf{X}}, \mathbf{U}, \lambda) = f_{j+H}(\hat{\mathbf{x}}_{j+H}) + \sum_{k=j}^{j+H} \left(f_k(\hat{x}_k, \mathbf{u}_k) + \lambda_k^T(\hat{\mathbf{x}}_{k+1} - \mathbf{A}_k\hat{\mathbf{x}}_k + \mathbf{B}_k\mathbf{u}_k + \hat{\mathbf{w}}_k) + \mu_k^T\mathbf{g}(\mathbf{u}_k) \right). \tag{2.65}$$

This problem is exactly the type that lends itself to revealing analysis and computable solutions in convex optimization.

2.5 Summary

The purpose of this chapter was to introduce the reader to the basic state space dynamical system model and show how optimization analysis and theory can be applied to estimation and control in such a system. Our next chapter outlines the model specifics for wireless sensor network communications and the on-board and between node resource models.

Chapter 3

Wireless Sensor Network Models

Engineering and other applied mathematics lives and dies on the assumption that models hold with some reliability. While certainly not appropriate for all conditions, we have attempted to outline some of the basic models and assumptions associated with the operation of a wireless sensor network, as well as when these models are valid.

Low-power wireless sensor networks (WSNs) usually assume a multi-hop communication framework with no centralized infrastructure due to the energy expense of transmitting over large distances. Individual sensors coordinate by forwarding packets from other nodes. However, centralization of the data is necessary for an end-user, the multi-hop nature of sensor networks is a practical requirement imposed by energy-consumption, as wireless transmissions attenuate superlinearly with respect to the propagation distance. This poses several interesting challenges including minimal energy availability (low power operation), appropriate power-saving sleep/wake-up schemes, scalability to many nodes, delayed data, communication failures, and network topology changes.

Each of the nodes in the wireless sensor network utilizes a radio transceiver, an energy harvester, a processing unit, and a sensing device. We will cover each and first outline the model for how the wireless devices communicate. The network and device models are then explained, including details about the quantization of the data. Lastly, we introduce some of the basic models for sensing used in this work, as well as what we look for in a metric which is considered across a wireless sensor network.

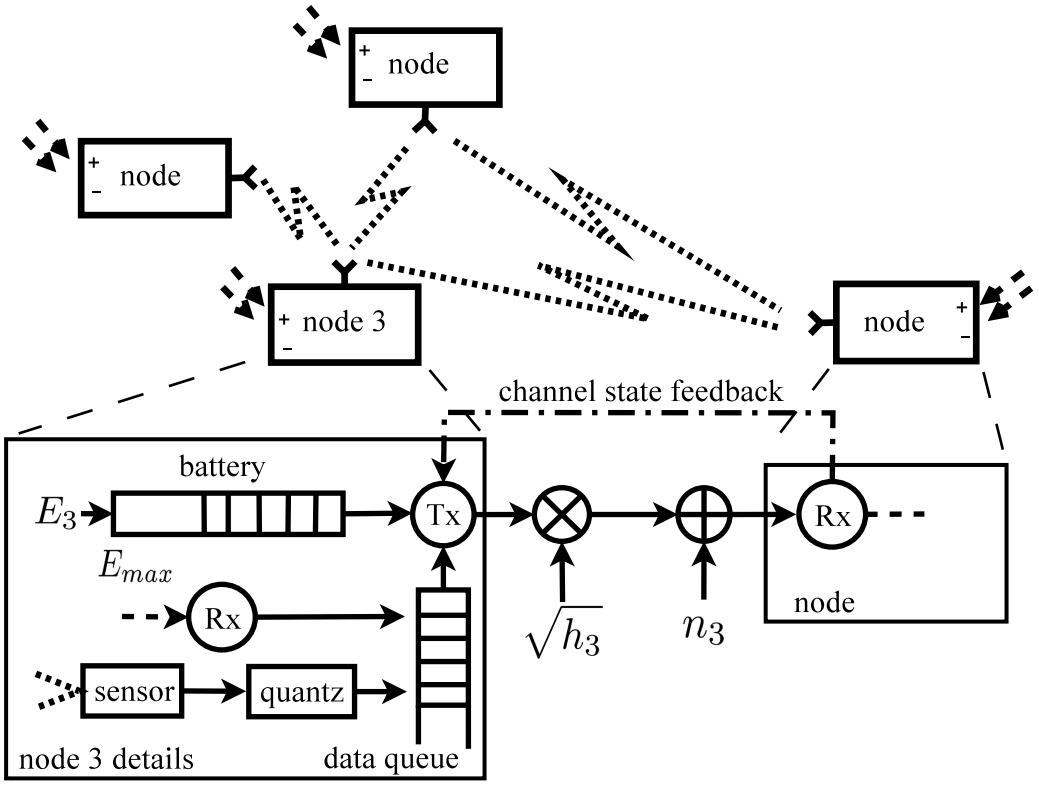


Figure 3.1: *An example of the energy harvesting wireless sensor network setup. Nodes transmit data through additive Gaussian fading channel with (causal) channel state feedback.*

3.1 Wireless Channel and Communication Models

The purpose of elaborating at all on the wireless channel and communication modes is to make explicit the characteristics of the functions we find therein. We consider a communication scheme where wireless sensors communicating over noisy fading channels. We consider the base path loss, $a_{nk} = |d_{nk}|^{\alpha}$ and fading coefficient, h_{nk}, where d is the d_{nk} is the distant between nodes and α is the path-loss constant. An illustration of an example WSN setup is shown in Figure 3.1. The signal at node m received from node n at time k is given by

$$y_{mk} = x_{nk} \cdot \frac{\sqrt{h_{nk}}}{a_{nk}} + n_{nk},$$

where x_{nk} signal transmitted with power p_{nk} during epoch k, where h_{nk} and n_{nk} are respectively the (squared) fading and additive Gaussian noise of the channel between nodes n and

m. We assume throughout the rest of our work that $\sigma_{nk}^2 = 1 \; \forall \; n, k$, although the methods herein are easily adapted to case of non-uniform noise variance. When we append the time dimension to our model, we assume that the the fading changes at most once per epoch. These assumptions fit for scenarios with slow and/or correlated fading across time. We also assume causal knowledge of this fading is available at the nodes. The indices $n, m, i, j, k \in \mathbb{N}$ unless otherwise noted.

3.1.1 Correlated Rayleigh Fading

In the description of the channel model above, we assume that for a fixed network of wireless sensor nodes, the fading coefficients are uncorrelated between nodes, but possible correlated in time. As there is little movement with respect to the communication transmissions, it is an appropriate assumption that the fading coefficients between short time intervals will have some correlation.

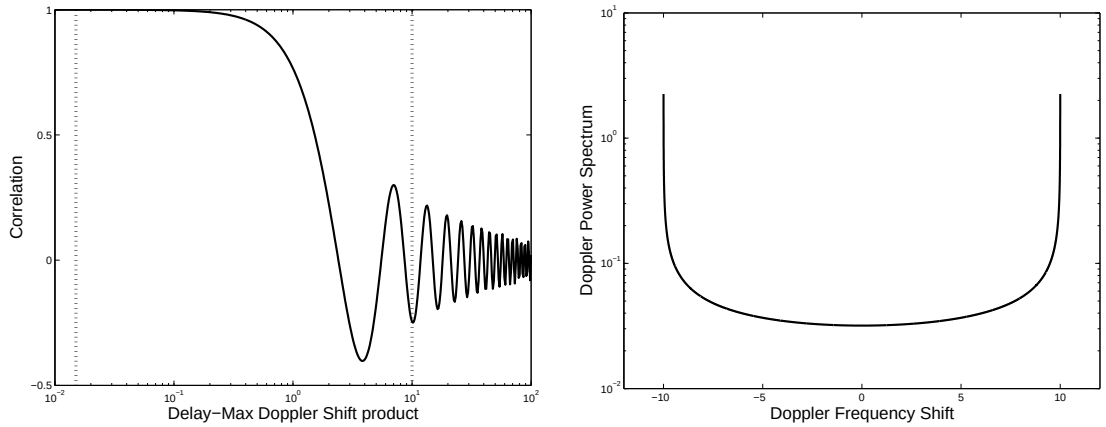
We adopt the Jakes model [92] for its simplicity in modeling systems with an expected limit on the maximum Doppler shift. The autocorrelation time-varying transfer function $h(f, t)$ is given as

$$R_h(\Delta t) = \frac{1}{2}\mathbf{E}[h^*(f,t)h(f,t+\Delta t)] = \mathcal{J}_0(2\pi f_D \Delta t),$$

where $\cdot^*$ is the complex conjugate, f_D is the maximum Doppler shift, and $\mathcal{J}_0$ is the zero-order Bessel function of the first kind. Since $f_D := vf_0/c$, with v the transmitter/receiver relative velocity, c the speed of the light, and f_0 the initial frequency of the transmission, we can expect small f_D values. This is a useful fact, as determining rapidly varying channel conditions usually requires adequate feedback to the transmitters, and thus, extra energy for communications. Correlated fading coefficients also allow us to accurately predict attenuation of future transmission and so effectively plan the utilization of wireless node energy.

The power spectral density of the Jakes model is given by

$$S_h(f) = \begin{cases} \frac{1}{\pi f_D \sqrt{1-(f/f_D)^2}} & |f| \leq f_D \\ 0 & |f| > f_D \end{cases}.$$



(a) (b)

Figure 3.2: *An illustration of the Jakes model for the (a) correlation of fading coefficients with a maximum Doppler spread of* $10Hz$ *in the (b) power spectral density. Notice the regions where the correlation is easily approximated (demarcated below the lowest and above the highest dotted lines.)*

An illustration of autocorrelation and PSD for the correlation of fading coefficients is in Figure 3.2.

The Bessel function can be easily approximated in particular regions of the correlation function, namely, for $f_D\Delta t << 1$ and for large arguments $f_D\Delta t >> 3/4$. The approximate characterization of the correlation can be summarize as

$$R_h(\Delta t) \approx \begin{cases} 1 & f_D\Delta t < 0.01 \\ \mathcal{J}_0(f_D\Delta t) & 0.01 < f_D\Delta t < 10 \\ \sqrt{\frac{2}{\pi f_D\Delta t}}\cos\left(f_D\Delta t - \frac{\pi}{4}\right) & f_D\Delta t > 10 \end{cases}$$

A point of interest which we will not tangent into much is the generation of correlated Rayleigh random variables [93]. Once we have a model characterizing the correlation, we would like to simulate system with such correlated fading coefficients. A series of zero-mean, uniform variance (σ^2) complex Gaussian signals

$$s_i = x_i + jy_i,\ i = 1, \ldots, N,$$

have envelopes

$$h_i = |s_i| = \sqrt{x_i^2 + y_i^2},\ i = 1, \ldots, N,$$

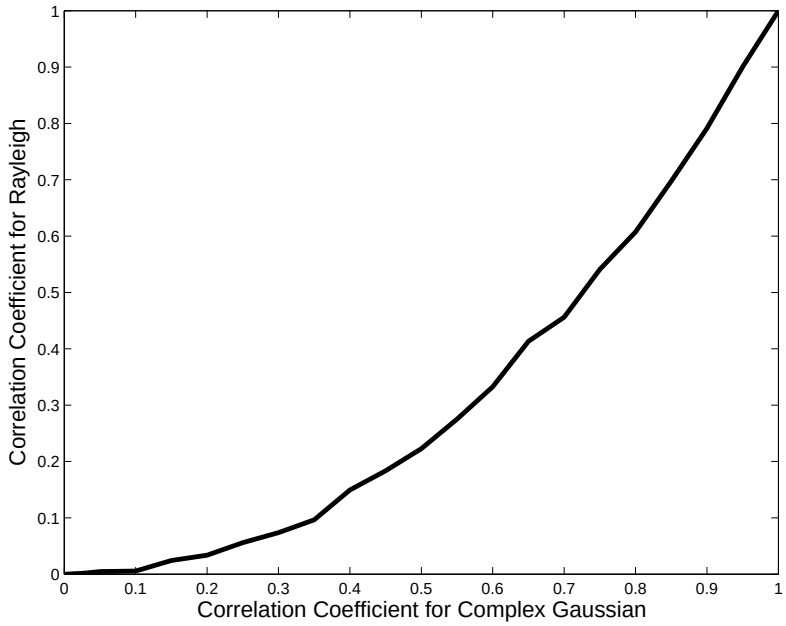


Figure 3.3: *Relationship between the correlation coefficients for complex Gaussian (absolute value) and Rayleigh random variables.*

which are Rayleigh distributed. If we assume we would like a normalized covariance matrix $\mathbf{C}_h$ (possible matching the Jakes model for correlated fading). By normalized we indicate that the diagonals of $\mathbf{C}_h$ are 1. We would like to write $\mathbf{C}_h$ in terms of the covariance matrix for the complex Gaussian signals, $\mathbf{C}_s$. The off-diagonal correlation coefficients between h_i and h_j, ρ_{ij}^h is exactly determined by the absolute value of the corresponding correlation of the complex Gaussian signals, $|\rho_{ij}^s|$. The exact analytical relationship between the two correlation coefficients is

$$\rho_{ij}^h = \frac{(1 + |\rho_{ij}^s| E_i(\eta) \left(\frac{2\sqrt{|\rho_{ij}^s|}}{1+|\rho_{ij}^s|} \right) - \frac{\pi}{2}}{2 - \frac{\pi}{2}}, \tag{3.1}$$

where $E_i(\eta)$ denotes the complete elliptic integral of the second with modulus η. Polynomial approximations can used to determine E_i, and a plot of this relationship between the Rayleigh correlation coefficient and absolute value of the complex Gaussian correlation coefficients is in Figure 3.3.

A basic algorithm can then be followed to generate a set of correlated Rayleigh random

variables [93].

1. Form the normalized covariance matrix, $\mathbf{C}'_h$ (unit values on diagonal), with entries $\rho^h_{ij} := \mathbf{E}[h^*_i h_j]/\sqrt{\sigma^2_i \sigma^2_j}$.

2. For each off-diagonal Rayleigh correlation coefficients, compute the value of the Gaussian correlation, $\mathbf{C}'_s$ from an interpolation of the relationship in Figure 3.3.

3. Generate N uncorrelated complex Gaussian random samples $\mathbf{s} = [s_1, \ldots, s_N]^T$.

4. Determine the eigen-quantities (using SVD *aut al.*) Λ and $\mathbf{U}$, where $\mathbf{C}_s = \mathbf{U}\Lambda\mathbf{U}^H$. Form the rooted-variance coloring matrix $\mathbf{L} = \Lambda^{1/2}\mathbf{U}^H$, where $\cdot^H$ is the Hermitian transpose.

5. Generate the correlated Rayleigh samples, $\mathbf{h} = [h_1, \ldots, h_N]$ from $h_i = |\hat{h}_i|$, where

$$\hat{h} = \mathbf{L}^{-1}\mathbf{s}.$$

3.1.2 Communication Schemes

There are many methods of breaking data into meaningful pieces for transmission across a wireless channel. We investing the following models for the unique convex functions they result in.

Binary Phase Shift Keying

Binary phase shift keying (BPSK) is a simple signal scheme which uses two signals, usually designated $-p$ and p, where p is the amplitude of the transmission [94]. Figure 3.4 illustrates the phase/amplitude-based signal schemes and features thereof. If we assume that we have a packet to be transmitted using BPSK over a channel from node n at time k with Rayleigh fading and additive Gaussian noise, then we can show the following.

The noise variance of the i^{th} packet to transmit is denoted by convex function $r^{c,i}_{nk}(p^i_{nk})$, which is derived as follows. Let us denote the probability of error of $\Pr^{i,\ell}_{nk}$ for the ℓ^{th} bit of

the transmission, then the noise due to the imperfect channel is

$$n_{nk}^{c,i} = \begin{cases} \pm 2^{\ell} \Delta_n^{i,\ell} & \Pr_{nk}^{i,\ell} \\ 0 & 1 - \sum_{\ell=0}^{b_{nk}^i - 1} \Pr_{nk}^{i,\ell} \end{cases} \tag{3.2}$$

where $\Delta_{nk}^{i,\ell} = \frac{2W}{2^{b_{nk}^i - 1}}$ is the quantizer step size, with W the dynamic range of the quantizer and b_{nk}^i the number of bits in the transmission. We now assume: (a) the bits in the transmit sequence have independently distributed probability of error, i.e., $\Pr_n^{i,\ell} = \Pr_{nk}^i$, (b) there is as most one bit error in each transmit sequence, and (c) the channel noise variance is unchanged during a transmission period. All of these are reasonable assumptions for a slow fading channel with adequate channel coding. Thus the noise variance contributed from the channel is

$$\begin{aligned} r_{nk}^{c,i} &= \sum_{\ell=0}^{b_{nk}^i - 1} (\pm 2^{\ell} \Delta_n^{i,\ell})^2 \overset{i,\ell}{\underset{nk}{\Pr}} \\ &= \overset{i}{\underset{nk}{\Pr}} \times (\Delta_n^{i,\ell})^2 \sum_{\ell=0}^{b_{nk}^i - 1} 4^{\ell} \\ &\approx \frac{4W^2}{3} \overset{i}{\underset{nk}{\Pr}}, \end{aligned}$$

which simplifies to

$$r_{nk}^{c,i} \approx \frac{4W^2}{3} \left(1 - \sqrt{\frac{0.5\Gamma_n^i}{1 + 0.5\Gamma_n^i}} \right). \tag{3.3}$$

Where $\Gamma_n^i = \frac{2p_n^i \overline{h_n}}{N_0}$ represents the average received signal-to-noise-ratio, $p_n^i \in [p_n^{min}\ p_n^{max}]$ is the transmit energy level for the i^{th} element of the n^{th} sensor. The minimum power level per bit is p_n^{min} which is necessary to achieve a minimum system SNR. The maximum power per bit in a transmission is p_n^{max}. The power level p_n^i considers only the radio-frequency (RF) power required at the node, and none of the power consumed by other circuits in the device, which are considered negligible for simplicity in our analysis. The average power of the Rayleigh fading channel coefficient is $\overline{h_n}$ and $N_0/2$ is the channel noise power spectral density.

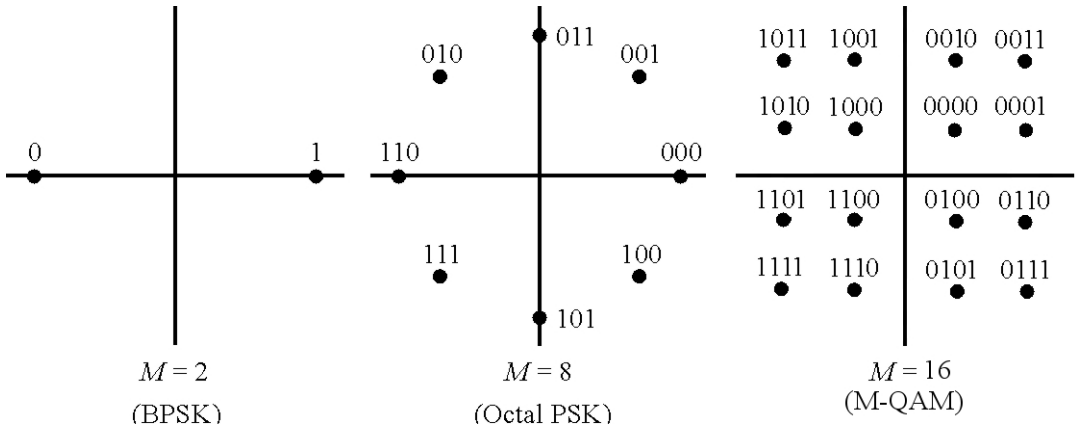


Figure 3.4: *An illustration of signal constellation maps for binary and multi-phase shift keying, as well as an example of a QAM signal constellation. The horizon axis represents the real (in-phase) part of the signal, while the vertical axis represents the imaginary (or quadrature) portion of the signal.*

M-ary Quadrature Amplitude Modulation

M-ary quadrature amplitude modulation (M-QAM) [94] is a signaling technique which utilizes the quadrature and in-phase components of the transmitted signal to generate a complex key corresponding to both a phase shift and amplitude adjustment such that the complex plane within the circle formed by $|p_{nk}^i| \leq \mathbf{p}_{\max}$. Considering this communication scheme, a simplified expression relates the transmission power and the rate as [95]

$$p_{nk}^i = c_n \alpha_n \sqrt{h_{nk}} \ln\left(\frac{2}{\mathrm{Pr}_{nk}}\right)(2^{b_{nk}^i} - 1) \tag{3.4}$$

where c_n is the constant related to the noise floor and tolerances of the hardware (gain, *etc.*), probability of error is Pr_{nk}, h_{nk} is the magnitude of the fading coefficient, and $\alpha_n = \delta_n^\nu$ is the path loss multiplier (where ν is the path loss exponent and δ_n the distance from the node to the fusion center). Here we have assumed MQAM constellation size, $M_n^i(k)$, and rate, b_{nk}^i, are equal.

We will consider the effect of channel noise on transmitted data using an M-QAM scheme in a later section. The analysis therein fits with the particular metric we use for optimization.

What we can say for now is that for a certain fixed probability of error, the effect of the channel noise variance on the final estimates can be bounded by $1 + p_0 > 1$ and reduce the problem to considering only the effects of quantization and local estimate inaccuracies.

3.2 Network and Device Models

The organization and handling of data traffic through the network is very important in networks, especially when there is a lot of data to move around. Fortunately for our modeling process, since WSNs are low-power and usually only transmit to a subset of neighboring nodes, we generally consider a standard periodic transmit to be free of packet drops (although we address packet drop in Chapter 6). Due to the assumption of low-power transmissions, we also assume that there is little or no interference between various nodes, and so we do not consider it.

We first discuss the dispersion and collection data in the network. Quantization of data is reviewed. Then we briefly discuss the energy storage and harvesting model.

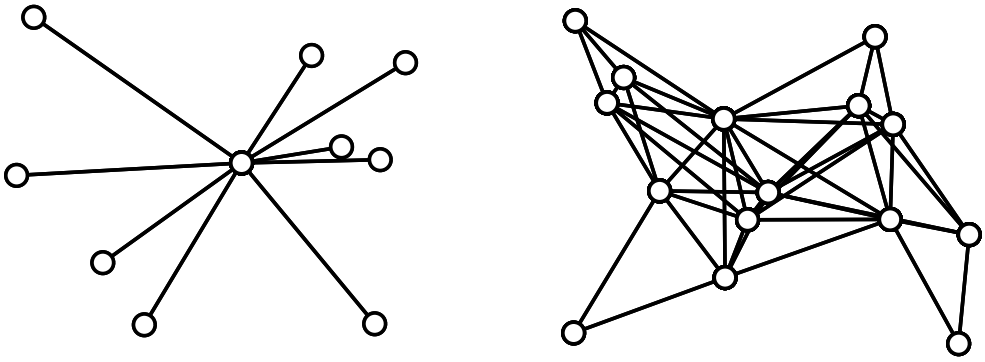
3.2.1 Data Collection

The N spatially distributed nodes in the WSN take measurements of a dynamic process and update their estimates. The estimate/measurement is quantized and transmitted to a neighboring node, which updates its estimate of the parameter or state based on received reports.

Centralized with Fusion Node

The first paradigm considered contains multiple remote sensor nodes which transmit data to a fusion node. It is the responsibility of the fusion node to instruct each node on how to send its update at each time instance via some orthogonal signaling scheme[1]. It is assumed that the fusion node is not energy constrained in its transmissions and that the energy of

[1]For example, if FDMA (frequency division multiple access) [94] is used then a fixed total bandwidth is divided among the nodes.



(a) (b)

Figure 3.5: *An illustration of the graph-like nature of a multi-node wireless sensor network for (a) centralized architecture, and (b) decentralized architecture. Each node is represented by a ◯ and the edges which join them are communication connections.*

receiving a transmission at a sensor node is negligible. As power is not constrained at the fusion node we assume practically noiseless feedback.

Decentralized

In the case that data does not flow to a particular central node, the nodes must determine amongst themselves how they should vary communication rates to a predefined (or dynamically changing) set of neighbors.

3.2.2 Quantization

We again must consider the various communication schemes we have allowed our communications to be modeled with. Let $x_i \in [-W, W]$ with $[-W, W]$ the dynamic range of the measurement source. If we use b_n bits then we will have $2^{b_n} - 1$ intervals of length $\Delta = \frac{2W}{2^{b_n}-1}$. Quantize x_n uniformly over the dynamic range requires rounding to the nearest interval centroid.

For a piece of data compressed using b_n bits, the noise variance for the uniform quanti-

zation is

$$r_n^q = \frac{W^2}{3(2^{b_n} - 1)^2}. \tag{3.5}$$

Each $b_n^i(k) \in [1, BW]$ is the number of bits used to quantize the i^{th} element of the state estimate from sensor n to the fusion node. BW is the rate constraint for the entire system. The quantization scheme is homogeneous across sensors for a quantization level, with a set dynamic range for components of position, velocity, et cetera.

3.2.3 Energy Storage and Harvesting Model

As we present various unique communication schemes, we also consider different models for energy storage and capabilities. Practically, there are some energy-sensitive wireless networks which are incapable of being fitted with energy harvesting devices, so an analysis of battery only devices is valuable. Problems solved using either model offer insight into the behavior of these devices.

We can consider an initially fixed amount of available energy to each sensor node (batteries only). A requirement that the communication system is that $p_{nk}^i \in [p_{\min}\ p_{\max}]$, where p_{nk}^i is the transmission power of the n node at time k and $[p_{\min}\ p_{\max}]$ are the minimum and maximum transmission power.

In our system model, we consider the energy costs associated with sensing and processing to be negligible relative to the wireless transceiver energy consumption [61, 65]. Signal reception energy usage is controlled by the MAC layer, using various bandwidth division schemes [5]. Therefore, we model the per epoch energy cost for signal reception as a constant, E_{rcv}. We proceed further to eliminate this complication by noting that the available energy, denoted E'_{nk}, in each epoch and at each sensor can have its receive energy extracted before optimization of transmit energies. That is, energy which can be allocated for transmission is

$$E_{nk} = \left[E'_{nk} - E_{\text{rcv}}\right]^+, \tag{3.6}$$

where $[\cdot]^+ = \max\{\cdot, 0\}$. In this way, our problems will only consider the assignment of

transmission rates and power levels.

Energy Harvester

When including the possibility of harvested energy, a time element can be surreptitiously assumed. That is, if a system is harvesting energy, it should estimate how much energy *will be* available and make decisions based on these predictions.

As we append time as a dimension when modeling energy harvesting, we will consider fixed epoch lengths while the authors in [57] model epochs as being of random size determined by some arrival distribution. The assumption of fixed epoch lengths caters to the implementation of many wireless sensor network systems, which utilize duty cycles of some practically fixed period. This reduces power consumption but can increase delay [3, 5]. It is a simple task to adapt the subsequent algorithm in the case of the non-uniform time step.

The energy harvested at each node could be from a variety of sources, e.g., generated by wind, solar, or seismic activity. We assume that the energy can be modeled as having discrete arrivals as in a Poisson process with rate λ (the energy arrival rate) for t, the elapsed time within epoch k. Additionally, we complicate this model slightly by employing a marked Poisson process for the energy arrival, then total energy in epoch k for node n is a compound Poisson process

$$E_{nk} = \sum_{j=0}^{N_n(t)} E_{A,nj}, \tag{3.7}$$

where $N_n(t) \sim \text{Poisson}(\lambda t)$ and the arrival energy quanta $E_{A,nj} \sim \text{unif}(p_{\min}, p_{\max})$ or $E_{A,nj} \sim \exp(\gamma)$. The uniform distribution and exponential modeling the per arrival quanta level is appropriate for the application [3, 57].

The collected energy is stored in a battery, super-capacitor, or other storage device which has a maximum transmit energy capacity denoted, $E_{\max} = E'_{\max} - E_{\text{rcv}}$, with $E'_{\max}$ the actual energy storage capacity. An illustration of the energy harvesting operating in conjunction with the wireless node is in Figure 3.1. We adopt the assumption made in [57] that all of the gathered energy quanta are less than the maximum storage capacity, i.e., $E_{nk} \leq E_{\max}$.

This assumption is intuitive since even if an energy harvesting device could harvest more than the capacity of its battery, the energy would be supplied to the transmitter through the battery, thus limiting the usable energy. We define epochs so that energy collected in epoch k is denoted E_{nk}, with allocated power p_{nk} which can use energy $E_{nj}, j = 0, \ldots, k$. We also follow the common assumption that the harvested energy is uncertain but predictable [3, 59–61]. Constraints imposed on the optimization by the causality of the harvested energy and by the limited storage capacity will be discussed in Chapter 5.

It should be noted that because of the intermittent (at times) availability of energy, delays can be introduced in the network so that the convergence of the WSN nodes to the same estimate or decision can be slowed. This will be covered more in Chapter 6.

3.3 Sensing Model

In this work, we consider a basic linear sensing model which is meant to approximate a range-angle measurement. Standard range-angle measurements involve non-linear functions of the state variables [96]. This would force us to either rely on Taylor series approximations of the Hessian of the measurement operation, or use Monte-Carlo methods to find an approximation of the resultant innovation covariance. Since these ideas are not the objective of this work, but rather to gain some analytic understanding and useful frameworks for optimization of resource in WSNs, we present a simple approximation for range-angle measurements. These measurements are linear position measurements corrupted by spatially correlated noise according to the following model.

We assume we are given some variance of the measurement in range and angle, σ_r^2 and σ_θ^2. We can then use basic geometry to find that the measurement, $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$, of a target on the x-axis (i.e., $(x, 0)$, see Figure 3.6) has the approximate covariance matrix

$$\mathbf{C}_\mathbf{z}(r, \theta)|_{\theta=0} = \begin{bmatrix} \sigma_r^2 + r^2 \sin^2(\sigma_\theta) & 0 \\ 0 & r^2\sigma_\theta^2 + \sigma_r^2\sigma_\theta^2 \end{bmatrix}. \tag{3.8}$$

Using rotation matrices from basic linear algebra, $\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$, we can find

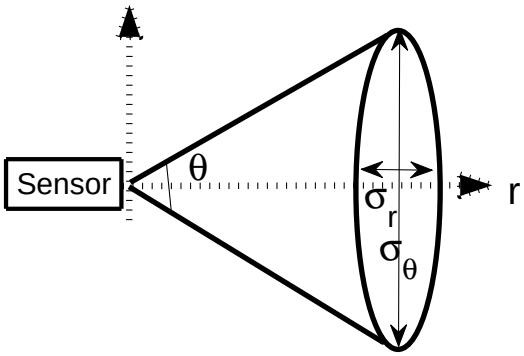


Figure 3.6: *An illustration of idea behind the noise variance approximation for a simple linear position measurement model.*

the covariance for a measurement thought to be at $(x, y) = (r\cos(\theta), r\sin(\theta))$ by find the product $\mathbf{R}^T(\theta)\mathbf{C_z}(r, 0)\mathbf{R}(\theta)$, resulting in

$$\mathbf{C_z}(r,\theta) = \tag{3.9}$$

$$r\begin{bmatrix} (\sigma_r^2 + r\sin^4(\sigma_\theta))\cos^2(\theta) + (r\sigma_\theta^2 + \sigma_r^2\sigma_\theta^2)\sin^2(\theta) & (\sigma_r^2 + r\sin^4(\sigma_\theta) - r\sigma_\theta^2 - \sigma_r^2\sigma_\theta^2)\sin(\theta)\cos(\theta) \\ (\sigma_r^2 + r\sin^4(\sigma_\theta) - r\sigma_\theta^2 - \sigma_r^2\sigma_\theta^2)\sin(\theta)\cos(\theta) & (\sigma_r^2 + r\sin^4(\sigma_\theta))\sin^2(\theta) + (r\sigma_\theta^2 + \sigma_r^2\sigma_\theta^2)\cos^2(\theta) \end{bmatrix}. \tag{3.10}$$

Thus, measurements are generated according to $\mathbf{z} = \mathbf{H}x + \mathbf{v}$, with $\mathbf{H} = [\mathbf{I}_d\ \mathbf{0}_{d\times d\eta}]$, where d is the dimension of the space and η is the number of derivatives tracked in the state (e.g., $\eta = 2$ for position-velocity, $\eta = 3$ for position-velocity-accerlation). The measurement noise is $\mathbf{v} = (\mathbf{R}(r,\theta))^{-1/2}\mathbf{u}$ with $\mathbf{u} \in \mathbb{R}^2$ and $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

3.4 Metrics

There are several metrics we can maximize (or minimize) when seeking to optimize some aspect of WSN performance. We seek a goal-oriented metric which illustrates the optimiza-

tion of the network decisions toward the completion of a cooperative task. Note that we will use metrics, utility functions, and objectives interchangeably. Examples of these metrics include:

- Mutual information [29],
- Rate [57, 61, 68],
- Distortion [19, 21, 43, 95, 97, 98].

Mutual information tells us how informative new data is with respect to correlated old data. For example, if we consider the Kalman filter updated by a measurement $\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$, then the mutual information between the current state and the measurement, given the previous measurements, is given by

$$\begin{aligned} I\left(\mathbf{x}_k; \mathbf{z}_k \left| \mathbf{z}_{k-1}, \ldots, \mathbf{z}_1 \right.\right) &= H\left(\mathbf{z}_k \left| \mathbf{z}_{k-1}, \ldots, \mathbf{z}_1 \right.\right) - H\left(\mathbf{z}_k \left| \mathbf{x}_k, \mathbf{z}_{k-1}, \ldots, \mathbf{z}_1 \right.\right) \\ &= H\left(\mathbf{z}_k \left| \mathbf{z}_{k-1}, \ldots, \mathbf{z}_1 \right.\right) - H\left(\mathbf{z}_k \left| \mathbf{x}_k \right.\right). \end{aligned} \tag{3.11}$$

Since conditioning on $\mathbf{x}_k$ (a Gauss-Markov process) removes the conditioning on the prior measurements. If we note that $\mathbf{z}_k|\mathbf{x}_k \sim \mathcal{N}(\mathbf{H}\mathbf{x}_k, \mathbf{C}_\mathbf{v})$, we have

$$H(\mathbf{z}_k|\mathbf{x}_k) = \frac{1}{2}\log\left|2\pi e \mathbf{C}_\mathbf{v}\right|.$$

Likewise, the measurement $\mathbf{z}_k|\mathbf{z}_{k-1}, \ldots, \mathbf{z}_1 \sim \mathcal{N}(\mathbf{H}\mathbf{A}\mathbf{x}_{k-1}, \mathbf{H}\mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T\mathbf{H}^T + \mathbf{C}_\mathbf{v})$ produces

$$H(\mathbf{z}_k|\mathbf{z}_{k-1}, \ldots, \mathbf{z}_1) = \frac{1}{2}\log\left|2\pi e(\mathbf{H}\mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T\mathbf{H}^T + \mathbf{C}_\mathbf{v})\right|.$$

Thus, the mutual information between an updated $\mathbf{x}_k$ and a measurement $\mathbf{z}_k$, given the previous measurements, is

$$I\left(\mathbf{x}_k; \mathbf{z}_k \left| \mathbf{z}_{k-1}, \ldots, \mathbf{z}_1 \right.\right) = \frac{1}{2}\log\left(|\mathbf{H}\mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T\mathbf{H}^T + \mathbf{C}_\mathbf{v}|/|\mathbf{C}_\mathbf{v}|\right), \tag{3.12}$$

which is the expected reduction in entropy given the updated measurement. Given several possible sensor with measurements, this metric would be useful in determining what is the

optimal schedule for activating/deactivating sensors. Despite this pleasing intuition, we do not employ mutual information, since we are looking more at how the varying of the communication and quantization parameters in our wireless nodes affects the final estimate quality.

The Shannon rate is is a metric which provides an upper bound on amount of usable data that can be accurately transmitted for a particular signal-to-noise ratio (SNR) and bandwidth. It is given by

$$C = BW \log\left(1 + \frac{hp}{N_0}\right), \tag{3.13}$$

where C is the maximum capacity in bits, BW is the bandwidth of the channel in Hertz, p is our transmitted signal power, h is the power of the fading, and N_0 is the noise power. This is an incredible useful metric in communication systems. However, even when maximized, may not produce a cooperative goal of providing useful information throughout the WSN, but rather, merely individual throughput. We do use rate maximization as an example application in a later chapter, but it is primarily in motivating the analysis for other optimization methods.

Distortion is a metric with which the nodes in the network cooperative to provide adequate results. See more in Chapter 2 about the BLU estimator. The original (superscript 'O') distortion for the BLU estimate results in a mean-squared error (or variance) in epoch k for a scalar estimate is

$$D_k^O = \left(\sum_{n=1}^{N} \frac{1}{\sigma_{nk}^2}\right)^{-1}, \tag{3.14}$$

where σ_{nk}^2, is the noise variance of the estimate from node n at time k. Borrowing from utility theory [99] we say that the BLUE variance, as a utility, has complementary flows.

One classification for utility functions specifies how the resource 'flows' behave in the functions. These include independent, complementary, substitutionary, and duplicate. For a general utility functions $U : \mathbb{R}^{N\times 1} \to \mathbb{R}$ and $U', U_n : \mathbb{R} \to \mathbb{R}$ with resource 'flow' variables

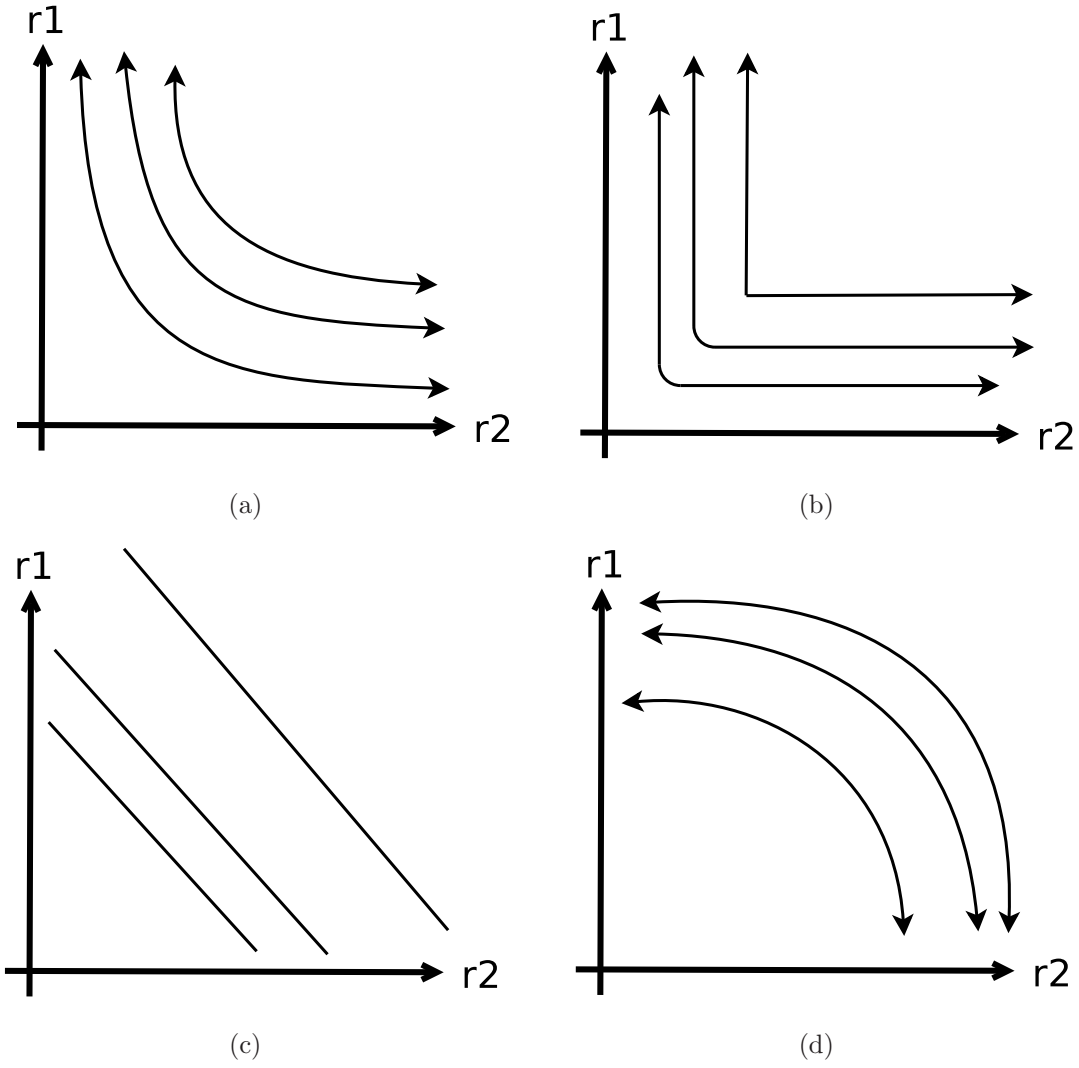


Figure 3.7: *Level curves for types of flow in different utility functions for resources r_1 and r_2: (a) independent, (b) complementary, (c) substitutionary, and (d) duplicate.*

$r_1, \ldots, r_N \in \mathbb{R}$, these (usually concave) functions are characterized by

$$
\begin{aligned}
U(r_1, \ldots, r_N) &= U_1(r_1) + \cdots + U_N(r_N) && \text{(Independent)} \\
U(r_1, \ldots, r_N) &= U'(\min\{r_1, \ldots, r_N\}) && \text{(Complementary)} \\
U(r_1, \ldots, r_N) &= U'(r_1 + \cdots + r_N) && \text{(Substitutionary)} \\
U(r_1, \ldots, r_N) &= U'(\max\{r_1, \ldots, r_N\}) && \text{(Duplicate)}
\end{aligned}
$$

Rate maximization and mutual information represent independent flows, as improvement in the rate from any of the nodes in the WSN improves the sum of the utilities. Complementary flows are of particular interest in WSN estimation, as is illustrated by the BLUE variance metric, and have the behavior that a combination of flows produce the utility value. That is, for a particular value of one flow, no additional utility is gained from increasing other flows. An illustration of the terrain of the level sets for these types of utility functions, for a pair of resources, is given in Figure 3.7. This corresponds to the case in the BLUE variance metric where, given a low variance estimate, the overall utility is not helped by small improvements in larger variance estimates. The independent flows are straightforward, an example of which is the objective

$$D_k^{RWF} = \sum_{n=1}^{N} \sigma_{nk}^2. \tag{3.15}$$

In substitutionary flows, flows can be substituted with no change in the utility. Duplicate flows are not considered in many optimization formulations, as they are non-concave (for maximization).

There are two notions of how to extend the BLU variance metric. If we seek to average the above metric across time, then we must understand whether our averaging produces 'difficult' objectives (non-convex/concave, duplicate flows, *etc.*) or objectives which provide us with an intuition and useful metric for optimization. The arithmetic and harmonic means of the original distortion metric produce

$$D^A = \sum_{k=1}^{K} \left(\sum_{n=1}^{N} \frac{1}{\sigma_{nk}^2} \right)^{-1}, \tag{3.16}$$

and

$$D^H = \left(\sum_{k=1}^{K} \sum_{n=1}^{N} \frac{1}{\sigma_{nk}^2} \right)^{-1}, \tag{3.17}$$

respectively. A similar averaged result occurs across epochs, k. The distinction that these averages point to is that of the idea of water-filling [100]. The allocation of resources in the original metric behaves like normal water-filling, estimates with better variance get more

resources/ higher weighting. On the other hand, the independent utility function example in equation 3.15, D_k^{RWF}, behaves like reverse water-filling, putting more resources into higher variance estimates. Following this, the averaging methods above have the interpretation that D^A executes water-filling among the sensors and reverse water-filling across the epochs. On the other hand, D^H performs water-filling across sensors and epoch indices. These distinctions have been made because, while both averages are non-convex, the reverse-water filling average cannot be simplified. Whereas the harmonic average of utilities has a simplification which will be advantaged in a later chapter.

3.5 Summary

This chapter was meant to give the reader some footing regarding some of the models related to communication between nodes in WSN and the basic operation of the devices. The reader should note carefully the amenability of the implicit metrics of interest and that they are convex. Additionally, we briefly covered a few types of utilities, how they relate to our various metrics, and which of these induce a more cooperative objective. The next chapter will introduce the battery only formulation of the WSN energy management and optimization in this work.

Chapter 4

Joint Resource Management in Wireless Sensor Networks: Target Tracking Case Study

We finally tackle the problem of optimizing the quantization and communication systems in a network of wireless sensors. We draw heavily on the models of the previous chapter, as well on on the methods of Chapter 2. We present the specific system model as well as the metric used, and develop an optimization framework for our system. In this chapter, we first consider the network paradigm of centralized estimation. We investigate various modifications to the original problem formulation including worst case simplification, energy-awareness, and a dual-like problem formulation where energy usage is minimized.

In the following, Section 4.1 details the setup of the problem of interest while Section 4.2 chronicles the formulation of the optimization problem and introduces the heuristic scaling parameter. As simulation results in Section 4.4 demonstrate, the lifetime of the WSN is extended with the inclusion this heuristic, maintaining some menial loss of estimation performance. The trade-off between consistent estimation accuracy and network lifetime is investigated using the heuristic scaling developed herein. We show that network lifetime can be extended to over 250% of the original by an appropriate choice of the heuristic scaling parameter.

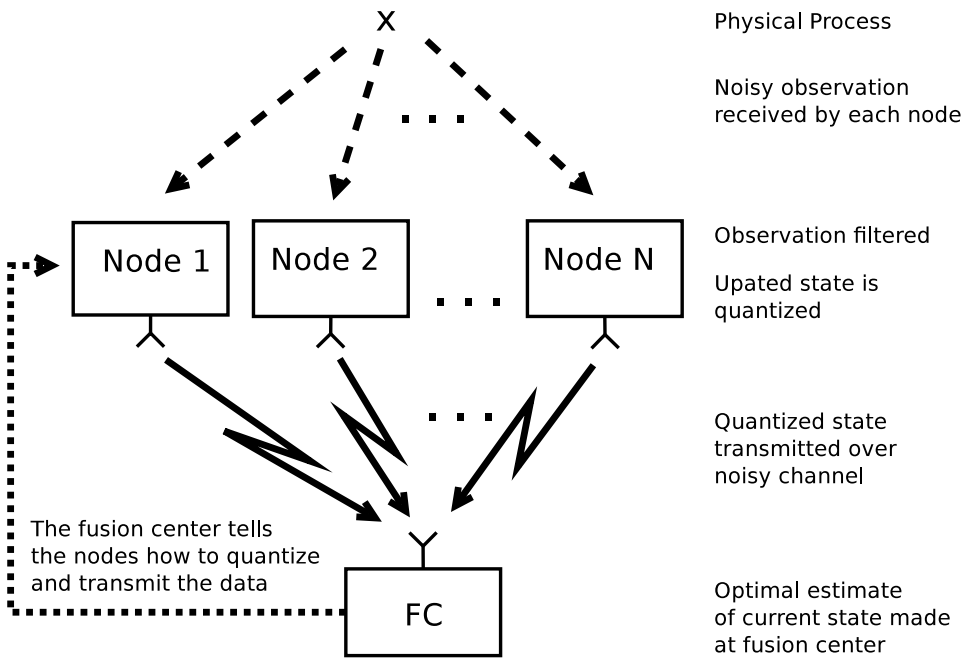


Figure 4.1: *An illustration of the specified distributed estimation system in which sensor nodes send Kalman filter updates.*

4.1 Centralized WSN Model

The considered paradigm contains multiple remote sensor nodes with power and bandwidth constraints which filter state estimates and transmit them to a fusion center for processing, as illustrated in Figure 4.1. It is the responsibility of the fusion node to instruct each node on how to send its update at each time instance via some orthogonal signaling scheme[1]. It is assumed that the fusion node is not energy constrained in its transmissions and that the energy required for receiving a transmission at a sensor node is negligible. As power is not constrained at the fusion node we assume practically noiseless feedback. Parameters which the fusion node uses to control the optimal reporting strategy or that it must take into account include: the accuracy of the state estimate of each node, the level of quantization, any effects of the noisy communications channel, as well as the remaining transmit energy (battery life) of each node. We consider a WSN with N spatially distributed nodes. These nodes take measurements of a dynamic process and update their local state estimates. The state estimate is quantized and transmitted to the fusion node, which estimates the state from the received reports.

4.1.1 Sensor Level Kalman Filtering

In these systems, the generalized state space model for the n^{th} sensor is of the form

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{w}_{k+1}, k) \tag{4.1}$$

$$\mathbf{z}_{nk} = \mathbf{h}_n(\mathbf{x}_k, k) + \mathbf{v}_{nk} \tag{4.2}$$

where $\mathbf{x}_k$ is the d-dimensional state of the true system at time instance k. The functions $\mathbf{f}$ and $\mathbf{h}_n$ are the generalized state transition and observation functions. The process noise vector $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ is assumed to be due to disturbances and modeling errors, $\mathbf{z}_{nk}$ is the observation vector and $\mathbf{v}_{nk} \sim \mathcal{N}(0, \mathbf{R}_{nk})$ is the measurement noise at node n, where $\mathbf{Q}_k$ and $\mathbf{R}_{nk}$ represent the process and measurement noise covariance matrices, respectively. We

[1] For example, if FDMA is used then a fixed total bandwidth is divided among the nodes. Thus, our method can be thought of as an adaptive bandwidth assignment FDMA scheme.

make the normal simplifying assumptions about the noise processes being zero mean, white, and uncorrelated. The goal for each local sensor is to estimate $\mathbf{x}_k$. Please see Section 2.3 for more information about the Kalman filter.

In this chapter, the state transition function modeled in the simulations of Section 4.4 is "nearly constant velocity" (NCV) propagation. The measurement model is a simple position-only observation function. The next section details the estimation of the state from the received Kalman updates.

4.1.2 Optimal Estimation from Kalman Updates

After a sensor node has measured the dynamical process and updated its local estimate, it will have obtained a state vector and covariance pair, $\{\hat{\mathbf{x}}_{n,k|k}, \mathbf{P}_{n,k|k}\}$. In practice, both the state estimate and its covariance matrix would be quantized and transmitted to the fusion node where a final estimate is formed. However, to maintain simplicity in presentation, we study the effect of quantizing and transmitting the state information while assuming that the fusion center has error-free knowledge of the covariance structure; this is a common assumption adopted by other authors [18]. The extension of the analysis to additionally transmitting the covariance structure simply involves sending more data (corruptible by quantization and transmission). Instead, we evaluate our approach by testing the sensitivity of it to the knowledge of the covariance information. This is done by adding a random perturbation to the covariance information and executing the forthcoming methods, as shown in Section 4.4.4. Given the state estimate at the node, the state vector is then quantized as

$$\check{\mathbf{x}}_{n,k|k} = \hat{\mathbf{x}}_{n,k|k} + \mathbf{n}^q_{nk} \;\forall\; n = 1, \ldots, N \tag{4.3}$$

where $\mathbf{n}^q_{nk} \sim \mathcal{N}(0, \mathbf{R}^q_{nk})$ is the quantization noise of sensor node n at time k, with $\mathbf{R}^q_{nk}$ the quantization noise covariance. The quantized data is mapped to a bit stream or other form suitable for transmission. Each bit of the data stream consisting of b_n bits is transmitted independently (by means of some orthogonal signaling scheme) through noisy wireless fading

channels to the fusion node. The final information received at the sensor fusion node is

$$\begin{aligned}\tilde{\mathbf{x}}_{n,k|k} &= \check{\mathbf{x}}_{n,k|k} + \mathbf{n}_{nk}^{c} \\ &= \hat{\mathbf{x}}_{n,k|k} + \mathbf{n}_{nk}^{q} + \mathbf{n}_{nk}^{c}, \end{aligned} \tag{4.4}$$

$\forall\ n = 1, \ldots, N$, where $\mathbf{n}_{nk}^{c} \sim \mathcal{N}(0, \mathbf{R}_{nk}^{c})$ is the channel noise of the n^{th} sensor at time k due to imperfect communication, with $\mathbf{R}_{nk}^{c}$ the channel noise covariance. We make the standard assumption that the internal noise of the state estimate, and the quantization and channel noises are all uncorrelated, since each of these noises could be considered as coming from independent sources. We justify this partition of the estimation process by the following principle.

Lemma 6 (Separation Principle)**.** *The final optimal estimate, $\tilde{\mathbf{x}} \in \mathbb{R}^{d\times 1}$, in the presence of random disturbance can be obtained by first finding the optimal estimate of the quantity of direct interest ($\hat{\mathbf{x}}$) to the user and then by communicating this estimate over the channel as if it were disturbance-free. The resulting mean-squared-error has the following form,*

$$D = \mathbf{E}||(\tilde{\mathbf{x}} - \hat{\mathbf{x}})||^2 + \mathbf{E}||(\hat{\mathbf{x}} - \mathbf{x})||^2,$$

where $\mathbf{E}[\cdot]$ denotes the expectation operation.

Proof. The proof of this is due to Sakrison [101, 102], but the principle is extended in [103, 104] for forming estimates of states in sensor networks. □

At the fusion node, received state updates, affected by various noise sources, are combined linearly to form an estimate of the actual process state. The predecessor of this estimation paradigm was introduced in [105] where scalar measurements of a deterministic source are used to make a single estimate at the fusion node. The issue of combining Kalman states by filtering is complicated by common process and measurement noise [106] of the sequential state estimate reports from a particular sensor. The methods typically employed are often practically infeasible [107] and it should be noted that they are avoided

in our work by simply estimating the current state from the only the most recently reported state estimates. We extend the simple scalar estimator of our previous work in [67] to send the elements of a vector state. In this case we consider each element of the state vector independently and this results in the centralized BLU-like estimator of the elements of $\mathbf{x}_k$ which are $x_k^i,\ i = 1, \ldots, d$. The BLUE at the fusion node represents a single-step control strategy (see Section 2.4) which determines an optimal number of bits and transmitting power for each of the sensor nodes in order to minimize the total variance of the elements of the state vector it is estimating.

We extend the simple scalar estimator in order to send the elements of a vector state. The BLUE method for vector estimates (cf. [86]) assumes the "measurements" (or reports in this case) of the true state are of the form

$$\tilde{\mathbb{X}}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{U}_k \tag{4.5}$$

where $\mathbf{H}_k$ is a linear combining matrix and $\mathbf{U}_k \sim \mathcal{N}(\mathbf{0}, \mathbb{P}_k)$ represents additive noise. In the notation we have given so far

$$\begin{aligned} \tilde{\mathbb{X}}_k &= [\tilde{\mathbf{x}}_{1,k|k}^T\ \tilde{\mathbf{x}}_{2,k|k}^T\ \cdots\ \tilde{\mathbf{x}}_{N,k|k}^T]^T \\ \mathbf{H}_k &= [\mathbf{I}_d \mid \mathbf{I}_d \mid \cdots \mid \mathbf{I}_d]_{d \times Nd}^T \\ \mathbf{U}_k &= [\mathbf{u}_{1,k}^T\ \mathbf{u}_{2,k}^T\ \cdots\ \mathbf{u}_{N,k}^T]^T \end{aligned} \tag{4.6}$$

where $\mathbf{I}_d$ is an identity matrix of size $d \times d$. To maintain unbiasedness, the linear combining matrix $\mathbf{W}_k$ must satisfy $\mathbf{W}_k \mathbf{H}_k = \mathbf{I}$. The resulting vector BLU estimate is given by

$$\begin{aligned} \hat{\mathbf{x}}_{BLUE,k} &= \mathbf{W}_k \tilde{\mathbb{X}}_k \\ &= [\mathbf{H}_k^T \mathbb{P}_k^{-1} \mathbf{H}_k]^{-1} \mathbf{H}_k^T \mathbb{P}_k^{-1} \tilde{\mathbb{X}}_k \end{aligned} \tag{4.7}$$

where $\mathbb{P}_k$ is a composite covariance matrix represented as

$$\mathbb{P}_k = \text{diag}\{[\mathbf{P}_{1,k|k} + \mathbf{R}_1^c + \mathbf{R}_1^q], \ldots, [\mathbf{P}_{N,k|k} + \mathbf{R}_N^c + \mathbf{R}_N^q]\}, \tag{4.8}$$

i.e., block diagonal, for the case where the individual reports are uncorrelated[2]. We define $\mathbf{P}^{(i)}_{n,k|k}$ as the variance of the i^{th} element of the update vector at time k from sensor node n. We assume these noise processes are uncorrelated in time, as well as spatially across vector elements. Noting the lack of a time index, this is reflected by $\mathbf{R}^q_n = E[\mathbf{n}^q_n \mathbf{n}^{q\,H}_n] = \sigma^2_q \mathbf{I}_d$, $\mathbf{R}^c_n = E[\mathbf{n}^c_n \mathbf{n}^{c\,H}_n] = \sigma^2_c \mathbf{I}_d \;\forall\; i = 1, \ldots, d,\; n = 1, \ldots, N$, where $E[\cdot]$ denotes the expectation operation. We represent the scalar variance terms as $r^{q,i}_{nk} = [\mathbf{R}^q_{nk}]_{i,i}$ and $r^{c,i}_{nk} = [\mathbf{R}^c_{nk}]_{i,i} \;\forall\; i = 1, \ldots, d$, which are the variances of the elements of the noise vectors. Note that the channel and quantization noise variances are functions of the power transmission level and bits used for quantization.

We use the mean squared error associated with this BLU estimator (also the variance), denoted D, as the metric of uncertainty to be minimized. The trace of the BLU estimate error covariance is taken to obtain a scalar quantity as

$$\begin{aligned} D_k &= \mathbf{tr}\left\{\mathrm{cov}(\hat{\mathbf{x}}_{BLUE,k} - \mathbf{x}_k)\right\} \\ &\overset{(a)}{=} \mathbf{tr}\left\{\left(\mathbf{H}^T_k \mathbb{P}^{-1}_k \mathbf{H}_k\right)^{-1}\right\} \\ &\overset{(b)}{=} \mathbf{tr}\left\{\left(\sum_{n=1}^{N} [\mathbf{P}_{n,k|k} + \mathbf{R}^c_n + \mathbf{R}^q_n]^{-1}\right)^{-1}\right\} \\ &\overset{(c)}{=} \sum_{i=1}^{d} \left(\sum_{n=1}^{N} \frac{1}{\mathbf{P}^{(i)}_{n,k|k} + r^{c,i}_{nk} + r^{q,i}_{nk}}\right)^{-1} \\ &= \sum_{i=1}^{d} D^i_k, \end{aligned} \tag{4.9}$$

where D^i_k is the BLUE error variance for the i^{th} element of the state estimate. The above definition of D_k follows since a) the second equality produces, by definition, the matrix form of the BLUE error covariance, b) the third equality holds when the reports from the nodes are uncorrelated, and c) the fourth equality is true when the noise terms are spatially uncorrelated. Since we have made these assumptions[2], (4.9) is our total uncertainty metric. Under these assumptions, the calculation of the optimal parameters for quantization and

[2]With the uncorrelated reports assumption, c), we recognize that not all process noise models possess the quality of being spatially uncorrelated. When not true, then the total uncertainty in (4.9) becomes an approximation.

transmission allow the elements of the state vector to be estimated individually as

$$\hat{x}^i_{BLUE} = \left(\sum_{n=1}^{N} \frac{1}{E[(\tilde{x}_{n,k|k} - x^i_k)^2]}\right)^{-1} \times \sum_{n=1}^{N} \frac{\tilde{x}_{n,k|k}}{E[(\tilde{x}_{n,k|k} - x^i_k)^2]}, \tag{4.10}$$

$\forall\ i = 1, \ldots, d$. Let $x_i \in [-W, W]$ with $[-W, W]$ the dynamic range of the measurement source. Then for a scalar element of the state vector,

$$r^{q,i}_{nk} = \frac{W^2}{3(2^{b^i_{nk}} - 1)^2} \tag{4.11}$$

is the uniform quantization noise variance. Each $b^i_{nk} \in [1, BW]$ is the number of bits used to quantize the i^{th} element of the state estimate from sensor n to the fusion node. BW is the rate constraint for the entire system. The quantization scheme is homogeneous across sensors for a quantization level, with a set dynamic range for components of position, velocity, et cetera. If the n^{th} sensor node communicates using BPSK modulation for a Rayleigh fading channel with a modulation scheme that produces a probability of error of $\Pr^{i,\ell}_{nk}$ for the ℓ^{th} bit of the transmission, then the noise due to the imperfect channel is

$$n^{c,i}_{nk} = \begin{cases} \pm 2^{\ell} \Delta^{i,\ell}_n & \Pr^{i,\ell}_{nk} \\ 0 & 1 - \sum_{\ell=0}^{b^i_{nk}-1} \Pr^{i,\ell}_{nk} \end{cases} \tag{4.12}$$

where $\Delta^{i,\ell}_{nk} = \frac{2W}{2^{b^i_{nk}-1}}$ is the quantizer step size. We now assume: (a) the bits in the transmit sequence have independently distributed probability of error, i.e., $\Pr^{i,\ell}_n = \Pr^i_{nk}$, (b) there is as most one bit error in each transmit sequence, and (c) the channel noise variance is unchanged during a transmission period. All of these are reasonable assumptions for a slow fading channel with adequate channel coding. Thus the noise variance contributed from the

channel is

$$\begin{aligned} r_{nk}^{c,i} &= \sum_{\ell=0}^{b_{nk}^{i}-1} (\pm 2^{\ell} \Delta_n^{i,\ell})^2 \Pr_{nk}^{i,\ell} \\ &= \Pr_{nk}^{i} \times (\Delta_n^{i,\ell})^2 \sum_{\ell=0}^{b_{nk}^{i}-1} 4^{\ell} \\ &\approx \frac{4W^2}{3} \Pr_{nk}^{i}, \end{aligned}$$

which simplifies to

$$r_{nk}^{c,i} \approx \frac{4W^2}{3} \left(1 - \sqrt{\frac{0.5\Gamma_n^i}{1 + 0.5\Gamma_n^i}} \right). \tag{4.13}$$

Where $\Gamma_n^i = \frac{2p_n^i \overline{h_n}}{N_0}$ represents the average received signal-to-noise-ratio, $p_n^i \in [p_n^{min}\ p_n^{max}]$ is the transmit energy level for the i^{th} element of the n^{th} sensor. The minimum power level per bit is p_n^{min} which is necessary to achieve a minimum system SNR. The maximum power per bit in a transmission is p_n^{max}. The power level p_n^i considers only the radio-frequency (RF) power required at the node, and none of the power consumed by other circuits in the device, which are considered negligible for simplicity in our analysis. The average power of the Rayleigh fading channel coefficient is $\overline{h_n}$ and $N_0/2$ is the channel noise power spectral density.

The previous discussion has introduced an objective for distributed estimation, which is optimal in the sense of minimizing an uncertainty metric which is a function of the variance of the estimate. The next section nestles this objective into the formal optimization statement which includes the constraints on the control variables.

4.2 Optimization of Estimation Formulation

It is desirable that optimization of quantization and transmit energy levels produce a balanced trade-off between estimate uncertainty and network lifetime. This trade-off is the subject of the following discussion. The optimization problem initially considers only the minimization of estimate variance under the given constraints, while lifetime is considered

when we discuss energy-aware optimization. Assuming a subset of sensors has already been selected, we want to find the optimal of number of bits and transmit power levels that produce the best linear unbiased estimate of the process state, given the maximum resources allowed to be utilized. We perform the optimization of the following method for all time steps in the scenario or until the network has no active nodes (non-zero remaining energy). The formal expression of the minimization problem for a single dimension state vector is

$$\begin{aligned} &\text{minimize } D_k \\ &\quad \text{subject to} \\ &\quad \textstyle\sum_{n=1}^{N} b_{nk} \leq BW \\ &\quad \Lambda_{nk} p_{nk} b_{nk} \leq p_{nk}^{rem} \\ &\quad -b_{nk} + 1 \leq 0 \\ &\quad p_{nk} - p_n^{max} \leq 0,\ p_n^{min} - p_{nk} \leq 0 \\ &\quad \forall\ n = 1, \ldots, N. \end{aligned} \tag{4.14}$$

Equivalently, the objective can be written as

$$minimize\ \ -D_k^{-1},$$

which results in a simpler objective function (by removing the inverse operating on the sum of inverted variance terms). The total power resources expended by each node is $p_{nk}b_{nk}$. Here p_{nk} and b_{nk} denote the power and bits used by the n^{th} sensor node at the k^{th} time instance; $\Lambda_{nk} \in [1, \frac{1}{\alpha}]$ is the weighting parameter, with fixed α a frugality parameter. Λ_{nk} is best defined as a weighting that reflects the resource policy of each sensor based on its operating state, i.e., its remaining energy in the battery (p_n^{rem} in the above formulation). The weighting adjustment parameter, α, determines how Λ_{nk} is updated. Low battery power would result in a large value for Λ_{nk} and vice versa, the role of Λ_{nk} is discussed next in Section 4.3.2. The requirement that every node transmit at least one bit (the $-b_{nk} + 1 \leq 0$ constraint) reflects the status that selection and scheduling has already happened. The maximum and minimum constraints defined for the p_{nk}^i and b_{nk}^i henceforth shall be referred to as "box-constraints". The above problem must be expanded to account for the information transmitted for each of the elements of a multi-dimensional state vector. Thus, the altered form for vector quantities

is written as

$$\begin{aligned}
&\text{minimize } -\textstyle\sum_{i=1}^{d} {D_k^i}^{-1} \\
&\quad \text{subject to} \\
&\quad \text{C1}: \ \textstyle\sum_{i=1}^{d}\sum_{n=1}^{N} b_{nk}^i \leq BW \\
&\quad \text{C2}: \ \textstyle\sum_{i=1}^{d} \Lambda_{nk}^i p_{nk}^i b_{nk}^i \leq p_{nk}^{rem} \\
&\quad \text{C3}: \ -b_{nk}^i + 1 \leq 0 \\
&\quad \text{C4}: \ p_{nk}^i - p_n^{max} \leq 0 \\
&\quad \text{C5}: \ p_n^{min} - p_{nk}^i \leq 0, \\
&\quad \forall\ n = 1, \ldots, N,\ i = 1, \ldots, d.
\end{aligned} \tag{4.15}$$

This formulation is by nature non-convex in the variables p_n^i and b_n^i, and in reality is a mixed-integer non-linear program (MINLP) with respect to the discrete values of the bits. It is known that solving a MINLP is NP-Hard [108]. We convert (4.15) to a difference of convex (DC) functions problem and solve the relaxed epigraph version of the problem by introducing new "uncertainty" variables $u_{nk}^i,\ n = 1, \ldots, N,\ i = 1, \ldots, d$, where

$$u_{nk}^i = \frac{1}{\mathbf{P}_{n,k|k}^{(i)} + r_{nk}^{c,i} + r_{nk}^{q,i}},$$

each of which is a scalar quantity. Recalling the dependency of the channel and quantization noise on the power level and number of bits variables, the new epigraph form of the optimization problem is

$$\begin{aligned}
&\text{minimize } -\textstyle\sum_{i=1}^{d}\sum_{n=1}^{N} u_{nk}^i \\
&\quad \text{subject to} \\
&\quad u_{nk}^i - \frac{1}{\mathbf{P}_{n,k|k}^{(i)} + r_{nk}^{c,i} + r_{nk}^{q,i}} = 0 \\
&\quad \forall\ n = 1, \ldots, N,\ i = 1, \ldots, d,
\end{aligned} \tag{4.16}$$

in addition to constraints C1-C5. We make additional simplifications to the uncertainty constraint by rewriting it as

$$\mathbf{P}_{n,k|k}^{(i)} + r_{nk}^{c,i} + r_{nk}^{q,i} - \frac{1}{u_{nk}^i} \leq 0$$

(noting that the function decreases in u_{nk}^i while the equality constraint increases with respect to it, the substitution is therefore adequate since the inequality introduced is strictly active at the minimum). This form of the constraint however, contains a convex function of power and bits, and concave function of the introduced uncertainty variables (i.e., the u's). A

first order Taylor approximation of the concave reciprocal uncertainty term can be used to transform the difference of convex (DC) functions constraint into an approximate convex constraint. Thus the final convex approximation formulation (abbreviated as CVX) is

$$\begin{array}{l} \text{minimize } -\sum_{i=1}^{d}\sum_{n=1}^{N} u_{nk}^{i} \\ \text{subject to} \\ \quad \mathbf{P}_{n,k|k}^{(i)} + r_{nk}^{c,i} + r_{nk}^{q,i} - \left(\frac{2\tilde{u}_{nk}^{i} - u_{nk}^{i}}{(\tilde{u}_{nk}^{i})^{2}}\right) \leq 0 \\ \quad \forall\ n = 1, \ldots, N,\ i = 1, \ldots, d, \end{array} \tag{4.17}$$

still subject to constraints C1-C5, where $\tilde{u}_{nk}^{i}$ is the iterated point about which the Taylor approximation is taken. One question which can arise with respect to this convex formulation, is whether the above is convex with respect to the product of two of the decision variables. In general, convexity of a function which is the product of convex functions cannot be assumed. However, we have the following.

Lemma 7. *If two functions, f and g are convex, positive, and non-decreasing, then the resulting product of functions, $h = f \cdot g$, is convex.*

This Lemma is illustrated in Figure 4.2. In our case we can consider two linear functions (linear functions are convex) of the power and bit variables, let $f(p) = p$ and $g(b) = b$, and their product $h = f \cdot g = p \cdot b$. This holds since in our formulation the variables are always constrained to be positive. (See [38], pages 84 and 119 (problem 3.32).)

We use (4.17) and sequential convex programming (SCP) [38] iterations find a stable upper bound to the original non-convex MINLP. Using SCP to obtain an approximation to the DC program comes attached to an increased computational effort as we must execute $O(nm^2)$ operations to solve a SQP at each iteration (where m is the number of constraints and n is the number of variables and $n \leq m$). A total computational cost of $O(nm^2L)$ results, where L is the number of SCP iterations. Since the KKT analysis does not provide any additional useful information concerning the problem behavior [105], we use simulation results to quantify the estimation performance of our relaxed convex approximation. Although this Taylor approximation is convex, the formulation experiences convergence issues

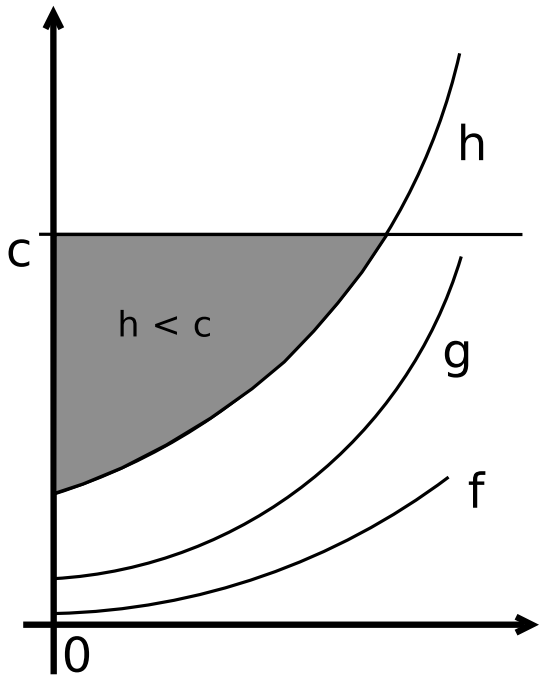


Figure 4.2: *Illustration of the convex hull of the constraint space produced by the product of convex functions when they are positive and non-decreasing. Here the region $h = f \cdot g < c$ non-empty implies that it is convex.*

because of the curvature of reciprocal function in the additional constraint. Next we investigate the use of a linear function in the added constraint to form a convex approximation which bypasses the need for SCP.

4.2.1 Low Complexity Formulation

A simpler formulation that matches this problem and circumvents sequential convex programming is presented. If, for a moment, we relax some of the notation for the sake of clarity in explanation, consider the objective obtained above (4.17): Minimizing the negative of the sum of the uncertainty terms implies minimizing each individually. The statement

$$\begin{aligned}(min\ -u)\ \&\ (P\ +\ r^q + r^c \leq \frac{1}{u}) \\ \Rightarrow max\ u \quad \rightarrow \quad min\ \{P + r^q + r^c\}\end{aligned} \tag{4.18}$$

maintains the truthfulness of what the substituted variable is meant to force. Now a parallel is drawn for a different substitutionary variable, y, playing an inverse role to u. The objective is simply to minimize positive uncertainty (or $-\frac{1}{u}$ in the current notation).

$$\begin{aligned}(min\ y)\ \&\ (P\ +\ r^q + r^c \leq y) \\ \Rightarrow min\ y \quad \rightarrow \quad min\ \{P + r^q + r^c\}\end{aligned} \tag{4.19}$$

which has the same resulting effect as the reciprocal substituted term. Thus u corresponds to inverse uncertainty in (4.18), and y simply to uncertainty in (4.19). This would now imply that the new uncertainty variable should be

$$y^i_{nk} = \mathbf{P}^{(i)}_{n,k|k} + r^{c,i}_{nk} + r^{q,i}_{nk}. \tag{4.20}$$

To draw out a little more intuition, we plot the constraints involving the substitutionary variables for the two formulations. Maximizing the u's will reduce the final objective of the original non-convex problem written as $(\sum_n u^i_n)^{-1}$ is minimized for maximum u^i_n. Similarly, the y's represent the noise variance directly (and not the reciprocal) which will relates the original objective when written as $(\sum_n \frac{1}{y^i_n})^{-1}$, which is minimized when y^i_n's are minimum.

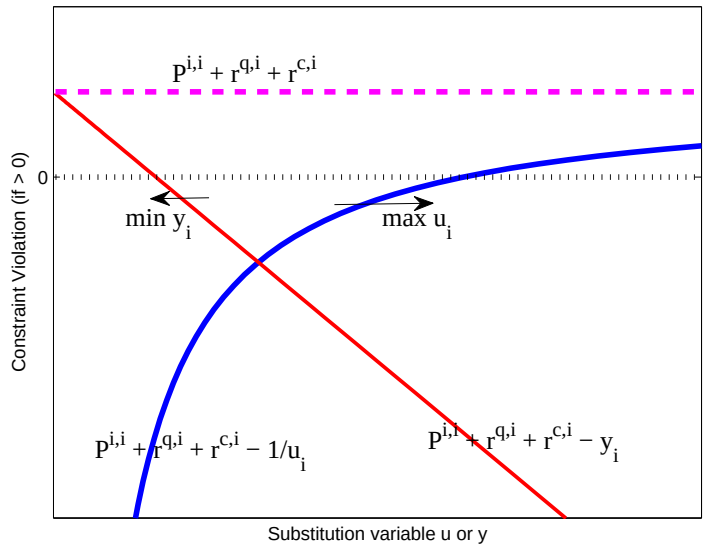


Figure 4.3: *An illustration of how the value produced by a negative linear function minimized produces a reciprocal value corresponding to the negative reciprocal function.*

Figure 4.3 illustrates that the graph of the two functions pursue reciprocal values of each other and that u and y are both valid substitutionary variables for the approximation which is the convexification of the problem. After inserting all of the notable changes above, the final problem formulation is

$$\begin{aligned} &\text{minimize } \textstyle\sum_{i=1}^{d}\sum_{n=1}^{N} y_{nk}^{i} \\ &\text{subject to} \\ &\quad \mathbf{P}_{n,k|k}^{(i)} + r_{nk}^{c,i} + r_{nk}^{q,i} - y_{nk}^{i} \leq 0 \\ &\quad \forall\ n = 1,\ldots,N,\ i = 1,\ldots,d, \end{aligned} \tag{4.21}$$

again still subject to constraints C1-C5. We call this new formulation in (4.21) the Linear Constraint Convex Approximation (LCVX) of the MINLP.

There are some convergence issues associated with the reciprocal constraint convex form (CVX). These issues can be mediated by tightening the tolerances which dictate the stopping criterion for the optimization routine. In addition to this, because the slope of the first-order Taylor approximation is dependent on the noise variance offset in the constraint, the difficulties associated with convergence vary from instance to instance so that in many

cases the reciprocal constraint convex approximation achieves equal results than linear constraint convex form. However, the swift convergence of linear constraint form provides the equivalent solution with more consistency. In some sense, these comparisons are a moot point since both approximations equally form convex upper bounds to the original problem. For the sake of comparison, the value of the objective functions of the previous convex approximations are compared in Section 4.4.1 with the same objective evaluated with integer bits (which are rounded or floored to satisfy the bandwidth constraint) from the continuous optimized values. Thus the relaxed convex approximations can be compared with the original mixed-integer non-convex problem (which is solved using a branch and bound based global solver). Next, the above approximations are further reduced so that the number of variables is decreased and the solutions to such simplifications provide a rapid upper bound to the original MINLP.

4.2.2 Worst-Case Formulation

In the present convex formulations, there are d bit and transmit energy parameters for each sensor node. It is desirable to formulate this problem with an objective function D which accounts for a single quantization and transmit energy parameter for each of the sensor nodes. The reduction of these parameters is done as follows. In the above (LCVX) formulation there exist a choice of the number of bits and power transmission level for each element of the state update for each node. If we consider a system which chooses the number of bits and power transmission level once for all elements of the state vector, the we reduce the number of needed variables by $2N(d-1)$. This transformation of the optimization problem is akin to requiring worst-case satisfaction of the chosen solution, i.e., only one of the uncertainty constraints will be active. An additional $N(d-1)$ variables can be removed from the problem by utilizing a single constraint with respect to the uncertainty term per sensor node, instead of d of them. This form is easily obtained by taking the maximum of the state covariances for all state elements, the constraints now include only one bit (p_{nk}),

Table 4.1: *Convex approximation algorithmic complexity using convex and sequential convex programming. L is the number of sequential convex programming iterations.*

	WC	**LCVX**	**CVX**
Variables	$2N$	$2Nd$	$3Nd$
Constraints	$1+7N$	$1+N(1+6d)$	
Runtime	$O(N^3)$	$O(N^3d^3)$	$O(N^3d^3L)$

transmit energy (b_{nk}), and uncertainty (y_{nk}) variable per sensor node. This reduced variable "worst-case" problem is written as

$$\begin{aligned} &\text{minimize } d\textstyle\sum_{n=1}^{N} y_k^n \\ &\quad \text{subject to} \\ &\quad \max_i\{\mathbf{P}_{n,k|k}^{(i)}\} + r_{nk}^c + r_{nk}^q - y_k^n \leq 0 \\ &\quad d\textstyle\sum_{n=1}^{N} b_k^n \leq BW \\ &\quad d\Lambda_k^n p_k^n b_k^n \leq p_{nk}^{rem} \\ &\quad \forall\ n = 1, \ldots, N. \end{aligned} \tag{4.22}$$

This representation of the convex program has a significant difference from the previous (LCVX) form: the uncertainty value being minimized is now the dimension of the state vector times the worst MSE term of that state estimate. Thus, the element of the state update vector with the largest variance determines the quantization and power levels for all of the elements of the state estimate for that node. Equivalently, for that node this determines the number of bits all data are encoded with and the transmission power level used. This formulation upper bounds the original MINLP as well.

Continuing the analysis of this simplification we have broken down the number of variables and constraints needed to evaluate the preceding problems. The formulation in equation (4.21) is the linear constraint convex formulation (LCVX) derived alongside the reciprocal constraint convex approximation (CVX) in (4.17). Both of these forms use the full number of variables and constraints (with the added uncertainty constraints) from the original problem. The form in (4.30) is the worst-case (WC) upper bound solution (worst-case in the sense of the sensor node state estimate covariances), and simplifies the problem by

using a single bit and transmit energy variable and the maximum covariance value of any of the elements of an estimate at a particular node, thus reducing the number of variables and constraints. Table I shows the number of variables and their respective number of constraints given N sensors, state vector dimension d, and L SCP iterations. From these approximations we can glean the fact that the worst-case upper bound will be considerably more efficient when optimizing wireless sensor networks with a large number of nodes, which might be running Kalman estimators for two or three dimensions with higher order terms (velocity or acceleration of states). Number of variables and constraints for different convex approximate formulations of the optimal power-quantization problem. N is the number of sensors, d is the dimension of the state vector, and L is the number of SCP iterations

4.2.3 Energy-Aware Optimization

The above formulations attempt to reduce the variance of the fusion node estimate of the state by minimizing over uncertainty as the free variable. Under this paradigm the system will blindly use resources at each time iteration without consideration of the need for future transmissions, for either the sensor locally or its neighbors in the network. It was also noted that the behavior of this configuration is unchanging for differing channel or measurement noises between sensors, but rather it continues in the presence of unequal noise levels to follow the strategy of maximizing the transmit energy level and bandwidth per transmission, within the permissibility of the constraints. However, when the remaining energy for a subset of the nodes is significantly disadvantaged from the rest of the network, a reduction in the number of active sensors may occur prematurely, i.e., node batteries are depleted. The results section will amply demonstrate this. As mentioned at the beginning of this section, we want to develop a strategy for preserving the energy of each node and thereby prolong the lifetime of the network. We use a heuristic scaling to alter the way in which a disadvantaged (in remaining energy) node is constrained in energy usage. A weighting function, $L(p_{nk}^{rem}, p^{init}, k)$, is used to decide the level of transmit energy frugality. This

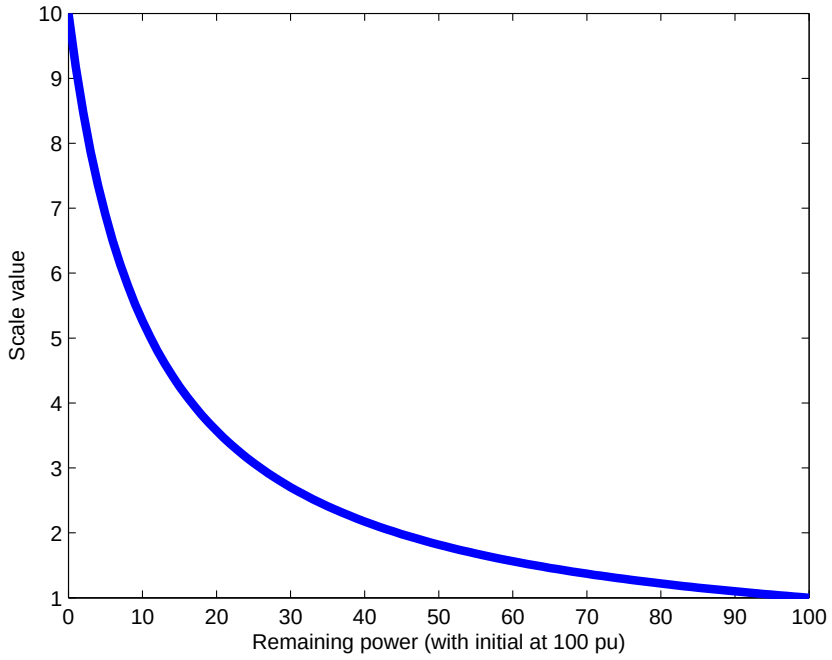


Figure 4.4: *The energy-aware scaling as a function of remaining energy.*

preserves the battery life of nodes with a dwindling operating states. A possible schema for this scaling function, currently implemented, is

$$\Lambda_{nk} = L(p_{nk}^{rem}, p^{init}, k) = \frac{1}{\alpha + (1-\alpha) \cdot \frac{p_{nk}^{rem}}{p^{init}}} \tag{4.23}$$

$\forall\ n = 1, \ldots, N,\ \forall\ k$, where p_n^{rem} is the remaining power at node n while p^{init} is the average initial power allocated to each node. These weights are updated at each iteration and used to create the constraint

$$\Lambda_{nk} \cdot \sum_i p_{nk}^i b_{nk}^i < p_{nk}^{rem} \tag{4.24}$$

$\forall\ k, n$, where p_n and b_n are the power per bit and number of bits used in the current transmission interval. This is used in place of the previous remaining energy constraint (C2). A plot of the scaling value versus current energy level for $\alpha = 0.1$ and $p^{init} = 100$ is shown in Figure 4.4. This illustrates how a lower remaining power will result in a more frugal energy policy. A difficultly which becomes apparent is how to gracefully let a node

deplete its energy. The constraints of the problem

$$p^{min} > p^i_{nk} \text{ and } \sum_i p^i_{nk} b^i_{nk} \Lambda_{nk} < p^{rem}_{nk} \tag{4.25}$$

are conflicting for low remaining energy with a large value of $\Lambda_{nk} > 1$, which causes the problem to become infeasible. Thus, it was determined that if

$$p^{min}_n \cdot d \cdot \Lambda_n > p^{rem}_n$$

then Λ_n is set to 1 and node n is allowed to be depleted. This feature is plainly visible in Section 4.4.2. The next section details the simulation of the above optimization problem with and without this heuristic scaling procedure.

4.3 Optimization of Energy Formulation

In this section, we formulate the distributed estimation problem so that energy usage is minimized. However, the final estimate variance is constrained to be beneath a maximum threshold. This approach is meant to extend the life of the WSN while maintaining adequate estimation performance, this constraint is on the uncertainty term in (4.9) for the vector estimate. Assuming a subset of sensors has already been selected, we want to find the bandwidth assignment and transmit power levels that produce the best linear estimator of the process state, with the given resource constraints.

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{d} \sum_{n=1}^{N} p^i_{nk} b^i_{nk} \\ \text{subject to} & \\ \text{C1:} & D_k \leq \Delta \\ \text{C2:} & \sum_{i=1}^{d} \sum_{n=1}^{N} b^i_{nk} \leq BW \\ \text{C3:} & \sum_{i=1}^{d} p^i_{nk} b^i_{nk} \leq p^{rem}_{nk} \\ \text{C4:} & -b^i_{nk} + 1 \leq 0 \\ \text{C5:} & p^i_{nk} - p^{max}_n \leq 0, \\ \text{C6:} & -p^i_{nk} + p^{min}_n \leq 0 \end{array} \tag{4.26}$$

$\forall\ n = 1, \ldots, N,\ i = 1, \ldots, d$, with δ the estimate variance/uncertainty threshold. The total power resources expended by node n at time k is $p_{nk} b_{nk}$ and p^{rem}_{nk} denotes the remaining power. The BW quantity represents the total rate for the network, with the requirement

that every node transmit at least one bit (constraint C4 defined next). Henceforth the maximum and minimum constraints defined here shall be referred to as "box-constraints". This formulation is inherently non-convex in the variables p_n^i and b_n^i, and in reality is a Mixed-Integer Non-Linear Program (MINLP) with respect to the discrete values of the bits.

We transform this to a "differences of convex functions" problem by solving the relaxed epigraph version of the problem. We first expand constraint C1 by letting each of the error variances for the elements of the estimate be constrained separately, thus having d constraints on uncertainty instead of one,

$$D_k^i \leq \delta_i \ \forall \ i = 1, \ldots, d, \tag{4.27}$$

with $\Delta = \sum_{i=1}^{d} \delta_d$. We then introduce new "uncertainty" variables $u_{nk}^i, \ n = 1, \ldots, N, \ i = 1, \ldots, d$, where $u_{nk}^i = \frac{1}{\mathbf{P}_{n,k|k}^{(i)} + r_{nk}^{c,i} + r_{nk}^{q,i}}$, each of which is a scalar quantity. For now we assume that the uncertainty thresholds are uniform, i.e., $\delta_i = \delta \ \forall \ i$. The uncertainty constraints can be trivially rewritten in terms of the uncertainty variables. Recalling the dependency of the channel and quantization noise on the power level and number of bits, the new epigraph form of the optimization problem is

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{i=1}^{d} \sum_{n=1}^{N} p_{nk}^i b_{nk}^i \\ \text{subject to} \quad & \\ & \tfrac{1}{\delta_i} - \textstyle\sum_{n=1}^{N} u_{nk}^i \leq 0 \\ & u_{nk}^i - \frac{1}{\mathbf{P}_{n,k|k}^{(i)} + r_{nk}^{c,i} + r_{nk}^{q,i}} = 0 \end{aligned} \tag{4.28}$$

$\forall \ n = 1, \ldots, N, \ i = 1, \ldots, d$, constraints C2-C6 still in effect. We make additional simplifications to the uncertainty constraint by rewriting it as

$$\mathbf{P}_{n,k|k}^{(i)} + r_{nk}^{c,i} + r_{nk}^{q,i} - \frac{1}{u_{nk}^i} \leq 0.$$

(noting that u_{nk}^i is constrained from below and decreases with a decreasing objective while the equality constraint increases with respect to it, the substitution is therefore adequate since the inequality introduced is strictly active at the minimum). This form of the constraint however, contains a convex function of power and bits, and concave function of the introduced uncertainty variables. We utilize a first order Taylor approximation of the concave

reciprocal uncertainty term to transform the "difference of convex functions" constraint into a approximate convex constraint. Thus the final convex approximation formulation (CVX) is

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{d}\sum_{n=1}^{N} p_{nk}^{i} b_{nk}^{i} \\ \text{subject to} & \\ & \frac{1}{\delta_i} - \sum_{n=1}^{N} u_{nk}^{i} \leq 0 \\ & \mathbf{P}_{n,k|k}^{(i)} + r_{nk}^{c,i} + r_{nk}^{q,i} - \left(\frac{2\tilde{u}_{nk}^{i} - u_{nk}^{i}}{(\tilde{u}_{nk}^{i})^2}\right) \leq 0 \end{array} \tag{4.29}$$

$\forall\ n = 1, \ldots, N,\ i = 1, \ldots, d$, with continued subjection to constraints C2-C6, where $\tilde{u}_{nk}^{i}$ is the iterated point about which the Taylor approximation is taken. Using Sequential Convex Programming (SCP) we quickly find a suitable approximation to the original non-convex MINLP. Using SCP to obtain an approximation to the DC program comes attached to a computational effort of $O(nm^2)$ operations to solve the SQP at each iteration (where m is the number of constraints and n is the number of variables and $n \leq m$). KKT analysis does not provide any additional intuition concerning the problem behavior [109], so we use simulation results to quantify the estimation performance of our relaxed convex approximation.

There are occasionally some convergence issues associated with this epigraph constraint convex form (CVX). These issues can be mediated by tightening the tolerances which dictate the stopping criterion for the optimization routine. However, the approach can quickly reach the relative floating point accuracy of the numeric solution. This is largely because the slope of the first-order Taylor approximation is dependent on the noise variance offset in the uncertainty variable constraint. Next, we discuss how the above approximation is reduced so the number of variables is decreased and the solutions to such simplifications provide a rapid approximation to the original MINLP.

4.3.1 Worst-Case Formulation

In the present convex formulation, there are d bit and transmit energy parameters for each sensor node. It is desirable to formulate this problem with an objective function D which accounts for a single quantization and transmit energy parameter for each of the sensor nodes. The reduction of these parameters is done choosing the number of bits and power

Table 4.2: *Number of variables and constraints for different convex approximations of the optimal energy-quantization problem. N is the number of sensors, d is the dimension of the state vector, and L is the number of SCP iterations.*

	WC	CVX
Variables	$3N$	$3Nd$
Constraints	$2+7N$	$1+N(1+6d)+d$
Runtime	$O(N^3L)$	$O(N^3d^3L)$

transmission level once for all elements of the state vector. This reduces the number of needed variables by $2N(d-1)$. An additional $N(d-1)$ variables can be removed from the problem by utilizing a single constraint with respect to the uncertainty term per sensor node, instead of d of them. This transformation of the optimization problem is akin to requiring worst-case satisfaction of the chosen solution, i.e., only one of the uncertainty constraints will be active. This form is easily obtained by taking the maximum of the state covariances for all state elements, the constraints now include only one bit (p_{nk}), transmit energy (b_{nk}), and uncertainty (u_{nk}) variable per sensor node. This reduced variable "worst-case" problem (WC) is written as

$$\begin{array}{ll} \text{minimize} & \sum_{n=1}^{N} p_{nk} b_{nk} \\ \text{subject to} & \\ & \frac{1}{\delta} - \sum_{n=1}^{N} u_{nk} \leq 0 \\ & \max_i\{\mathbf{P}^{(i)}_{n,k|k}\} + r^c_{nk} + r^q_{nk} - \left(\frac{2\tilde{u}_{nk}-u_{nk}}{(\tilde{u}_{nk})^2}\right) \leq 0 \\ & d\sum_{n=1}^{N} b_{nk} \leq BW \\ & d p_{nk} b_{nk} \leq p^{rem}_{nk} \end{array} \tag{4.30}$$

$\forall\ n = 1, \ldots, N$, in addition to the box constraints (i.e. C4-C6). This representation of the convex program has a significant difference from the previous (CVX) form: the uncertainty value being constrained is now the worst MSE term of the node state estimate. Thus, the element of the state update vector with the maximum variance determines the quantization and transmit energy level for all of the elements of the state estimate for that node.

We have broken down the number of variables and constraints needed to evaluate the

preceding problems in Table 4.2. The formulation in equation (4.29) is the convex constraint approximation (CVX). This form uses the full number of variables and constraints (with the added uncertainty constraints) from the original problem. The form in (4.30) is the worst-case (WC) solution (worst-case in the sense of the sensor node state estimate covariances), and uses a single bit and transmit energy variable.

4.3.2 Energy Aware Optimization

The formulation presented in (4.29) implies that for a fixed performance on estimate variance, minimizing over energy usage as the free variable will not take into account possible any disparate distribution of resources at each time iteration, without consideration of the need for future transmissions at a disadvantaged node. Thus, non-uniform remaining battery power can cause a rapid reduction in the number of active sensors. The results section will demonstrate this. As in the last formulation, a heuristic-based scaling creates variable cost of consuming energy resources at each sensor node dependent on sensor health, causing it to more conservatively use battery power when it is close to depletion. As in the estimation formulation, Λ_{nk} is updated (Eq. (4.23)) using the current battery level and α, the heuristic scaling parameter. Different from the energy-aware formulation of the original form, this schema for fairness scaling (similar to (4.23) replace the value of the objective with a weighted form. The Λ_{nk} weights are updated at each iteration and the objective in (4.29) is replaced with

$$\text{minimize} \sum_{n=1}^{N} \Lambda_{nk} \sum_{i=1}^{d} p_{nk}^{i} b_{nk}^{i}. \tag{4.31}$$

Thus the heuristic weight $\Lambda_{nk} \in [1, \frac{1}{\alpha}]$ with α the heuristic scaling parameter. Λ_{nk} is best defined as a dynamic control parameter that reflects the resource policy of each node based on remaining battery power. Low battery power would result in a large value for Λ_{nk}) and the vice versa.

4.4 Simulation Results

We consider the distributed estimation of an object moving in two dimensions[3]. Each sensor maintains a position-velocity (PV) state estimate of the object. Power, time, and distance values are given in generic pu, tu, and du units, respectively, and are not necessarily equivalent to any standard units, but merely serve as a reference. Likewise, the number of bits should not be reflective of any particular wireless communication standard, but is only a part of the illustration of the optimization strategy. The sensor takes position measurements with uncorrelated measurement noise, which has variance $\sigma_n^2 = 1$ along both axes. The true trajectory starts at $[0, 0, v_x, v_y]^T$ (where $v_x = v_y = 1du/tu$) and evolves with a nearly constant velocity (NCV) or Velocity Wiener process model [86]. The true process noise variance level is 0.8 as well as the assumed process noise of the local filters. The number of sensors in each of the following scenarios is $N = 4$. The communications parameters are as follows: The total allowable rate is set to $BW = 60$ bits per channel use and the dynamic range of observation is $W = 10$, with an offset region of $[0, 20]$ for the $x-$ and $y-$axis. The Rayleigh fading channel coefficient is $\overline{|h_n|^2} = 1$ and $N_0 = 0.5$ is the channel noise power spectral density coefficient. The maximum and minimum transmit energy are $p^{max} = 40pu$ and $p^{min} = 5pu$, respectively. We first illustrate that all methods perform comparably and their differences are measured against the single-instance global solution. Then, a pair of single run scenarios are shown for the purpose of illustrating the performance of the system for a typical run. Finally, multiple Monte Carlo runs are executed to characterize average performance.

[3] As noted in Section 4.1.2, the objective in (4.9) becomes an approximation of the MSE for spatially correlated noise. For a moving object in a field of sensors, the process noise among nodes is correlated. The application here was not selected to match the uncorrelated case observed in our objective, but rather for illustration purposes. Other work breaches the correlated process for these scenarios, e.g., in [110].

4.4.1 Comparison of Convex Formulations to a Single-Instance Global Solution

The results for the linear constraint convex (LCVX) approximation should theoretically be very close if not equal to the reciprocal constraint convex approximation (CVX). This pair of formulations along with the worst-case (WC) approximations of the original problem are shown in the single instance optimization results given in Table II along with the exact global solution. The relaxed and integer value solutions are shown for comparison. This table shows the objective values computed from the original objective in (4.9) using the final decision variables determined by the convex approximations. Performance is compared to the exact (branch and bound) global solution[4]. In order to make an accurate comparison, the relaxed solution (rlxd) of the optimization problem is rounded (or floored, to maintain the total bandwidth constraint) and the original objective is then computed for comparison. This comparison also illustrates that the effect of integer relaxation is minimal in terms of achieved cost function values. The two cases considered in this juxtaposition of solutions represent uniform operating states, with readily available energy for each node (Case A); and a highly energy constrained scenario with uneven operating states where improper utilization of resources would cause sensor nodes to become inoperable (Case B). The heuristic parameter is set to $\alpha = 0.1$ for both cases. As expected, the linear constraint convex (LCVX) and reciprocal constraint convex (CVX) formulations both achieve nearly the same value. We note that the reciprocal constraint formulation converges very slowly and convergence can be improved by providing tighter tolerances and more iterations. However, we quickly reach the relative floating point accuracy of the numeric solution. This is a result of the difficulties explained in Section 4.2.1. The worst-case approximation (WC) also does well but with performance inferior to the other convex approximations. This difference is due to non-uniformity in the covariance of the state estimate. For uniform state estimate variance, the LCVX and WC methods will be exactly the same. However, if there are differences

[4]LINGO® was used to find the globally optimal branch and bound solution. The optimal solution was returned with a dual certificate equal to it with sufficient numerical precision.

Table 4.3: *Objective values for convex formulations versus globally optimal exact solutions.*

	WC		LCVX		CVX		Global
	rlxd	int	rlxd	int	rlxd	int	
Case A	6.533	7.522	5.499	6.036	5.605	5.658	3.588
Case B	10.674	11.211	9.318	9.213	9.372	9.324	5.665

in the covariance of the elements of the state estimate the WC will still uniformly split the bandwidth and power chosen for a particular sensor between the elements of the state estimate to be transmitted. Thus, the minimum BLU-like objective is only achieved for the most noisy element of the state estimate. Whereas the LCVX method can increase bits and transmit energy for noisier elements of the state estimate and thus maintain an overall lower BLU-like objective. This could be thought of as a reverse-water-filling-like property, i.e., attempting to match the estimate variance with more precision in the channel and quantization variance. The WC treats all elements of the state estimate covariance equally when they are not equal, resulting in using more bandwidth and power necessary for some estimates and not enough for others. The advantage of WC gained in computational speed is implied by the runtimes shown previously in Table I. We next analyze the behavior some typical single run time-based results common for LCVX.

4.4.2 Single Object Tracking Run for Estimation Formulation

For the following scenarios linear constraint convex (LCVX) and worst-case (WC) approximations are shown here, as the LCVX and CVX methods perform similarly. The initial power allocated for this scenario is $p_n^{rem} = 4000pu$ for $n = 1, 4$ and $p_n^{rem} = 2000pu$ for $n = 2, 3$. The methods are shown both with and without the energy-aware heuristic (i.e. $\alpha = 1$ and $\alpha = 0.1$, respectively). For $\alpha = 1$ this implies $\Lambda_{nk} = 1, \ \forall \ k, n$. We additionally compare our methods to some static policies utilizing the minimum and maximum transmit energy levels. The minimum benchmark utilizes half the bandwidth and evenly allocates it amongst all active sensors. The maximum benchmark utilizes the entire bandwidth evenly

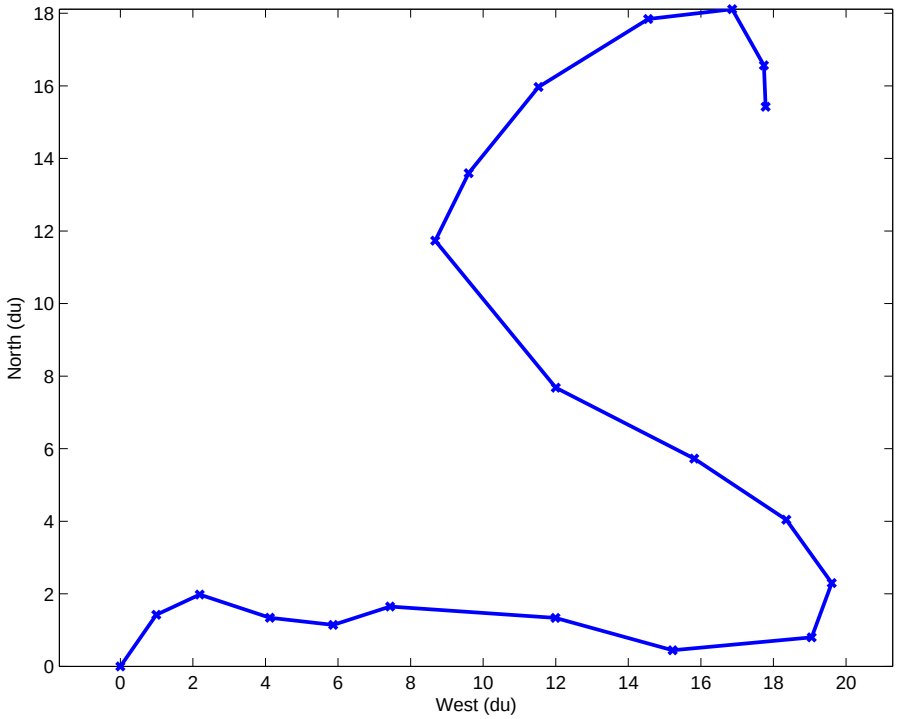
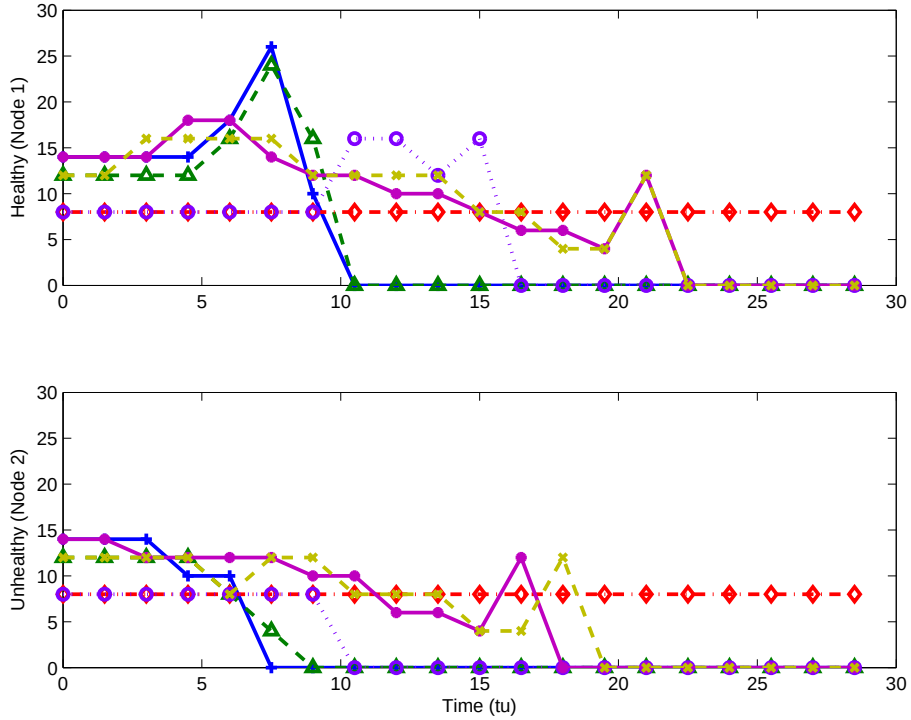


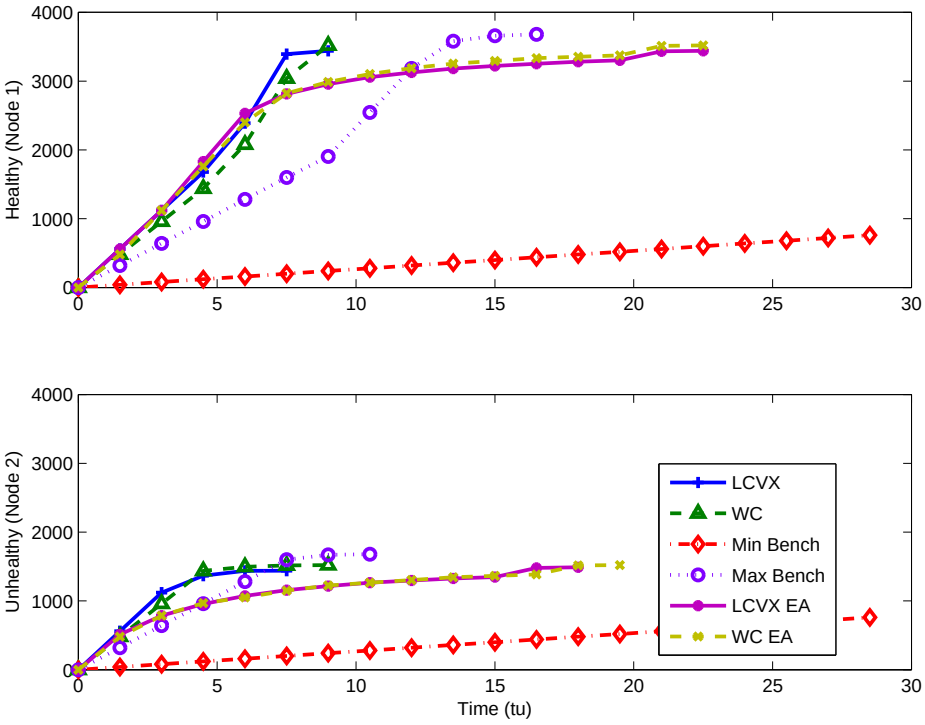
Figure 4.5: *Simulation scenario for example truth trajectory for single object tracking run.*

allocated to all sensors. These are denoted "Min Bench", and "Max Bench" in the legends. The plot in Figure 4.5 shows a two dimensional plot of the object true trajectory. Figure 4.6 shows the individual power usage, number of bits used by a healthy and unhealthy sensor node in the system, in this case, sensors 1 and 2, respectively. Note that for the LCVX and CVX methods, the maximum bits and energy levels are selected at each time instant until the unhealthy nodes (2 and 3) deplete their energy resources and stop functioning.

After this, the remaining collaborating nodes have twice as much available bandwidth with which to transmit their states. This being the case, the remaining sensors use their energy reserves quickly since the optimization greedily uses all available bandwidth, also maximizing transmit energy levels, where possible. It is plain that the sensors deplete their energy resources quickly and stop tracking at about $9tu$. The benchmark methods provide us an intuitive upper and lower bound on resource usage and possible sensor node lifetimes. While the LCVX and WC follow the usage patterns of the maximum usage benchmark, the



(a) Individual sensor node quantization levels.



(b) Individual sensor node running total energy usage.

Figure 4.6: *Single object tracking run: Transmit energy and quantization levels for a healthy and unhealthy node of the network, with and without the energy-aware heuristic ($\alpha = 1$ vs. $\alpha = 0.1$).*

allocation is not optimal because it does not consider the influences of any noise sources, as we will see in the next section and the comparison of the error performance. When the energy-aware heuristic is applied to LCVX and WC (denoted in the legend with an appended "EA") for $\alpha = 0.1$, nodes 2 and 3 no longer deplete their energy resources, but rather these nodes function to almost $20tu$. It is easy to see the conditions near the depletion of sensor battery which result in setting $\Lambda_{nk} = 1$, as mentioned in Section 4.3.2. We next compare the LCVX and WC methods alongside our benchmark methods for multiple Monte Carlo runs and apply some performance metrics.

4.4.3 Monte Carlo Simulations for Estimation Formulation

The following results are obtained by executing multiple Monte Carlo (MC) runs. The starting energy in the single tracking run is the same here but results are averaged over 50 runs. We will now define several metrics by which to compare the methods of interest. In the case of a static policy, the *lifetime* is simply the number of iterations that can be run before the battery is depleted, i.e., $LT = \left\lfloor \frac{p_{rem}}{\sum_{i=1}^{d} p_n^i b_n^i} \right\rfloor$. However, when power and quantization are determined dynamically based on the noise levels, then the lifetime is a probabilistic quantity, and no explicit function can be offered. Instead the lifetime value is determined during simulations and its probabilistic occurrence represented by Monte Carlo averages. We define the lifetime as

$$LT = arg\max_{k} \left\{ \frac{1}{N} \sum_{n=1}^{N} I(p_{nk}^{rem}) > \epsilon \right\}, \tag{4.32}$$

where

$$I(p) = \begin{cases} 0 & p < p_{min} \\ 1 & p \geq p_{min} \end{cases} \tag{4.33}$$

and ϵ is the node outage threshold. We also define the error measure by which we will approve our methods. We use a normed measure since we are consider a vector state. Define the Root Mean Square Normed Error (RMSNE) to be

$$RMSNE_k = \sqrt{E[||\mathbf{x}_k - \hat{\mathbf{x}}_{BLUE,k}||^2]}, \tag{4.34}$$

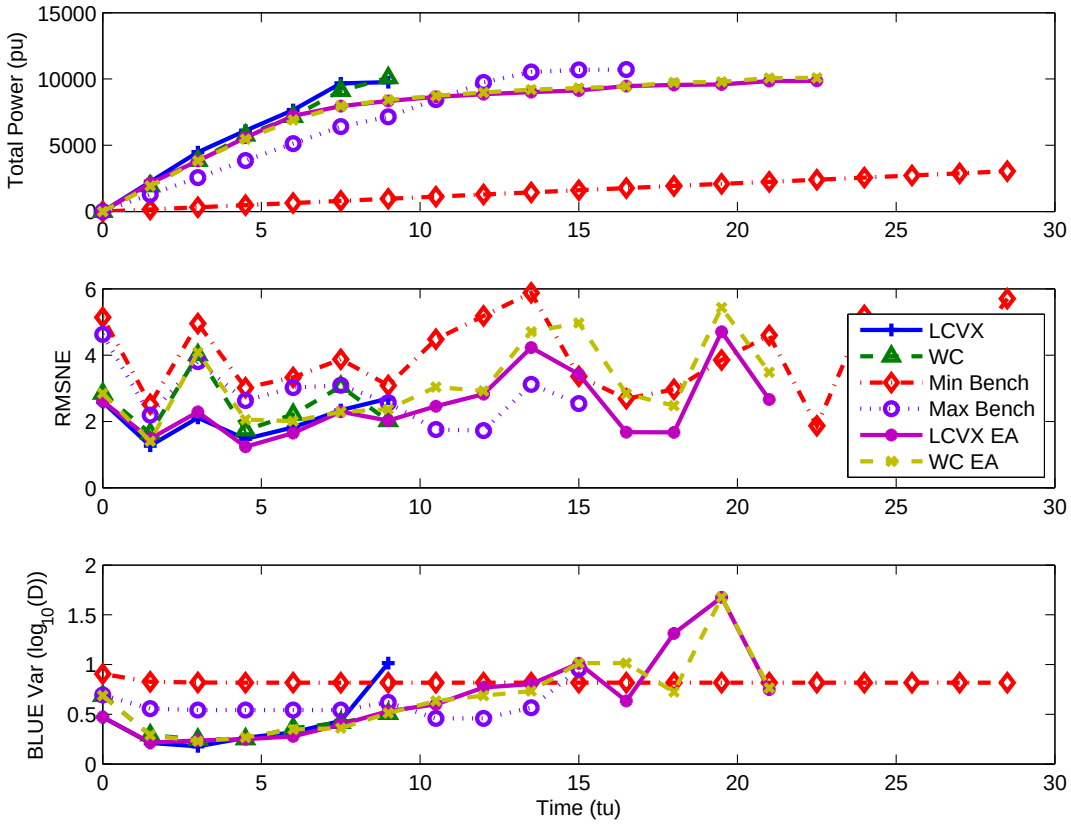


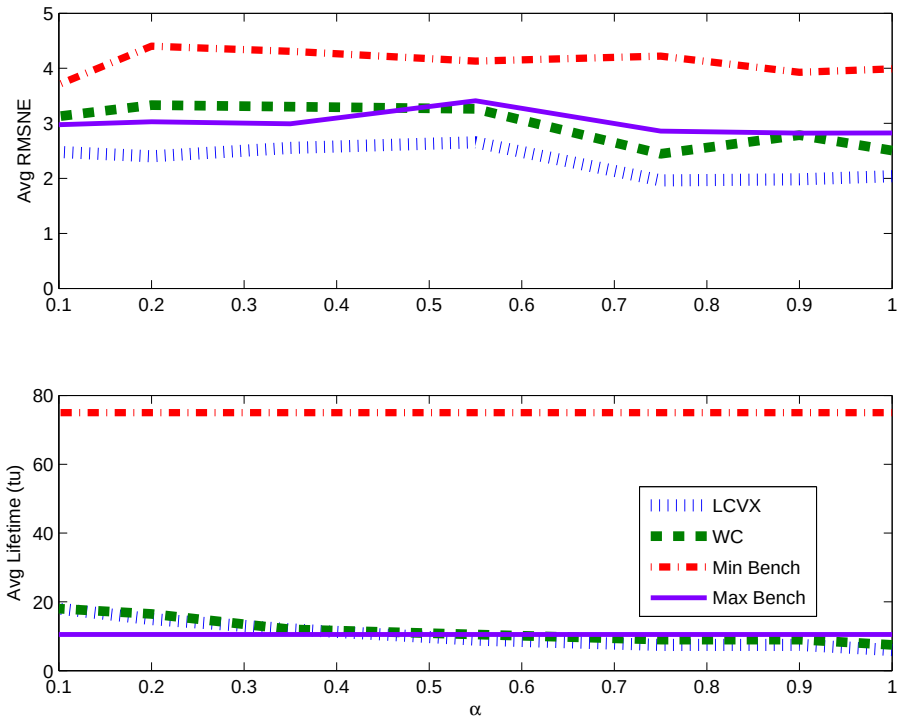
Figure 4.7: *Single object tracking run: Comparison of total consumed energy, RMSNE, and analytic BLUE variance for the various methods.*

where $||\cdot||$ is the ℓ_2-norm. The performance of the optimization methods versus the benchmark methods introduced in the previous section in terms of total power usage and RMSNE is shown in Figure 4.7. We clearly see that the LCVX and WC methods are comparable with the maximum benchmark, with LCVX having slightly better error performance than WC, as expected. The methods are nearly equivalent with and without the energy-aware heuristic up to about $9tu$ when the methods not employing the heuristic deplete their energy reserves. While the energy-aware enabled methods continue tracking at a somewhat reduced error performance, since they are conserving energy resources. The minimum benchmark maintains a higher error throughout the simulations, as it uses the fewest resources, with the exception of the very end of the energy-aware methods as they are forced to use fewer bits than even the minimum benchmark method. We have also include the analytic value of

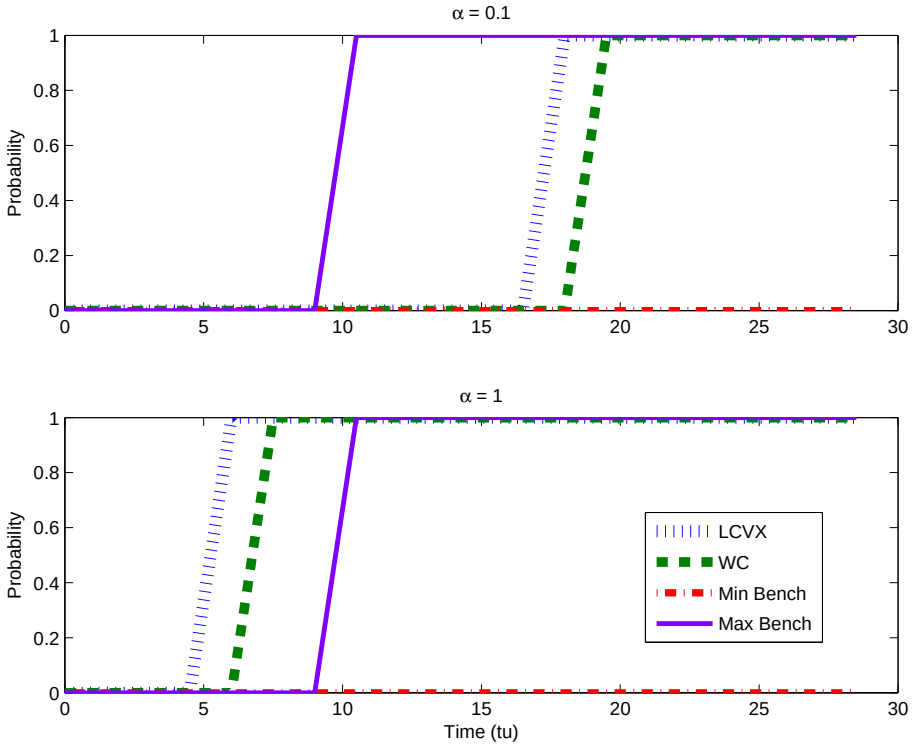
D from equation (4.9), which is dependent direct on the various noise variances determined by the choice of transmission power and quantization. While not directly proportional, this metric instructs us on what we should expect in the relative values of the RMSNE for the different methods.

Our evaluation of the energy-aware optimization can be performed by observing how average RMSNE and lifetime are affected by the heuristic scaling parameter, α. To do this, we find the RMSNE_k, and then average across time $k = 1, \ldots, K$. For these comparisons we also find the average lifetime for each α value. Figure 4.15 clearly shows that the lifetime is greatly extended by using reasonably small values of α. It also demonstrates that the increase in error is moderate relative to the naïve approach. As expected, the LCVX always has much lower error than the WC method, while the WC method maintains slightly longer lifetimes for the same values of α. The probability of outage is the probability that the remaining percentage of nodes at any given time instance falls below a threshold. In our case, the running lifetime of each node is recorded and the WSN is considered inoperable when the fraction of nodes with remaining energy drops below the threshold number of nodes ($\epsilon = 0.75$ in this case). For a single run the WSN is either operable or not (0 or 1) at each time instance. However, this is averaged over 50 MC runs, thus this provides a probability of outage, $Pr_k \in [0, 1]$ at each time instance, k. We note in Figure 4.16 that the probability of outage for the network is extended for $\alpha = 0.1$ to 350% for LCVX and to 250% for WC. The LCVX and WC formulation have differing probability of outage performance for the reason given in Section 4.4.1, and it follows that slightly worse error performance would indicate a more energy-conservative network.

At this juncture it is important to point out that we do not explicitly investigate the scalability problem because of the assumption that sensor scheduling has been completed. The scalability of distributed estimation in wireless sensor networks in terms of either number of sensors or percentage of active sensors has been extensively explored in other works (e.g., [17–19]). These metrics of performance for a scheduling algorithm are indicators of

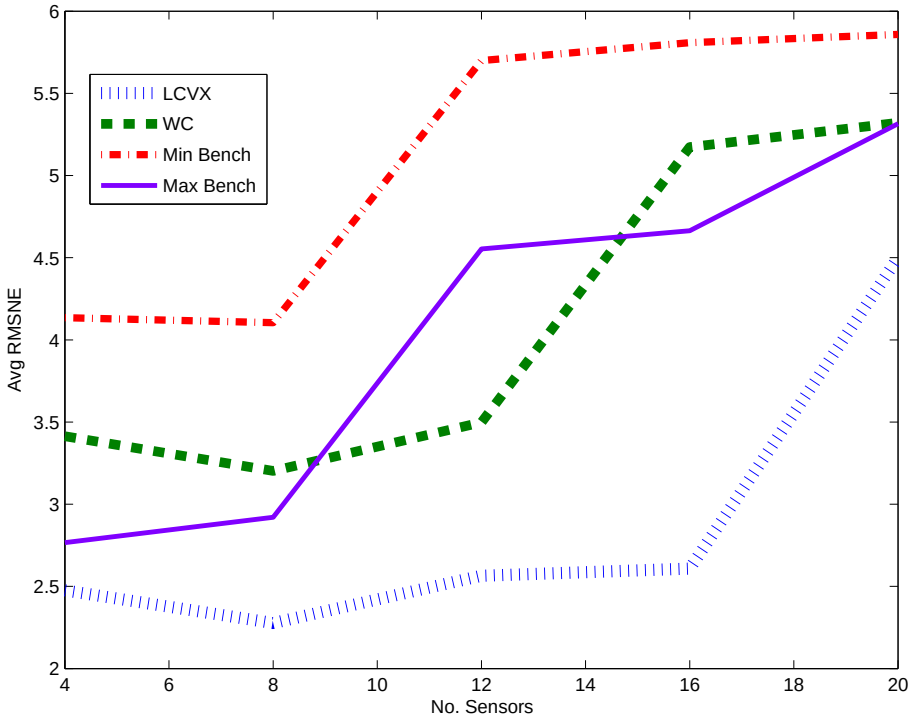


(a) WSN lifetime and error plotted for varying frugality parameter, α; the unequal initial energy scenario (1 & 2) with $N = 4$ is considered for the LCVX and WC formulations.

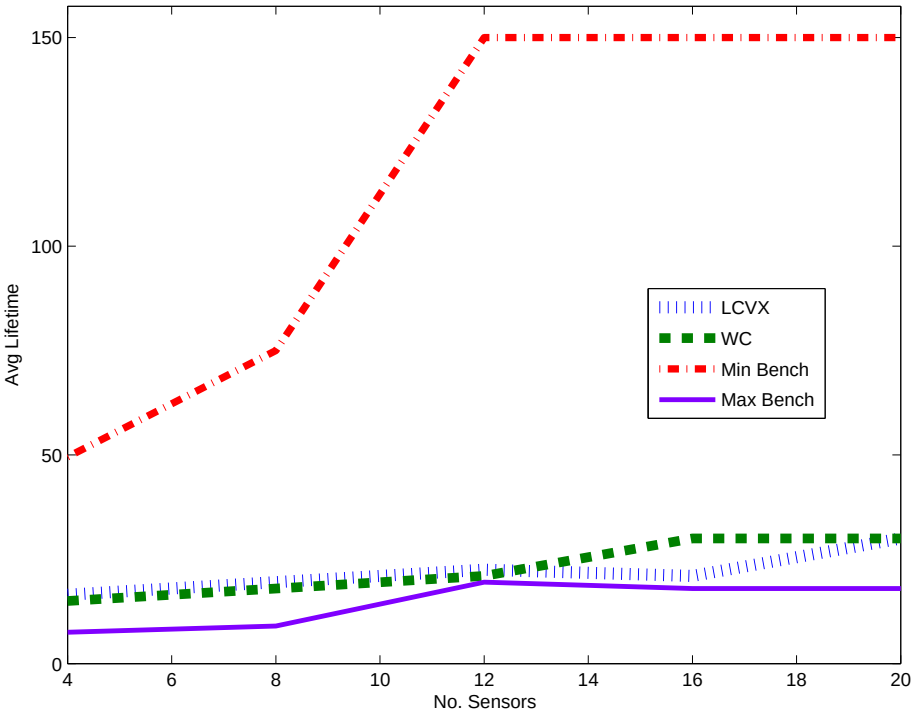


(b) System probability of outage for each of the formulation types, considering a threshold of 75% of the operating sensors still functioning. With (top) $\alpha = 0.1$ and (bottom) $\alpha = 1$.

Figure 4.8: *The trade-off of lifetime versus average error performance and its affect on the system probability of outage.*



(a) RMSNE vs. number of sensors.



(b) Lifetime vs. number of sensors.

Figure 4.9: *The scalability of the scheduled subset size.*

the energy savings offered by the algorithm. As a thought experiment, if we assume the lifetime of a single node to be the number of (not necessarily consecutive) operational time instances, then it is intuitive if we schedule one sensor at each time step then we would have the network lifetime proportional to the product of the number of sensors and the lifetime of a single node. For our illustrations, if we assume a schedule that cycles between sets of sensor nodes of the same size and LT is the lifetime of a subset of $N < M$ sensors in a M node network, the relationship of overall WSN lifetime to that of the subset will be

$$LT_{WSN} \propto \frac{M}{N} LT. \tag{4.35}$$

This is actually a lower bound since smarter scheduling could further increase the WSN lifetime by preventing disadvantaged nodes from being depleted (e.g. [111]). The vital issue with respect to the scalability of the network in this case is chiefly that of how to divide up the limited bandwidth. Indeed, there is a maximum number of scheduled nodes such that each node can transmit all components of the local updated state vector. As the network increases in size, the optimization is constrained is a smaller feasible space, as each element of each state vector must have at least one bit (since the nodes have been scheduled). The maximum network size based on single bit (minimum) quantization is

$$N_{max} = \frac{BW}{d}, \tag{4.36}$$

where d is the dimension of the state vector. In addition to this upper bound on scheduled node subset size, single bit quantization can introduce significant errors, especially for a large dynamic range. Thus, we expect with increasing network size that the errors increase until no more sensors can be utilize in the scheduled set. We have included an illustration of this in Figure 4.9, where we have shown results for the LCVX and WC methods (since LCVX and CVX have equivalent results for the appropriate optimizing parameters). The setup is the same as the basic scenario described in Section 4.4.2 (but with the adjustment of $BW = 100$). Figure 4.9 demonstrates the increasing average RMSNE for increasing number of sensors, particularly as the number of sensors approaches the limit in (4.36), which is 25

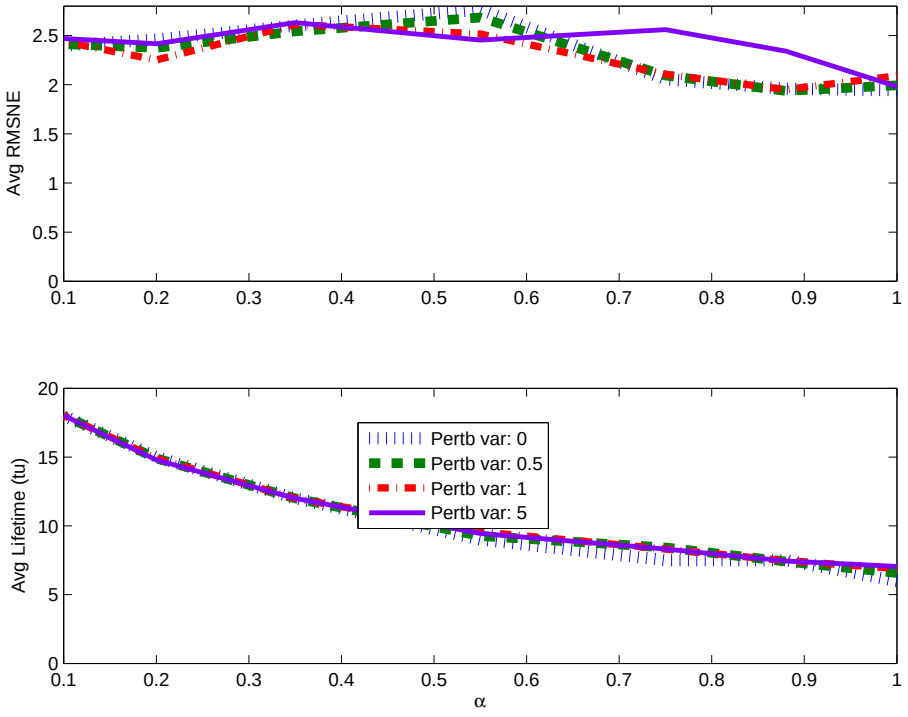
for Figure 4.9.

4.4.4 Sensitivity Analysis for Unknown Covariance

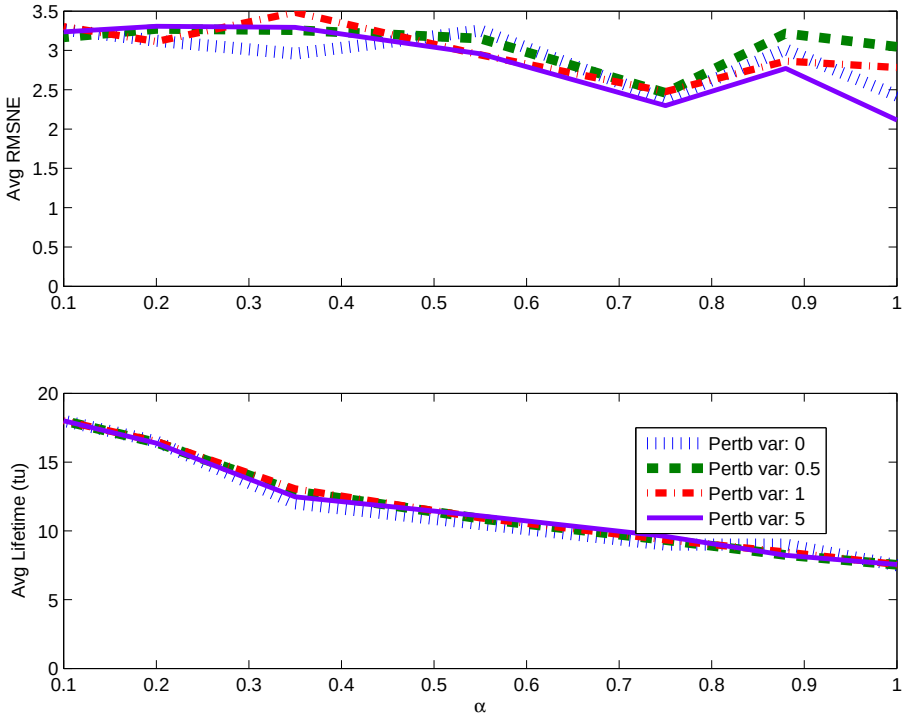
The nature of the problem set forth in this chapter demands knowledge of the covariances of the estimates at each node. This extra information is not considered in the transmission of data from the sensor nodes to the fusion center. We therefore wish to determine the sensitivity of the quantization and transmit energy level decision variables to random perturbations in the covariances, which are used to determine them. These perturbations are generated from Chi-squared distribution with one DOF, i.e., $\chi^2(1,\rho) = \sum_{i=1}^{k} Z_i^2$ for $k = 1$ with independent $Z_i \sim \mathcal{N}(0,\rho)$. In Figure 4.10 the optimization is carried out with randomly perturbed covariance information and the results are averaged over 50 Monte Carlo runs. The variance parameter (ρ) of the perturbation is swept from zero (no perturbation) to five. Figure 4.10 clearly illustrates the small average error which is introduced for the linear constraint convex (LCVX) and worst-case (WC) approximations at various values of α. The same also demonstrates that the lifetime remains largely unaffected by perturbations to the covariance information. For all the methods the original objective (4.9) decreases for non-zero covariance perturbation. This is because more bits and higher transmit energy levels are selected to reduce noise and compensate for the additional noise introduced by the variance perturbation. However, the final estimation accuracy depends on the actual state covariance. Thus, more accurate estimates are obtained for perturbed covariance values, while lifetimes for such perturbations are decreased.

4.4.5 Simulation Setup for Energy Formulation

The setup for the dual-like energy formulation utilizes a slightly different scenario. We still consider the distributed estimation of an object moving in two dimensions with sensor nodes maintaining a position-velocity (PV) state estimate of the object. The process and measurement models are the same and the number of sensors in each of the following scenarios is $N = 4$. The communications parameters are as follows: The total allowable rate is set



(a) LCVX Sensitivity to Covariance data perturbation.



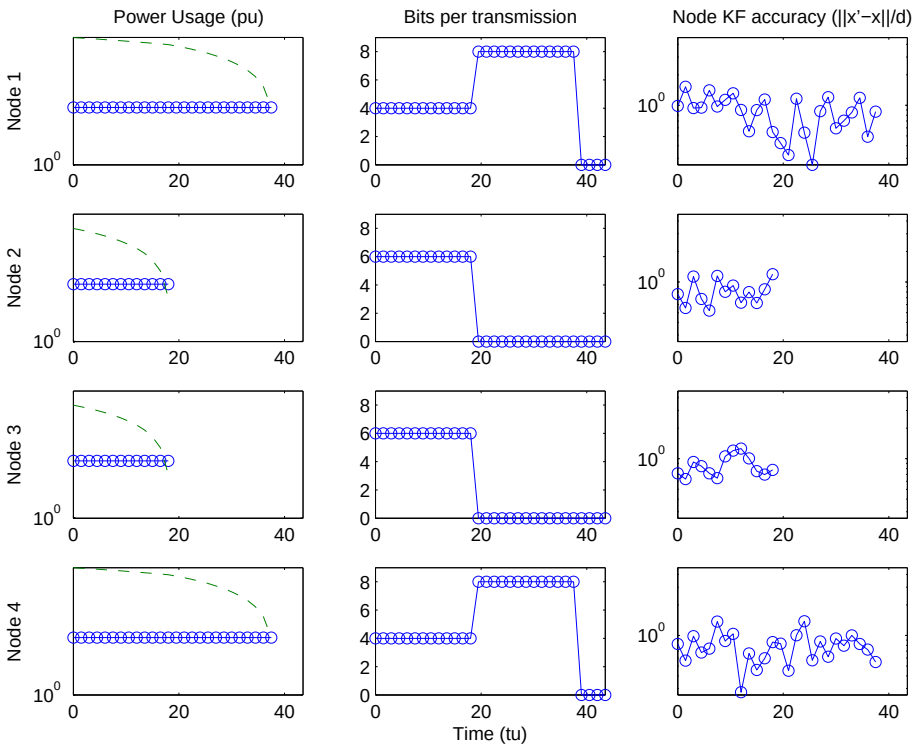
(b) WC Sensitivity to Covariance data perturbation.

Figure 4.10: *Covariance sensitivity tests for the LCVX and WC approximations for several values of* α.

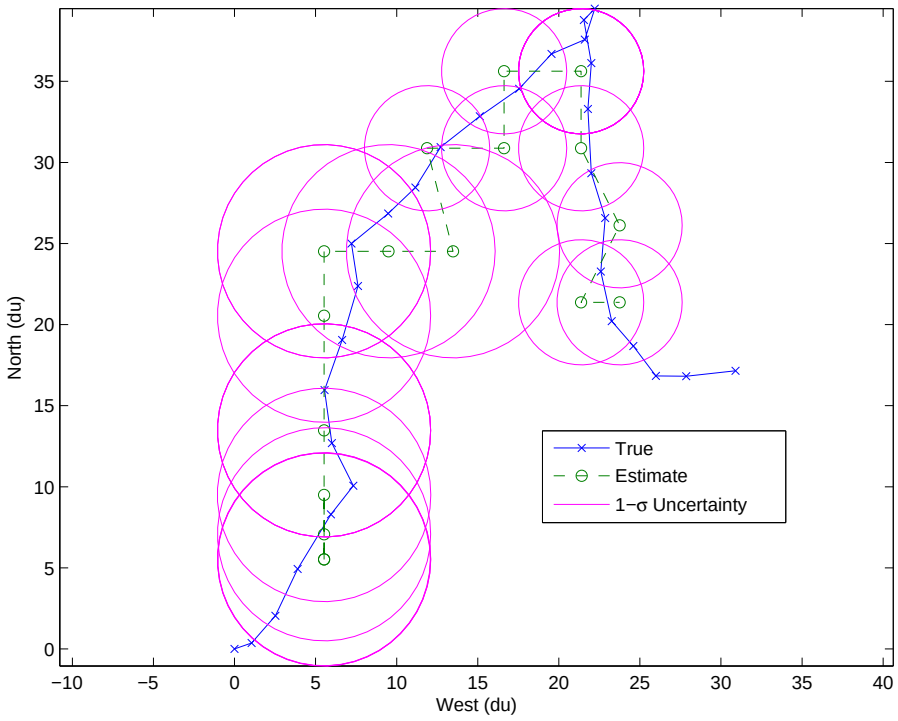
to $BW = 60$ bits per channel use and the dynamic range of observation is $W = 20$, with an offset region of $[0, 40]$ for the x- and y-axis. The Rayleigh fading channel coefficient is $\overline{|h_n|^2} = 1$ and $N_0 = 0.25$ is the channel noise power spectral density coefficient. The maximum and minimum transmit energy are $p^{max} = 40pu$ and $p^{min} = 5pu$, respectively. The allowable uncertainty level is $\delta = 8$. It should be noted that in order to properly simulate the quantization and transmission of data, the relaxed solution of the optimization problem is rounded (or floored, to maintain the total bandwidth constraint) and the original objective is then computed for comparison. We first show the results on a pair of single run scenarios which illustrate the performance of the system for a typical run. Following this multiple Monte Carlo runs are executed to characterize average performance.

Non-uniform Initial Energy Resource Scenario for Energy Formulation

The following scenarios demonstrate the epigraph convex constraint approximation (CVX) as an example. The initial power allocated for the first scenario in Figure 4.11 is $p_n^{rem} = 800pu$ for $n = 1, 4$ and $p_n^{rem} = 400pu$ for $n = 2, 3$. No energy-aware heuristic is used and we set $\alpha = 1$, which means $\Lambda_{nk} = 1, \ \forall \ k, n$. Figure 4.11(a) shows the individual power usage, number of bits selection, and the local MSE of the filter. Note that the minimum bits and energy levels necessary to maintain the MSE performance are selected at each time instant until nodes 2 and 3 deplete their energy resources and stop functioning around $20tu$. After this, the remaining collaborating nodes must make transmission decisions (by increasing bits and energy usage) so as maintain error performance without aid of the two depleted nodes. This being the case, the remaining sensors use their energy reserves quickly since the optimization uses more of the available bandwidth, in addition to higher transmit energy levels. The plot in Figure 4.11(b) shows a two dimensional plot of the object true trajectory, the fusion node estimate of the object, and a one standard deviation uncertainty ellipse around each estimate. Figure 4.12 shows the Lagrange multipliers and that the necessary constraints (bandwidth, power, MSE) satisfy complementary slackness in the solutions to the relaxed linear constraint convex approximation for each time instance



(a) Individual sensor node metrics.



(b) Two-dimensional uncertainty plot of trajectory and estimates.

Figure 4.11: *Scenario 1 - no energy-aware heuristic ($\alpha = 1$).*

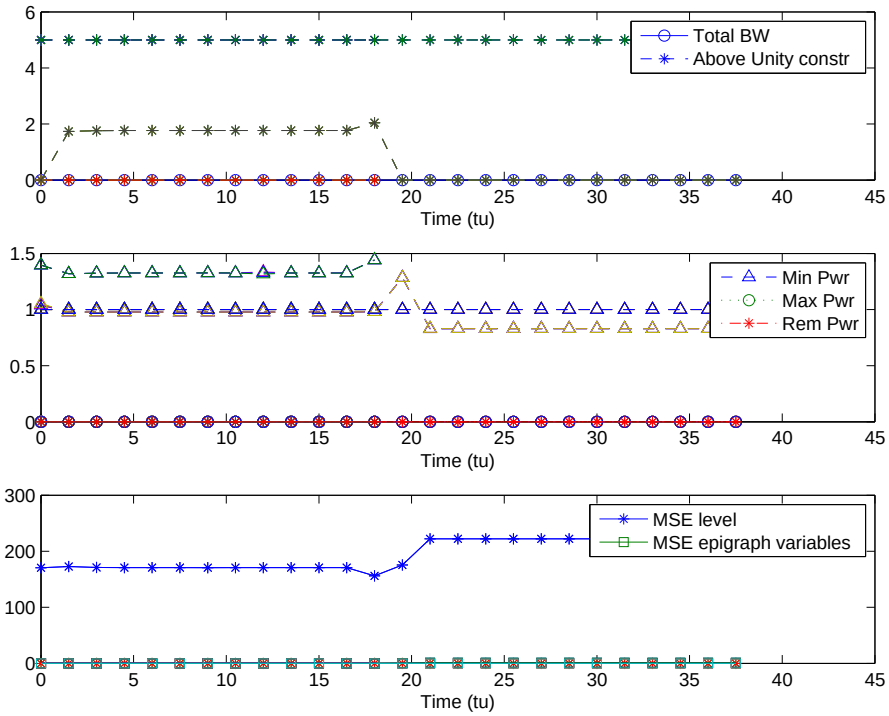


Figure 4.12: *Lagrange multipliers resulting from the optimization problems in scenario 1.*

during the scenario.

Non-uniform Initial Energy Resource Scenario for Energy Formulation with Energy-Awareness

This scenario uses the energy-aware optimization of Section 4.3.2. The initial conditions are the precisely same as the previous scenario, with the initial energy again allocated as $p_n^{rem} = 800pu$ for $n = 1, 4$ and $p_n^{rem} = 400pu$ for $n = 2, 3$. This scenario is shown in Figure 4.13 and sets $\alpha = 0.1$ to enable the energy-aware optimization. Figure 4.13(a) shows that the energy resources of nodes 2 and 3 do not deplete as rapidly and function until around $35tu$. Here, the healthier nodes compensate and contribute more to the estimation performance. The two dimensional uncertainty plot in Figure 4.13(b) shows that the fusion node maintains a track of the object, with stable uncertainty regions throughout the scenario. Figure 4.14 shows the scaling values of the energy-aware heuristic which are used in the optimization over time. Here a lower value of Λ_{nk} (cfr. nodes 1 and 4) implies that more of a node

resources can be utilized in a particular time instance, larger values (cfr. nodes 2 and 3) are more limiting of energy usage. The following results compare the performance of the CVX and WC methods with the uniform usage benchmark over multiple Monte Carlo runs.

4.4.6 Monte Carlo Simulation Analysis for Energy Formulation

The following results are obtained by executing multiple Monte Carlo (MC) runs. The starting energy in the single run scenario is the same here but results are averaged over 50 runs. Our evaluation of the energy-aware optimization can be performed by observing how error and lifetime are affected by the heuristic scaling parameter, α. To do this, we find the mean of the mean-squared normed error from the MC ensemble and then average across time, i.e.,

$$E_k[MSNE_k] = E_k[E[||\mathbf{x}_k - \mathbf{x}_{BLUE,k}||_2^2]],$$

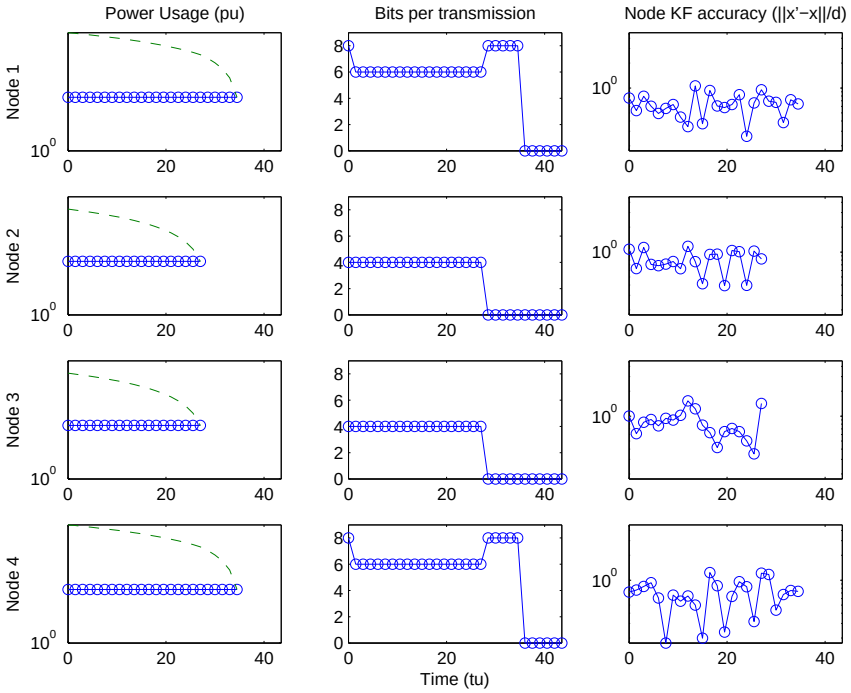
where $E[\cdot]$ is the expectation across the ensemble set and $E_k[\cdot]$ is the expectation across the time instances. For these comparisons we also find the average lifetime for each α value. In the case of a static policy lifetime is simply the number of iterations that can be run before the battery is depleted, i.e. $LT = \left\lfloor \frac{p_{rem}}{dp_n^i b_n^i} \right\rfloor$. However, when power and quantization are determined dynamically based on the noise levels, then the lifetime is a probabilistic quantity, therefore no explicit function can be offered. Instead the lifetime value is determined during simulations and its value is the average over Monte Carlo runs. We define the lifetime as

$$LT = arg\max_k \left\{ \frac{1}{N} \sum_{n=1}^{N} I(p_{nk}^{rem}) > \epsilon \right\}, \tag{4.37}$$

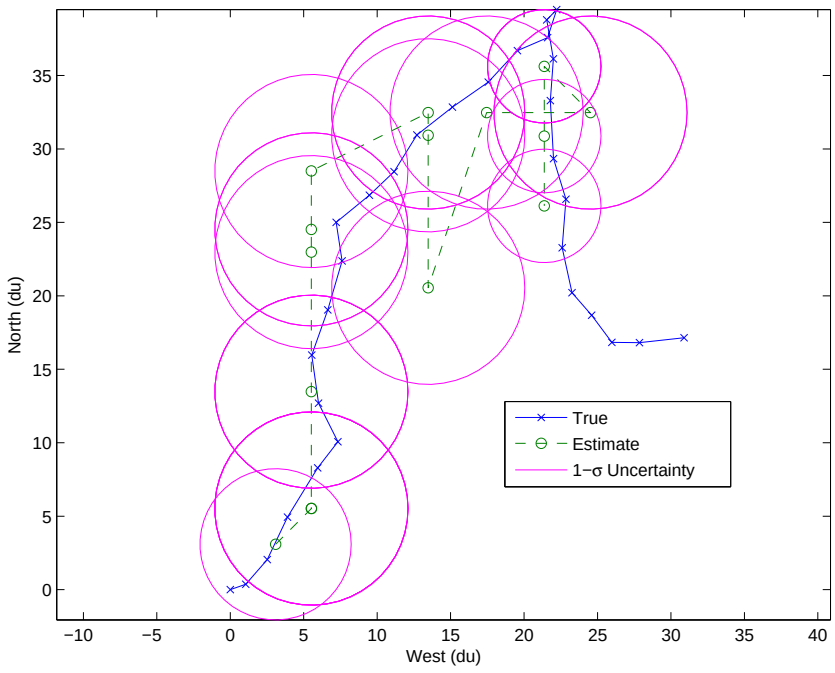
where

$$I(p) = \begin{cases} 0 & p < p_{min} \\ 1 & p \geq p_{min} \end{cases}$$

and ϵ is the node outage threshold. We also compare the CVX and WC formulations with that of uniform bit and transmission energy usage. In this benchmark method, we instruct the nodes to send quantize data with 50% of their individual allotment of bandwidth, and



(a) Individual sensor node metrics.



(b) Two-dimensional uncertainty plot of trajectory and estimates.

Figure 4.13: *Scenario 2 - using energy-aware heuristic ($\alpha = 0.1$).*

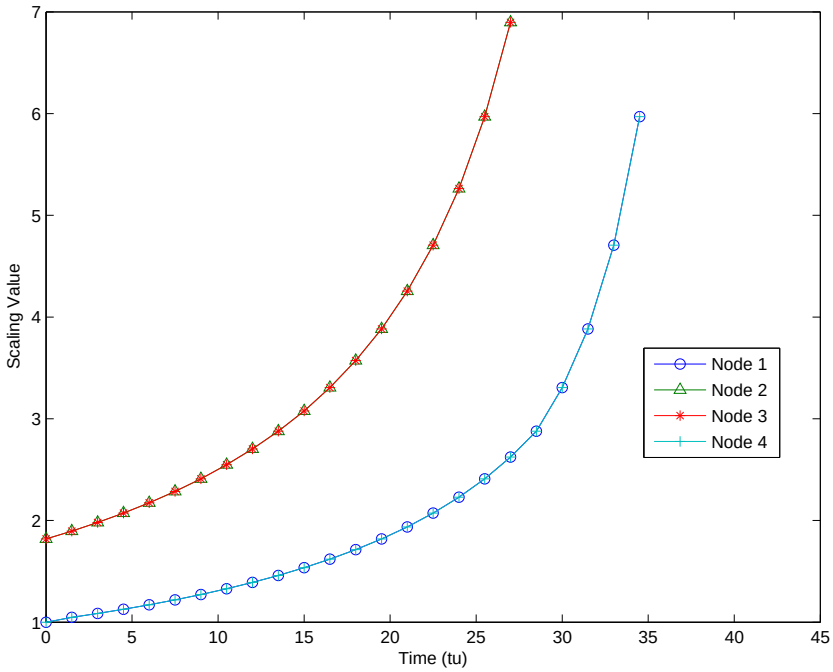


Figure 4.14: *Energy-aware heuristic scaling coefficients for scenario 2.*

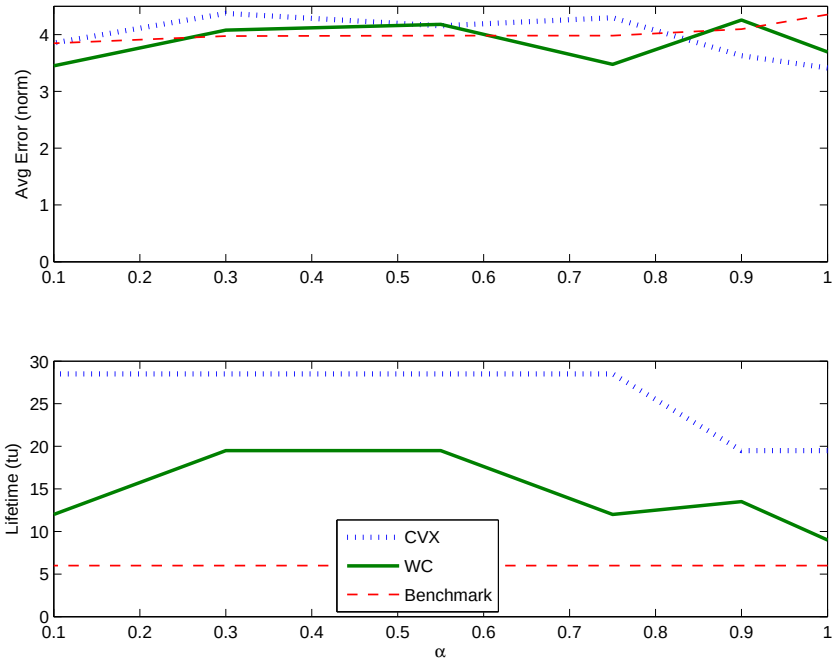


Figure 4.15: *WSN lifetime and error plotted for varying α values, the unequal initial energy scenario (1 & 2) with $N = 4$ is considered for the CVX and WC formulations, and compared with the uniform benchmark decisions.*

25% of their maximum allowable transmit energy level. This represents the supposed desire of an operator to achieve adequate performance which is traded-off for network lifetime. For a outage threshold of $\epsilon = 0.9$, Figure 4.15 clearly shows that the lifetime is greatly extended by using reasonably small values of α. It also demonstrates that the increase in error is moderate relative to the naïve approach.

The performance differences between the two approximations is due to non-uniformity in the covariance of the state estimate. For state estimate variance uniform across the vector, the CVX and WC methods will be the same. However, if there are differences in the covariance of the elements of the state estimate the WC will still uniformly split the bandwidth and power for a particular sensor between the elements of the estimate. The BLUE uncertainty constraint is maintained for the most noisy element of the state estimate, whereas the CVX method can decrease bits and transmit energy for less noisy elements of the state estimate and thus maintain an overall lower energy usage objective. This could be thought of as a reverse-water-filling-like property, i.e., attempting to match higher variance with more precision in the channel and quantization variance. The WC treats all elements of the state estimate as with equal variance, resulting in using more bandwidth and power that necessary for some estimates and not enough for others.

The probability of outage is the probability that the remaining percentage of nodes at any given time instance falls below a threshold. In our case, the lifetime of each node is recorded and the WSN is considered inoperable when the fraction of nodes with remaining energy drops below the threshold number of nodes (90% in this case, or $\epsilon = 0.9$). For a single run the WSN is either operable or not (0 or 1) at each time instance. However, this is averaged over 50 MC runs, thus this provides a probability of outage, $Pr_k \in [0, 1]$ at each time instance, k. We note in Figure 4.16 that the probability of outage for the network is extended for $\alpha = 0.1$ to 150% for CVX and up to 200% for WC. The CVX and WC formulation have differing probability of outage performance for the reason mentioned in the previous paragraph, and it follows that slightly worse error performance would induces

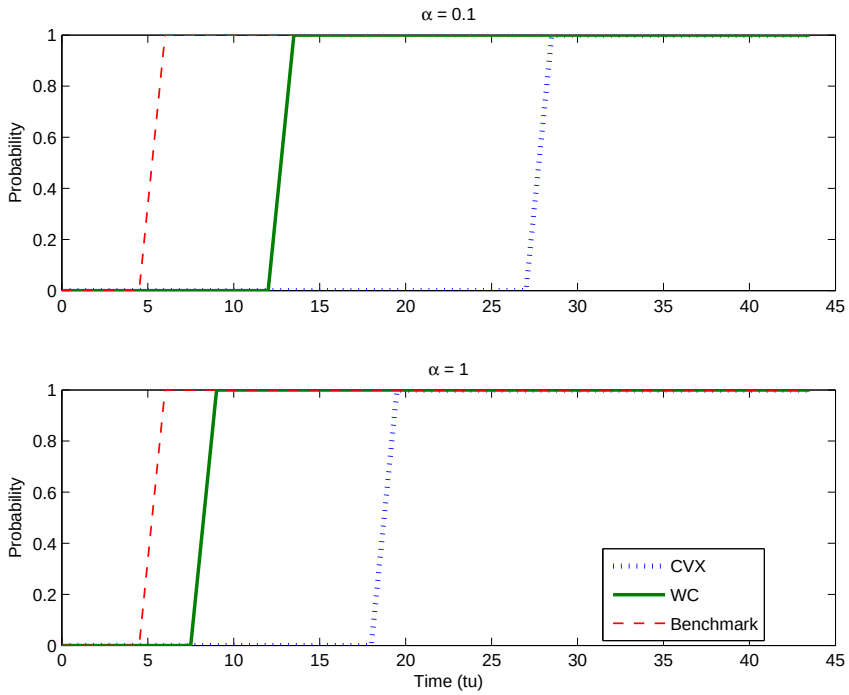


Figure 4.16: *System probability of outage for each of the formulation types, considering a threshold of 90% of the sensors still functioning. With (top)* $\alpha = 0.1$ *and (bottom)* $\alpha = 1$.

a more energy-conservative network.

4.5 Summary

The distributed estimation scheme presented in this chapter focuses on the decisions made after sensor selection and scheduling has been completed. The first formulation utilizes a resource constrained uncertainty objective which is a non-convex MINLP, the second examines the network lifetime by seeking to maximize network lifetime by minimizing the number of bits and total transmit energy with a constraint on estimation performance. We propose the variance of the Best Linear Unbiased Estimate (BLUE) on the estimation metric. Formulations from the relaxed, convex approximated, scalar case were applied in estimating a state vector from distributed nodes. It was found that the linear constraint convex approximation and its worst-case upper bound are well suited for solving this problem, both in performance and relative runtime. The energy-aware heuristic introduced allowed for the

extension of network lifetime while delivering adequate estimation results, clearly expressing the trade-off between lifetime and estimation accuracy. Monte Carlo runs demonstrated that lifetime increases for decreasing heuristic parameter, which is also reflected in the probability of outage. While error also increases, it is only minute and does not significantly affect the estimation results.

In the next chapter, we address a similar problem, but make simplifying assumptions which allow us to understand more of the underlying aspects of distributed constrained resource allocation.

Chapter 5

A General Framework for Resource Management: Energy Harvesting Case Study

While the the last chapter provided an adequate numerical solution to an optimization framework for joint bandwidth and power allocation decisions, our goal now is to study the various problems presented by different kinds of simplifying assumptions. A simplified framework allows for more intuition in the subsequent analysis. We attempt to extract the underlying behavior of these simplified problems and develop an analytical understanding, as well as methods for solving them.

Although concave utility maximization problems are tractable and the associated numerical techniques accurate, they are based on optimality conditions which, while providing a solution, do not offer intuition into the analytical behavior of the problem or solution steps. We develop this intuition through applicable examples, showing them to be special cases of our algorithm solution. In contrast to previous work, we present an intuitive algorithm for solving constrained utility maximization problems based on a fresh perspective of specific problem structures and principle of optimality which we generalize. We generalize a principle of optimality associated with the idea of diminishing returns for concave network communication utilities and analyze specific energy harvesting wireless sensor network problems to show the appeal of exploiting the underlying structure of these types of

problems.

Our method relies upon a type of 'per constraint' optimality which is achieved for the next most restrictive constraint. Subsequently, increments in resource allocations are made in such a way that the improved utility for active agents/users are optimal. In the previous sections, we have introduced the various models necessary to understand the setup for the energy harvesting problem and elaborations are made as necessary. We then provide motivating examples, followed by some guiding principles and the general solution to the utility maximization problem. The method is shown to be applicable to our examples. The optimality of the algorithm is then given, and simulation results follow which verify the correctness of our method.

5.1 Review of Basic System Model

We consider a cooperative network of wireless sensors communicating via some orthogonal signaling scheme over independent communication channels. Unique from our last chapter, our analysis considers two applications, (1) rate allocation/maximization which models these channels with fading and additive Gaussian noise, whereas (2) distributed estimation which only considers the channels in terms of its base path loss (we use the term 'loss' to indicate pathloss (α_{nk}) and the inverse of the fading coefficient ($\frac{1}{h_{nk}}$)). Each of these nodes utilizes an energy harvester, processing unit, and sensing devices. The N nodes communicate data either directly to a centralized base station [65], or along a predefined routing path to the base station [61]. An illustration of an example WSN setup is shown in Figure 4.1.

The transmissions are modeled according to Chapter 3. The received signal, in the case of the rate allocation application at one node from another is given by $y_{mk} = x_{nk} \cdot \sqrt{h_{nk}} + n_{nk}$ where x_{nk} is the transmitted signal during epoch k, with h_{nk} and n_{nk} respectively the (squared) uncorrelated fading and additive Gaussian noise of the channel between nodes n and m. In this model, whenever a signal is transmitted for duration L, $\frac{L}{2}\log\left(1+\frac{h_{nk}p_{nk}}{\sigma_{nk}^2}\right)$ bits of data are sent to the receiving node from the queue of data at node n, where σ_{nk}^2 is

the variance of the zero mean Gaussian random noise. We assume throughout the rest of our work that $\sigma_{nk}^2 = 1 \ \forall \ n, k$, although the methods herein are easily adapted to case of non-uniform noise variance. This transmission costs Lp_{nk} units of energy from the battery. Chapter 3 contains most of the relevant information on the modeling of the energy systems with the WSN nodes.

5.1.1 Prediction Horizon

One aspect of the energy modeling not mentioned in Chapter 3 is the prediction horizon. The prediction horizon, K, is the number of epochs wherein the harvested energy and correlated fading can be predicted reasonably well [61]. Our approach considers the optimal selection of transmission power levels over such a horizon of epochs with the assumption that the prediction of energy arrivals and fading coefficients has already been made. We will qualify the use of 'optimal' in the next section. The use of our algorithm as an online method can easily be reached by such simple extensions as stochastic model predictive control [69], in which the action for the first epoch is taken, after which predictions and allocations are recalculated again for the remaining/moving horizon.

5.2 Problem Statement

There are some commonly encountered problems in energy-constrained communication systems which can be generalized into a common optimization framework. As motivation, we first present these specific optimization problem formulations which arise in communications and especially wireless sensor networks. Simplifying these problems results in constrained concave maximization programs which show a similar structure during analysis. These similarities have been distilled, and an understanding of their structure and a generalized algorithm for solving them is presented. While most aficionados in optimization refer to solving these problems as convex optimization, we are discussing *concave* programs because the idea of utility *maximization* [39, 99] is common in the literature. We could just as easily

minimize the negative concave function, it is a matter of notation.

Our illustrative problems have been addressed with varying degrees of similarity (modeling differences, inclusion of the time dimension, consideration of stochasticity, *etc.*) in [19, 57, 61, 63, 95]. Recent approaches to the rate maximization problem, specifically for energy harvesting systems can be found in [57, 61, 63, 68]. We introduce such a harvesting system for the inherent constraints which it produces in the problem, allowing for intuition to be gained from the analysis therein.

It is important to note that various representations of the variable of interest p (power allocated), or q (quantization level) may be given, (or alternatively r for generic resource) but they all refer to the resource of interest for the application of current interest in the discussion. We use the (node, epoch) index $\cdot_{nk}$ or sometimes only the generic single index $\cdot_n$ when explaining parts of our analysis. Similarly, we use bold face ($\mathbf{p}$. for example) to denote the vector version of these quantities, and the plain math text with indices (p_{nk} or p_n for example) to indicate the elements of these vectors.

Weighted Rate Maximization. The problem is to select the power levels for all nodes across a finite time horizon, $p_{nk}\ \forall\ n \in [1, N],\ k \in [1, K]$, achieving the optimal sum rate:

$$\max_{\mathbf{p}}\ U_T = \max_{\mathbf{p}}\ \sum_{k=1}^{K}\sum_{n=1}^{N} \frac{w_{nk}L}{2} \log\left(1 + h_{nk}p_{nk}\right), \tag{5.1}$$

subject to the constraints induced by wireless network communications and the energy harvesting systems; U_T is the total utility, $\mathbf{p}$ is the non-negative (an implicit constraint) vectorization of all power allocations, across sensors and epochs, $h_{nk} > 0$ are the magnitudes of the fading coefficients of the communication channels, L is the length of the transmission period, and w_{nk}. The inclusion of interference between communication devices in this problem has been looked at by others (e.g., [112]), and we restrict ourselves to the above concave approximation of the SINR throughput utility. This is a reasonable assumption, since WSNs, being energy restricted, are designed to use low-power, multi-hop transmissions which reduce node-to-node interference.

The problem of distributed estimation has also received significant recent interest as a useful application of WSNs (see [19, 95] for examples of such work).

Distributed Estimation. The problem is to select the quantization dynamic levels for all sensors across a finite time horizon, $q_{nk}\ \forall\ n \in [1, N],\ k \in [1, K]$, achieving the optimal BLUE (Best Linear Unbiased) estimate [65] of the measured quantity of interest. The terms W and $\mathbf{P}_{n,k|k}$ are constants for the optimization procedure and represent the estimation and quantization noise variances. D_k^B is a bound on the BLUE estimation variance which comes from the following [19, 95], where the MQAM constellation size, $M_n^i(k)$, and rate $b_{n,k}^i$ are equal.

Lemma 8. *If the probabilities of error,* $\Pr_{n,k}$*, are uniform and bounded, then for a dynamic range of* W *and* N *sensor nodes, it holds that* $\Pr_{n,k} = \Pr \leq \min_n \frac{3}{16NW^2} p_0 \cdot \mathbf{P}_{n,i|i}^{(k)}$ *and follows that*

$$D_k^B \leq (1 + p_0) \left(\sum_{n=1}^{N} \frac{1}{\mathbf{P}_{n,k|k} + r_{n,k}^q} \right)^{-1}, \tag{5.2}$$

where $p_0 > 0$ *is a constant.*

Proof. The proof is straightforward and provided in [95]. □

The purpose of this last elucidation was to point out the fact that the communication model can be reduced to a function of only the number of dynamic levels squared, $q_{nk} = (2^{b_{nk}} - 1)^2$. We use this reciprocal BLUE variance bound as the total utility to be maximized,

$$\max_{\mathbf{q}}\ U_T = \max_{\mathbf{q}}\ \sum_{k=1}^{K} (D_k^B)^{-1} = \max_{\mathbf{q}}\ \sum_{k=1}^{K} \sum_{n=1}^{N} \frac{q_{nk}}{\mathbf{P}_{n,k|k}\ q_{nk} + W^2} \tag{5.3}$$

subject to the constraints induced by the wireless network communications and the energy harvesting systems; $\mathbf{q}$ is the non-negative (an implicit constraint) vectorization of all quantization decisions, across sensors and epochs. The scalar per sensor estimation variance is given by $\mathbf{P}_{n,k|k}$ and the dynamic range of the quantization is $[-W, W]$.

The EH-WSN problems given in the antecedent are based on the assumption of some underlying concave, non-decreasing function $U : \mathbb{R}_+ \to \mathbb{R}$. If we assume fixed constants

$a_n, b_n, c_n, d_n \in \mathbb{R}\ \forall\ n = 1, \ldots, N$ and resource levels $r_1, \ldots, r_N \geq 0$, then a generic optimization problem can be given as

$$\max_{\mathbf{r}} \sum_{n}^{N} a_n U(b_n r_n + c_n) \tag{5.4}$$

subject to some constraints which are linear functions of the resource to be allocated, e.g., a total energy constraint: $\sum_n^N d_n r_n \leq Q_{\text{tot}}$, where Q_{tot} is the limit on the sum of the scaled resource levels. It is this general formulation that motivates our forthcoming analysis. Notice that we can simply define distinct utility functions indexed by n without the simplification of using only linear modifications. That is, our problem could be put forward as

$$\max_{\mathbf{r}} \sum_{n}^{N} U_n(r_n), \tag{5.5}$$

but since our results still apply in this case, the simplified form was considered an advantage with regard to intuition gained.

5.3 Analysis of Problem Structure

An important motivation for studying concave utility maximization theory is that the concave reward for resource allocation is a near-perfect representation of practical systems. Examples include the diminishing rate for increasing transmission power, as well as improved error reduction at a decreasing return for better quantizations of data [44, 61]. We draw on well-known solutions to motivate intuition and show a progression of thought toward a more general type of solution. The analysis of these problems provides us with a structured approach to finding resource allocations resulting in maximal utility. Our algorithm results from this approach.

The novelty of our work is the analysis of a general optimization framework which can be reduced to various commonly encountered problems in energy-constrained communication systems. These optimization problems can often be formulated as the minimization of convex (or maximization of concave) functions. While there is an abundance of numerical

solutions including active set, log-barrier, *et al.*, which easily solve these problems [38, 64, 79], they do not offer any immediate intuition into the problem or grant easy extension to distributed decision making. We attempt to illuminate the problem structure and offer a straight-forward solution based on this analysis. We also specifically apply this optimization framework to constrained energy systems utilizing energy-harvesting.

Next we present the general concave utility maximization analytic solution and explore properties of concave functions which uncover our optimal iterative method. Following this, the aforementioned examples in Section 5.2 are analyzed and cast into the general solution framework.

5.3.1 Generalized Concave Utility Maximization

For the general concave utility maximization problem in (5.4), we have the following result.

Lemma 9. *Given a concave, non-decreasing function* $U : \mathbb{R}_+ \to \mathbb{R}$*, constants* $a_n, b_n, c_n, d_n \in \mathbb{R}\ \forall\ n = 1, \ldots, N$*, and the optimization problem*

$$\text{maximize } \sum_n^N a_n U(b_n r_n + c_n) \text{ subject to } \sum_n^N d_n r_n \leq Q_{\text{tot}}, \tag{5.6}$$

where Q_{tot} *is the sum limit on the resource levels* $r_1, \ldots, r_N \geq 0$*. If there exist continuously differentiable invertible functions*

$$V(r_n) = \frac{dU}{dr_n} \text{ and } \mathcal{W}(\lambda) = \sum_n^N \frac{1}{b_n}\left[V^{-1}\left(\frac{-d_n \cdot \lambda}{a_n b_n r_n}\right) - c_n\right]^+, \tag{5.7}$$

where λ *is the Lagrange multiplier (or resource price) of the constraint (and the water-filling level). Then the solution to the maximal utility problem is*

$$r_n^* = \frac{1}{b_n}\left[V^{-1}\left(\frac{-d_n \mathcal{W}^{-1}(Q_{\text{tot}})}{a_n b_n}\right) - c_n\right]^+. \tag{5.8}$$

Proof. The proof of this Lemma follows circuitously from the Lagrange analysis. But we illustrate a few of the details here. The first order necessary condition from the Lagrangian requires

$$\frac{dU}{dr_n} = a_n b_n V(b_n r_n + c_n) - \lambda = 0.$$

Solving for the resource

$$r_n = \frac{1}{b_n}\left[V^{-1}\left(\frac{-d_n \cdot \lambda}{a_n b_n r_n}\right) - c_n\right] \;\forall\; r_n \neq 0$$

and adding all non-zero allocations together we find the total is constrained as

$$\mathcal{W}(\lambda) = \sum_n \frac{1}{b_n}\left[V^{-1}\left(\frac{-d_n \cdot \lambda}{a_n b_n r_n}\right) - c_n\right] \leq Q_{\text{tot}}.$$

Setting $\mathcal{W}(\lambda) = Q_{\text{tot}}$ and solving for λ achieves the optimal Lagrange multiplier and thus reveals the remaining the unknowns. □

It is granted that $\mathcal{W}$ is usually not directly invertible, but there are usually ways to find the 'water level', λ, iteratively.

As a special case of the above Lemma, we provide the optimal allocation of constrained resources which maximizes uniform utility. We leverage the following principle to augment our understanding of the problem. Here we relax some of the index notation.

Lemma 10 (Optimal Concave Assignment)**.** *Given concave non-decreasing function* $U : \mathbb{R}_+ \to \mathbb{R}$ *and variables* $r_1, \ldots, r_N \geq 0$ *such that* $\sum_{n=1}^N r_n \leq Q_{\text{tot}}$*, with* Q_{tot} *a non-negative constant, then* $\sum_n U(r_n)$ *is maximized when* $r_1 = \cdots = r_N = \frac{Q_{\text{tot}}}{N}$.

Proof. This is a special case of the previous Lemma and we follow a similar constructive proof. The first order condition with respect to each resource provides

$$\frac{dU}{dr_n} - \lambda = 0.$$

Finding an inverse to the derivative function results in $r_n = V^{-1}(\lambda)$, which is uniform for all resources allocations. Adding up the values of the resource allocation we get $\sum_{n=1}^N r_n = \sum_{n=1}^N V^{-1}(\lambda) = NV^{-1}(\lambda) = Q_{\text{tot}}$. Thus, $r_n = V^{-1}(\lambda) = \frac{Q_{\text{tot}}}{N}\;\forall\; n$ results. □

This principle will guide the development of our algorithm.

The first-order condition for the general constrained concave utility maximization problem is

$$\begin{aligned} \frac{dU_n}{dr_n} &= \frac{dU_m}{dr_m} = \lambda \\ \frac{a_n b_n}{d_n} V(b_n r_n + c_n) &= \frac{a_m b_m}{d_m} V(b_m r_m + c_m) = \lambda \end{aligned} \tag{5.9}$$

for any non-zero allocations r_n, r_m. If we have $\frac{dU_n}{dr_n} \geq \frac{dU_m}{dr_m}$ (arbitrarily), we can write value of r_n which "levels" the marginal utilities (makes the derivatives of the utility functions equal) as

$$r_n = \frac{1}{b_n} V^{-1}\left(\frac{a_m b_m d_n}{a_n b_n d_m} V(b_m r_m + c_m)\right) - c_n. \tag{5.10}$$

If we seek an iterative solution, then the conditions in (5.9) and Lemma 10 dictate that the non-zero allocations should have evaluated derivatives which are equal. In the case where a constraint is not an issue, the optimal point would find the marginal utilities at zero. As it is, concave functions under constraints will never find such a scenario, and leveling the derivatives for all utilities associated to a common constraint suffices as the optimality criterion. If we initiate all resource allocations at zero, then the derivatives of the above concave functions (i.e., the marginal utilities) evaluated at $r_n = 0 \; \forall \; n$ will provide a starting point for incrementing the resource allocation in an intelligent way. If U is non-trivial, then $V_n(r_n)|_{r_n=0} > 0$ for at least one n. Thus, we seek to find the increase in resource $r_n = 0$ to r'_n (given $\left.\frac{dU_n}{dr_n}\right|_{r_n=0} > \left.\frac{dU_m}{dr_m}\right|_{r_m=0}$) which will reduce the marginal utility with respect to r_n until it is equal to the next largest marginal utility,

$$\left.\frac{dU_n}{dr_n}\right|_{r_n=r'_n} = \left.\frac{dU_m}{dr_m}\right|_{r_m=0}.$$

Two iterations of this type of incremental improvement are illustrated in Figure 5.1. This incremental approach does not take into account the resource constraints.

Another important aspect of solving the utility maximization problems is taking into account various energy restrictions of the problem. Beyond a total usage constraint, the system, in general, can have constraints on how the energy resources (or other resources) of

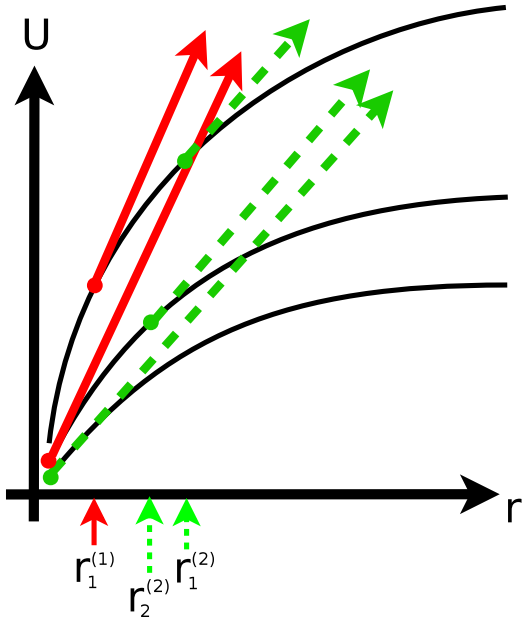


Figure 5.1: *An illustration of two iterations (first: solid, second: dashed) of the incremental improvement, which is found by increasing resource r_n (given $\frac{dU}{dr_n} > \frac{dU}{dr_m}$) which will reduce the marginal utility with respect to r_n until it is equal to the next largest marginal utility.*

the system are utilized. Next we introduce a nomenclature for a general representation of these resource constraints.

5.3.2 Generalized Energy Flow Constraints

The additional constraints of the system dictate the flow of energy between epochs, which introduce a more complexly constrained problem which our method solves. Examples of other types of constraints (beyond total usage) are found in the energy harvesting aspect which induces causality and storage limitation constraints as

$$\sum_{k=1}^{j} Lp_{nk} \leq \sum_{k=0}^{j} E_{nk} \tag{5.11}$$

$$\sum_{k=0}^{j} E_{nk} - \sum_{k=1}^{j} Lp_{nk} \leq E_{\max}, \tag{5.12}$$

$\forall\ n \in [1, N], j \in [1, K]$, with $E_{n,0}$ the initial battery level, $E_{\max}$ the battery capacity, and $E_{n,1}$ the initial collected energy available at node n. The causality constraint in (5.11) requires that energy cannot be utilized until it has been gathered. Similarly, the battery constraint in (5.12) signifies the limited energy that can be saved for future epochs.

The constraints for the determination of the optimal quantization levels for the BLUE (Best Linear Unbiased) estimate [65] in an energy harvesting WSN can be approximately (since we require a power-squared form) written using the Cauchy-Schwarz inequality as

$$\sum_{n=1}^{N} (\alpha_n^2 / h_{nk}^2) q_{nk} \leq (E_{\text{tot}}/L)^2, \tag{5.13}$$

$$(\alpha_n^2 / h_{nk}^2) q_{nk} \leq \min \left\{ E_{\max}, \left(\frac{1}{L} \sum_{j=0}^{k} E_{n,j} \right)^2 - \sum_{j=1}^{k-1} h_{n,j}^2 \alpha_n^2 q_{n,j} \right\}, \tag{5.14}$$

with α_n the pathloss, and h_{nk} the fading coefficients. We will represent a general energy (or resource) constraint as in the following. By shifting all of the quantities in the constraint equations (for example, equations (5.11) and (5.12) to the 'less than' side, we can write our constraints as: $f_{nk}^z(\mathbf{r}, \mathbf{E}) \leq 0$. We have denoted $\mathbf{r}$ as the vector quantity of the resource allocations (power or quantization level), and $\mathbf{E}$ as the vector of available energy, both nodes and epochs. The superscript represents the type of constraint. We consider causality, storage, total, or link budget constraints, but do not refer to them explicitly from here on. This general notion of a convex inequality constraint will aid us in describing the methods later presented.

We present the optimization applications of energy harvesting wireless sensor networks, problems which our method is well-suited to solve. These constrained communication problems particularly illustrate the structure which we will generalize.

When a communications system attempts to maximize rate by distributing power (or resources) to various channels with disparate losses under a resource constraint, the optimal solution obeys the following Lemma. Notice that for our concave function we have taken the log rate utility [57, 61].

Lemma 11 (Water-filling)**.** *Given N nodes, a finite constant, and $\mathbf{h} \in \mathbb{R}^{N\times 1}$ the non-negative vector of fading coefficients, the optimization problem across sensor nodes is*

$$\max_{\mathbf{p}} \sum_{n=1}^{N} \frac{L}{2} \log(1 + h_n p_n) \text{ subject to } \sum_{n=1}^{N} L p_n \leq E_{\text{tot}} \tag{5.15}$$

with respect to non-negative power allocations $\mathbf{p} \in \mathbb{R}^{N\times 1}$, has the well known water-filling solution (when the total constraint (E_{tot}) is met with equality). The optimal power allocation is

$$p_k^* := \left[\frac{1}{L\lambda(m)} - \frac{1}{h_n}\right]^+ = \left[\frac{1}{|\mathcal{A}|}\left(E_{\text{tot}} + \sum_{\sigma(i)\in\mathcal{A}} \frac{L}{h_{\sigma(i)}}\right) - \frac{1}{h_n}\right]^+,$$

$\forall\ n$*, where*

$$\mathcal{A} := \left\{ j \;\middle|\; \begin{array}{c} m \leq j \leq N \text{ where} \\ \sum_{i=m}^{N} \frac{1}{L\lambda(m)} - \frac{1}{h_{\sigma(i)}} \leq E_{\text{tot}} \\ \text{and } \sum_{i=m-1}^{N} \frac{1}{L\lambda(m)} - \frac{1}{h_{\sigma(i)}} > E_{\text{tot}} \\ \text{with } h_{\sigma(1)} \leq h_{\sigma(2)} \leq \cdots \leq h_{\sigma(N)} \end{array} \right\},$$

with $\lambda(m) = (N-m)/(E_{\text{tot}} + L\sum_{i=m}^{N} \frac{1}{h_{\sigma(i)}})$ the constraint Lagrange multiplier, $\mathcal{A}$ the set of indices of the non-zero power allocations, $|\cdot|$ the number of elements in a set, $\cdot^*$ indicating optimal, E_{tot} the total-energy-usage constraint, and $\sigma(\cdot)$ some permutation of the natural numbers $1, \ldots, N$ such that the elements of $\mathbf{h}$ are ordered as in the definition of $\mathcal{A}$ above (i.e., an increasing rearrangement of $\mathbf{h}$).

A conceptual proof is as follows. The loss levels, $\frac{1}{h_n}$, are arranged in decreasing order by the permutation $\sigma(\cdot)$. The solution can be found by iterating: adding the channel with the smallest loss, and then the next smallest, and up to the largest. The set $\mathcal{A}$ is found by excluding the loss level (and all greater than it) that produce a violation of the total energy constraint. Another proof is provided by applying Lemma 9. As water-filling is a well explored subject [100], we will elaborate no further. The point is to grasp the intuition behind this particular representation, viz., the iterative addition of nodes in the system in a manner that optimally improves utility while satisfying the constraint.

We apply a similar analysis (as in Section 5.3.1) to the weighted log rate total-energy-constrained water-filling problem, and take the problem in (5.15). Rewriting the objective as

$$U = \sum_{n=1}^{N} \frac{w_n L}{2} \log\left(\frac{1}{h_n} + p_n\right) - \frac{w_n L}{2} \log\left(\frac{1}{h_n}\right),$$

we see that the first order condition dictates that

$$\frac{w_n}{\frac{1}{h_n} + p_n} = \frac{w_m}{\frac{1}{h_m} + p_m} = \frac{\lambda}{L} \; \forall \; n, m \in \mathcal{A}, \tag{5.16}$$

where λ is the total energy constraint Lagrange multiplier and $\mathcal{A}$ is the set of non-zero power allocations. Thus, the first iteration towards an improved (and optimal according to Lemma 10) allocation would be

$$p_m = \frac{w_n}{w_m} \frac{1}{h_m} - \frac{1}{h_n}$$

where $\frac{1}{w_m h_m} > \frac{1}{w_n h_n}$ in the initial problem.

Our work in [68] analyzes the water-filling problem for systems with energy-harvesting constraints. We now develop further intuition about concave utility maximization techniques.

Application 1: Sum Rate Maximization

The objective of this section is to understand the structure of the water filling solutions with various different types of constraints by looking at the KKT conditions [38]. We employ the energy-causality and storage constraints associated with energy harvesting (in (5.11) and (5.12)). What we hope to find is a general rule for how these water-filling solutions can be obtained. Some of this analysis was presented in [68]. The marginal utility for the rate utility is

$$V_{nk}(p_{nk}) = \frac{1}{1 + h_{nk} p_{nk}}, \tag{5.17}$$

and we utilize the result of the KKT analysis for a total constraint, following the form of Lemma 11. If we focus on the energy causality and limited storage constraints (i.e., for a single node, n) the Lagrangian becomes

$$\begin{aligned}\mathcal{L}_n &= \sum_{k=1}^{K} w_{nk}\log(1+h_{nk}p_{nk}) \\ &- \sum_{j=1}^{K}\lambda_{nj}\left(\sum_{k=1}^{j}Lp_{nk}-\sum_{k=1}^{j}E_{nk}\right) \\ &- \sum_{j=1}^{K}\mu_{nj}\left(\sum_{k=1}^{j}E_{nk}-\sum_{k=1}^{j}Lp_{nk}-E_{max}\right),\end{aligned} \tag{5.18}$$

where λ_{nj} and μ_{nj} are the Lagrange multipliers of the causality and energy storage constraints, respectively. The first order condition produces the optimal power allocation (across epochs),

$$p_{nk}^{*} = w_{nk}\left[\frac{1}{\sum_{j=k}^{K}L(\lambda_{nj}-\mu_{nj})}-\frac{1}{w_{nk}h_{nk}}\right]^{+}. \tag{5.19}$$

Motivated by this solution and the complementary slackness conditions, a result detailed in [57, 63] is the following.

Lemma 12. *For $E_{\max}=\infty$ (which implies $\mu_{nj}=0\ \forall\ n,j$) and uniform w_{nk}, regardless of whether all the energy in each epoch is used in that epoch (whether $\lambda_{nj}=0$ for any $k<K$), the water levels $\nu_{nk}=\frac{w_{nk}}{L\sum_{j=k}^{N}\lambda_{nj}-\mu_{nj}}$ will monotonically increase, since $\lambda_{nj}\geq 0\ \forall\ n,j$.*

The proof of this is found in [57] and does not necessarily hold for finite $E_{\max}$. This Lemma provides a packet of intuition which exposes a unique aspect of the the water filling problem in Lemma 11, as we explain in the following example.

Example If we are given a scenario satisfying $E_{n1}\leq E_{n2}\leq\cdots\leq E_{nk}$ and $w_{n1}h_{n1}\leq w_{n2}h_{n2}\leq\cdots\leq w_{nK}h_{nK}$, with $\sum_{k=1}^{K}E_{nk}\leq E_{\max}$. Such an ordering provides us with a solution where energy only flows to future epochs. Since by construction, the battery constraint is never violated, epochs with the largest $w_{nk}h_{nk}$ will contain the largest power levels. In fact, the allocation mimics that of the solution in Lemma 11. This is because the

fading levels and energy availability are 'pre-ordered' such that the there is no re-ordering necessary to find the optimal common 'water level' over the epochs. Thus, when reordering is not possible due to causality constraints, these constraints become actively involved any algorithmic solution.

Similarly, if the available energy is ordered the same but with the weighted fading coefficients in reverse order, we have the following.

Corollary 1. *If $\lambda_{nj} \neq 0 \ \forall \ n, j$ then the optimal power allocation is*

$$p_{nk}^{*} = \frac{E_{nk}}{L} \ \forall \ n, k \tag{5.20}$$

and this solution is only possible when the following hold:

$$E_{n1} \leq E_{n2} \leq \cdots \leq E_{nK}$$

and

$$w_{n1}h_{n1} \geq w_{n2}h_{n2} \geq \cdots \geq w_{nK}h_{nK}.$$

The conceptual proof is as follows. Since the weighted fading loss $(1/(w_{nk}h_{nk}))$ is least in the beginning epochs, more energy should be allocated to them. However, energy availability is also the least in the beginning epochs, and since causality prevents energy from reaching past epochs, only E_{nk} is available. No energy flows further since this would move away from the optimal solution of Lemma 11 (without causality), thus all of the $\lambda_{nj} \neq 0$ since the constraints are strict. We next perform a similar analysis the distributed estimation application.

Application 2: Distributed Estimation

The analysis of the distributed estimation problem takes a similar feel to the sum rate problem, except that the Lagrangian is

$$\begin{aligned}\mathcal{L} &= \sum_{nk} \frac{q_{nk}}{\mathbf{P}_{n,k|k} q_{nk} + W^2} \\ &+ \sum_{k} \lambda_k \left(\sum_{n} (\alpha_n^2 / h_{nk}^2) q_{nk} - (E_{\text{tot}}/L)^2 \right) \\ &+ \sum_{nk} \mu_{nk} ((\alpha_n^2 / h_{nk}^2) q_{nk} \\ &- \min \left\{ E_{\max}, \left(\frac{1}{L} \sum_{j=t}^{t+k} E_{nj} \right)^2 - \sum_{j=t}^{t+k-1} h_{nj}^2 \alpha_n^2 q_{nj} \right\}).\end{aligned} \tag{5.21}$$

and the marginal utility function is

$$V_{nk}(q_{nk}) = \frac{1}{h_{nk}^2 \alpha_{nk}^2} \frac{W^2}{(\mathbf{P}_{n,k|k} q_{nk} + W^2)^2}. \tag{5.22}$$

Differentiating the Lagrangian results in the following solution

$$q_{nk} := \frac{1}{\mathbf{P}_{n,k|k}} V_{nk}^{-1} \left(\left[\frac{W}{\sqrt{\nu \cdot (\alpha_n^2 / h_{nk}^2)}} - W^2 \right]^+ \right) \tag{5.23}$$

where ν is a collection of non-zero Lagrange multipliers for the active constraints (this could be λ's or μ's). In the case that only the per epoch total resource constraint [19] is active in epoch k, then

$$\nu(n') := \frac{W \sum_{n=n'}^{N} (\alpha_n / h_{nk}) / \mathbf{P}_{n,k|k}}{W^2 \sum_{n=n'}^{N} (\alpha_n / h_{nk}) / \mathbf{P}_{n,k|k} + (E_{\text{tot}}/L)^2} \ \forall \ k. \tag{5.24}$$

If we assume without loss of generality that the values $\frac{\alpha_1}{h_{1k}} > \cdots > \frac{\alpha'_n}{h_{n'k}} > \cdots > \frac{\alpha_N}{h_{Nk}}$, then the values of q_{nk} are non-zero when $\nu(n') \frac{\alpha_{n'}}{h_{n'k}} W < 1$. The solution to the above is calculated in a similar way to the previous example for the case of multiple active constraints. These solutions however can be found with the method in the next section.

The primary purpose of these examples is to illustrate the role of the marginal utility, along with the evaluation of the Lagrange dual variable, in the calculation of the resource

usages. Using our understanding of how to best maximize a sum of concave utilities, along with the extraction of the typical form for the marginal utility, allows us to utilize the following iterative algorithm.

5.4 Algorithmic Solution

In order to complete the description of our algorithm for iterative general utility function resource allocation, we must look at feasibility and convergence. As opposed to other types of concave function maximization schemes with updates which must often project resource allocation decisions back into the feasible space [59], we propose a solution which achieves constraints (or remains feasible) exactly in each update. It could be considered a type of minimum cost network flow problem [113, 114], but for the use of concave reward functions. Therefore, our method does not require gradient-based updates or other such iterations, however, convergence is understood from the following Lemmas.

Our method is presented in Algorithm 1 for maximizing generalized constrained concave utility generalized problems. The Minimum-Increment Marginal Matching method initiates all power allocations to zero. It then finds the minimum increment which is the difference between the maximum marginal utility and *next maximum* marginal utility. The method then computes the hypothetical largest resource flow allowed while maintaining the strict (or active) feasible constraints. The maximum dual price is found from amongst all the utilities for the resources associated with the largest feasible increment toward meeting a constraint. The maximum between the maximum price increment (MPI) from associated constraints and the next maximum marginal utility (MMU) is the new price and is used to find the new allocation. It is essential to note that the maximum marginal utilities found on line 3 of Algorithm 1 are almost never singletons, which is the approach of the method: when common maximums are found, these resources are incremented until the marginal utility matches the next largest marginal utility, or they are increment *evenly* based on available resources.

Algorithm 1 Minimum-Increment Matched Marginal (MIMM) for General Concave Utility Maximization

Init. $\mathbf{r} = \mathbf{0}_{NK}$, $\mathcal{M} \leftarrow \varnothing$, $\mathcal{B} \leftarrow \mathcal{Z}$
while there exists some resource allocations for which there is slack in all of its associated constraints **do**
    (MMU) Store in $\mathcal{M}$ the indices of the allocations associated with *the* maximum marginal utility; find the 'next' maximal marginal utility (λ^{MMU+}) price decrement to satisfy to Eq. (5.9)
    (MPI) Find the maximum price (λ^{MIP+}) for the constrained largest allowable increase (as in Eq. (5.26) from Eq. (5.28)) among the active allocations in $\mathcal{M}$
    Update the active resource allocations (for each index in $\mathcal{M}$) using the maximum price between the MMU and MPI steps, and then assign resources using Eq. (5.25)
    Update the constraints to reflect remaining resources
    if a resource constraint is met with equality **then**
        remove any resources indices from $\mathcal{B}$ so that they do not update further
    end if
    if a lower bound constraint (e.g., battery) is achieved **then**
        split remaining nodes to allocate into $\mathcal{B}$ and $\mathcal{B}'$ independently solvable subproblems and execute further iterations on each
    end if
end while

For the next discussion, similar to the set $\mathcal{A}$, we define $\mathcal{M}$ as the set of 2-tuples (n, k) (could be any number of dimensions) indexing the resource allocations which have maximal marginal utilities all matched according to Lemma 9 (following from 10). In the case of the sum rate maximization example, if any constraints have slack for the non-zero power allocations, then there exists for each allocation in $\mathcal{M}$ a maximum increment in flow restricted by the associated constraints. Mathematically put, an iteration of the algorithm at such a juncture finds

$$r_{nk}^{MIP+} := r_{nk} + \min_{f_{mj}^z : (nk) \in \mathcal{F}(f_{mj}^z)} \left\{ \frac{-f_{mj}^z(\mathbf{r}, \mathbf{E})}{|\mathcal{M} \cap \mathcal{F}(f_{mj}^z)|} \right\} \tag{5.25}$$

for all $(n, k) \in \mathcal{M}$. $\mathcal{F}(\cdot)$ extracts the set of indices associated with the constraint supplied in the argument, $|\cdot|$ is the cardinality of a set, and $\cap$ is the set intersection. In the case of the sum rate problem, if the maximum price increment (MPI) marginal utility,

$$\lambda^{MIP+} = V_{nk}(r_{nk}^{MIP+}) = V_{mj}(r_{mj}^{MIP+}) \tag{5.26}$$

$\forall\ (n,k),(m,j) \in \mathcal{M}$, is greater than the marginal utility found in step 3 (MMU), then the MIP allocation is selected. When the MIP update is selected, a constraint becomes active and the allocations associated to this constraint are optimal and complete. If the MMU price is greater, then it is selected for the update. Summarizing, the new resource allocation is

$$r_{nk}^{+} := V_{nk}^{-1}(\max\{\lambda^{MMU+}, \lambda^{MIP+}\}). \tag{5.27}$$

In the case of more general utility functions, we must find the maximum dual variable price restricted by remaining resource by solving a smaller problem, we cannot simply divide the resource uniformly. We use the computation

$$r_{nk}^{MIP+} := \arg \left\{ \begin{array}{l} \max_{r_{nk},(n,k)\in\mathcal{M}} \ \sum_{(n,k)\in\mathcal{M}} U_{nk}(r_{nk}) \\ \text{subject to} \\ \ f_{mj}^{z} \leq 0\ \forall\ (m,j) : (n,k) \in \mathcal{F}(f_{mj}^{z}) \end{array} \right\}, \tag{5.28}$$

to solve for the resource allocation in the subproblem considering only $(n,k) \in \mathcal{M}$, and then calculate the maximum dual price as before. The maximum among MMU and MIP prices is taken and this price is then used to increment the resource allocations as in Eq. (6.18).

Using the example of rate maximization, it is intuitive that if, for example, any sensor and epoch (n,j) achieves its causal energy constraint, i.e., $\sum_{k=1}^{j} Lp_{nk} = \sum_{k=1}^{j} E_{nk}$, then all of the previous epoch power levels cannot be increased without violating the constraint. When a such condition is met, then the epochs $1,\ldots,j$ for that sensor are removed from future updating and the $p_{nj},\ j = 1,\ldots,k$ values are final. There is a similar procedure followed for any constraints which become active, wherein all of the resource allocations associated with the achieved constraint are optimal and removed from further updates.

Another important aspect of the algorithm is how to deal with a battery at maximum capacity. Since the method has found a partial solution where no more energy can flow to later epochs, the problem is decomposed in the epochs before, between, and after the achieved constraints. These new subproblems can be solved optimally since resource usage before and after the epoch independent, i.e., the marginal utilities of these sub-problems need

not match, but only follow the optimality conditions within themselves. This algorithmic formulation can readily be applied to the energy-harvesting distributed estimation problem in (5.3), as well as being a generalization of several previous results [19, 57, 61, 63, 95]. A sub-class of this algorithm was presented in [68]. It is assumed that the utilities in this maximization problem are concave. From this we offer the following.

Theorem 5. *Algorithm 1 converges to an optimal solution. That is, the sequence generated by the algorithm satisfies sufficient conditions of optimality.*

Proof. See Appendix A. □

The algorithm complexity for solving the sum rate maximization is $O(N^3K^3)$ as the upper bound. This follows since the deepest `for` loop include K possible operations done possibly N times (each sensor and time instance meets a constraint, each one at a time), subsumed by an possible NK iteration loop within the final possible NK iteration loop. Scenarios with close fading levels or not achieving battery limits much produce much quicker runtimes. Convex programming methods have complexity $O(N^3K^3)$. The benefit gained is a model-specific algorithm that proceeds from a greater understanding of the underlying behavior. The general solution can require slightly more computation as the subproblem in 5.28 must be solved in each update. However, this decomposition of the problem by associated constraints is appealing in the case of a decentralized implementation, where it is already required that more iterations are required in obtaining a solution simply because there is no single controller making a centralized decision.

5.5 Simulation Results

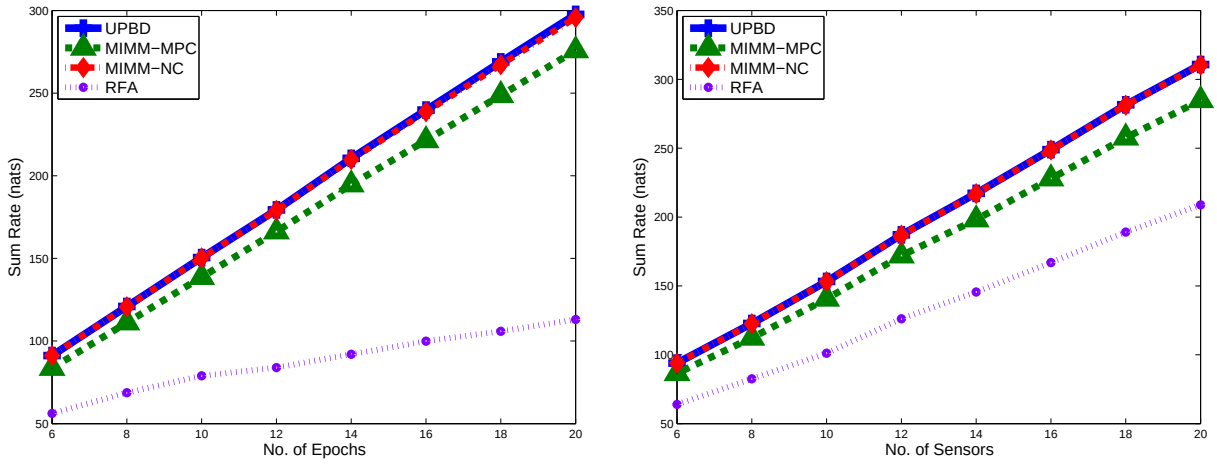
We demonstrate the use of Algorithm 1 on the sum rate maximization and the distributed estimation problems. In both cases we adopt the energy harvesting model in [57] with Poisson harvested energy arrivals. However, we 'collect' energy until the beginning of the next epoch to use for transmission. We vary the number of sensors, number of epochs, and

the maximum Doppler frequency to illustrate behavior of the problem and compare various solutions. When not being varied, the default values of these variables of interest are five sensor/transmitters and five epochs in the scenario. The default maximum Doppler spread is $1Hz$, and a time step of $1s$ (moderately correlated) is used in the generation correlated, Rayleigh distributed, fading coefficients across time [93]. We utilize a linear Wiener predictor [86] to estimate the future coefficients. The mean of the harvested energy stochastic process is used to estimate future collected energy. In each of our test scenarios a fixed number of epochs is used.

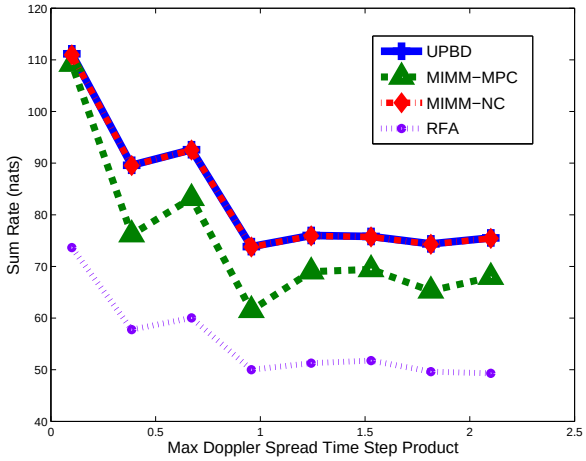
5.5.1 Rate Maximization

We compare our method (MIMM) against the numerical convex programming solution with non-causal information applied to the problem of maximal rate throughput over fading channels. The numerical convex programming solution serves as an upper bound (UPBD) (see Section 2.2 for more information and references concerning standard techniques and algorithms). We use a Model Predictive Control method [69] as a computationally viable alternative to full, infinite-horizon stochastic dynamic programming. See more on the model predictive control strategy in Section 2.4.3. Thus, we execute Algorithm 1 with causal information with MPC (MIMM-MPC) using the above predictors. We also compare the solution obtained using our method with non-causal information (MIMM-NC). A Random Feasible Allocation (RFA) heuristic is included for comparison. The RFA method randomly selects a node/epoch to update with a maximum improvement such that all associated constraints remain feasible.

We execute the above scenario over 100 Monte Carlo runs to obtain the averaged results in Figure 5.2. Figure 5.2(a) shows the sum rate performance of the MPC and non-causal MIMM solution versus the upper bound as a function of increasing epoch length for five sensors ($N = 5$). Figure 5.2(b) illustrates variation of the sum rate versus sensors for $K = 5$ epochs. Preserving the time step of $1s$, Figure 5.2(c) shows sum rate versus maximum



(a) (b)



(c)

Figure 5.2: *Application 1: sum rate maximization performance with respect to variation of (a) number of epochs, (b) number of sensors, (c) maximum Doppler spread.*

Doppler spread for $N = 5$ and $K = 5$. Smaller Doppler-spread produces more highly correlated fading and better sum rate performance. We see that the MPC-like method does reasonable well using only the mean from the statistics of the harvested energy and fading coefficients. We also see that MIMM method achieves the upper bound when using non-causal information.

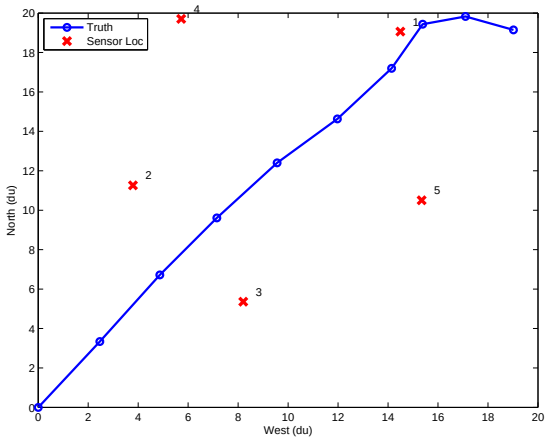
5.5.2 Distributed Estimation

Similar to the previous application, we compare our MIMM method, with the non-causal numerical convex programming solution (the upper bound (UPBD)) applied to the energy-harvesting distributed estimation problem. The MPC-based MIMM is applied to the distributed estimation utility function, and Algorithm 1 is executed with causal information (MIMM-MPC) using predictors as described at the beginning of the results section. The non-causal version of our algorithm is again denoted MIMM-NC. The same Random Feasible Allocation (RFA) heuristic is included for comparison.

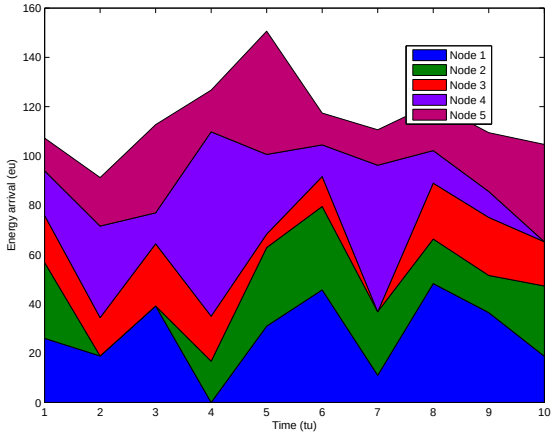
The distributed estimation scenario uses the following setup: a single, nearly-constant-velocity target (cf. [86]) starts at $(0, 0)$ in a two dimensional 'field' of $20du$ by $20du$ ($W = 10$) and is propagated with random process noise under the constraint that it stays within the field. The positions of the WSN nodes are uniformly randomly generated. Both the correlated fading and the path loss from the sensor nodes to the fusion center at the center of the field are used in calculated the cost of communication. We assume linearized range-angle measurements for the sensors. An example single run is shown in Figure 5.3 which demonstrates the dependence of the resource allocation decisions on predicted local filtered state accuracy, available energy, and the fading-over-pathloss (h_{nk}/α_n: how energy efficient it is to transmit). Figure 5.3(a) shows the positions of the sensor nodes with $\times$'s and the trajectory of the moving object with a line marked with $\circ$'s at the sample points. At the beginning of the example, nodes 2 and 3 observe the sensor best and have inexpensive transmissions, while at the end nodes 1 and 5 both have low estimate error and but node

4 has the most inexpensive cost of transmission and communicates the most information about the target.

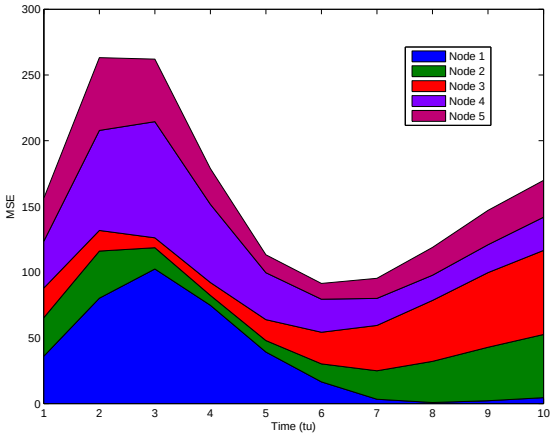
The scenario was evaluated for 100 Monte Carlo runs to obtain the averaged results in Figure 5.4. The results in these figures show a similar behavior as the rate maximization problem, with increasing accuracy for more sensors. For multiple time instances we also find an increased accuracy for more epochs in the scenarios (since from the definition of our utility, we are adding up the inverse error across time instances). Figure 5.4(a) shows the accuracy performance of the MPC and non-causal MIMM solution versus the upper bound as a function of increasing epoch length for five sensors ($N = 5$). Figure 5.4(b) illustrates variation of the estimation accuracy versus sensors for $K = 5$ epochs. Preserving the time step of $1s$, Figure 5.4(c) shows the distributed estimation accuracy versus maximum Doppler spread for $N = 5$ and $K = 5$. As in the rate maximization problem, smaller Doppler-spread produces more highly correlated fading, and therefore a more predictable communication channel which improves the estimation accuracy. Note also, that since the correlation of fading coefficients has a Bessel function-like behavior, the resulting correlation affects the coherence time (and therefore overall performance) in a Bessel function-like periodic decay for increasing maximum Doppler spread time step product. We see that the MPC-like method does extremely well (using only the predicted mean of the random quantities). We also notice that for an increasing number of sensors, there is a slight difference in performance from the non-causal solutions, while for increasing number of epochs the performance of the MPC-like method is also only slightly different from the upper bound. The model predictive control method reevaluates the solution at each epoch for a limited horizon (applying only the decision for the first epoch of the horizon). Thus, it is inferred (and corroborated by the simulations) that the overall performance is only mildly worsened by scenarios with an increasing number of epochs. We again see that MIMM method achieves the upper bound when using non-causal information. In summary, the new method achieves the same solutions (numerically) as standard optimization methods and can be used adeptly



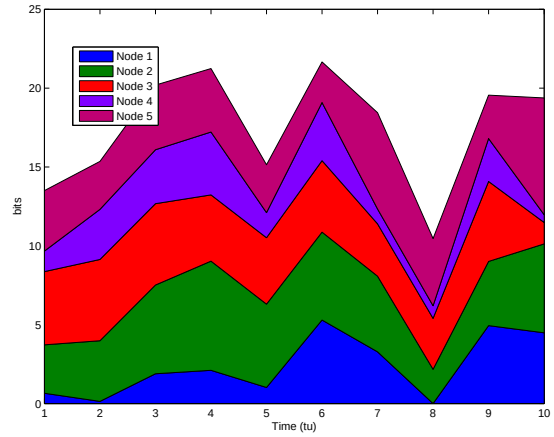
(a) Object Trajectory



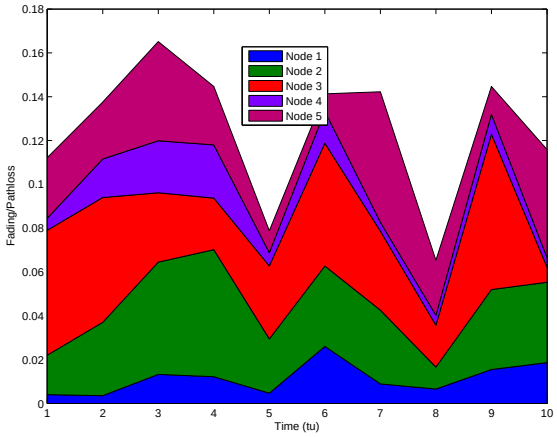
(b) Energy Harvested



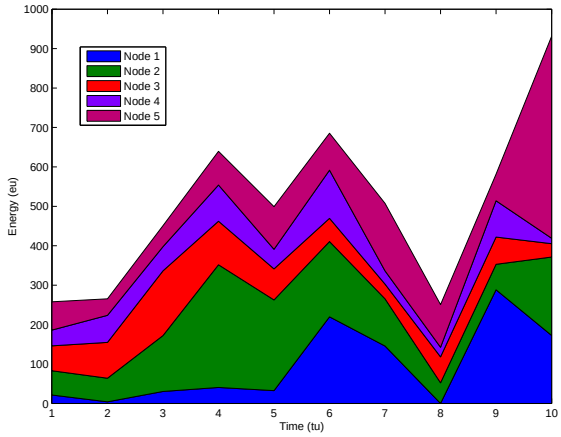
(c) Local Filter State Uncertainty



(d) Bits Utilized



(e) Fading over Pathloss



(f) Energy Utilized

Figure 5.3: *An example single run in which illustrates the dependence of the optimal resource allocation decisions ((d) bandwidth and (f) energy spent) on (b) available energy, (c) predicted local filtered state accuracy, and (e) the fading-over-pathloss. The values are layered node 1 to 5 from bottom to top. If viewing in color: node 1) blue, 2) green, 3) red, 4) purple, 5) maroon.*

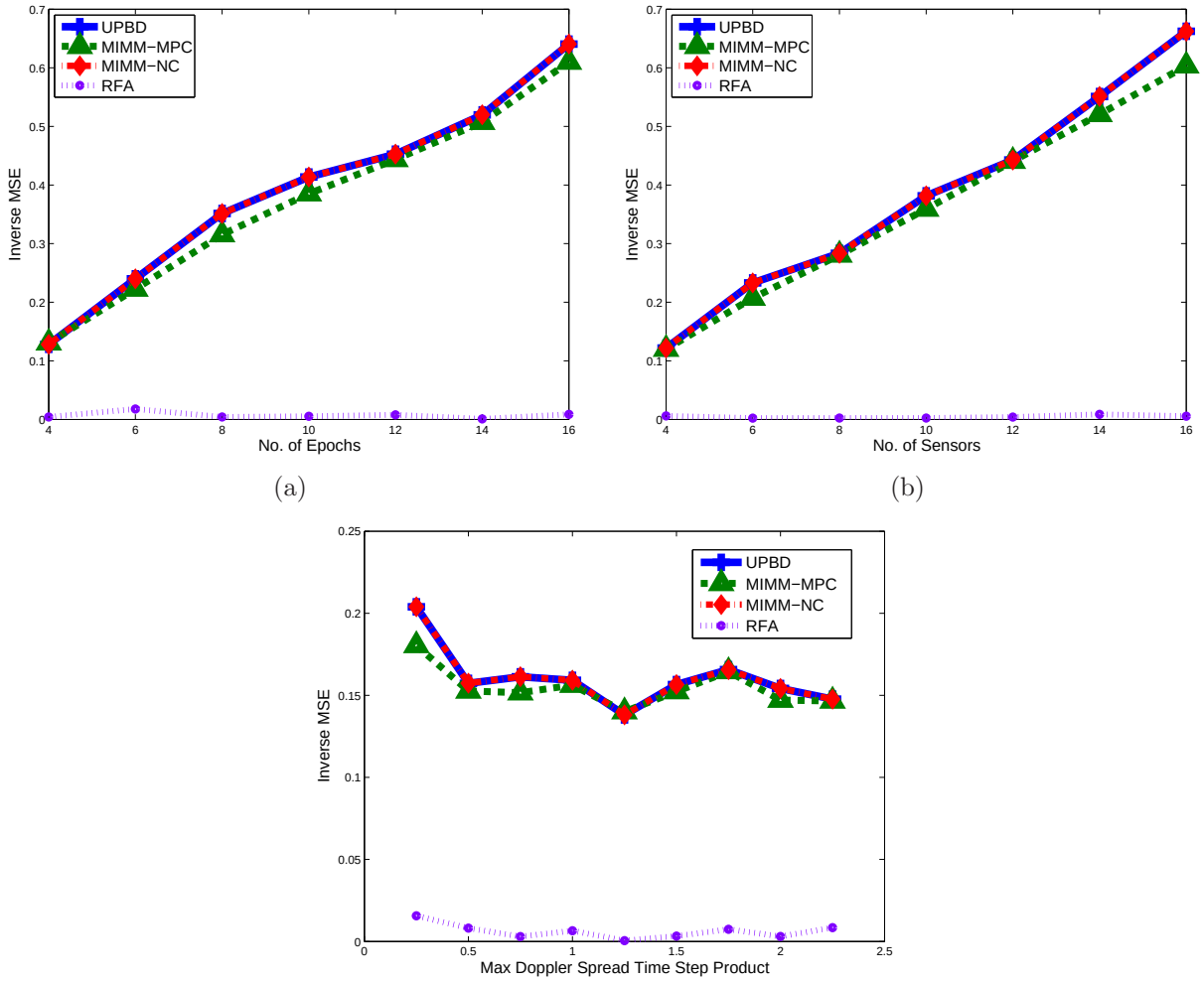


Figure 5.4: *Application 2: distributed estimation performance with respect to variation of (a) number of epochs, (b) number of sensors, (c) maximum Doppler spread.*

in conjunction with approximate stochastic control techniques to produce viable causal resource allocation solutions.

5.6 Summary

This chapter has presented an alternative perspective for resource allocation in energy harvesting wireless sensor networks. Motivated by the analysis of structure of such problems an intuitive algorithm has also been presented. This chapter attempts to combine and balance the intuition gained from simple analytic solutions and finding of solutions to more complex constrained problems with non-uniform utilities. We demonstrate the application of a general solution to the specific EH-WSN problems, resulting in well-known problems/solutions (i.e., extended cases of water-filling). The next stage of this work is to develop an algorithm for decentralized utility maximization problems. Our approach will allow significant problem decomposition. We have shown by proof and equivalence to numerical solutions that our concave utility maximization algorithm is optimal. More importantly, we have attempted to provide intuition into how constrained utility maximization solutions can be obtained.

Our next chapter will analyze how the above analysis and algorithm can be put to work in decentralized WSN optimization, i.e., no central controller dictates the actions of the individual nodes. We will analyze the convergence of the methods, as well as the delay and error associated with utilizing such distributed optimization techniques in energy-harvesting wireless sensor networks.

Chapter 6

Decentralized Optimization in Energy Harvesting WSNs

Optimization across a set of agents or nodes connected by a wireless sensor network is difficult, but can be an even more daunting enterprise without a centralized controller. This problem of decentralized decision making, in a general sense, is a well analyzed problem [45]. Many of the facets of the decentralized system are also present in centralized ones, therefore, we move beyond some of those which are overlapping, and look at the aspects unique to decentralized systems. In particular, communications has a significant impact on the performance of decentralized optimization. If perfect and instantaneous communications is assumed (all information about the problem is known to all the agents), then the decentralized case is equivalent to centralized optimization. The communication system adds a complicating component to the decision process, e.g., delay, quantization noise, additional resources for communication, *et cetera*. Even simple problems become quite complex and difficult to solve when the costs and constraints related to communication are included in the optimization problem.

The effort of this chapter is to extend the results of the previous chapter to decentralized optimization of wireless sensor network communications. We first reintroduce the dual formulation presented in Section 2.2 and discuss the decomposition of relaxed objectives for individual agents and explain how the constraint multipliers (dual variables) link these

individual objectives so that an iterative algorithm can reach the final solution of a constrained resource problem. We also provide a modified version of our matched marginal utility algorithm for decentralized optimization. Finally, we give some analysis of the bound on the optimization solution with respect to delay and quantization, and provide a probabilistic analysis of this bound with respect to an energy harvesting WSN utilizing various communication policies.

6.1 Dual Decomposition for Distributed Networks of Agents

We again consider an optimization problem, where a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (Figure 6.1) of N interconnected agents are maximizing the sum of their individual utility functions, $U_n : \mathbb{R}^d \to \mathbb{R}$, as

$$U^* := \max_{\mathbf{r} \in \mathcal{C}} \sum_{n=1}^{N} U_n(\mathbf{r}). \tag{6.1}$$

Assumption 1. *The utilities $U_n(\cdot)$ are concave (potentially non-smooth) functions and the constraint set $\mathcal{C} \subseteq \mathcal{D} \subseteq \mathbb{R}^d$ is non-empty and convex, where $\mathcal{D}$ is the non-empty convex domain of the utility functions.*

The constraint set $\mathcal{C}$ represents the bounds the states in $\mathbf{r}$ can take, where r_n is an element of $\mathbf{r}$ and each U_n depends on a subset of the r's. This maximization is usually subject to a set of constraints among the nodes, whether between neighboring nodes, or constraints on communication routes between source and sink nodes. The parameters of each objective function U_n are assumed to be known to neighboring nodes $m \in \mathcal{N}_n$, thus the agents $\mathcal{V} = \{1, \ldots, N\}$ need to coordinate their decisions through a limited set of communication links $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The goal is to solve the maximization problem in a decentralized fashion, where the agents cooperatively find the optimal solution without a central coordinator. We propose a procedure based on a modification of dual decomposition and the alternating direction method of multipliers (ADMM). Toward this end, let us discuss the dual decomposition of our problem.

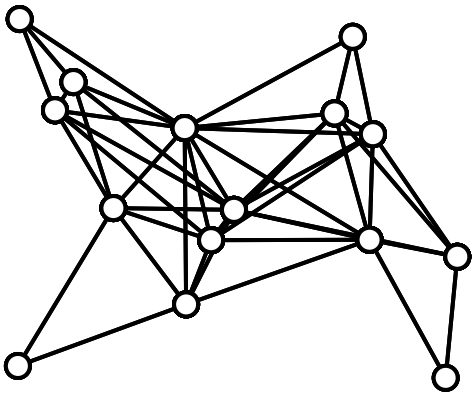


Figure 6.1: *An illustration of the graph-like nature of a multi-agent system. Each agent is represented by a ◯ and the edges which join them are communication connections.*

We first introduce local decisions (or estimates) $\mathbf{r}^n \in \mathcal{C}$. While this setup could define the problem of aligning decisions across distributed agents, in this work, we do not limit the scope to of this problem alone. The optimization problem in (6.1) can then be written as

$$\begin{aligned} &\max_{\mathbf{r}^1,\ldots,\mathbf{r}^N \in \mathcal{D}} \sum_n U_n(\mathbf{r}^n) \\ &\text{subject to } h_{ni}(\mathbf{r}^n) = 0,\ g_{nj}(\mathbf{r}^n) \leq 0 \\ &\qquad \forall\ i = 1,\ldots,I_n,\ j = 1,\ldots,J_n\ \forall\ n, \end{aligned} \tag{6.2}$$

where the equality and inequality constraints now represent the definition of the region $\mathcal{C}$. Notice also that any possible coupling between the objective functions is moved from the decision variable to the consistency constraints ($h_{ni}(\mathbf{r}^n)$ for some i and each n). We utilize vector notation for the set of constraints associated with each agent, i.e, the constraints

$$h_{ni}(\mathbf{r}^n) = 0,\ g_{nj}(\mathbf{r}^n) \leq 0\ \forall\ i = 1,\ldots,I_n,\ j = 1,\ldots,J_n$$

can be written more concisely as $\mathbf{h}_n(\mathbf{r}^n) = \mathbf{0},\ \mathbf{g}_n(\mathbf{r}^n) \leq \mathbf{0}$. Next, we partition the decision variable $\mathbf{r} \in \mathcal{C}$ into N parts such that each part can be associated with a unique agent,

$$\mathbf{r} = [r_1^T,\ r_2^T, \ldots,\ r_N^T]^T, \tag{6.3}$$

where $\cdot^T$ is the matrix/vector transpose and, in general, each $r_n \in \mathbb{R}^{M_n}$ can be any size, with arbitrary partitioning for $\sum_{n=1}^N M_n = d$. For simplicity in the following, we assume

$M_n = 1 \; \forall \; n$, since in many applications this is a natural partitioning, where r_n represents the internal state of agent n. In general, $r_m^n, \;\; n, m \in \mathcal{V}$ is the local decision/estimate at agent n of the internal state of agent m.

Remark 5. *Later in our setup, we consider the case where the local decisions in* $\mathbf{r}^n$ *include only the local agent,* n*, and the neighboring nodes* $m \in \mathcal{N}_n \backslash \{n\} \subset \mathcal{V}$ *(connected by an edge in* $\mathcal{E}$*) in its set of local estimates, i.e.,* $\mathbf{r}^n = \{[r_m^n, \ldots, r_{m'}^n]^T |\; m, m' \in \mathcal{N}_n\}$ *(where, for simplicity we assume* $\mathcal{N}_n$ *includes agent* n*).*

The dual variables $\lambda_{ni}, \;\; i = 1, \ldots, I_n$ and $\mu_{nj}, \;\; j = 1, \ldots, J_n$, are associated with each of the equality and inequality constraints, respectively. For notational simplicity, we write all of these Lagrange multipliers as

$$\nu := [\lambda_{11}, \ldots, \lambda_{ni}, \ldots, \lambda_{NI_N}, \mu_{11}, \ldots, \mu_{nj}, \ldots, \mu_{NJ_N}]^T. \tag{6.4}$$

The Lagrangian $\mathcal{L}$ is defined as [79]

$$\begin{aligned} \mathcal{L}(\mathbf{r}^1, \ldots, \mathbf{r}^N, \nu) \;\; &:= \;\; \sum_{n=1}^{N} U_n(\mathbf{r}^n) \\ &- \sum_{n=1}^{N} \left(\sum_{i=1}^{I_n} \lambda_{ni} h_{ni}(\mathbf{r}^n) + \sum_{j=1}^{J_n} \mu_{nj} g_{nj}(\mathbf{r}^n) \right), \end{aligned} \tag{6.5}$$

and the corresponding Lagrange dual function is q the supremum (remember our problem involves maximization in the primal formulation) with respect to the primal variables $\mathbf{r}^1, \ldots, \mathbf{r}^N$,

$$q(\nu) := \sup_{\mathbf{r}^1, \ldots, \mathbf{r}^N \in \mathcal{D}} \mathcal{L}(\mathbf{r}^1, \ldots, \mathbf{r}^N, \nu). \tag{6.6}$$

We can now rewrite the Lagrange dual function by formulating subproblems $\phi^n(\nu)$ as

$$\begin{aligned} \phi^n(\nu) \;\; &:= \;\; \sup_{\mathbf{r}^n \in \mathcal{D}} \mathcal{L}_n(\mathbf{r}^n, \nu) \\ &:= \;\; \sup_{\mathbf{r}^n \in \mathcal{D}} \; U_n(\mathbf{r}^n) - \lambda_n^T \mathbf{h}_n(\mathbf{r}^n) - \mu_n^T \mathbf{g}_n(\mathbf{r}^n). \end{aligned} \tag{6.7}$$

We have utilized the vector form of the inequality constraint mentioned previously and λ_n and μ_n (note the single index) are the dual variable vectors of the associated vector constraint

functions. From this definition, the subproblems only depend on the dual variable, ν, and is the supremum over the local variable $\mathbf{r}^n$. Hence, agent n computes $\phi^n(\nu)$ locally, independent of all other agents. Further, the Lagrange dual function is the sum of all the subproblems,

$$q(\nu) = \sum_{n=1}^{N} \phi^n(\nu). \tag{6.8}$$

Concluding our short review of dual formulations, the Lagrange dual problem is the minimization of the Lagrange dual function,

$$q^* := \min_{\nu} q(\nu) = \min_{\nu} \sum_{n=1}^{N} \phi^n(\nu). \tag{6.9}$$

Under Assumption 1 (U_n's concave) Slater's condition [38] guarantees zero duality gap, or equivalently, that the dual and primal solutions will be the equal, i.e., $q^* = U^*$. Thus, we can solve the Lagrange dual problem instead of the original primal optimization problem.

6.2 Modified Dual Optimization Algorithm

We can relieve ourselves of the necessity of coupling constraints by considering the availability of information about the utility in neighboring agents. This is a reasonable assumption for some applications. For example, in a network of wireless sensor nodes which communicate over slow fading channels, the achievable rate over a channel when transmitting from node n to m is the same when m replies to n. The problem that remains focuses on the interaction of the internal states of the agents through other constraints, namely, inequality constraints. The individual Lagrangian subproblems thus become

$$\phi^n(\nu) := \sup_{\mathbf{r}^n \in \mathcal{D}} \; U_n(\mathbf{r}^n) - \lambda_n^T \mathbf{h}'_n(\mathbf{r}^n) - \mu_n^T \mathbf{g}_n(\mathbf{r}^n),$$

where $\mathbf{h}'_n(\mathbf{r}^n)$ is the reduced equality constraint vector (i.e., not including consistency constraints). We have kept the internal states of neighboring agents in the evaluation of the constraint functions.

We implement an augmented Lagrangian to deal with the maximization with respect to inequality constraints. This necessitates the inclusion of a set of slack variables, $\mathbf{u}^n = [u_{n1}, \ldots, u_{nJ_n}]^T,\ u_{nj} \geq 0$, for the inequality constraints of agent n.

A subgradient-based alternating direction multipliers method has update steps which take the form

$$\begin{aligned}
\mathbf{r}^{n^{(k+1)}} &:= \max_{\mathbf{r}^n \in \mathcal{D}} \{\ \mathcal{L}_n(\mathbf{r}^n, \nu) \\
&\quad -\frac{1}{2}\alpha^{(k)}||\mathbf{h}_n'(\mathbf{r}^{n^{(k)}})||_2^2 \\
&\quad -\frac{1}{2}\alpha^{(k)}||\mathbf{g}_n(\mathbf{r}^{n^{(k)}} + \mathbf{u}^{n^{(k)}})||_2^2\} \ \forall\ n \qquad (6.10)\\
\mathbf{u}^{(k+1)} &:= \max\{\mathbf{0}, -[\mu^{(k)}/\alpha^{(k)} + d_\mu^{(k)}]\} \qquad (6.11)\\
\lambda^{(k+1)} &:= \lambda^{(k)} + \alpha^{(k)} d_\lambda^{(k)} \qquad (6.12)\\
\mu^{(k+1)} &:= \mu^{(k)} + \alpha^{(k)} d_\mu^{(k)}, \qquad (6.13)
\end{aligned}$$

where $||\cdot||_2^2$ is the squared two-norm and $\alpha^{(k)}$ is a variable scalar penalty parameter at step k. The maximum operation without a variable subscript, $\max\{\cdot,\cdot\}$, is taken to be the element-wise maximum between the vectors. The vectors $d_\lambda^{(k)}$ and $d_\mu^{(k)}$ are subgradients to q which satisfy

$$q(\bar{\nu}) \geq q(\nu^{(k)}) + (\bar{\lambda} - \lambda^{(k)})^T\ d_\lambda^{(k)} + (\bar{\mu} - \mu^{(k)})^T\ d_\mu^{(k)}, \qquad (6.14)$$

for all $\bar{\nu}$ (remembering that $\nu^T = [\lambda^T\ \mu^T]$, λ and μ are the vectors of the entire equality and inequality constraint multipliers). Using the following projections into the inequality constraint set, we choose subgradients

$$\begin{aligned}
d_\mu^{n^{(k)}} &:= \max\{\mathbf{g}_n(\mathbf{r}^{n^{(k+)}}), -\mu^{(k)}/\alpha^{(k)}\} \\
d_\lambda^{n^{(k)}} &:= \mathbf{h}_n'(\mathbf{r}^{n^{(k+1)}}), \qquad (6.15)
\end{aligned}$$

to adequately satisfy the feasibility for the inequality constraints. The appropriate elements of the subgradient, $d_\mu^{n^{(k)}}$ and $d_\lambda^{n^{(k)}}$, are calculated locally at each agent for solving the subproblem at agent n.

Algorithm 2 Alternating Update Augmented Dual for Multi-Agent Optimization

1: *Init.* Each active agent sets an initial internal state, slack variable, and multipliers to suitable (not necessarily feasible) values
2: *Exchanging Information:* To each neighbor involved with a common constraint, the agent communicates current state
3: *Update slack and dual variables:* An agent updates its local state, slack, and dual variables
4: *Exchanging slack variables:* To each neighbor, an agent sends current slack and dual variable updates for any common constraints
5: *Updating the local state:* An agent updates its own internal state pursuant to better utility fixing new dual variables in local Lagrangian
6: *Iterate:* All agents simultaneously execute steps 2 through 5 and repeat until convergence to desired tolerance

Remark 6 (Convergence)**.** *The values of the penalty step size, $\alpha^{(k)}$, can be constant. However, super-linear convergence can be obtained for $\alpha^{(k)} \to \infty$ [115]. This complicates the proof of convergence, but as long as $\alpha^{(k)}$ becomes fixed after a finite number of iterations, the standard convergence results apply [56].*

The pseudo-code description of the alternating update augmented dual method is shown in Algorithm 2. The idea is that the subgradient updates advance the solution in the direction of improving utility, under the quadratic penalization of any constraint violation. The updates with respect to the inequality constraint slack are always modified to be a feasible projection of the initial update.

6.2.1 Matched Marginal Utility Modification for Inequality Constraints

This section details a unique incremental updating technique which leverages basic Lagrange analysis and provides initial convergence improvements over the beginning steps of the aforementioned augmented Lagrange dual formulation.

While we have shown the above algorithm formulation and included equality constraints, the modification we propose next involves updates with respect to the inequality constraints amongst neighborhoods of agents. While the utilization of projections would easily allow the application of our modified algorithm to problems which consider equality constraints,

Algorithm 3 Minimum-Increment Marginal Utility Matching (MIMUM) Method

Init. $\mathbf{r} = \mathbf{0}$, $\mathcal{M} \leftarrow \varnothing, \mathcal{B} \leftarrow \mathcal{V}$
while there exists any agents for which there is slack in all associated constraints OR until maximum 'priming' iterations reached **do**
    (MMU) Instruct the agents associated with the maximum marginal utility (indices $\mathcal{M}$) to update; inform the updating agents of the 'next' maximal marginal utility (μ^{MMU}) price to satisfy to Eq. (5.9)
    Exchange local state and slack variable information with neighbors
    (MCP) Locally find the maximum price (μ^{MCP}) for the constrained largest allowable increase from Eq. (6.17)) for all the active agents, $n \in \mathcal{M}$
    Each agent $n \in \mathcal{M}$ updates internal state according to Eq. (6.18) using the multiplier which is the maximum between μ^{MMU} and μ^{MCP}
    Update the constraints to reflect remaining slack
    if an inequality constraint is met with equality **then**
        Remove any agent indices from $\mathcal{B}$ so that they do not update further
    end if
end while
Execute Algorithm 2 for further convergence of solution

we do not consider them here for the sake of clarity.

Assumption 2. *The concave utility functions, U_n, are non-decreasing and smooth (with at least continuous first derivatives), that is, $U_n \in \mathbf{C}^p$, $p \geq 1$.*

Given the simple optimization problem with uniform utility for all agents

$$\max_{\mathbf{r}} \sum_{n=1}^{N} U(r_n) \text{ subject to } \sum_{n=1}^{N} r_n \leq Q, \tag{6.16}$$

we follow the same result as obtained in Chapter 5 and propose a modified algorithm for decentralized optimization.

The modified algorithm is presented in Algorithm 3 for maximizing inequality constrained concave utility problems. This Minimum-Increment Marginal Utility Matching method initiates all power allocations to zero. It then finds the minimum increment which produces the reduction in the maximum marginal utility toward the *next maximum* marginal utility. The method then computes the hypothetical largest values of the internal states allowed while maintaining the strict (or active) feasible constraints. The maximum dual price is found from amongst all the utilities for the internal states associated with the largest

feasible increment toward meeting a constraint. The maximum price between the maximum constrained price (MCP) from associated constraints and the next maximum marginal utility (MMU), is the new price and is used to update the internal states. It is essential to note that the maximum marginal utilities found on line 3 of Algorithm 3 are almost never singletons, which part of the value of the method. When common maximum marginal utilities are found, these internal states are simultaneously updated while maintaining matched marginal utilities.

We must find the maximum dual variable price restricted by remaining inactive constraints through the solution of a smaller problem. We compute (using a dual formulation over a sub-network or local predictions of neighboring utilities)

$$r_n^{MCP} := \arg_n \left\{ \begin{array}{l} \max_{r_m \in \mathcal{M} \cap \mathcal{N}_n} \sum_{m \in \mathcal{M} \cap \mathcal{N}_n} U_m(r^m) \\ \text{subject to} \\ \mathbf{g}_n(\mathbf{r}^n) \preceq 0 \ \forall \ n \in \mathcal{M} \end{array} \right\} \tag{6.17}$$

for each $n \in \mathcal{M}$, the currently active agents, and then calculate the dual price, $\mu^{MCP} = \max_{n \in \mathcal{M}} V_n(r_n^{MCP})$. The new internal state is

$$r_n^{(k+1)} := V_n^{-1}(\max\{\mu^{MMU}, \mu^{MCP}\}). \tag{6.18}$$

If $\mu_{MCP} > \mu_{MMU}$, then the MCP allocation is selected and a constraint becomes active and the allocations associated to this constraint are optimal and complete.

This decentralized method calculates the maximum marginal utility across the system of agents and updates according to the specifications in Algorithm 3. The current marginal utility to match is exchanged, from this an agent is selected to update. The agents exchange state and slack variables with those in the local neighborhood. The active agent then performs a local check (in (6.17)) for the achievement of an active constraint before updating the local state obtained from the new marginal utility and (6.18).

Interpretations

This dual decomposition modification could be considered the dual of a minimum cost flow problem [113, 114], except for the use of concave reward functions. Alternatively, we could

liken our modification to a block coordinate descent method [116] in which marginal utility (not normally a feature of BCD) is used and the block decomposition of optimization variables updates in each iteration. The decentralized nature of the optimization problem is maintained by our assumption that in some applications knowledge of the utility of neighboring agents will be available. Thus, with the exchange of state and slack variables, the agents can predict the states of neighboring agents.

Remark 7. *It should be especially noted that this modification is still providing updates according to requirements of (6.10) through (6.13). The gradients, $d_\mu^{(k)}$, for updating the Lagrange multipliers and constraint slack are selected 'intelligently' according to the optimality condition posited in Lemma 9. Instead of a descent-like combination of the previous update, each new $d_\mu^{(k)}$ is immediately taken as the new multiplier.*

6.2.2 Convergence Analysis of Matched Marginal Utility Method

The convergence of the modified algorithm relies primarily on the convergence of Algorithm 2 which is documented in [49, 56] and others. However, we must still show the convergence of the 'priming' MIMUM method. A centralized sub-class of this algorithm was presented in [68]. The following results rely on Assumption 2.

Proposition 1. *Given decision variables for the N distributed agents at iteration k, $\mathbf{r}^{(k)}$, the update $\mathbf{r}^{(k+1)}$ ($\succeq \mathbf{r}^{(k)}$) from the maximum multiplier in step 6, the $k+1$ iteration of Algorithm 3 provides an optimal improvement to the solution.*

Proof. Given the slack utilized in the update, $R^{(k)} = ||\mathbf{r}^{(k+1)} - \mathbf{r}^{(k)}||_1$, there is no alternate state $\bar{\mathbf{r}}^{(k+1)}$ such that $U(\mathbf{r}^{(k+1)}) < U(\bar{\mathbf{r}}^{(k+1)})$. This is seen from the utilization of the optimality condition discussed in Lemma 9 and Lemma 10, and by the matching of the active marginal utilities, optimality is maintained in the update. □

The idea is that the marginal utility of each resource should be 'balanced' to the price (Lagrange multiplier) of the most strict associated constraint, so implicitly also to each other through said constraints.

Proposition 2. *Algorithm 3 maintains feasibility in each iteration, as the increment of any resource allocation is limited by the most strict constraint associated with it.*

Proposition 5 follows simply from the fact that our algorithm *constructs* allocations such that no infeasibility is ever incurred.

Lemma 13. *Algorithm 3 converges to an optimal solution. That is, the sequence generated by the algorithm satisfies sufficient conditions of optimality.*

Proof. Since each iterative update in step 6 is optimal, given in Proposition 4, and feasibility is maintain as postulated in Proposition 5, then the final update in which no further allocation can be updated produces the final optimal point, i.e., there are a finite number of improving iterations. The updates to the multipliers balance the resources among the agents according to (5.9) so that, given Assumption 2, the standard Lagrangian for the original constrained problem satisfies the appropriate first order necessary and sufficient conditions [38]. □

6.2.3 Algorithm Complexity and Communication Cost

The local complexity for solving the distributed dual utility maximization problem with the MIMUM modification is $O(L\mathcal{N}_{\max}^3)$ as the upper bound, where $\mathcal{N}_{\max}$ is the upper bound on the size of the agent neighborhoods, L is the number of iterations taken, (remembering the assumption that $M_n = 1 \; \forall \; n$, i.e., $d = N$). This follows since the `while` loop includes $\mathcal{N}_{\max}^3$ possible operations done possibly L times (the state of each agent becomes 'level' with another marginal utility *or* a part of an active constraint). A convex programming method for the centralized version of this problem has complexity $O(N\mathcal{N}_{\max}^2)$. However, this method is appealing when the problem is decomposed for decentralized implementation, where it is already required that more iterations are required for standard methods in obtaining a solution, since there is no single controller making a centralized decisions. In this case, each of the decentralized agents only see a complexity of $O(\mathcal{N}_{\max}^3)$, with the minimum number of $N+1$ iterations necessary to meet all inequality constraints. Thus, while

the neighborhoods may be small and the computations efficient, the number of necessary iterations for a large decentralized network may be impractical. However, Algorithm 3 provides a 'partially complete' optimal solution, as each of the updates renders the system balanced with respect to the marginal utilities. The agents offering the most improvement to the overall utility are allocated resources first. The algorithm then can be stopped after the desired number of iterations and provide a suitable 'primed' state for the agents which can then be optimized to finality within the standard augmented dual framework.

In some distributed systems, such as wireless sensor networks, the communication resources are often highly limited. It is therefore important to analyze the optimization routine with respect to the necessary communication. For simplicity, we define the communication cost as being constant per transmission of a real valued scalar with cost c. There are two contributions to the communication cost: exchange of the decision variables, and the updating of slack variables and constraint multipliers. In general, the exchange of decisions variables amongst all agents involves a cost of

$$c\sum_{n=1}^{N}\sum_{m\in\mathcal{H}_n} dim(r_n) = cN(N-1)d,$$

where $\mathcal{H}_n$ is the set of agents needed to disseminate decision variable r_n via multi-hop, $|\mathcal{H}_n| = N-1$ is the maximum number of hops possible, and d is the dimension of the state variable. This includes the case of any consistency constraints. According to Remark 5, we can assume there is a sparse dependency of the local utility functions on neighboring decision variables. In this case, the cost is reduced to

$$c\sum_{n=1}^{N} dim(r_n) = cd.$$

It is assumed that a single unidirectional transmission from agent n can reach all neighboring agents. The cost of communicating the slack and dual variables across all agents is

$$\begin{array}{c}\sum_{n=1}^{N}\left(\sum_{i=1}^{I_n}\sum_{m\in\mathcal{H}_{ni}} c + \sum_{j=1}^{J_n}\sum_{m\in\mathcal{H}_{nj}} 2c\right) \\ = c(N-1)(dim(\nu)+dim(\mu))\end{array},$$

and while there are possibly more constraint multipliers than decision variables, they must only be communicated to specific nodes. Similar to the decision variables, this cost reduces when only neighborhood constraints are considered.

$$\sum_{n=1}^{N} (cI_n + 2cJ_n) = c(dim(\nu) + dim(\mu))$$

Proposition 3. *The dual decomposition optimization communication cost for all agents in the systems (neglecting routing overhead) is*

$$c(d + dim(\nu) + dim(\mu)). \tag{6.19}$$

The communication cost of the MIMUM method is similar to that of the dual decomposition, it is necessary, however, to exchange the maximum price update instead of the typical set of Lagrange multipliers, a cost of $cN(N-1)$ given one neighborhood constraint per agent. The assumption of prior information about the utilities of neighboring agents allows local optimization to reduce communication cost, which, for all agents combined, is

$$c(d + N(N-1) + N + dim(\lambda)).$$

6.2.4 Simulation Results

We illustrate the effect of augmenting the dual decomposition with the MIMUM method. In the following simulations, the state of the system is first optimized using MIMUM, and after a limited number of iterations, the dual decomposition method is ran on the solution obtained from MIMUM to produce a final optimal answer. Given sufficient iterations, the MIMUM will converge to the optimal solution (as shown in the previous section). However, for a large system of agents, it may take many iterations. The value, then, of using the MIMUM is not the convergence of the states to the precise optimal. Rather, it is in the fact that if the system is running with limited communication or expects outages, a solution which has quickly converged to the optimal values will offer better performance than the standard augmented dual decomposition method for a similar number of iterations. Thus,

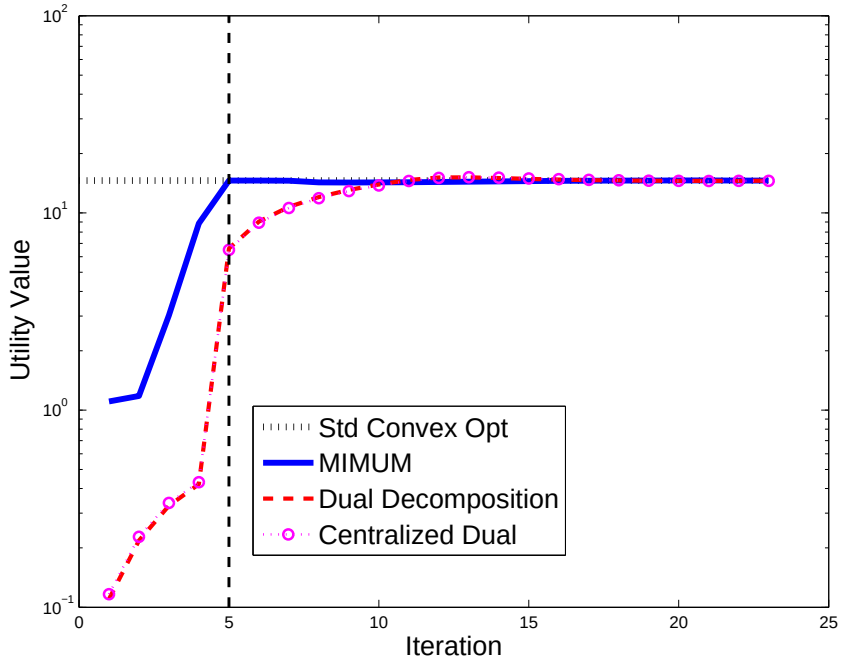


Figure 6.2: *Objectives values versus iteration for the different methods, the vertical dashed line represents where the MIMUM method stops and provides augmented dual algorithm with an 'primed' initial solution.*

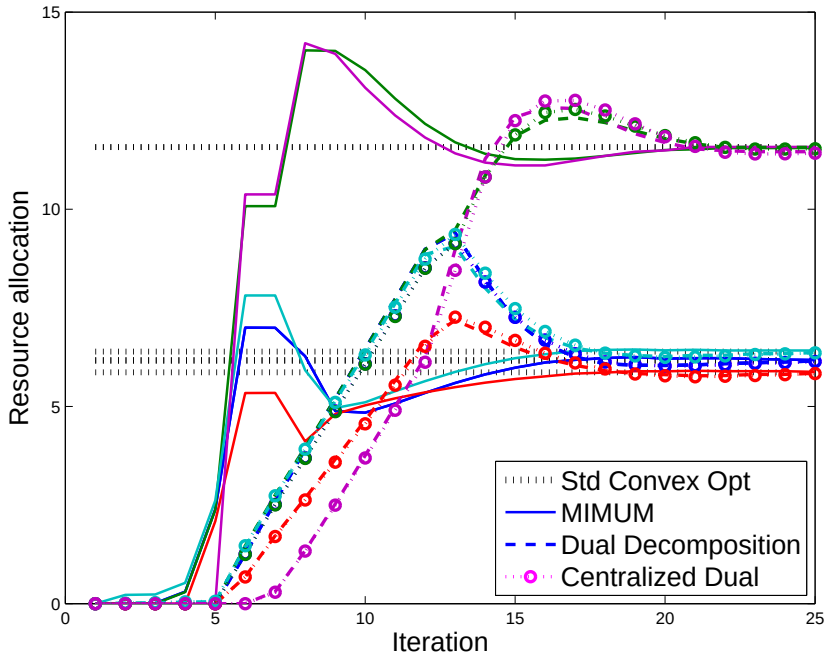


Figure 6.3: *Agent local decisions versus iteration for the different methods. (The legend is color indifferent, only the line type matters)*

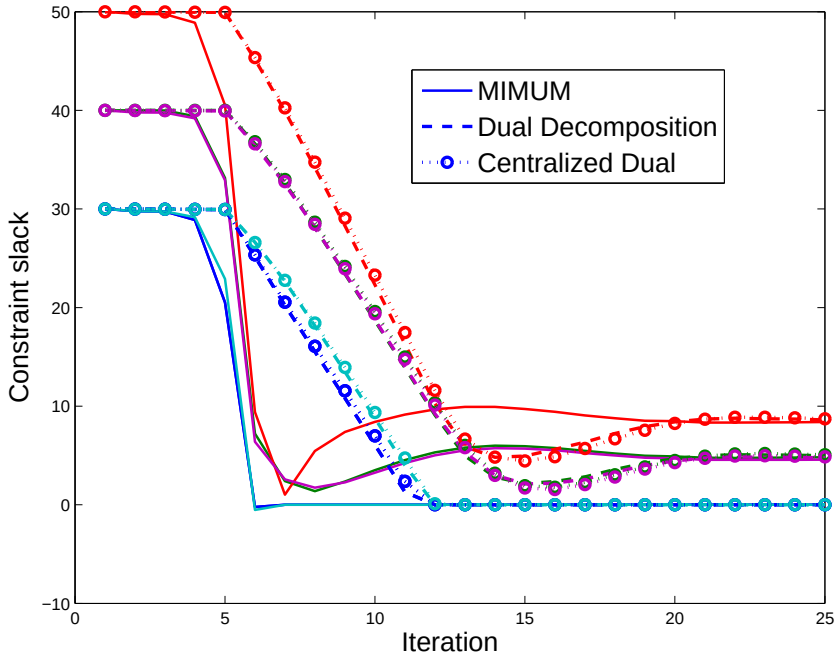


Figure 6.4: *Inequality constraint slack versus iteration for the different methods.*

by 'priming' the system with the MIMUM method, acceptable performance can be obtained in few iterations, and if desired, the optimal solution can be achieved using the augmented dual method introduced in Section 6.2.

The above description of the benefit of the MIMUM method is clear in the simulation of a simple system of five nodes with non-uniform log utilities, of which the neighboring agents have information concerning. The resulting objectives (sum of the utility for all agents) versus iteration are shown in Figure 6.2. It is plain that the MIMUM method converges quickly in very few iterations as compared to the dual decomposition method. Figure 6.3 shows a similar contrast, but specifically, the quick decisions made by the MIMUM method which could prove valuable if an early decision on the states of the agents is needed. Lastly, Figure 6.4 shows the slack of the neighborhood usage constraints place on the agents and further illustrates the behavior of the MIMUM versus the standard dual decomposition method.

6.3 Analysis of Gradient Method Convergence with Delay and Quantization

In Algorithm 2, we see an iterative exchange of dual variables which can be updated by find the subgradients in (6.15). However, because of transmission delays and quantization of the subgradient updates, the convergence of the dual function to the optimal will have a modified form. Following the results of [49] and [117], we give the analysis of delayed quantized gradient updates, but will elaborate these results in the context of energy harvesting wireless sensor networks.

Recalling the relationship

$$q(\bar{\nu}) \geq q(\nu(t)) + (\bar{\lambda} - \lambda(t))^T \, d_\lambda(t) + (\bar{\mu} - \mu(t))^T \, d_\mu(t), \tag{6.20}$$

with

$$\begin{aligned}\lambda^{(k+1)} &:= \lambda(t) + \alpha(t) d_\lambda(t) \\ \mu^{(k+1)} &:= \mu(t) + \alpha(t) d_\mu(t),\end{aligned}$$

where $d_\lambda(t)$ and $d_\mu(t)$ are subgradients to q for a gradient-based update of the dual function $q(\nu)$, with $\nu^T = [\lambda^T \ \mu^T]$, λ and μ the vectors of the entire equality and inequality constraint multipliers.

The problem we now address is when the updates of the dual variables are not received with the same delay. The total communication delay between two nodes, n and m, is $d_{nm}(t)$. In the case that data must be exchanged forward and backward (a gradient sent, an updated multiplier received), then the round-trip delay is given by $d_{nm}(t) = \delta_{nm} + \delta_{mn} \leq D$, where D is a bound on the total delay. The delay bound means that there are no packet losses in the network, but that all updates eventually reach their destinations within time interval D. We also assume that the value of the gradient is bounded by G, which means we assume sudden changes do not occur in the optimization updates. These assumptions are suitable for a network with many small local neighborhood constraints with relatively low traffic. Lastly, we assume that the gradient updates are quantized with a bounded quantization noise, where $Q = \frac{W^2}{(2_n^b - 1)^2}$ is the uniform quantization noise variance. In Algorithm 2, local agent n computes the gradient update for the constraint multipliers associated with agents n and m as , $g_{nm} = [d_{\mu,nm}^T \ d_{\lambda,nm}^T]^T$ so that the overall update to the constraint multipliers across all nodes is $\nu(t+1) = \nu(t) + \alpha(t) g(t)$, where again, $\alpha(t)$ is the variable update step size. However, the gradient updates are delayed by the multi-hop communication in the network. We define the projection $\Pi_\Delta(\bar{x}) : \mathbb{R}^{dim(\nu)} \to \mathbb{R}^{dim(\nu)}$ as a vector which extracts the delayed components of the updated gradient or dual variable. That is,

$$[\Pi_\Delta(x(t))]_{nm} = \begin{cases} x_{nm}(t - d_{nm}) & d_{mn} = \Delta \\ 0 & \text{otherwise,} \end{cases},$$

that is, the n-m components of the constraint multipliers which have a total delay Δ and

were sent at time $t-\Delta$. We can express the dual variable update as

$$\nu(t+1) = \nu(t) + \sum_{\Delta=0}^{D} \Pi_{\Delta}(\alpha(t)\mathcal{Q}(g(t))),$$

where $\mathcal{Q}(g(t))$ is the vector gradient update which is quantized to be sent to neighboring agents. We bound the distance of the initial dual function $q(\nu(0)$ to the optimal $q^*(\nu^*)$ for the multipliers as $||\nu(0) - \nu^*|| \leq R$. This holds for the delayed version of $\nu(0)$. Using these definitions, we give the following bound on the convergence of Algorithm 2 (see also [49, 117]),

Theorem 6 (Bound on Dual Problem Convergence)**.** *Assuming the existence of a minimizer q^*, with time delays bounded by D, bounded quantization errors below Q, and T update iterations, the dual function value satisfies*

$$\begin{aligned} &||\min_{t=0,\ldots,T-1} q(\nu(t)) - q^*||_2^2 \leq \\ &\qquad \frac{R + (D+1)(G+Q)[3DG + DQ + G + Q]\sum_{t=0}^{T}\alpha^2(t)}{2\sum_{t=0}^{T}\alpha(t)} \end{aligned} \tag{6.21}$$

and simplifies for uniform α to

$$\begin{aligned} &||\min_{t=0,\ldots,T-1} q(\nu(t)) - q^*||_2^2 \leq \\ &\qquad \frac{R + (T+1)(D+1)(G+Q)[3DG + DQ + G + Q]\alpha^2}{2(T+1)\alpha} =: B(D,Q) \end{aligned} \tag{6.22}$$

Proof. As demonstrated in previous works [49], the assumption of the existence of the optimal ν^* provides $||\nu(T+1) - \nu^*||_2^2 \geq 0$. This allows us to write

$$\left|\left|\nu(T) - \sum_{\Delta=0}^{D} \Pi_{\Delta}(\alpha(t)\mathcal{Q}\left(g(t)\right)) - \nu^*\right|\right|_2^2 \geq 0.$$

The above equation can be expanded,

$$||\nu(0)-\nu^*||_2^2 + 2\sum_{t=0}^{T}\sum_{\Delta=0}^{D}[\Pi_{\Delta}(\alpha(t)\mathcal{Q}(g(t)))]^T(\nu(t)-\nu^*) + \sum_{t=0}^{T}\alpha(t)\sum_{\Delta=0}^{D}||\Pi_{\Delta}(\alpha(t)\mathcal{Q}(g(t)))||_2^2 \geq 0,$$

where the recursive definition of $\nu(t)$ is used until $t = 0$. We derive the bounds of the individual terms of the above in Appendix B. Using a uniform α, the inequality results in

$$0 \leq R + 2\alpha \sum_{t=0}^{T} (\alpha GD(G+Q)(D+1) + q(\nu(t)) - q^*) + \sum_{t=0}^{T} (D+1)^2 (G+Q)^2 \alpha^2. \quad (6.23)$$

Rearranging the above terms, we find the bound in 6.22. □

For a constant step size α and no delay or quantization error, the convergence bound becomes

$$|| \min_{t=0,\ldots,T-1} q(\nu(t)) - q^* ||_2^2 \leq \frac{R + G^2\alpha^2(T+1)}{2\alpha(T+1)}.$$

This is similar to the convergence of the ordinary gradient method. Thus, dual decomposition gradient-based methods still converges, but into a larger neighborhood about the optimal solution.

6.4 Probabilistic Distribution of Convergence Bounds in Networks of Energy Harvesting Agents

In the case of deterministic delays and quantization, the results of the previous section are sufficient for illustrating the convergence behavior of a distributed optimization algorithm operating over a communications network [49, 117]. However, in wireless sensor networks utilizing energy harvesting, the availability of energy for communication depends on uncertain, but usually predictable, harvesting processes. There is an inherent difficultly in using communication resources to optimize the utilization of these resources. Using the example of the above gradient method of multipliers and matched marginal utility algorithms, the allocations of energy resources for communication could be the decision variables, but the current state of the decision variables must be communicated iteratively in order to find the optimal solution. Thus, we do not suggest an optimization method for how to communicate decisions concerning the communication, rather, we introduce some heuristic policies for when and how agents in the wireless sensor network should communicate. This also applies

in the case where the decision variables represent the resource allocation for a generic cooperative task, and the communication of the dual variables promotes the achieving of the optimal decisions.

6.4.1 Modeling Energy Harvesting with Compound Poisson Processes

The energy harvested at each node could be from a variety of sources, e.g., generated by wind, solar, or seismic activity. We assume that the energy can be modeled as having discrete arrivals as in a Poisson process with rate λ (the energy arrival rate) for t, the elapsed time since the previous transmission. Additionally, we complicate this model slightly by employing a marked Poisson process for the energy arrival. That is, each energy arrival instance drawn from the Poisson process also has a independent random draw from another (usually different) distribution indicating the quantity of energy harvested. The total energy in a particular epoch is given by the compound Poisson process (for agent n)

$$E_n(t) = \sum_{k=0}^{N_n(t)} E_{A,nk}, \tag{6.24}$$

where $N_n(t) \sim \text{Poisson}(\lambda_n t)$ and the arrival energy quanta $E_{A,nk} \sim \text{unif}(p_{\min}, p_{\max})$, $E_{A,nk} \sim \exp(\gamma)$, or some other appropriate distribution of virtually small support ($f_E > \varepsilon > 0$). The uniform and exponential distributions for modeling the per arrival quanta level are appropriate for the application [3, 57], so we do not consider any alternatives.

While our previous model in Chapter 3 assumes the collected energy is stored in a battery, super-capacitor, or other storage device with a maximum capacity, we assume now that the harvested energy quanta is significantly less than the capacity of the battery. This assumption that the maximum is never achieved is valid for some models [3], and especially since the subsequent policies we employ do not rely on energy storage over long periods of time.

In general, the compound Poisson process is the cumulative value process [78, 118]

$$Z(t) = \sum_{k=1}^{X(t)} Y_k, \quad \forall\ t \geq 0,$$

where $X(t) \sim \text{Poisson}(\lambda t)$ and $\Pr(Y_k \leq y) = G(y)$ the distribution function of the Y_k's. If $\lambda > 0$ is the rate for the process $X(t)$ and $\mu_Y = \mathbf{E}[Y_k]$ and $\sigma_Y^2 = \text{var}(Y_k)$ are the common moment for $Y_k\ \forall\ k$, then the moments of $Z(t)$ can be determined to be

$$\mathbf{E}[Z(t)] = \lambda \mu_Y t, \qquad \text{var}(Z(t)) = \lambda(\sigma_Y^2 + \mu_Y^2)t.$$

The distribution of the compound Poisson process can be explicitly written, conditioned on the knowledge of $X(t)$. We denote the n^{th} convolution of the mark distribution (this assumes independent Y_k's) as

$$\begin{aligned} G^{(n)}(y) &= \Pr(Y_1 + \cdots + Y_n \leq y) \\ &= \int_{-\infty}^{\infty} G^{(n-1)}(y - z) dG(z) \end{aligned} \tag{6.25}$$

where $G^{(0)} = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases}$. Then

$$\begin{aligned} \Pr(Z(t) \leq z) &= \Pr\left(\sum_{k=1}^{X(t)} Y_k \leq z \right) \\ &= \sum_{n=0}^{\infty} \Pr\left(\sum_{k=1}^{X(t)} Y_k \leq z \,|X(t) = n \right) \frac{(\lambda t)^n e^{-\lambda t}}{n!} \\ &= \sum_{n=0}^{\infty} \frac{(\lambda t)^n}{n!} e^{-\lambda t} G^{(n)}(z), \end{aligned} \tag{6.26}$$

since $X(t)$ is independent of the Y_k's.

In the following, we are interested in utilizing this distribution for two interesting cases. In one case, we are interesting in the distribution of the energy arrivals for a fixed time period. In the second, we want to know the length of the delay, T, with respect to the compound process thresholded at some z_{th}. That is, we want to know the time how long

the value of the compound process fails to meet this threshold,

$$\Pr(Z(t) \leq z_{\text{th}}) \;\Leftrightarrow\; \Pr(T \geq t),$$

where T is the time at which the threshold is reached. From 6.26 we can write

$$\Pr(T > t) = \sum_{n=0}^{\infty} \frac{(\lambda t)^n e^{-\lambda t}}{n!} G^{(n)}(z_{\text{th}}).$$

Now the expected time to reach the threshold is

$$\begin{aligned}
\mathbf{E}[T] &= \int_0^{\infty} \Pr(T > t) dt \\
&= \sum_{n=0}^{\infty} \left(\int_0^{\infty} \frac{(\lambda t)^n e^{-\lambda t}}{n!} dt \right) G^{(n)}(z_{\text{th}}) \\
&= \lambda^{-1} \sum_{n=0}^{\infty} G^{(n)}(z_{\text{th}}).
\end{aligned} \tag{6.27}$$

Using these two formulations (distribution on compound value, and distribution on time), we provide in the following two strategies for communication of the dual variable updates. The first fixes quantization and produces variable delay, while the second fixes delay and allows quantization to vary.

6.4.2 Communication Policy: Fixed Quantization with Variable Delay

We introduce this policy for the case when the wireless agents in the system have some minimum level of energy, E_{th}, when they are required to harvest before transmission. This threshold is set to ensure that the probability of error in communication is sufficiently low so that virtually no packets are lost in communication, consequently, we set the required the energy per transmitted bit, E_b, to achieve this. The quantization noise variance then becomes

$$Q(E_{\text{th}}) = \frac{W^2}{(2^{(E_{\text{th}}/E_b)} - 1)^2}, \tag{6.28}$$

so that Q is fixed in the convergence bound of the previous section. The delay, however, for multiple-hop communication of dual variable updates, maintains the distribution

$$\Pr(\delta_{mr^1_{mn}} + \cdots + \delta_{r^h_{mn}n} \leq d) = \Pr(D_{mn} \leq d), \tag{6.29}$$

where agents $m, r^1_{mn}, \ldots, r^h_{mn}, n$ represent multi-hop communication sequence for h hops so that the total delay for agent n to update is D_{nm}, and the maximum delay amongst all updates is $D_n = \max_{m \in \mathcal{N}_n} D_{mn}$. We say that the system completes a gradient update when every agent has received the dual variables update from all neighboring agents involved in common constraints. This delay is represented by the maximum order statistic with distribution

$$\Pr(\max_n\{D_n\} \leq d) = \Pr(\max_n\{\delta_{mr^1_{mn}} + \cdots + \delta_{r^h_{mn}n}\} \leq d) = F_D^N(d),$$

where $F_D(d)$ is the delay distribution for a single node, identical for N nodes in the network. In the sequel, we analyze various distributions of the mark of the marked Poisson process. Additionally, the nature of the problem becomes intensely complex by the consideration of multiple hops for communication. That is, in the most general case, not only is the convolution of the mark distributions required for the definition of the energy harvesting process, but to consider the cumulative delays across agents requires further convolutions of the delay distribution associate with each of the agents cooperative in the inter-node multi-hop communication. We therefore apply some simplifying assumptions for various cases to discover the approximate underlying behavior.

Single Hop Deterministically Marked

Lemma 14. *If we consider a deterministic mark of unit value (unit value since without loss of generality, the level of a deterministic mark can be absorbed by the threshold), then for the case of single hop updates (suitable for systems with local neighborhood constraints), has the maximum order statistic distribution for the maximum delay amongst all communications*

in the N node network of

$$F_{\max(D)}(d) = F_D^N(d) = \left(1 - \sum_{k=0}^{\lfloor E_{\text{th}} \rfloor} \frac{(\lambda d)^k e^{-\lambda d}}{k!} \cdot 1\right)^N.$$

Proof. The proof follows from the definition of the Poisson cumulative distribution function (CDF). The single agent delay distribution simplifies as

$$\begin{aligned} F_D(d) &= \Pr(D_n \leq d) \\ &= \Pr(E_n(d) \geq E_{\text{th}}) \\ &= 1 - \Pr(E_n(d) \leq E_{\text{th}}) \\ &= 1 - \sum_{k=0}^{\lfloor E_{\text{th}} \rfloor} \frac{(\lambda d)^k e^{-\lambda d}}{k!}. \end{aligned}$$

We rely on the fact that the the maximum over all nodes m updating to node n simplifies according to the following,

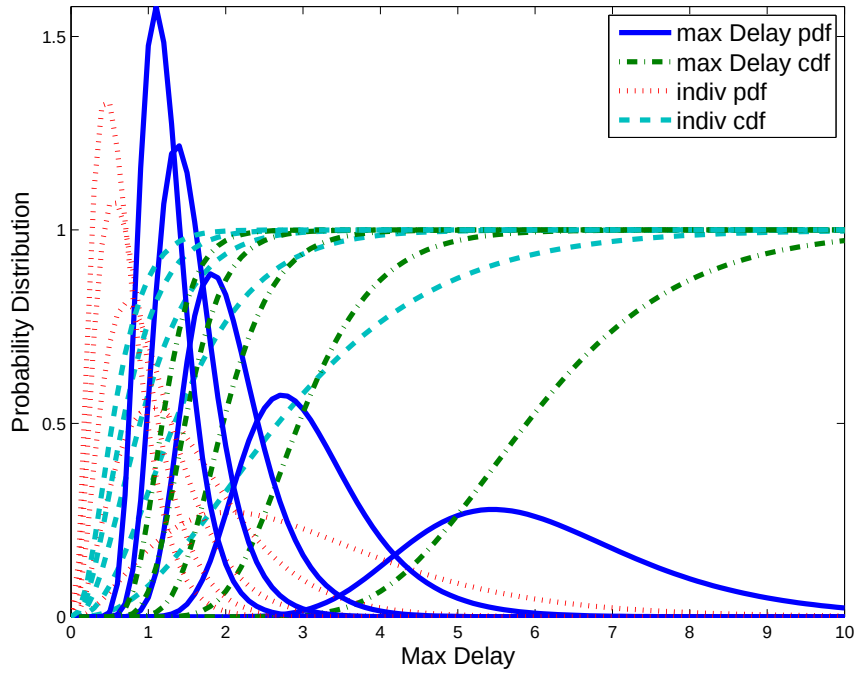
$$D_{\max} = \max_n D_n = \max_n \{\max_{m \in \mathcal{N}_n} D_{mn}\} = \max_n \delta_n,$$

since a single transmission of the gradient update is necessary to inform nodes in the neighborhood of n regarding the common constraint. Therefore, the maximum order statistic distribution is the distribution raised to the N^{th} power (for identically distributed RVs) [119]. □

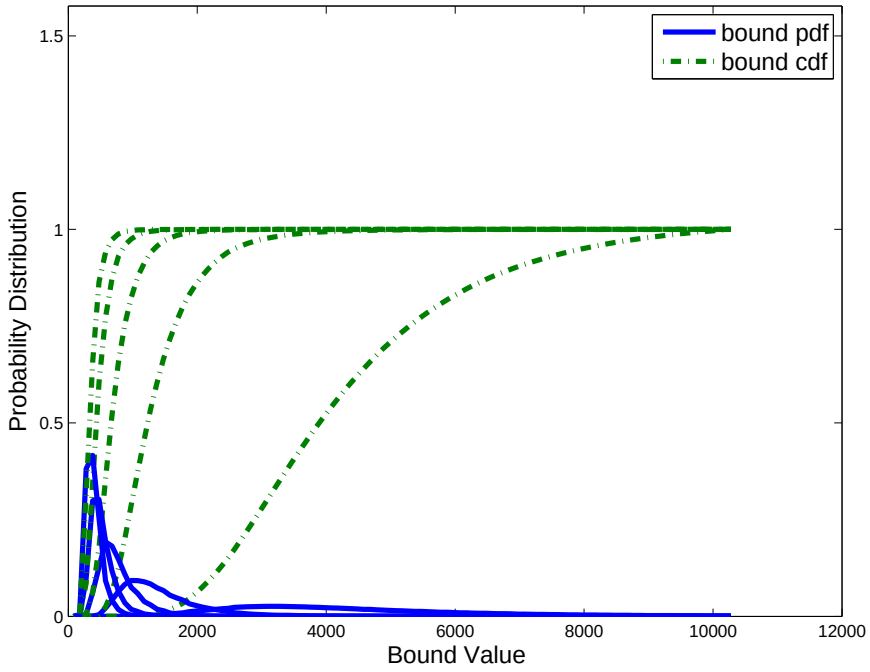
Using this distribution on delay, we can find a distribution on the convergence bound $B(D)$ (in (6.22)) using standard probability theory techniques and relying on the fact that $B(\cdot)$ is monotonic in D,

$$F_{B(D)}(b_0) = \Pr(B \leq b_0) = \int_0^{b_0} f_{\max(D)}(B^{-1}(\beta)) \left| \frac{\partial (B^{-1})}{\partial b} \right|_{b=\beta} d\beta,$$

where $f_{\max(D)}(B^{-1}(\beta))$ is the density function derived from the distribution function $F_{\max(D)}(d)$. Now we can observe the distribution of the bound when when we sweep various parameters of the networked energy harvesting system.

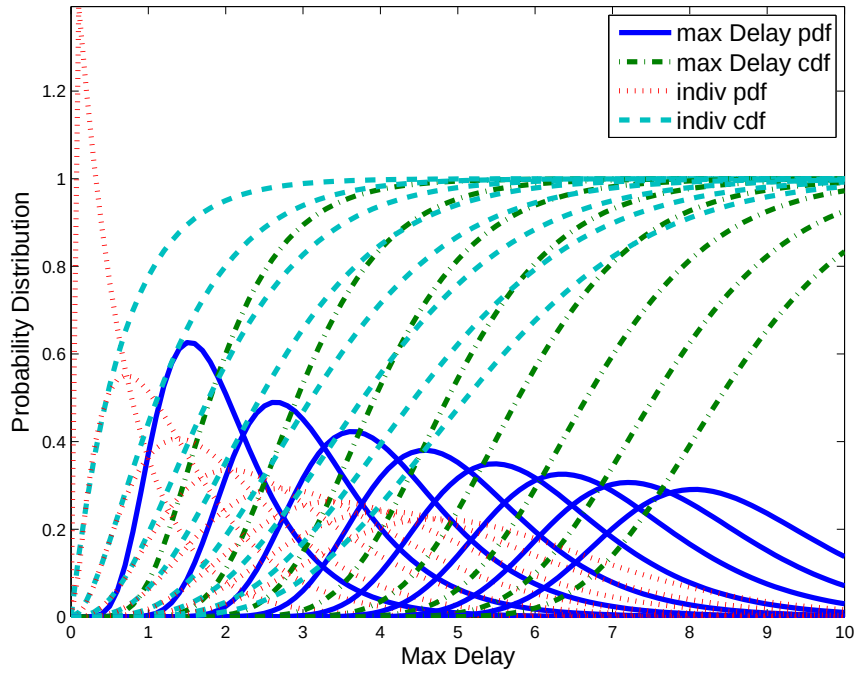


(a)

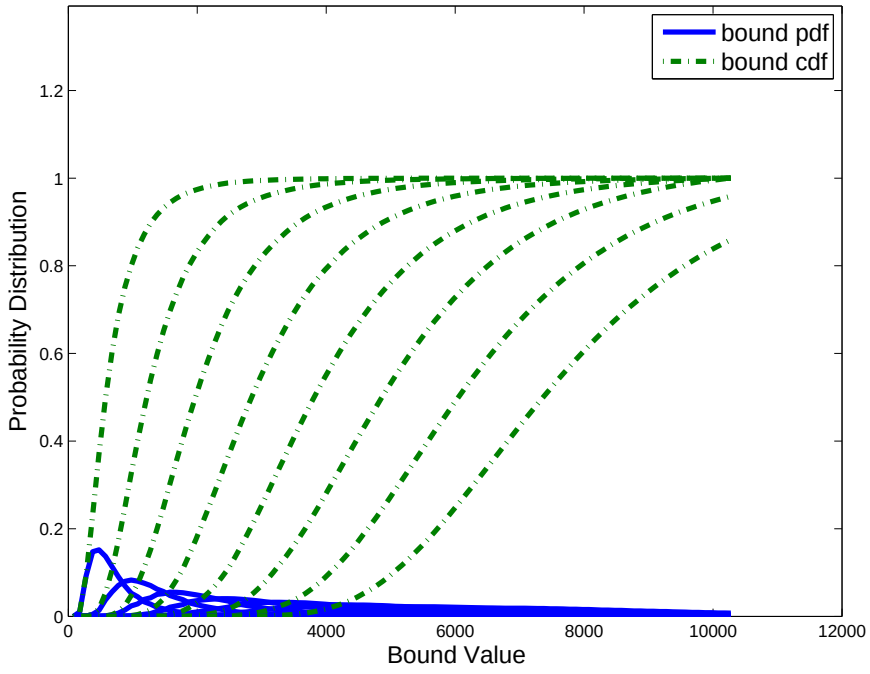


(b)

Figure 6.5: *Example distribution of (a) delay and (b) convergence bound for Poisson arrival rate* $\lambda = (2, 4, 6, 8, 10)$, $N = 10$, $h = 1$, E_{th}*=3, nominal values for optimization bound parameters.*

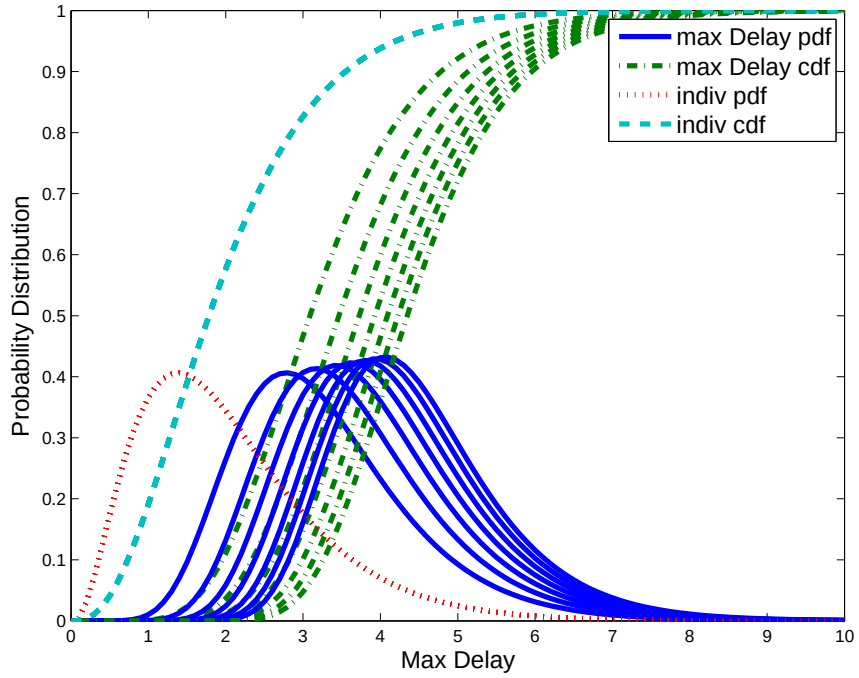


(a)

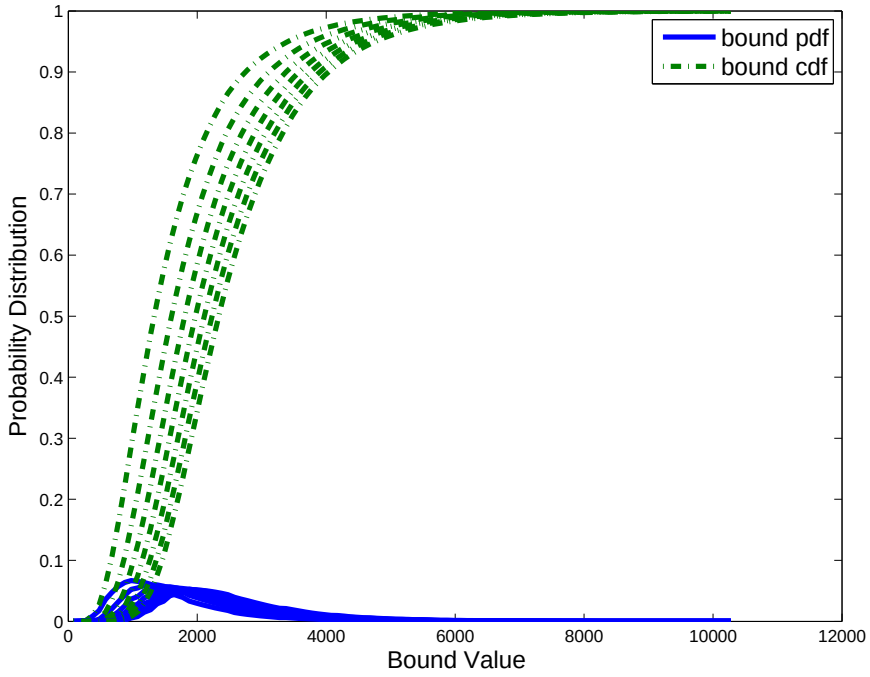


(b)

Figure 6.6: *Example distribution of (a) delay and (b) convergence bound for Poisson arrival rate* $\lambda = 3$, $N = 10$, $h = 1$, $E_{\text{th}} = (1, 2, \ldots, 8)$, *nominal values for optimization bound parameters.*



(a)



(b)

Figure 6.7: *Example distribution of (a) delay and (b) convergence bound for Poisson arrival rate* $\lambda = 3$, $N = (4, 6, \ldots, 16)$, $h = 1$, $E_{\text{th}} = 3$, *nominal values for optimization bound parameters.*

We plot the distribution of the delay and the resulting distribution on the decentralized algorithm bound for various parameter variations. Figures 6.5 sweep λ from 2 to 10, with the result that delay and bound CDFs shows that for rapid energy arrivals a high probability that delay will be around 1 for this set of scenario parameters. Figure 6.6 show the effect of varying E_{th} from 1 to 8, so that increasing energy threshold results in heavy tailed delay distributions. Lastly, Figure 6.7 demonstrates the resulting convergence bound for sweeping N from 4 to 16, where the results indicate that for increasing number of sensors, there is a practical limit on how much delay is introduced in the single hop scenario.

Single Hop Stochastically Marked

When considering single hop updates which are stochastically marked, we must apply a slightly different approach from directly finding the distribution on the maximum delay. The convolution of the mark distributions provides us with an understanding of the nature of the sum of the random mark values ($G^{(n)}$ is as in (6.25)). This distribution takes the form

$$F_D(d) = \Pr(D > d) = \sum_{n=0}^{\infty} \frac{(\lambda d)^n e^{-\lambda d}}{n!} G^{(n)}(E_{\text{th}}).$$

Thus, attempting to find a maximum delay among all N nodes in the network requires the max order statistic distribution, $F_{\max(D)}(d) = (F_D(d))^N$. Even in the case of identically distributed delays this expression is quite complex, involving and N^{th} order multiplicative expansion of the infinite sum. If we instead consider the expected value of the delays, we can get a more reasonable set of expressions.

If we have exponentially distributed mark values, i.e., $E_A \sim \exp(\gamma)$ with

$$g_E(\epsilon) = \gamma e^{-\gamma\epsilon}, \ \forall \ y \geq 0,$$

then n^{th} convolution[1] of the distribution is

$$G^{(n)}(\epsilon) = \sum_{k=n}^{\infty} \frac{(\gamma\epsilon)^k e^{-\gamma\epsilon}}{k!}.$$

[1] n is a summation index associated with n^{th} arrival in the considered epoch, not the sensor node index.

Following the result of the expectation of wait-time in (6.27), the expected delay simplifies to the following form [118]

$$\begin{aligned}\mathbf{E}[D] &= \lambda^{-1}\sum_{n=0}^{\infty} G^{(n)}(E_{\text{th}}) \\ &= \lambda^{-1}\sum_{n=0}^{\infty}\sum_{k=n}^{\infty}\frac{(\gamma E_{\text{th}})^k e^{\gamma E_{\text{th}}}}{k!} \\ &= \lambda^{-1}\sum_{k=0}^{\infty}\sum_{n=0}^{k}\frac{(\gamma E_{\text{th}})^k e^{\gamma E_{\text{th}}}}{k!} \\ &= \lambda^{-1}\sum_{k=0}^{\infty}(1+k)\frac{(\gamma E_{\text{th}})^k e^{-\gamma E_{\text{th}}}}{k!} \\ &= \frac{1+\gamma E_{\text{th}}}{\lambda}, \end{aligned} \tag{6.30}$$

the mean delay for a transmission from one energy harvesting sensor node. Additionally, we have the following result.

Lemma 15. *The variance of the delay for a single node with exponential marked Poisson arrivals is equal to*

$$\begin{aligned}\sigma_D^2 = \text{var}(D) &= \mathbf{E}[D^2] - (\mathbf{E}[D])^2 \\ &= \frac{1}{\lambda^2}\left(3(\gamma E_{\text{th}})^2 + 6\gamma E_{\text{th}} + \frac{1}{3\sqrt{\gamma E_{\text{th}}}}\right). \end{aligned} \tag{6.31}$$

Proof. The proof of this is shown in Appendix C. □

Armed with these first and second order statistics, we pursue a bound on the maximum order statistic for delay.

Lemma 16. *For a set of random variables, $X_1, X_2, \ldots, X_n$, drawn independently from identical distributions with mean μ and variance σ^2, we can bound the maximum among these values with*

$$\sup_{X_i,\ 1\le i\le n} \mathbf{E}[X_{n:n}] = \mu + \sigma\sqrt{n-1} \tag{6.32}$$

Proof. The proof of this result and other maximum order statistic bounds can be found in [120]. □

Using this expectation bounding result gives us the upper limit on the maximum delay,

$$\bar{D} := \sup \ \mathbf{E}[\max_n\{D_n\}] = \mu + \sigma\sqrt{N\mathcal{N}_{\max} - 1}$$
$$= \frac{1 + \gamma E_{\text{th}}}{\lambda} + \sigma_D\sqrt{N\mathcal{N}_{\max} - 1}, \tag{6.33}$$

where N is the number of nodes and $\mathcal{N}_{\max}$ is the maximum number of neighbors with which each node exchanges updates. Using this upper limit in the convergence result, we can obtain an upper bound on the distance of the solution from the optimum. It should be noted, that we cannot obtain the expected value of the convergence bound from only knowledge of the first moment. Since the convergence bound expression contains D and D^2 terms, inserting the supremum bound for $\mathbf{E}[D]$ will result in an approximation.

Multi-Hop Deterministically Marked

In the case that we allow constraints and interactions of dual variables to be between nodes which are not single-hop neighbors, then we must account for the delay associated with multiple hop transmissions. The distribution we are interested in involves the sum of identically distributed random delay variables, $\delta_{mr^1_{mn}} + \cdots + \delta_{r^h_{mn}n}$, so that we can find

$$\Pr(\max_n\{D_n\} \leq d) = \Pr(\max_n\{\delta_{mr^1_{mn}} + \cdots + \delta_{r^h_{mn}n}\} \leq d).$$

For the case when the number of hops, h, is large, we can use the central limit theorem (CLT) to approximate the behavior of this sum of random delays. The CLT says the following.

Theorem 7 (Central Limit Theorem (CLT))**.** *For independent RVs, $X_1, X_2, \ldots, X_K$, with identical mean and variance (finite), μ and σ^2 (can be from any distribution), the sum of these random variables*

$$X_1 + X_2 + \cdots + X_K \sim \mathcal{N}(K\mu, K\sigma^2) \ \text{ as } K \to \ \infty$$

Proof. This theorem is a well-known result for random quantities, see [78] for details. The approximation converges as $O(\frac{1}{K^{3/2}})$ or $o(\frac{1}{K})$. □

Lemma 17. *In case of the unit deterministically marked compound Poisson process, the mean and variance are*

$$\mu = \mathbf{E}[D] = \frac{\lfloor E_{\text{th}} \rfloor}{\lambda}, \qquad \sigma^2 = \text{var}(D) = \frac{\lfloor E_{\text{th}} \rfloor}{\lambda^2}. \tag{6.34}$$

Proof. The density of the delay is related to the Gamma distribution since

$$\begin{aligned} f_D(d) &= \frac{\partial}{\partial d} F_D(d) \\ &= \frac{\partial}{\partial d}\left(1 - \sum_{k=0}^{\lfloor E_{\text{th}} \rfloor} \frac{(\lambda d)^k e^{-\lambda d}}{k!}\right) \\ &= -e^{-dt}\left[\lambda + \lambda\frac{\lambda d}{1!} + \lambda\frac{(\lambda d)^2}{2!} + \cdots + \lambda\frac{(\lambda d)^{\lfloor E_{\text{th}} \rfloor - 2}}{(\lfloor E_{\text{th}} \rfloor - 2)!}\right] \\ &\quad + \lambda e^{-\lambda d}\left[1 + \frac{\lambda d}{1!} + \frac{(\lambda d)^2}{2!} + \cdots + \frac{(\lambda d)^{\lfloor E_{\text{th}} \rfloor - 1}}{(\lfloor E_{\text{th}} \rfloor - 1)!}\right] \\ &= \frac{\lambda^{\lfloor E_{\text{th}} \rfloor} d^{\lfloor E_{\text{th}} \rfloor - 1} e^{-\lambda d}}{(\lfloor E_{\text{th}} \rfloor - 1)!}. \end{aligned}$$

Extracting the mean and variance of this Gamma density produces the desired result. □

We then find the distribution of the sum of the delays for a particular node by applying the CLT. Thus, as approximation of the sum delay for sufficiently large number of hops.

Lemma 18. *For sufficiently large number of hops, h, with local node delays identically distributed with mean, μ, and variance, σ, the maximum delay among all updates from node m (among $\mathcal{N}_{\max}$ possible nodes) sent to node n (among N possible nodes) is well approximated by*

$$F_{\max(D)}(d) = \left(\int_0^\infty \frac{1}{\sqrt{(2\pi)h\sigma^2}} \exp\left\{-\frac{(d - h\mu)^2}{2h\sigma^2}\right\} \mathrm{d}d\right)^{\mathcal{N}_{\max} N}. \tag{6.35}$$

Proof. From the assumption of identically distributed delays and sufficiently large number of hops for the multi-hop update we can apply the CLT directly. The sum delay for an update from node m sent to node n is approximated by

$$D_{mn} = \delta_{m r^1_{mn}} + \cdots + \delta_{r^h_{mn} n} \sim \mathcal{N}(h\mu, h\sigma^2).$$

The distribution of the maximum sum delay is given by

$$F_{D_h}(d) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{(2\pi)h\sigma^2}} e^{-\frac{(d-h\mu)^2}{2h\sigma^2}} \mathrm{d}d. \tag{6.36}$$

For large h and reasonable σ^2, the tail probabilities for $d < 0$ are close to zero, so we alter the lower limit of integration to be 0. Since this distribution on delay is the same among all nodes, the maximum order statistic among all updates ($\forall$ m to any n) can be given as

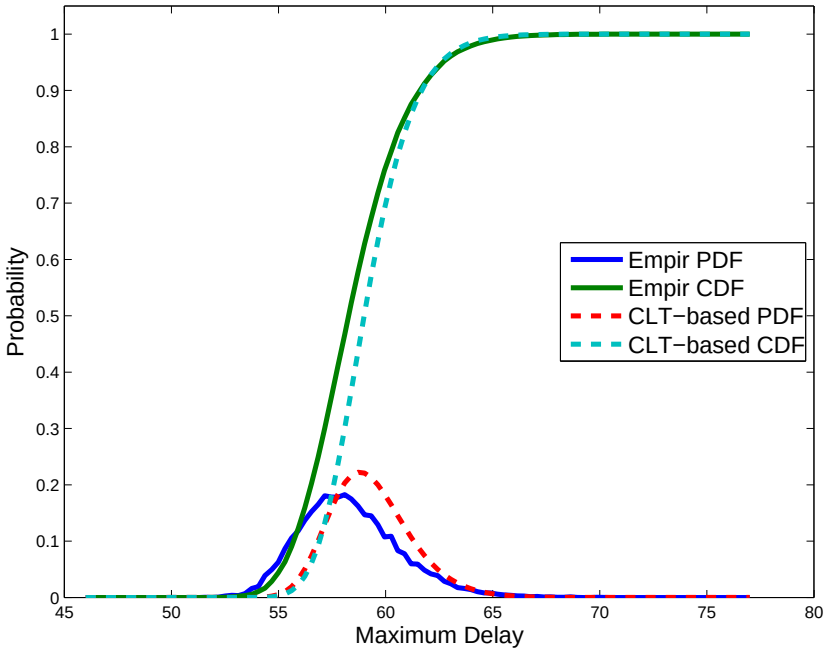
$$\begin{aligned} F_{\max(D)}(d) &= (F_{D_h}(d))^{N\mathcal{N}_{\max}} \\ &= \left(\int_{0}^{\infty} \frac{1}{\sqrt{(2\pi h\sigma^2)}} e^{-\frac{(d-h\mu)^2}{2h\sigma^2}} \mathrm{d}d \right)^{N\mathcal{N}_{\max}}, \end{aligned} \tag{6.37}$$

which is the desired result. Maximum number of neighboring nodes, $\mathcal{N}_{\max}$, is the partition of nodes in the network with which a node n among the total N nodes must update. □

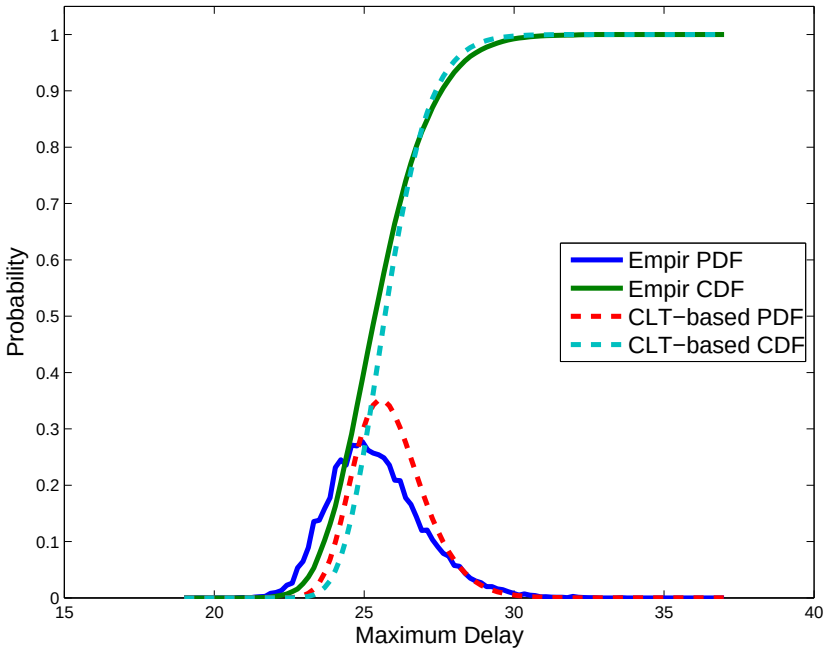
We can use the previous Lemma to obtained the approximate maximum delay distribution for various mark distributions. By substituting in the mean and variance for the deterministically marked compound Poisson process we get

$$F_{\max(D)}(d) = \left(\int_{0}^{\infty} \frac{1}{\sqrt{(2\pi)h\left(\frac{\lfloor E_{\mathrm{th}} \rfloor}{\lambda^2}\right)}} \exp\left\{ -\frac{\left(d - h\left(\frac{\lfloor E_{\mathrm{th}} \rfloor}{\lambda}\right)\right)^2}{2h\left(\frac{\lfloor E_{\mathrm{th}} \rfloor}{\lambda^2}\right)} \right\} \mathrm{d}d \right)^{N\mathcal{N}_{\max}}. \tag{6.38}$$

We can evaluate this approximate distribution by looking at repeated simulations of the deterministically marked energy arrivals and constructing a histogram of the maximum delay. These empirical distributions can be compared to the approximate distribution for multi-hop update maximum delay constructed in Lemma 18. Figure 6.8 demonstrates that the approximate CLT-based max delay distribution closely follows the empirical distribution for the maximum delay. Where it does not follow the empirical distribution, the CLT-based approximate distribution seems to over-estimate the maximum delay. The empirical distributions are normalized histograms constructed from 300 Monte Carlo runs. This approximate distribution can similarly be used to observe the distribution of the convergence region for the decentralized optimization algorithm with respect to the energy harvesting process.



(a)



(b)

Figure 6.8: *Normalized empirical histogram of maximum delay (solid line) in an energy harvesting wireless sensor network. For the case of unit deterministically marked arrivals,* $N = 50$ *nodes,* $\mathcal{N}_{\max} = 25$ *nodes with which to communicate updates,* $\lambda = 3$*,* $E_{\text{th}} = 3$*, with (a) maximum multi-hop length* $h = 50$*, (b) maximum multi-hop length* $h = 20$*. The results are averaged over 300 Monte Carlo runs and shown in comparison to the CLT-based approximation of the maximum delay distribution (dashed line).*

Multi-Hop Stochastically Marked

We continue the use of the idea that given sufficient hops, the sum of the delays is approximated distributed Gaussian according to the CLT. When the mean and variance of the delay distribution are modified by the inclusion of the distribution of the random marks, a similar result can be obtained using these modified statistics. Using the mean and variance of the exponential distribution obtained in (6.30) and (6.31), we can apply the CLT and find the distribution of the maximum sum delay. and use this to find an approximate distribution of the bound on the convergence for the dual decomposition decentralized optimization algorithm as

$$F_{B(D)}(b_0) = \Pr(B \leq b_0) = \int_0^{b_0} f_{\max(D)}(B^{-1}(\beta)) \left| \frac{\partial (B^{-1})}{\partial b} \right|_{b=\beta} d\beta,$$

where $f_{\max(D)}(d)$ is the density associated with the distribution

$$F_{\max(D)}(d) = \left(\int_0^{\infty} \frac{1}{\sqrt{(2\pi) h \sigma_D^2}} \exp \left\{ - \frac{\left(d - h \left(\frac{1+\gamma E_{\text{th}}}{\lambda} \right) \right)^2}{2 h \sigma_D^2} \right\} \mathrm{d}d \right)^{N \mathcal{N}_{\max}}. \tag{6.39}$$

where we have applied Lemma 18 and substituted the mean and variance in (6.30) and (6.31) for the exponentially distributed mark values.

In order to ease our conscience regarding the utilization of this approximated distribution, we again provide some simulation results on the empirical maximum delay versus the approximate CLT-based distribution provided by the above result. We simulated results for a random network and the maximum sums of delays associated with multi-hop communication. We execute 300 Monte Carlo runs and show the normalized histograms in Figure 6.9, the results of which demonstrate that the CLT approximation is appropriate for modeling the maximum delay distribution when multi-hop updates are utilized in energy harvesting networks.

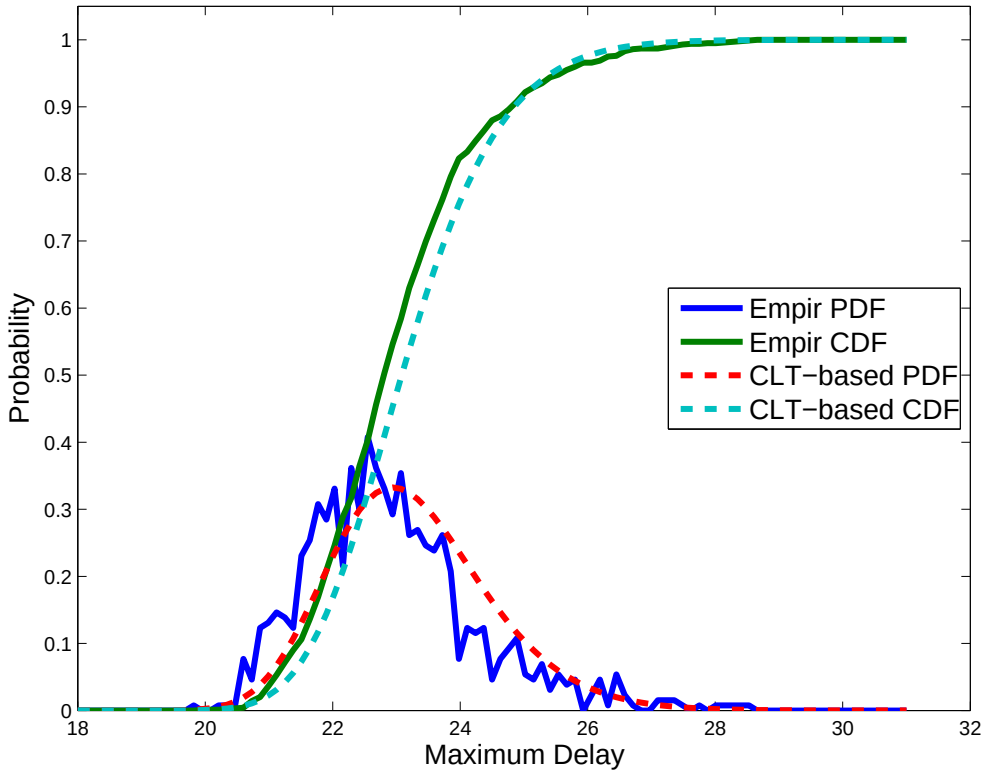


Figure 6.9: *Normalized empirical histogram of maximum delay (solid line) in an energy harvesting wireless sensor network. For the case of exponentially marked arrivals, for* $N = 50$ *nodes,* $\mathcal{N}_{\max} = 25$ *nodes with which to communicate updates,* $\lambda = 3$*,* $E_{\text{th}} = 3$*, with maximum multi-hop length* $h = 20$*. The results are averaged over 300 Monte Carlo runs and shown in comparison to the CLT-based approximation of the maximum delay distribution (dashed line).*

6.4.3 Communication Policy: Fixed Delay with Variable Quantization

We approach the decentralized optimization problem from the perspective of fixing the delay while allowing the quantization to vary. For a fixed time interval, τ, such a distribution on the arrival of energy for agent n can be written as

$$\begin{aligned}\Pr(E_n(\tau) \leq \epsilon) &= \Pr\left(\sum_{j=1}^{N(\tau)} E_{A,j} \leq \epsilon\right) \\ &= \sum_{k=0}^{\infty} \frac{(\lambda\tau)^k}{k!} e^{-\lambda\tau} G^{(k)}(\epsilon).\end{aligned}$$

As in the previous sub section, we analyze various distributions of the mark of the marked Poisson process. Since the nature of the problem for multiple hop communication is inherently complex, we therefore apply some simplifying assumptions for various cases to discover the underlying behavior

Single hop Deterministically Marked

If we assume deterministic unit marks, then this distribution is simplified to

$$\Pr(E_n(\tau) \leq \epsilon) = \sum_{k=0}^{\lfloor \epsilon \rfloor} \frac{(\lambda\tau)^k}{k!} e^{-\lambda\tau},$$

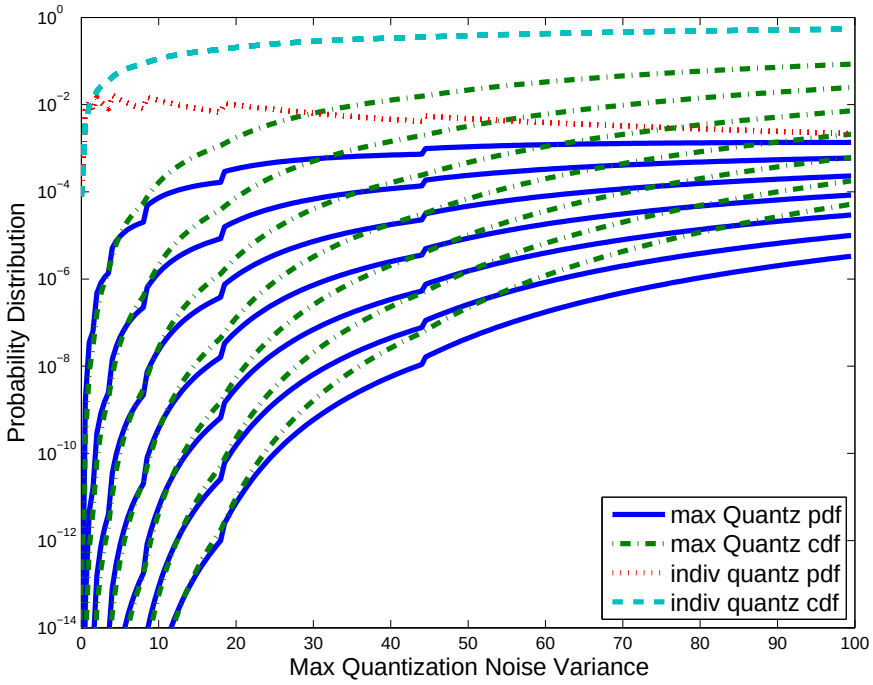
where the associated density is $f_E(\epsilon)$. Utilizing the definition of the quantization function in (6.28), we can say the following about the distribution of the quantization noise variance with an underlying energy arrival process.

Lemma 19. *The distribution function for the maximum quantization noise variance is*

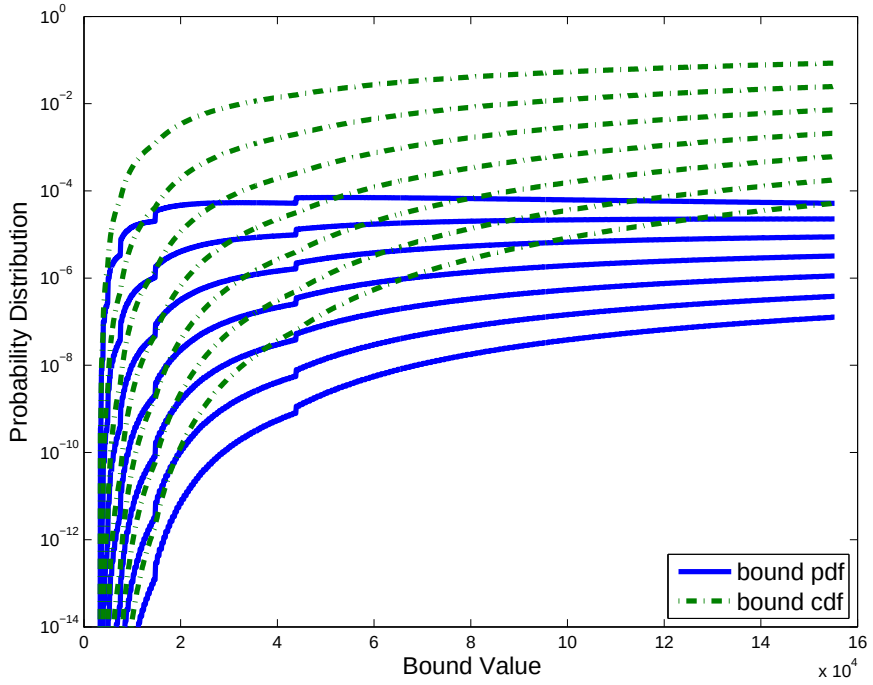
$$F_{\max(Q)}(q) = F_Q^N(q), \tag{6.40}$$

where

$$F_Q(q_0) = \int_0^{q_0} f_E\left(Q^{-1}(\beta)\right) \left|\frac{\partial(Q^{-1})}{\partial q}\right|_{q=\beta} d\beta. \tag{6.41}$$

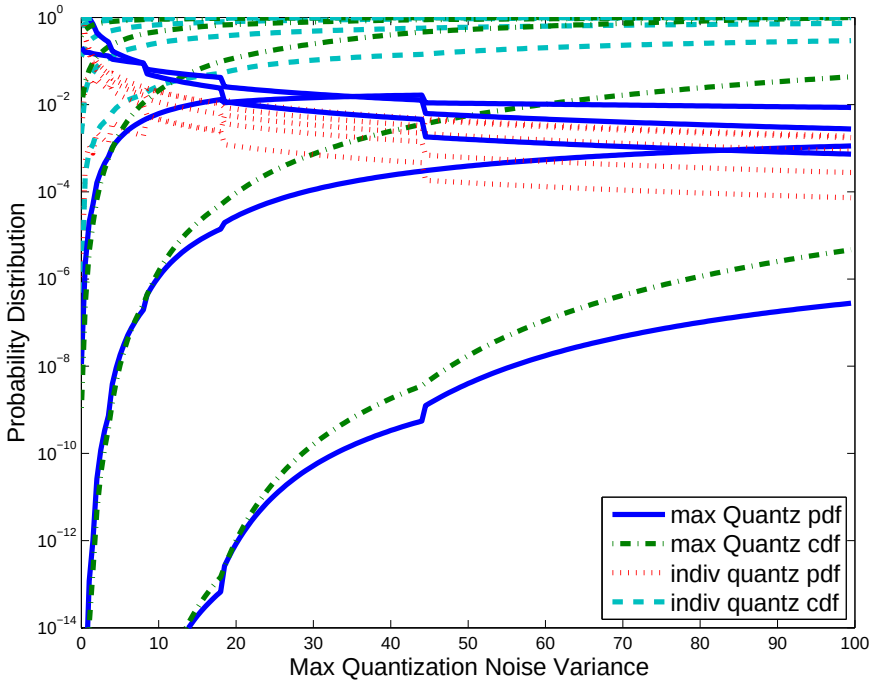


(a)

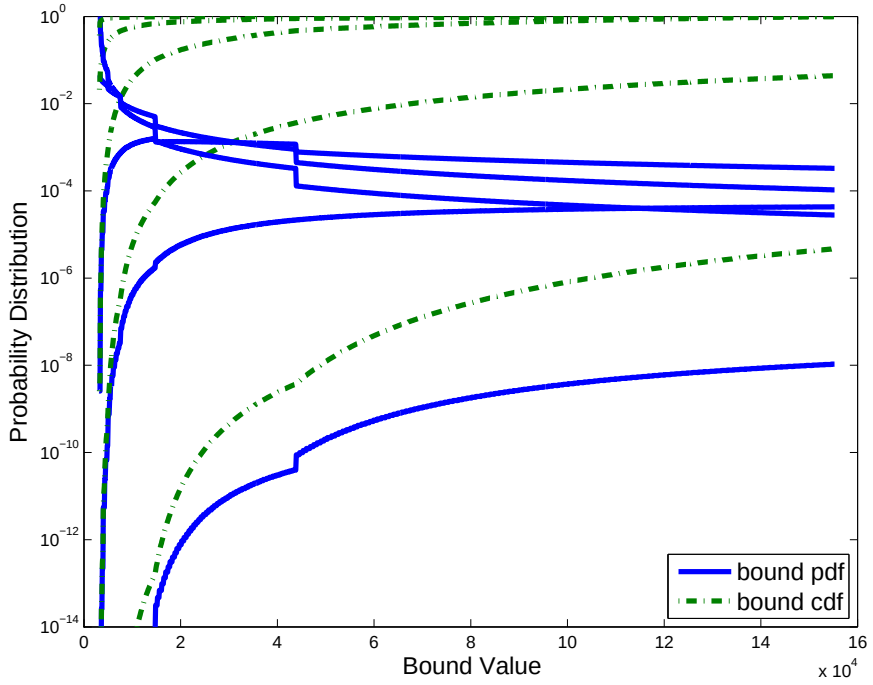


(b)

Figure 6.10: *Example distribution of (a) quantization noise variance and (b) convergence bound for Poisson arrival rate* $\lambda = 3$, $N = (4, 6, \ldots, 16)$, $h = 1$, $\tau = 1$, *nominal values for optimization bound parameters.*



(a)



(b)

Figure 6.11: *Example distribution of (a) quantization noise variance and (b) convergence bound for Poisson arrival rate* $\lambda = (2, 4, \ldots, 10)$*,* $N = 10$*,* $h = 1$*,* $\tau = 1$*, nominal values for optimization bound parameters.*

Proof. This follows by using the standard method from probability for formulating the distribution of a function of a random variable, this requires $Q(\cdot)$ to be a monotonic function, which it is. We also need the assumption of local neighborhood constraints with a single transmission of information to the neighboring agents. □

Using this distribution on the quantization, we obtain a similar result as the previous sub section by looking at the maximum order statistic on all of the quantization noise variances. This maximum order statistic on the quantization is

$$F_{B(Q)}(b_0) = \Pr(B \leq b_0) = \int_0^{b_0} f_{\max(Q)}(B^{-1}(\beta)) \left| \frac{\partial B^{-1}}{\partial b} \right|_{b=\beta} d\beta,$$

where $f_{\max(Q)}(q)$ is the probability density associated with the max quantization distribution, $F_{\max(Q)}(q)$ and $B^{-1}(b) = q$ produces quantization values which produce the provided bound value.

We can again illustrate the probabilistic behavior of the convergence bound by sweeping various system and policy parameters. Figure 6.10 shows the effect of sweeping N from 4 to 16, with the result that the maximum quantization becomes increasingly heavy tailed for large N. Figure 6.11 shows the effect of varying λ from 2 to 10. The resulting CDFs become more sharp with increasing λ, where the quantization and bound distributions become effectively constant at about 1 and 5000, respectively.[2]

Single Hop Stochastically Marked

If we consider stochastically marked compound Poisson process for the energy harvesting process. We are in a similar situation as in the case of finding the delay distribution for single hop stochastic marks. The problem is that the maximum energy distribution involves an N-fold multiplicative expansion of the infinite sum. In order to find a reasonable expression, we attempt to bound the maximum order statistic. First, we note that the complementary

[2]This probability one value of the convergence bound is dependent on the other non-probabilistic parameters of the bound, e.g., T, G, etc.

distribution on the energy takes the form

$$\Pr(E(\tau) \geq \epsilon) = \sum_{k=0}^{\infty} \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!}(1 - G^{(k)}(\epsilon)), \tag{6.42}$$

where $G^{(k)}(\epsilon)$ is the k^{th} convolution of the marked distribution.

Lemma 20. *For τ is a fixed interval, then for an exponentially marked compound Poisson process, the expected value of the arrival energy in the interval τ is*

$$\mathbf{E}[E] = \frac{\lambda\tau}{\gamma}. \tag{6.43}$$

Proof. The general expectation of the generically marked compound process in (6.42) follows as

$$\begin{aligned}
\mathbf{E}[E] &= \int_0^{\infty} \Pr(E(\tau) \geq \epsilon) d\epsilon \\
&= \int_0^{\infty} 1 - F_E(\epsilon) d\epsilon \\
&= \sum_{k=0}^{\infty} \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!} \int_0^{\infty} (1 - G^{(k)}(\epsilon)) d\epsilon,
\end{aligned}$$

where the integral of the complementary CDF produces the expectation whenever $\Pr(E \geq 0) = 1$, which is satisfied for this distribution. For exponentially distributed mark values we can simplify the integral as

$$\begin{aligned}
\mathbf{E}[E] &= \sum_{k=0}^{\infty} \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!} \sum_{j=0}^{k-1} \int_0^{\infty} \frac{(\gamma\epsilon)^j e^{-\gamma\epsilon}}{j!} d\epsilon, \\
&\overset{(a)}{=} \sum_{k=0}^{\infty} \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!} \frac{1}{\gamma} \sum_{j=1}^{k} \int_0^{\infty} \frac{\gamma^j \epsilon^{j-1} e^{-\gamma\epsilon}}{(j-1)!} d\epsilon \\
&= \sum_{k=0}^{\infty} \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!} \frac{1}{\gamma} \sum_{j=1}^{k} \int_0^{\infty} f_\Gamma(\epsilon; j, \gamma) d\epsilon
\end{aligned}$$

where $f_\Gamma(\epsilon; j, \gamma)$ is the Gamma density. In (a) we notice the Gamma distribution form in the terms of the sum over j. Evaluating the integral and continuing our simplification, we

find

$$\begin{aligned}\mathbf{E}[E] &= \sum_{k=0}^{\infty} \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!} \frac{1}{\gamma} \sum_{j=1}^{k} 1 \\ &= \sum_{k=0}^{\infty} \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!} \frac{1}{\gamma} k \\ &= \frac{1}{\gamma} \sum_{k=0}^{\infty} (\lambda\tau) \frac{(\lambda\tau)^{k-1} e^{-\lambda\tau}}{(k-1)!} \\ &\overset{(b)}{=} \frac{\lambda\tau}{\gamma}.\end{aligned}$$

The equality in (b) is made by noting the equivalence of the summation with the Poisson distribution (since adding all masses of a probability distribution should equal one). □

The result is intuitive since, for large γ, the exponential marks will be close to zero. The variance similarly can be derived using the law of total variance [121] to find

$$\sigma^2 = \text{var}(E) = \frac{2\lambda\tau}{\gamma^2}.$$

Using the mean and variance of the probabilistic energy availability, we can use this to find what the max energy across the network is by employing Lemma 16 on page 172. Substituting these values into our quantization function, we can bound the worst quantization across the network from this expected energy availability. Which provides us with a value of Q we can insert into the convergence bound of the decentralized optimization algorithm (Eq. 6.22).

Multi-Hop Deterministically and Stochastically Marked

Employing a similar technique as the case of the maximum delay for single hop stochastically marked processes, we can use the CLT and the maximum order statistic bound (Lemma 16, page 172) to find the limit on the expected available energy. The distribution for harvested energy is used to construct a distribution on the quantization and is incorporated to create an approximate distribution of the convergence bound of the decentralized algorithm.

6.5 Summary

This chapter has addressed some of the aspects of decentralized optimization in energy harvesting wireless sensor networks. For perfect and unlimited communication, these results revert to the case of a centralized controller. However, limited communication introduces delays and quantization errors with affect solution convergence, especially for algorithms utilizing multi-hop updates. We have proposed a modification to dual-decomposition-based decentralized multi-agent optimization. The MIMUM method is developed and based on an iterative optimality condition. While not viable for large networks with many heavily interacting decision variables, the algorithm modification improves the initial results of the dual decomposition for local neighborhoods of decisions by offering a marginal utility selective initial solutions as a 'primed' solution in the case of communication limitations or failures. The proposed modification passes its solution to the decentralized multiplier method for obtain the final solution, which, as simulation results demonstrate, only requires a handful of additional iterations for reasonable stable convergence of the final solution. We have also provided a probabilistic analysis of the effect of using energy harvesting in these decentralized optimization frameworks. This analysis demonstrates the effect of delays and quantization errors on the convergence of decentralized optimization in an energy harvesting multi-agent system.

Chapter 7

Conclusion

In this chapter, we provide concluding remarks on the results of our inquires, explain how the loose ends will wrap up, and propose possible future research directions based on this work.

7.1 Conclusions

This dissertation addresses the modeling and resource optimization of wireless sensor networks, and in particular for energy harvesting equipped WSN. We have put particular focus on poignant questions concerning the optimal utilization of energy resources in a WSN executing a cooperative task. We propose an appropriate metric for WSNs accomplishing estimation tasks. We then propose novel centralized optimization formulations for distributed estimation and tracking with consideration of the energy state of the sensor nodes. This first set of formulations relies only on fixed amounts of stored energy (a battery or similar energy storage device), and attempts to determine the optimal transmission power and rate levels the nodes in the network should communicate. The problem formulation results in a Mixed-Integer Nonlinear Program. Using exhaustive search only as a comparison, we explore the reduction in problem complexity from NP-Hard to polynomial time by a relaxed integer convex formulation of the original non-convex problem. The effectiveness of the proposed algorithms is evaluated using a one-shot (single time instance) solution with and without energy state awareness (battery only). We also benchmark these algorithms

against fixed strategies and offer some performance bounds as well as simulation results. An improvement over uniform bandwidth/energy allocations is seen, and an energy-aware heuristic is used to improve network lifetime.

Our next contribution initiates using a simplification of the communication model which allows us to assume a concave utility function as the distortion metric of interest. We then analyze the solutions to a class of "water-filling" problems, of which the WSN estimate-distortion-minimizing problem is a specific example. Based on a few principles obtained from this analysis, we provide a matched marginal utility algorithm for general problems within our utility function calculus. We then apply this algorithm to the estimation problem by designing and analyzing a WSN-specific, energy-harvesting/aware, distortion minimization. Simulation results demonstrate the performance of the new algorithm. The new method with non-causal information using MPC and prediction of uncertain quantities has a comparable (given only causal information) performance to the standard optimization technique with non-causal information on the uncertain quantities.

Finally, we address the preference of multi-hop communication exhibited by low-power WSNs. We evaluate different decentralized optimization strategies which are limited by the energy-constrained communication. Within this category, we explore the use of optimization decomposition techniques and propose an algorithmic modification. The first stage method of the two-stage algorithm applies a optimality condition-based optimization solution. The second stage is a decentralized version of the method of multipliers using subgradients. The 'priming' algorithm allows the initial decision variables with the 'most importance' in the final solution to receive initial allocations. This provides a better initial 'guess' in the case that communication is limited in the network of agents or if the network experiences communication failures. And lastly, we present an analysis of the convergence bound for the aforementioned dual-decomposition-based decentralized optimization algorithm. Specifically, we investigate how the convergence bound varies probabilistically with respect to the random resources available through energy harvesting. The result is a telling

summary of the trade-offs of energy-conservation and system performance in EH-WSNs, as well as demonstrating the effectiveness and limitations of the algorithms presented herein.

In summary, the following topics were address with novel approaches and analyses:

- Energy harvesting/storage and communication channel models appropriate for wireless sensor networks
- Adoption of appropriate metric for cooperative estimation tasks; formulation and simplification of centralized estimation and tracking problem
- Analysis of simplified concave utility problems and a general utility maximization algorithm based on underlying optimal conditions from the Lagrangian with multiple (linear) inequality constraints
- Extension of utility maximization algorithm to decentralized optimal resource utilization as augmentation of subgradient-based iterative dual decomposition techniques
- Probabilistic analysis of delay and quantization effects on the convergence region of subgradient-based optimization using decentralized dual decomposition

7.2 Future Work

In this section, we propose extensions to, and more complete explorations, of our existing research.

- In Chapter 3 there are several modeling aspects which we have not addressed. Particularly, we have not considering interference in the network communication. While we justify the lack of interference as a feature of low-power communication, this is perhaps only valid in particular scenarios. For example, a deployed sensor network in a open field, versus in a building. The attenuation due to the structure of a building may prevent interference from having an effect, whereas in an open field nodes may be able to interfere more easily with each others' transmissions.

- Chapter 4 considers a linear model for the tracking and measure of the observed process. More realistic measurement and process models could be employed to obtain a optimization solution which more accurately reflects existing hardware capabilities [29].

- The optimization algorithm in Chapter 5 works well with linear inequality constraints and concave non-decreasing differentiable utility functions. In order to speed up convergence, it would be useful to employ some type of generic gradient- (or subgradient- for non differentiable) method, while maintaining the desirable feature of the MIMUM method which keeps the active marginal utilities matched.

- The probabilistic analysis of energy harvesting WSNs in Chapter 6 considers the arrival rates and mark distributions to be identical for all nodes. While this has some credence in system distributed across small regions, WSNs with a dramatic spatial spread may experience non-uniform stochastic availability of harvested energy resources. A more complete analysis of the effects of delay and quantization error on decentralized optimization in these cases is desirable.

- Additionally, energy harvesting WSNs in Chapter 6 operate on assumptions about the energy harvesting process, and, while backed by previous research [3, 61, 122], it would be useful to employ non-parametric statistics on actual energy harvesting data to extract a useful data-based statistical model of the energy harvesting capabilities.

Some future research directions which could have their start from the work in this dissertation include the following.

- Since networks rely on multiple-access (MA) schemes to prevent/recover from data transmission collisions. Our analysis of the decentralized optimization convergence bounds does not consider collisions and packet losses. How could these MA algorithms in EH-WSNs be advantaged by incorporating an underlying model of the energy harvesting process? Particularly when executing some decentralized, data critical tasks,

packet losses can be extremely detrimental to convergence, thus it is advantageous to find methods which minimizes these types of errors. Previous work which has looked at similar problems include [123–125].

- A large network of nodes relying on EH may not be able to keep every node active. What type of techniques can be used to adjust the sleep/wake-up duty cycle so as to accomplish the network cooperative task? In particular, when the task to be accomplished involves signals which have a sparse representation in some basis. With such an assumption, only a subset of the nodes need observe the process and share their information with any neighbors which were previously inactive. We would make the distinction [126, 127] in work by continuing the theme of explicitly considering the energy state of the nodes and any energy harvesting capability available to them.

- Considering the WSN from the graph-theoretic perspective, we want to know how the availability of the energy from the harvesting processes affects the connectivity of the graph structure defined by the sensor nodes and their associated communication links. For example, we could ask: How likely is it for a disjoint subgraph which has not communicated to any other subgraphs to remain disconnected (in a communication-link sense) in the next update step? That is, how will the limited energy from the harvesting process affect the ability of the sub-graphs of the network to stay in communications with the rest of the network? The authors in [128] have looked at a similar problem, but have not considered energy harvesting. The authors in [129] approach the problem of how robust a network of agents is in completing a task. It would be interesting to extend this concept of robustness to account for the effects of uncertain resource availability in energy harvesting systems.

- Lastly, we would like to ask questions about the effects of energy harvesting on other types of tasks in distributed systems (e.g., distributed computing tasks). Cooperative signal processing does not necessarily always involve optimization (at least not from a

centralized perspective), and we would like to know what effect unpredictable resources has on the performance of such tasks. See for example [130].

Bibliography

[1] E. Serpedin, H. Li, A. Dogandžić, H. Dai, and P. Cotae, "Distributed Signal Processing Techniques for Wireless Sensor Networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, pp. 1–3, 2008.

[2] S. Y. Sim, H. S. Jeon, G. S. Chung, S. K. Kim, S. J. Kwon, W. K. Lee, and K. S. Park, "Fall detection algorithm for the elderly using acceleration sensors on the shoes." in *33rd Annual International Conference of the IEEE EMBS*, vol. 2011, Jan. 2011, pp. 4935–8.

[3] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, pp. 32–es, Sep. 2007.

[4] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *4th ACM/IEEE international symposium on Information processing in sensor networks, (IPSN'05)*, 2005.

[5] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *ACM International Conference on Embedded Networked Sensor Systems*, 2004, pp. 95–107.

[6] J. L. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Transactions on Automatic Control*, vol. AC-24, no. 2, pp. 266–269, 1979.

[7] A. Willsky, M. Bello, D. A. Castanon, B. Levy, and G. Verghese, "Combining and updating of local estimates and regional maps along sets of one-dimensional tracks," *IEEE Transactions on Automatic Control*, vol. 27, no. 4, pp. 799–813, Aug. 1982.

[8] D. A. Castanon and D. Teneketzis, "Distributed estimation algorithms for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 30, no. 5, pp. 418–425, May 1985.

[9] E. Ayanoğlu, "On optimal quantization of noisy sources," *Information Theory, IEEE Transactions on*, vol. 36, no. 6, pp. 1450–1452, Nov. 1990.

[10] J. A. Gubner, "Distributed estimation and quantization," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1456–1459, 1993.

[11] W.-M. Lam, A. R. Reibman, and R. Reibman, "Quantizers for Decentralized Estimation Systems," *IEEE Transactions on Communications*, vol. 41, no. 11, pp. 1602–1605, Nov. 1993.

[12] H. Papadopoulos, G. Wornell, and a.V. Oppenheim, "Sequential signal encoding from noisy measurements using quantizers with dynamic bias control," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 978–1002, Mar. 2001.

[13] V. Megalooikonomou and Y. Yesha, "Quantizer design for distributed estimation with communication constraints and unknown observation statistics," *Communications, IEEE Transactions on*, vol. 48, no. 2, pp. 181–184, Feb. 2000.

[14] S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51–60, Mar. 2002.

[15] A. Ribeiro, Z.-q. Luo, and G. B. Giannakis, "Distributed Compression-Estimation using wireless sensor networks," *IEEE Signal Processing Magazine*, no. July, pp. 27–41, 2006.

[16] J. Fang and H. Li, "Adaptive Distributed Estimation of Signal Power from One-Bit

Quantized Data," *IEEE Transactions On Aerospace And Electronic Systems*, vol. 46, no. 4, 2010.

[17] Z.-Q. Luo, "An isotropic universal decentralized estimation scheme for a bandwidth constrained ad hoc sensor network," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 735–744, Apr. 2005.

[18] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith, "Joint estimation in sensor networks under energy constraints," *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, vol. 00, no. c, pp. 264–271, 2004.

[19] V. Aravinthan, S. K. Jayaweera, and K. Al Tarazi, "Distributed estimation in a power constrained sensor network," in *IEEE Vehicular Technology Conference*. IEEE, 2006, pp. 1048–1052.

[20] J. Li and G. Al Regib, "Rate-Constrained Distributed Estimation in Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1634–1643, May 2007.

[21] ——, "Energy-constrained distributed estimation in Wireless Sensor Networks," in *Military Communications Conference, 2007 - MILCOM 2007*, 2007, pp. pp. 1 – 7.

[22] ——, "Maximizing Network Lifetime for Estimation in Multi-Hop Wireless Sensor Networks," *2008 Proceedings of 17th International Conference on Computer Communications and Networks*, pp. 1–6, Aug. 2008.

[23] A. Sripad and D. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 5, pp. 442–448, Oct. 1977.

[24] Y. Mostofi and R. Murray, "Effect of time-varying fading channels on the control performance of a mobile sensor node," *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, vol. 00, no. c, pp. 317–324, 2004.

[25] S. Balasubramanian, S. Jayaweera, and K. Namuduri, "Energy-aware, collaborative tracking with ad-hoc wireless sensor networks," *IEEE Wireless Communications and Networking Conference, 2005*, pp. 1878–1883, 2005.

[26] M. Cardei and D.-Z. Du, "Improving Wireless Sensor Network Lifetime through Power Aware Organization," *Wireless Networks*, vol. 11, no. 3, pp. 333–340, May 2005.

[27] H. Zhu, I. Schizas, and G. Giannakis, "Power-efficient dimensionality reduction for distributed channel-aware kalman tracking using WSNs," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3193–3207, Aug. 2009.

[28] O. Ozdemir, R. Niu, and P. K. Varshney, "Channel Aware Target Localization With Quantized Data in Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1190–1202, Mar. 2009.

[29] J. L. Williams, J. W. Fisher III, and A. S. Willsky, "Approximate Dynamic Programming for Communication-Constrained Sensor Network Management," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4300–4311, 2007.

[30] S. Aeron, V. Saligrama, and D. A. Castanon, "Efficient Sensor Management Policies for Distributed Target Tracking in Multihop Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2562–2574, Jun. 2008.

[31] R. Olfati-Saber and R. M. Murray, "Consensus Problems in Networks of Agents With Switching Topology and Time-Delays," vol. 49, no. 9, pp. 1520–1533, 2004.

[32] ——, "Consensus Protocols for Networks of Dynamic Agents," *Proceedings Of The American Control Conference*, 2003.

[33] R. Olfati-Saber, "Distributed Kalman Filter with Embedded Consensus Filters," *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 8179–8184, 2005.

[34] ——, "Distributed Kalman filtering for sensor networks," *2007 46th IEEE Conference on Decision and Control*, pp. 5492–5498, 2007.

[35] H. Hindi, "A tutorial on convex optimization," *2006 American Control Conference*, no. 1, pp. 686–696, 2006.

[36] D. P. Palomar, "A Tutorial on Decomposition Methods for Network Utility Maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[37] S. Kalyanasundaram, E. K. P. Chong, and N. B. Shroff, "Optimal resource allocation in multi-class networks with user-specified utility functions," *Computer Networks*, vol. 38, no. 5, pp. 613–630, Apr. 2002.

[38] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, 7th ed. Cambridge Univ. Press, Feb. 2004.

[39] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for Communication networks: Shadow prices , Pproportional Fairness and Stability," *The Journal of the Operational Research Society*, vol. 49, no. 3, 1998.

[40] S. H. Low and D. E. Lapsley, "Optimization Flow Control, I : Basic Algorithm and Convergence," *IEEE/ACM Trans. on Networking*, vol. 7, no. 6, pp. 1–16, 1999.

[41] S. H. Low, "A Duality Model of TCP and Queue Management Algorithms," *IEEE/ACM Trans. on Networking*, vol. 11, pp. 525–536, 2003.

[42] M. Mehyar, D. Spanos, and S. H. Low, "Optimization Flow Control with Estimation Error," *Source*, vol. 00, no. C, 2004.

[43] L. Zhang and Q. Gao, "Performance Analysis of Universal Distributed Estimation in Bandwidth-Constrained Sensor Networks," *IEEE Signal Processing Letters*, vol. 15, pp. 473–476, 2008.

[44] E. K. P. Chong and B. Brewington, "Decentralized rate control for tracking and surveillance networks," *Ad Hoc Networks*, vol. 5, no. 6, pp. 910–928, Aug. 2007.

[45] J. N. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D. dissertation, MIT, 1984.

[46] N. A. Lynch, *Distributed algorithms*. Morgan Kaufmann, 1996.

[47] A. Nedić and A. Ozdaglar, "On the Rate of Convergence of Distributed Subgradient Methods for Multi-Agent Optimization," in *46th IEEE Conference on Decision and Control*, no. 1, 2007, pp. 4711–4716.

[48] ——, "Distributed subgradient methods for multi-agent optimization," Technical Report 2575, MIT LIDS., Tech. Rep., 2007.

[49] H. Terelius, U. Topcu, and R. M. Murray, "Decentralized Multi-Agent Optimization via Dual Decomposition," 2011.

[50] D. Zheng, "Physical-Layer Aware Control and Optimization in Stochastic Wireless Networks," Ph.D. dissertation, 2007.

[51] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," *2007 46th IEEE Conference on Decision and Control*, pp. 5492–5498, 2007.

[52] D. P. Spanos and R. M. Murray, "Motion planning with wireless network constraints," *Proceedings of the 2005 American Control Conference*, pp. 87–92, Jun. 2005.

[53] J. Solanki and N. Schulz, "Using Intelligent Multi-Agent Systems for Shipboard Power System Reconfiguration," in *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*. Ieee, 2005, pp. 212–214.

[54] V. Jojic, S. Gould, and D. Koller, "Accelerated dual decomposition for MAP inference," in *Proceedings 27th International Conference on Machine Learning*, 2010.

[55] A. F. T. Martins, M. A. T. Figueiredo, P. M. Q. Aguiar, N. A. Smith, and E. P. Xing, "Alternating Directions Dual Decomposition," *CoRR*, no. 2007, pp. 1–47, 2010.

[56] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[57] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with Energy Harvesting Nodes in Fading Wireless Channels : Optimal Policies," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1732–1743, 2011.

[58] K. W. Fan, Z. Zheng, and P. Sinha., "Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks," in *6th ACM conference on Embedded network sensor systems, (SenSys'08)*, 2008.

[59] R.-S. Liu, P. Sinha, and C. E. Koksal, "Joint Energy Management and Resource Allocation in Rechargeable Sensor Networks," in *IEEE Conference on Information Communication (INFOCOM '10)*, 2010.

[60] C. Moser, L. Thiele, D. Brunelli, and L. Benini, "Adaptive Power Management for Environmentally Powered Systems," *IEEE Transactions on Computers*, vol. 59, no. 4, pp. 478–491, 2010.

[61] B. Zhang, R. Simon, and H. Aydin, "Maximal Utility Rate Allocation for Energy

Harvesting Wireless Sensor Networks," in *ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, 2011, pp. 7–16.

[62] C. Rusu, R. Melhem, and D. Moss, "Multi-version Scheduling in Rechargeable Energy-aware Real-time Systems," in *Proc. of 15th Euromicro Conference on Real-Time Systems*, 2003.

[63] C. Moser, J.-J. Chen, and L. Thiele, "Reward Maximization for Embedded Systems with Renewable Energies," *2008 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 247–256, Aug. 2008.

[64] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. New York: Athena Scientific, 1997.

[65] N. J. Roseveare and B. Natarajan, "Distributed Tracking with Energy Management in Wireless Sensor Networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3494–3511, Oct. 2012.

[66] ——, "Optimizing Network Lifetime for Distributed Tracking with Wireless Sensor Networks," in *ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, 2011, pp. 41–48.

[67] ——, "Energy-Aware Distributed Tracking in Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference, WCNC 2011*, 2011, pp. 363–368.

[68] ——, "A Structured Approach to Optimization of Energy Harvesting Wireless Sensor Networks," in *IEEE Consumer Communications and Networking Conference, CCNC 2012 - Wireless Communications Track*, 2013, pp. 420–425.

[69] D. P. Bertsekas, "Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC," *European Journal on Control*, vol. 11, no. 4-5, pp. 1–50, 2005.

[70] N. J. Roseveare and B. Natarajan, "An Alternative Perspective on Utility Maximization in Energy Harvesting Wireless Sensor Networks," *[under review in] IEEE Transactions on Vehicular Technology*, 2013.

[71] B. Wahlberg, S. P. Boyd, and M. Annergren, "An ADMM Algorithm for a Class of Total Variation Regularized Estimation Problems," in *Proceedings 16th IFAC Symposium on System Identification*, 2012.

[72] N. J. Roseveare and B. Natarajan, "A Modified Distributed Dual Algorithm for Decentralized Optimization in Multi-Agent Systems," in *(process for submission) in IEEE Global Communications Conference, Globcom 2013.*, 2013.

[73] ——, "Probabilistic Analysis of Communication Strategies for Decentralized Signal Processing in Energy Harvesting Systems," *[to be submitted] in IEEE Transactions on Signal Processing*, 2013.

[74] W. L. Brogan, *Modern control theory*, 3rd ed. Prentice-Hall, 1991.

[75] R. F. Stengel, *Optimal Control and Estimation.* Dover, 1994.

[76] H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[77] G. B. Folland, *Real Analysis: Modern Techniques and Applications.* Wiley-Interscience, 1999.

[78] J. A. Gubner, *Probability and Random Processes for Electrical and Computer Engineers.* Cambridge University Press, 2006.

[79] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.

[80] J. B. Hiriart-Urruty and C. Lemaréchal, "Fundamentals of Convex Analysis." Springer, Abridged version of Convex Analysis and Minimization Algorithm volumes 1 and 2., 2001.

[81] J. Demmel, *Applied Numerical Linear Algebra*. SIAM, 1997.

[82] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. Johns Hopkins University Press, 1996.

[83] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, Dec. 1984.

[84] Y. Nemiravski and A. Nerterov, *Interior-Point Polynomial methods in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[85] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. Springer, 1994.

[86] J. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. New Jersey: Prentice Hall, 1995.

[87] E. Todorov, *Optimal Control Theory*. MIT Press, 2006.

[88] R. E. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

[89] Y. Li, L. Krakow, E. K. P. Chong, and K. Groom, "Approximate stochastic dynamic programming for sensor scheduling to track multiple targets," *Digital Signal Processing*, vol. 19, no. 6, pp. 978–989, Dec. 2009.

[90] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Volume I*, 3rd ed. Belmont, MA: Athena Scientific, 2005, vol. II.

[91] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[92] W. C. Jakes, *Microwave Mobile Communications*. New York: John Wiley & Sons, 1975.

[93] B. Natarajan, C. Nassar, and V. Chandrasekhar, "Generation of correlated Rayleigh fading envelopes for spread spectrum applications," *IEEE Communications Letters*, vol. 4, no. 1, pp. 9–11, 2000.

[94] J. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2000.

[95] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith, "Power scheduling of universal decentralized estimation in sensor networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 413–422, Feb. 2006.

[96] Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs, CT: YBS Publishing, 1995.

[97] T. Wu and Q. Cheng, "Distributed estimation over fading channels using one-bit quantization," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5779–5784, Dec. 2009.

[98] A. Ribeiro and G. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor Networks-part I: Gaussian case," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 2784–2796, Mar. 2006.

[99] C. Menger, *Principles of Economics*. Auburn, AL: Ludwig von Mises Institute, 2004.

[100] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley-Interscience, 2006.

[101] D. J. Sakrison, "Source Encoding in the Presence of Random Disturbance," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 165–167, 1968.

[102] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[103] N. Goela and M. Gastparl, "Linear Compressive Networks," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, 2009, pp. 159–163.

[104] C. Yu and G. Sharma, "Distributed Estimation Using Reduced Dimensionality Sensor Observations: A Separation Perspective," in *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, 2008, pp. 150–154.

[105] R. Krishnan and B. Natarajan, "Joint Power and Quantization Bits Optimization for Distributed Estimation in Wireless Sensor Networks," *IEEE GLOBECOM 2008*, vol. 2, no. 785, pp. 2–4, 2008.

[106] Y. Bar-shalom and L. Campo, "The effect of common process noise on the two-sensor fused-track covariance," *IEEE Transactions On Aerospace and Electronic Systems*, no. 10, pp. 803–805, 1986.

[107] O. E. Drummond, "Tracklets and a hybrid fusion with process noise," *Proceedings of SPIE*, vol. 3163, pp. 512–524, 1997.

[108] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.

[109] R. Krishnan and B. Natarajan, "Joint Power and Quantization Optimization for Target Tracking in Wireless Sensor Networks," *IEEE GLOBECOM 2008*, no. 785, pp. 1–5, 2008.

[110] A. D'Costa, V. Ramachandran, and A. M. Sayeed, "Distributed Classification of Gaussian Space-Time Sources in Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1026–1036, Aug. 2004.

[111] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, Feb. 2006.

[112] K. Eriksson, S. Shi, N. Vucic, M. Schubert, and E. G. Larsson, "Globally Optimal

Resource Allocation for Achieving Maximum Weighted Sum Rate," *Communications Society*, no. 4, 2010.

[113] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*, 1st ed. Prentice Hall, 1993.

[114] V. A. Nguyen and Y. P. Tan, "On Network Flow Problems with Convex Cost," *Journal of ICT*, vol. 3, no. 1, pp. 33–51, 2004.

[115] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, p. 877, 1976.

[116] P. Tseng, "Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, 2001.

[117] S. Bose, "Cyber-Physical Modeling, Analysis, and Optimization - A Shipboard Smartgrid Reconfiguration Case Study," Ph.D. dissertation, Kansas State University, 2012.

[118] H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, 3rd ed. Academic Press, 1998.

[119] H. A. David and H. N. Nagaraja, *Order Statistics*. Wiley, 2004.

[120] D. Bertsimas, K. Natarajan, and C.-P. Teo, "Tight Bounds on Expected Order Statistics," *Probability in the Engineering and Informational Sciences*, vol. 20, pp. 667–686, 2006.

[121] N. A. Weiss, *A Course in Probability*. AddisonWesley, 2005.

[122] A. Kansal and M. Srivastava, "An environmental energy harvesting framework for sensor networks," *Proceedings of the 2003 International Symposium on Low Power Electronics and Design, 2003. ISLPED '03.*, pp. 481–486, 2003.

[123] B. Gurakan, O. Ozel, J. Yang, and S. Ulukus, "Two-way and multiple-access energy harvesting systems with energy cooperation," in *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. Ieee, Nov. 2012, pp. 58–62.

[124] K. Tutuncuoglu and A. Yener, "Multiple Access and Two-way Channels with Energy Harvesting and Bi-directional Energy Cooperation," Wireless Communications and Networking Laboratory, Pennsylvania State University, Tech. Rep., 2013.

[125] R. A. Raghuvir, D. Rajan, and M. D. Srinath, "Capacity of the Multiple Access Channel in energy harvesting wireless networks," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. Ieee, Apr. 2012, pp. 898–902.

[126] Q. Ling, Z. Tian, and S. Member, "Decentralized Sparse Signal Recovery for Compressive Sleeping Wireless Sensor Networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 7, pp. 3816–3827, 2010.

[127] X. Jia and G. Chen, "Extending lifetime of sensor surveillance systems in data fusion model," in *IEEE Wireless Communications and Networking Conference, 2011*, 2011, pp. 1452–1457.

[128] T. Hossmann, F. Legendre, and T. Spyropoulos, "From Contacts to Graphs: Pitfalls in Using Complex Network Analysis for DTN Routing," in *IEEE INFOCOM Workshops 2009*. Ieee, Apr. 2009, pp. 1–6.

[129] V. Gupta, C. Langbort, and R. M. Murray, "On the robustness of distributed algorithms," *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 3473–3478, 2006.

[130] N. Batouma and J.-l. Sourrouille, "Dynamic Adaption of Resource Aware Distributed Applications," *International Journal of Grid and Distributed Computing*, vol. 4, no. 2, pp. 25–42, 2011.

Appendix A

Algorithm Convergence

The purpose of this appendix is to show the convergence of the matched marginal utility method, that is, to prove Theorem 5. We first established some useful propositions to aid in said objective. The reader is reminded that we are dealing with a problem involving the maximization of concave utilities.

Proposition 4. *Given resource allocation at iteration κ, $\mathbf{r}^{\kappa}$, the update $\mathbf{r}^{\kappa+1}$ ($\geq \mathbf{r}^{\kappa}$) from the minimum in step 6, Algorithm 1 provides an optimal improvement to the solution.*

Proof. Given amount of resource used in the update $R^{\kappa} = ||\mathbf{r}^{\kappa+1} - \mathbf{r}^{\kappa}||_1$, there is no allocation of resource $\bar{\mathbf{r}}^{\kappa+1}$ such that $U(\mathbf{r}^{\kappa+1}) < U(\bar{\mathbf{r}}^{\kappa+1})$. This is seen from the utilization of the optimality condition discussed in Lemma 9 and Lemma 10, and by the matching of the active marginal utilities, optimality is maintained in the update. □

The idea is that the marginal utility of each resource should be 'balanced' to the price (Lagrange multiplier) of the most strict associated constraint, so implicitly also to each other through said constraints.

Proposition 5. *Algorithm 1 maintains feasibility in each iteration, as the increment of any resource allocation is limited by the most strict constraint associated with it.*

Proposition 5 follows simply from the fact that our algorithm *constructs* allocations such that no infeasibility is ever incurred.

We now offer the proof of the convergence of the matched marginal utility method.

Proof. (of Theorem 5)

Since each iterative update in step 4 is optimal, given in Proposition 4, and feasibility is maintain as postulated in Proposition 5, then the final update in which no further allocation can be updated produces the final optimal point, i.e., there are a finite number of improving iterations. All of the levels (Lagrange multipliers) are balanced among the resources and their associated constraints so that the standard Lagrangian for the constrained problem satisfies the appropriate first order necessary and sufficient conditions [38]. □

Appendix B

Convergence Bounds for Dual Decomposition in a WSN System

This appendix proves the composition of the terms in Theorem 6 on page 161. For no packet losses, the network delivers all gradient updates, $g(t)$, to the appropriate nodes, but with some delays bounded by D. These updates are quantized, $\mathcal{Q}(g(t))$, and have a maximum quantization noise variance of Q. For the sake notational ease, let us define the vector of delayed dual variables as

$$\bar{\nu}(t) = [\nu_{11}^T(t - d_{11}) \ \ldots \ \nu_{nm}^T(t - d_{nm}) \ \ldots \ \nu_{NN}^T(t - d_{NN})]^T,$$

so that the update rule accounting for delays may be concisely written as: $\bar{\nu}(t+1) = \bar{\nu}(t) + \alpha(t)g(t)$. After T steps, the bound on the current to optimal dual variable requires $||\nu(T+1) - \nu^*||_2^2 \geq 0$ and results in

$$\left|\left|\nu(T) - \sum_{\Delta=0}^{D} \Pi_\Delta(\alpha(t)\mathcal{Q}\left(g(t)\right)) - \nu^*\right|\right|_2^2 \geq 0.$$

The above equation is expanded in Theorem 6 as

$$||\nu(0) - \nu^*||_2^2 \tag{B.1}$$

$$+ 2\sum_{t=0}^{T}\sum_{\Delta=0}^{D}[\Pi_\Delta(\alpha(t)\mathcal{Q}(g(t)))]^T(\nu(t) - \nu^*) \tag{B.2}$$

$$+ \sum_{t=0}^{T}\alpha(t)\sum_{\Delta=0}^{D}||\Pi_\Delta(\alpha(t)\mathcal{Q}(g(t)))||_2^2 \geq 0, \tag{B.3}$$

but we left the simplification of the terms in the bound for this appendix.

Here we show the simplification of the above terms, but first there are several small lemmas which we list here for convenience and clarity.

Lemma 21. *From the assumption of bounded gradients*

$$||\bar{\nu}(t+1) - \bar{\nu}(t)|| = \alpha(t)||g(t)|| \leq \alpha(t)G.$$

Lemma 22. *From the assumption of bounded gradients*

$$||\nu(t) - \bar{\nu}(t)|| \leq G \sum_{\Delta=t}^{t+D-1} \alpha(\Delta).$$

Proof. This lemma follows from Lemma 21 and from the definition of the projection operator. □

Lemma 23. *Similar to result in [49], under the assumption of bounded time delays and bounded gradients*

$$g(t)^T(\bar{\nu}(t) - \nu^*) \leq G^2 \sum_{\Delta=t}^{t+D-1} \alpha_\Delta + q(\nu(t)) - q^*$$

Proof. This lemma follows from Lemma 22 by adding and subtracting $\nu(t)$ inside the right argument of the inner product, and then by applying Cauchy-Schwarz inequality. □

Now we finish the task of simplifying the bound terms. The second term of the bound in B.2 can be rewritten by applying the projection operator to the dual variable and then adding and subtracting $\bar{\nu}(t)$ inside $\Pi_\Delta(\cdot)$ to get the sum

$$\sum_{t=0}^{T} \mathcal{Q}(g(t))^T \sum_{\Delta=0}^{D} \Pi_\Delta(\nu(t) - \bar{\nu}(t)) + \sum_{t=0}^{T} \mathcal{Q}(g(t))^T \sum_{\Delta=0}^{D} \Pi_\Delta(\bar{\nu}(t) - \nu^*).$$

Applying Lemma 23 to the first sum over Δ we get

$$\begin{aligned} \mathcal{Q}(g(t))^T \sum_{\Delta=t}^{t+D-1} \Pi_\Delta(\nu(t) - \bar{\nu}(t)) &\leq (||\mathcal{Q}(g(t))|| \sum_{\Delta=0}^{D} ||(\nu(t) - \bar{\nu}(t))|| \\ &\leq \alpha G D(G+Q)(D+1). \end{aligned} \tag{B.4}$$

The second sum over Δ can be bounded as

$$\mathcal{Q}(g(t))^T \sum_{\Delta=t}^{t+D-1} \Pi_\Delta(\bar{nu}(t) - \nu^*) \leq \mathcal{Q}(g(t))^T(\nu(t) - \nu^*)$$
$$\leq q(\nu(t)) - q^*. \tag{B.5}$$

Finally, the last term in the original bound expression, (B.3), can be written as

$$\alpha^2 \sum_{t=0}^{T} \left|\left| \sum_{\Delta=0}^{D} \Pi_\Delta(\mathcal{Q}(g(t))) \right|\right|_2^2 = \sum_{t=0}^{T} ||\nu(t+1) - \nu(t)||_2^2.$$

Simplifying the terms in the above using the triangle inequality and the definition of the projection operator, we obtain

$$\begin{aligned} ||\nu(t+1) - \nu(t)||_2^2 &= \alpha^2 \left|\left| \sum_{\Delta=0}^{D} \Pi_\Delta(\mathcal{Q}(g(t))) \right|\right|_2^2 \\ &\leq \alpha^2 \sum_{\Delta=0}^{D} ||\Pi_\Delta(\mathcal{Q}(g(t)))||_2^2 \\ &\leq \alpha^2 \sum_{\Delta=0}^{D} ||\mathcal{Q}(g(t))||_2^2 \\ &\leq \alpha^2 (D+1)^2 (G+Q)^2, \end{aligned} \tag{B.6}$$

where the added effect of Q due to the quantization of the gradient updates is motivated by [117].

Appendix C

Variance of Delay in Exponentially Marked Compound Poisson Process

The purpose of this appendix is to show the derivation of the variance of an compound Poisson process with exponentially marked random arrivals (the proof of Lemma 15). We show a generic derivation,using t and z_{th} in place of application-specific notation.

Let us review the preliminaries. The compound Poisson process is given by $Z(t) = \sum_{k=1}^{X(t)} Y_k, \quad \forall\ t \geq 0$, where $X(t) \sim \text{Poisson}(\lambda t)$ and $\Pr(Y_k \leq y) = G(y)$ the distribution function of the Y_k's.

We denote the n^{th} convolution of the mark distribution (this assumes independent Y_k's) as

$$\begin{aligned} G^{(n)}(y) &= \Pr(Y_1 + \cdots + Y_n \leq y) \\ &= \int_{-\infty}^{\infty} G^{(n-1)}(y-z)dG(z) \end{aligned} \tag{C.1}$$

where $G^{(0)} = \begin{cases} 1 & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases}$. Then

$$\begin{aligned} \Pr(Z(t) \leq z) &= \Pr\left(\sum_{k=1}^{X(t)} Y_k \leq z\right) \\ &= \sum_{n=0}^{\infty} \Pr\left(\sum_{k=1}^{X(t)} Y_k \leq z \,|X(t) = n\right) \frac{(\lambda t)^n e^{-\lambda t}}{n!} \\ &= \sum_{n=0}^{\infty} \frac{(\lambda t)^n}{n!} e^{-\lambda t} G^{(n)}(z), \end{aligned} \tag{C.2}$$

since $X(t)$ is independent of the Y_k's.

We want to know the length of the delay, d, with respect to the compound process thresholded at some z_{th}. That is, we want to know the time how long the value of the compound process fails to meet this threshold,

$$\Pr(Z(t) \leq z_{\text{th}}) \;\Leftrightarrow\; \Pr(T \geq t),$$

where T is the time at which the threshold is reached. From C.2 we can write

$$\Pr(T > t) = \sum_{n=0}^{\infty} \frac{(\lambda t)^n e^{-\lambda t}}{n!} G^{(n)}(z_{\text{th}}).$$

Proof. We begin the proof by noting that we need to the density with respect to delay from the distribution at a threshold z_{th}. That is,

$$\begin{aligned} f_{T(z_{\text{th}})}(t) &= \frac{d}{dt}\left(1 - \Pr(T > t)\right) \\ &= -\sum_{n=0}^{\infty} \frac{(\lambda n(\lambda t)^{n-1} - \lambda(\lambda t)^n)e^{-\lambda t}}{n!} G^{(n)}(z_{\text{th}}) \\ &= \sum_{n=0}^{\infty} \frac{(\lambda^{n+1}(t^n e^{-\lambda t}) - n\lambda^n(t^{n-1}e^{-\lambda t})}{n!} G^{(n)}(z_{\text{th}}). \end{aligned}$$

From this density we would like to find the second moment,

$$\mathbf{E}[T^2] := \int_0^{\infty} t^2 f_{T(z_{\text{th}})}(t)dt.$$

This requires a great deal of algebra. We can now rewrite the internal integral as an expectation over the Gamma density,

$$\begin{aligned} \mathbf{E}[T^2] &= \int_0^{\infty} t^2 f_{T(z_{\text{th}})}(t)dt \\ &= \sum_{n=0}^{\infty} \frac{1}{n!}\left[\Gamma(n+1)\int_0^{\infty} \frac{t^2\lambda^{n+1}t^n e^{-t\lambda}dt}{\Gamma(n+1)} - n\Gamma(n)\int_0^{\infty} \frac{t^2\lambda^n t^{n-1}e^{-t\lambda}dt}{\Gamma(n)}\right] G^{(n)}(z_{\text{th}}) \\ &= \sum_{n=0}^{\infty} \frac{1}{n!}\left[\Gamma(n)(n+1)\mathbf{E}_{\Gamma(n+1)}[T^2] - \Gamma(n)n\mathbf{E}_{\Gamma(n)}[T^2]\right] G^{(n)}(z_{\text{th}}) \\ &= \sum_{n=0}^{\infty} \left[(n+1)\left(\frac{n+1}{\lambda^2} + \left(\frac{n+1}{\lambda}\right)^2\right) - n\left(\frac{n}{\lambda^2} + \left(\frac{n}{\lambda}\right)^2\right)\right] G^{(n)}(z_{\text{th}}), \end{aligned}$$

since, for the gamma density $f_\Gamma(t;\beta,n) := \frac{\beta^n t^{n-1} e^{-\beta t}}{\Gamma(n)}$, the second moment is

$$\mathbf{E}[T^2] = \sigma^2 + \mu^2 = \frac{n}{\beta^2} + \left(\frac{n}{\beta}\right)^2.$$

Simplifying and substituting the k^{th} convolution of the exponential mark distribution: $F_E(z_{\text{th}}) := \sum_{k=n}^{\infty} \frac{(\gamma z_{\text{th}})^k e^{-\gamma z_{\text{th}}}}{k!}$, we obtain

$$\begin{aligned}
\mathbf{E}[T^2] &= \sum_{n=0}^{\infty} \left[(n+1)\left(\frac{n+1}{\lambda^2} + \left(\frac{n+1}{\lambda}\right)^2\right) - n\left(\frac{n}{\lambda^2} + \left(\frac{n}{\lambda}\right)^2\right)\right] G^{(n)}(z_{\text{th}}) \\
&= \sum_{n=0}^{\infty} \left[(n+1)\left(\frac{n+1+(n+1)^2}{\lambda^2}\right) - n\left(\frac{n+n^2}{\lambda^2}\right)\right] G^{(n)}(z_{\text{th}}) \\
&= \frac{1}{\lambda^2} \sum_{n=0}^{\infty} \sum_{k=n}^{\infty} \left[3n^2+5n+2\right] \frac{(\gamma z_{\text{th}})^k e^{-\gamma z_{\text{th}}}}{k!} \\
&= \frac{1}{\lambda^2} \sum_{k=0}^{\infty} \sum_{n=0}^{k} \left[3n^2+5n+2\right] \frac{(\gamma z_{\text{th}})^k e^{-\gamma z_{\text{th}}}}{k!}.
\end{aligned}$$

Using properties of finite sums we can simplify to

$$\begin{aligned}
\mathbf{E}[T^2] &= \frac{1}{\lambda^2} \sum_{k=0}^{\infty} \sum_{n=0}^{k} \left[3n^2+5n+2\right] \frac{(\gamma z_{\text{th}})^k e^{-\gamma z_{\text{th}}}}{k!} \\
&= \frac{1}{\lambda^2} \sum_{k=0}^{\infty} \left[3\frac{k(k+1)(2k+1)}{6} + 5\frac{k(k+1)}{2} + k + 1\right] \frac{(\gamma z_{\text{th}})^k e^{-\gamma z_{\text{th}}}}{k!} \\
&= \frac{1}{\lambda^2} \sum_{k=0}^{\infty} \left[k^3+4k^2+4k+1\right] \frac{(\gamma z_{\text{th}})^k e^{-\gamma z_{\text{th}}}}{k!}.
\end{aligned}$$

The sum and terms in the sum constitute a linear combination of the first through third moments of the Poisson distribution with parameter γz_{th}, thus, the second moment of the delay becomes

$$\begin{aligned}
\mathbf{E}[T^2] &= \frac{1}{\lambda^2}(M_3 + 4M_2 + 4M_1 + 1) \\
&= \frac{1}{\lambda^2}\left(\frac{\gamma z_{\text{th}} - (\gamma z_{\text{th}} + \frac{1}{3})}{\sqrt{\gamma z_{\text{th}}}} + 4(\gamma z_{\text{th}} + (\gamma z_{\text{th}})^2) + 4\gamma z_{\text{th}} + 1\right) \\
&= \frac{1}{\lambda^2}\left(4(\gamma z_{\text{th}})^2 + 8\gamma z_{\text{th}} + 1 + \frac{1}{3\sqrt{\gamma z_{\text{th}}}}\right).
\end{aligned}$$

Here, for mean μ, median ν, and variance σ^2, we have used the definition of the second and third moments

$$M_2 = \mathbf{E}[k^2] = \sigma^2 + \mu^2 \qquad M_3 = \mathbf{E}[k^3] = \frac{\mu - \nu}{\sigma}.$$

We apply a standard definition of the variance and find it by subtracting the square of the mean (derived previously in Chapter 6) from the second moment,

$$\begin{aligned}
\sigma^2 &= \mathbf{E}[(T-\mu)^2] \\
&= \frac{1}{\lambda^2}\left(4(\gamma z_{\text{th}})^2 + 8\gamma z_{\text{th}} + 1 + \frac{1}{3\sqrt{\gamma z_{\text{th}}}}\right) - \left(\frac{1+\gamma z_{\text{th}}}{\lambda}\right)^2 \\
&= \frac{1}{\lambda^2}\left(3(\gamma z_{\text{th}})^2 + 6\gamma z_{\text{th}} + \frac{1}{3\sqrt{\gamma z_{\text{th}}}}\right),
\end{aligned}$$

producing the final result. □