A STUDY ON CERTAIN THEORETICAL AND PRACTICAL PROBLEMS IN WIRELESS NETWORKS

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

BY

MEHMET AKİF ANTEPLİ

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

SEPTEMBER 2010

Approval of the thesis:

A STUDY ON CERTAIN THEORETICAL AND PRACTICAL PROBLEMS IN WIRELESS NETWORKS

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

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Elif Uysal-Bıyıkoğlu
Supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Elif Uysal-Bıyıkoğlu
Electrical and Electronics, METU _____

Assoc. Prof. Dr. Ali Özgür Yılmaz
Electrical and Electronics, METU _____

Assoc. Prof. Dr. Çağatay Candan
Electrical and Electronics, METU _____

Dr. Sevgi Zübeyde Gürbüz
Space Technologies Research Institute, TUBITAK _____

Date: **13.09.2010**

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

Name, Last Name: MEHMET AKİF ANTEPLİ

Signature :

ABSTRACT

A STUDY ON CERTAIN THEORETICAL AND PRACTICAL PROBLEMS IN WIRELESS NETWORKS

Antepli, Mehmet Akif
M.S., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Elif Uysal-Bıyıkoğlu

September 2010, 61 pages

The aim of the thesis is to investigate the design of efficient wireless networks through practical as well as theoretical considerations.

We constructed a wireless sensor network (WSN) testbed with battery operated nodes capable of RF communication. The system is a centralized tree-based WSN to study challenges of target modeling, detection, and localization. The testbed employed magnetic sensors, on which relatively few results have been reported in the literature. A ferrous test target is modeled as magnetic dipole by validating experimentally. The problem of sensor sensitivity variation is addressed by including sensitivity estimates in model validation. After reliably detecting the target, maximum-likelihood and least-squares techniques are applied for localization. Practical considerations of constructing a WSN utilizing magnetic sensors addressed.

Maximum-lifetime operation of these networks requires joint consideration of sensing and communication. Energy harvesting is promising to overcome this major challenge for energy-constrained systems. In the second part of the thesis, we considered the minimization of transmission completion time for a given number of bits per user in an energy harvesting multiuser

communication system, where the energy harvesting instants are known beforehand. The two-user case with achievable rate region having structural properties satisfied by the AWGN Broadcast Channel is studied. It is shown that the optimal scheduler ends transmission to both users at the same time while deferring a nonnegative amount of energy from each energy harvest for later use. The problem is formulated as an optimization problem and solved by exploiting its special structure.

Keywords: wireless sensor network, magnetic sensing, localization, packet scheduling, energy harvesting

ÖZ

KABLOSUZ AĞLARDA BELİRLİ KURAMSAL VE PRATİK PROBLEMLER ÜZERİNE BİR ÇALIŞMA

Antepli, Mehmet Akif
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Elif Uysal-Bıyıkoğlu

Eylül 2010, 61 sayfa

Bu tezin amacı verimli kablosuz ağ tarasımının kuramsal açıdan olduğu kadar pratik açıdan da incelenmesidir.

Bir kablosuz algılayıcı ağı (KAA) test düzeneği, RF haberleşmesi yapabilen ve batarya ile çalışan düğümler kullanılarak hayata geçirilmiştir. Sistem, hedefin modellenmesindeki, tespit edilmesindeki, ve lokalize edilmesindeki zorlukları incelemek üzere merkezi ağaç yapılı bir KAA olarak kurulmuştur. Test düzeneği, üzerinde nisbeten az sonuçlar bulunan manyetik algılayıcıları kullanmıştır. Ferromanyetik bir test hedefi, deneysel olarak doğrulanarak manyetik dipol olarak modellenmiştir. Manyetik algılayıcının hassasiyet kestirimi, model doğrulama sürecine eklenerek, hassasiyetteki değişimlerin yarattığı problem işaret edilmiştir. Güvenilir hedef tespiti yapıldıktan sonra, hedefi lokalize etmek üzere enbüyük olabilirlik ve en küçük kareler kestirim yöntemleri uygulanmıştır. KAA larda manyetik algılayıcıların kullanılmasındaki pratik anlayışlar işaret edilmiştir.

Bu ağların en uzun yaşam süresiyle çalışabilmesi, algılamanın ve haberleşmenin bir arada düşünülmesini gerektirmektedir. Enerji hasatı, enerji kısıtlı sistemlerde varolan bu temel

zorluğu aşmak için umut vaadeden bir seçenektir. Tezin ikinci kısmında, enerji hasatı yapabilen ve enerji hasat zamanları önceden bilinen çokkullanıcılı bir haberleşme sisteminde, her bir kullanıcıya gönderilmesi gereken belirli sayıda bit varken, iletim tamamlanma süresinin enküçültülmesi incelenmiştir. Yapısal özellikleri AWGN çoğagönderim kanalı tarafından sağlanan ulaşılabilir bir hız bölgesine sahip iki kullanıcılı durum çalışılmıştır. En iyi çizelgeleyicinin, her bir enerji hasatından negatif olmayan bir miktar enerjiyi sonradan kullanılmak üzere erteleyerek, her iki kullanıcı için de iletimi aynı anda bitirdiği gösterilmiştir. Problem, bir eniyileme problemi olarak yazılmıştır ve içerisindeki özel yapı kullanılarak çözülmüştür.

Anahtar Kelimeler: kablosuz algılayıcı ağı, manyetik algılama, lokalizasyon, paket çizelgeleme, enerji hasatı

To my dear fiancée Burcu, and my mother

ACKNOWLEDGMENTS

I am deeply grateful to my supervisor Assoc. Prof. Dr. Elif Uysal-Bıyıkoğlu not only for her valuable guidance, encouragement, motivating ideas in the technical sense but also for her friendly attitude, and smiling face throughout this thesis work. It is my fortune to work with her and I appreciate her being the role model for every graduate student during their future academic studies, as well as me.

During the thesis work, I had the chance to meet with Dr. Sevgi Zübeyde Gürbüz and I am sincerely thankful to her for her guidance and encouragement during the tiring experimental work in this thesis. Her friendly approach gave me the opportunity to have long conversations on lots of social and political topics beyond our discussions on technical issues.

I would also like to sincerely thank to Burcu, my dear fiancée, for her support, and tolerance even during my stressful and unbearable times. Her sweet nature, gentle voice, and beautiful look make life worth to live for me.

My mother is the person in the backstage to whom I owe everything, my every achievement for her tolerating me for 27 years and never complaining about it, smiling all the time with full of love, always doing exactly the thing in her mind even though seeming to listen my thoughts beforehand and hence making me smile, trying to feed me at any time and anywhere, falling asleep while praying to God for the sake of my goodness, as a result, performing with all her effort to make life easier for me.

I would also like to thank to Muhammet Fatih Bayramoğlu for his wise advices and friendship starting from the times of his being the assistant of the Circuit Theory Lab. class during my undergraduate studies.

Lastly, I acknowledge sponsorship of TUBITAK under Kariyer grant 106E119.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Efficient ad-hoc wireless network design requires judicious use of limited resources such as memory and energy under the performance requirements of specific applications. Memory needs to be allocated efficiently between communication layers and signal processing modules as most distributed algorithms are running in ad-hoc networks requiring to store network-wide state and data in limited memory spaces. On the other hand, energy is the most critical resource as most of today's ad-hoc wireless networks are powered from batteries. However, since battery life is inherently limited, this can create maintenance problems, especially for larger networks.

Specifically, in wireless sensor networks, nodes have low processing capabilities, and memory on the order of a few kBytes, and they are strictly dependent on limited batteries as they are are designed to be compact and low cost, and hence manufactured with low-complexity processors with small amount of internal memory, and low-power radios to work with AA-size batteries. These networks demand employing efficient routing, collaborative signal processing and data fusion techniques to preserve robust operation and long network lifetime with minimum maintenance. However, techniques used to reach this aim may vary depending on the application requirements.

Magnetic sensing with wireless ad-hoc sensor networks is a rather new research area. Beyond its commercial applications such as traffic monitoring, and parking lot space detecting systems, magnetic sensors are more attractive for large scale surveillance applications as most intruders have ferromagnetic ingredients and magnetic sensors are passive devices. However, saturation of magnetic sensors under strong magnetic fields, differences in sensor sensitivities due to variations in sensor hardware, susceptibility of magnetic sensors to ambient noise,

and highly nonlinear signal characteristics of ferromagnetic bodies are the major drawbacks of magnetic sensor networks. Hence, along with the above challenges, providing ubiquitous coverage, efficient signal processing for continual network operation, target detection, localization, tracking and energy-efficient sensor scheduling should be more investigated in wireless magnetic sensor networks.

Using infrastructures capable of ambient energy harvesting is another option to power wireless nodes. The nodes may harvest energy through solar cells, piezoelectricity, temperature gradients, etc. Even though energy efficient scheduling in wireless networks has been well investigated during the last decade, considering physical layer, routing and scheduling in a cross-layer framework in energy harvesting systems is a rather new concept. In these systems, energy harvests as well as data arrivals can be represented as two independent random processes and hence, transmission power and rates need to be adaptively adjusted to make optimal data scheduling. However, such a general problem formulation is rather difficult to solve. Hence, understanding optimal scheduling in the general problem requires ideal assumptions and the investigation of basic problem formulations.

The first part of this thesis work focuses on the problems of constructing a wireless magnetic sensor network and practical solutions to these. In this context, a WSN to detect and localize a ferromagnetic target is implemented. The focus of the second part of the thesis is to solve an idealized scheduling problem in energy harvesting systems i.e., offline packet scheduling to minimize the total transmission completion time in a two user broadcast link where the sender is capable of harvesting energies at known time instants.

1.1 Outline of the Thesis

In Chapter 2, we present a wireless magnetic sensor network implementation to detect and localize a ferromagnetic test target. The network is constructed as a tree-based wireless network with a gateway to establish connection of the network with a PC acting as a fusion center. After investigating the magnetic sensor properties, we emphasize soft-calibration of magnetic sensors to obtain reliable magnetic readings. The chapter continues with modeling the magnetic signature of a simple ferromagnetic test target (an iron bar, 20 cm in height and 3.5 cm in diameter) and validation of the model. Experiments conducted to model the test

target followed by target parameter estimation jointly with magnetic sensor sensitivity which varies across sensors due to variations in sensor hardware have been explained. A detection scheme is developed which effectively handles ambient magnetic field tracking, and fast magnetic sensor calibration to sustain continual operation under saturation conditions caused by relative closeness of sensors and the moving target. Following detection of the target, localization is performed centrally on the PC side via Simulated Annealing algorithm using the magnetic readings sent by wireless sensor nodes. The performance of localization is investigated in the mean localization error sense. Finally sequential localization is applied on a specified path to have an understanding on the combinatorial effect of target modeling, sensor sampling frequency, target motion, detection and localization, hence on the overall network functionality.

An idealized version of the general scheduling problem in an energy harvesting system is presented in Chapter 3. The problem is the minimization of packet transmission completion time to two users in a broadcast link in which sender harvests energies at time instants which is known beforehand. After presenting the system model, properties of optimal schedule is investigated through a sequence of Lemmas. First of all, it is shown that in the optimal schedule, the scheduler need not change the transmission rates of each user (hence, transmission power) between energy harvest. Then, the reason behind ending transmission to both users at the same time is presented followed by the formulation of optimal offline packet scheduling as an optimization problem. The chapter continues with the adaptation of the algorithm FlowRight, which is presented in the earlier literature, to solve the scheduling problem in an energy harvesting broadcast link. The complexity of the algorithm is investigated and a numerical example is provided to illustrate the operation of the algorithm.

Chapter 4, summarizes the key points addressed throughout this thesis work and provides possible future studies.

CHAPTER 2

FERROMAGNETIC TARGET DETECTION AND LOCALIZATION WITH A WIRELESS SENSOR NETWORK

2.1 INTRODUCTION

The material in this chapter is partially accepted to appear in Military Communications Conference 2010 [1] and has been performed as a part of the project [24].

Magnetic sensors have been used in a variety of applications, the most ubiquitous being vehicle detection, tracking and classification. Simple vehicle detection schemes have been proposed for aiding cars in locating parking spaces in [2], and for traffic measurement in [3], where a WSN with magnetic sensors was used to process experimentally observed vehicle signatures. In [4], an air traffic control application was introduced for multiple target tracking and classification of aircraft and cars. Yet another application of magnetometers is magnetic anomaly detection for hidden ferromagnetic objects, presented in [5] and [6].

While previous magnetic sensing applications with WSNs reported in the literature focus on the detection of large ferromagnetic bodies, such as vehicles, our work focuses on a more fine-tuned detection regime utilizing a much smaller test target. Moreover, in addition to detection, we explore the possibilities of signal exploitation, and more specifically, localization. Although some work on localization of a ferromagnetic target with known parameters has been conducted [7], our work seeks to explore the challenges involved in modeling the properties of a target in the context of localization in magnetic WSNs. The theme of this chapter is to unify the problems of modeling, detection, and localization of unknown ferromagnetic targets.

In this chapter, we also discuss the key practical issues that arise when dealing with the implementation of magnetic WSNs, especially challenges presented by the magnetic sensors themselves. In particular, issues such as sensitivity estimation and sensor calibration, which are critical to the realization of a magnetic sensing system, are discussed.

A brief description of the system architecture and the essentials of MTS310CB magnetometer circuitry, especially in regards to sensitivity estimation and sensor calibration via software, is presented in Section 2.2. In Section 2.3, the modeling of the ferrous test target is discussed, followed by details of our detection algorithm in Section 2.4. Sections 2.5 and 2.6 present the methodology and results of the localization and sequential localization algorithms utilized, respectively. Section 2.7 presents our conclusions.

2.2 SYSTEM ARCHITECTURE

The ferromagnetic target detection and tracking system constructed for this work is composed of MICAz motes equipped with MTS310CB [8] sensor boards (Fig.2.1), a gateway MICAz mote with a MIB520 programming board from Crossbow Technology, and a PC acting as the fusion center. Each MICAz mote is equipped with an IEEE 802.15.4 compliant, Chipcon CC2420 RF transceiver and Atmega 128L microcontroller. The MTS310CB sensor board includes a Honeywell HMC1002 2-axis magnetometer [9].

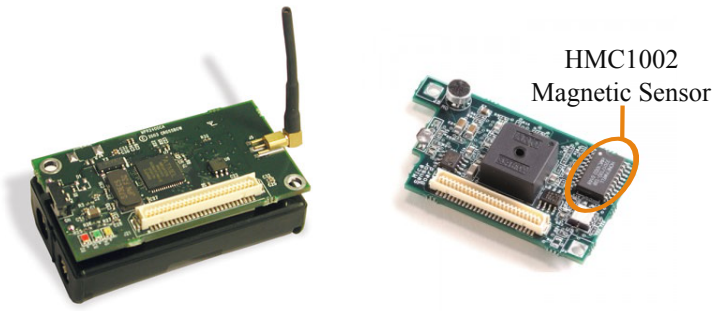


Figure 2.1: MICAz wireless sensor node is on the left and MTS310CB sensor board with HMC1002 two-axis magnetic sensor is on the right.

To program the MICAz motes, the open-source operating system TinyOS-2.1.0 was used. TinyOS uses the component based programming language called NesC (Network Embedded Systems-C), which is a modified version of C programming language. Code generation is per-

formed by using Eclipse IDE v3.5.0 with Yeti2 [10] which is a TinyOS plug-in for Eclipse. On the PC side, Ubuntu 9.10 was used as the host operating system (OS) with TinyOS-2.x toolchain installed. TinyOS program images are disseminated through the gateway node to the remote nodes over the air to provide network-wide programming using the protocol Deluge [11].

Besides MICAz, we have tested two other hardware platforms, TelosB from CrossBow Inc. and Sensenode from Genetlab Inc, in terms of ease in programming and communicating with PC, application development environment, compatible sensor boards with magnetic sensors, and communication performances. TelosB and Sensenode both use the same transceiver with MICAz, i.e., CC2420, but different microcontrollers. They both have TI MSP430 micro-controller with 10kB RAM (a critical resource for embedded systems) which is more than twice than that of Atmega 128L micro-controller (4kB RAM). TelosB has an on-board USB connector to program or communicate with a PC which makes it suitable for rapid application development. Also, any node can be used as the gateway in a network composed of TelosB nodes which brings flexibility. However, the major drawback of TelosB is that there is no magnetic sensors on it and Crossbow does not have any sensor board compatible with TelosB. On the other hand, Sensenode has a sensor board with a three-axis magnetic sensor. However, Genetlab only supports TinyOS-1.x the older version of the OS. This makes application development difficult, especially using improved routing protocols developed for TinyOS-2.x, since there are structural differences between the two versions of the OS. Even though we have made the necessary software patch for Sensenode to support Tinyos-2.x, yet we have faced with hardware problems while programming and communicating with PC, as well as network-wide programming. Hence, we have decided to use MICAz and MTS310CB with HMC1002 magnetic sensor in this work.

A centralized tree-based network was established using the Collection Tree Protocol (CTP) [12, 13], and Four Bit Link Estimator (4BitLE) [14] implemented by the TinyOS Network Protocol Working Group (Net2WG) to reliably route packets to the PC via gateway node and efficiently estimate wireless link quality, respectively [1]. CTP uses data traffic as active network topology probes to detect and recover from routing loops as fast as possible and adaptive beaconing depending on the network topology changes to minimize broadcasts. 4BitLE com-

[1] During our experiments with CTP and 4BitLE, we have made bug fixes and reported to Net2WG for the necessary updates in the TinyOS-2.x CVS repository.

putes single-hop bidirectional expected transmission count (ETX) [15] metric of communication with single-hop neighbours. This is a hybrid ETX value in which ETX in the forward direction is estimated by using the received ACK packets in reply to unicast data packets whereas ETX in the reverse direction is estimated by using the received periodic broadcast packets from neighbouring nodes. For messages required for network-wide maintenance, such as rebooting the network, the dissemination protocol, DRIP [16], by Net2WG, was used. On the PC side, a java application, WSN Visualizer, is adapted from the open-source project, Octopus [17], to visualize network topology and magnetic coverage via enabling user configurable sensor node locations, collect magnetic sensor readings from the network and perform target localization. Our system architecture performing ferromagnetic target detection and localization is shown in Fig. 2.2.

2.2.1 Magnetometer Measurements

The Honeywell HMC1002 magnetometer is comprised of two Wheatstone bridges, whose resistance change according to the magnetic field applied to each magnetometer axis. The change in resistance, triggered by a change in magnetic field, is measured by the MICAz's analog-to-digital converter (ADC) circuit, which converts the sensor output into a 10 bit ADC reading in each axis.

The ADC may be related to the applied magnetic flux as follows [2]. First, the differential output voltage of each Wheatstone bridge is expressed in terms of the applied magnetic flux as [18]

$$V_{diff} = S \times V_b \times B_s + V_{offset} \tag{2.1}$$

where S is the sensitivity (mV/V_{ex}/Gauss), V_b is the bridge supply voltage (V), B_s is the magnetic flux applied to the bridge (Gauss), and V_{offset} is the bridge offset voltage (mV). Here, the sensitivity is an especially important factor as the precise sensitivity value was seen in our experiments to vary from sensor to sensor. The HMC1002 data sheet [9] states that the sensor sensitivity varies between 2.5 to 4 mV/V_{ex}/Gauss. However, due to individual variations, the sensitivity must be estimated, as explained in more detail in Section IV.

[2] The following analysis has been performed to present the reasons behind the conversion of magnetic field (in mGauss) to sampled values by MICAz (in ADC count) which is left untouched in the literature.

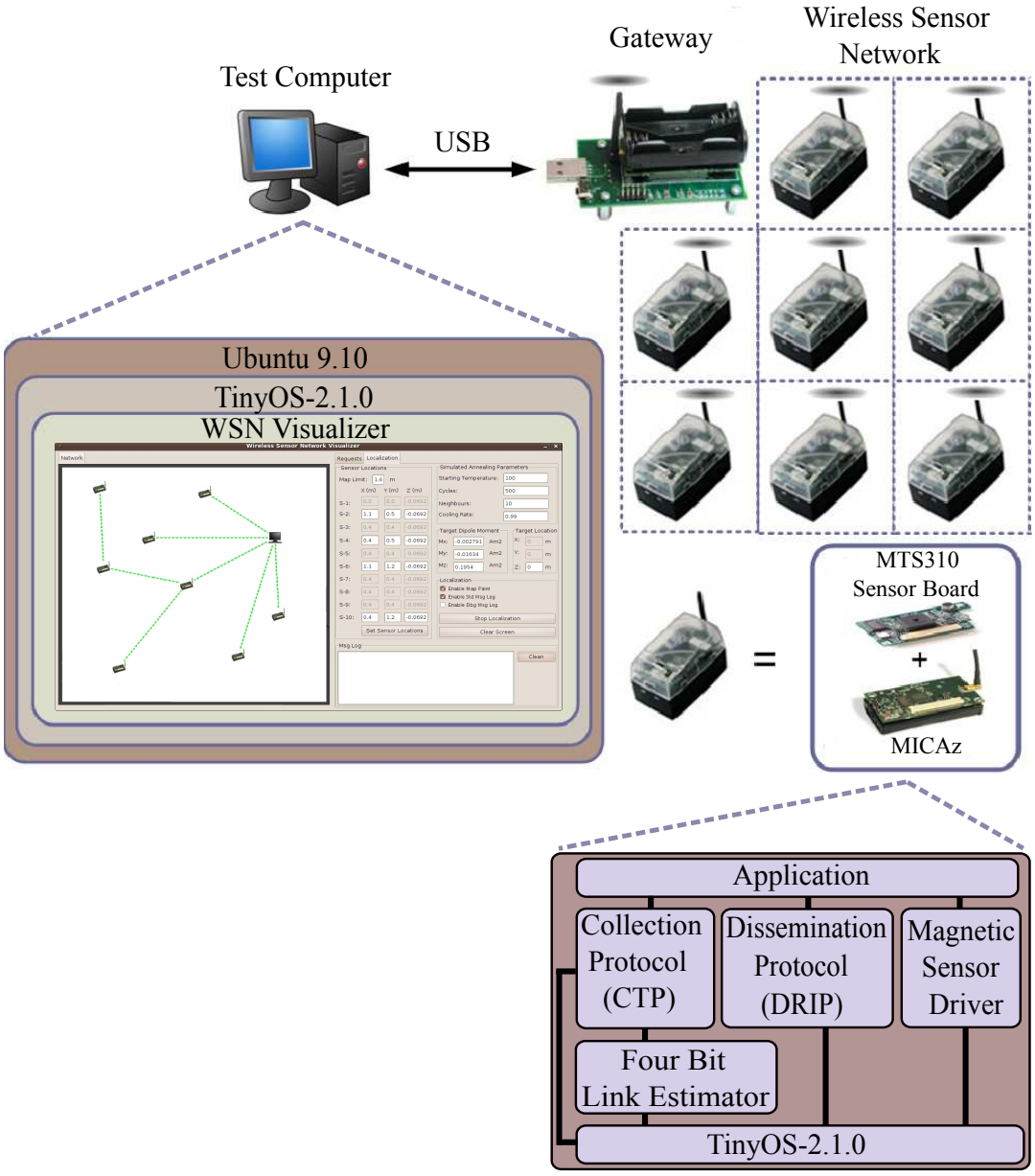


Figure 2.2: System architecture for ferromagnetic target detection and localization. Target detection is performed by MICAz motes whereas target localization is by WSN Visualizer application in PC. Wireless sensor network is connected to PC through a gateway node comprised of a MICAz mote and MIB520 programming board.

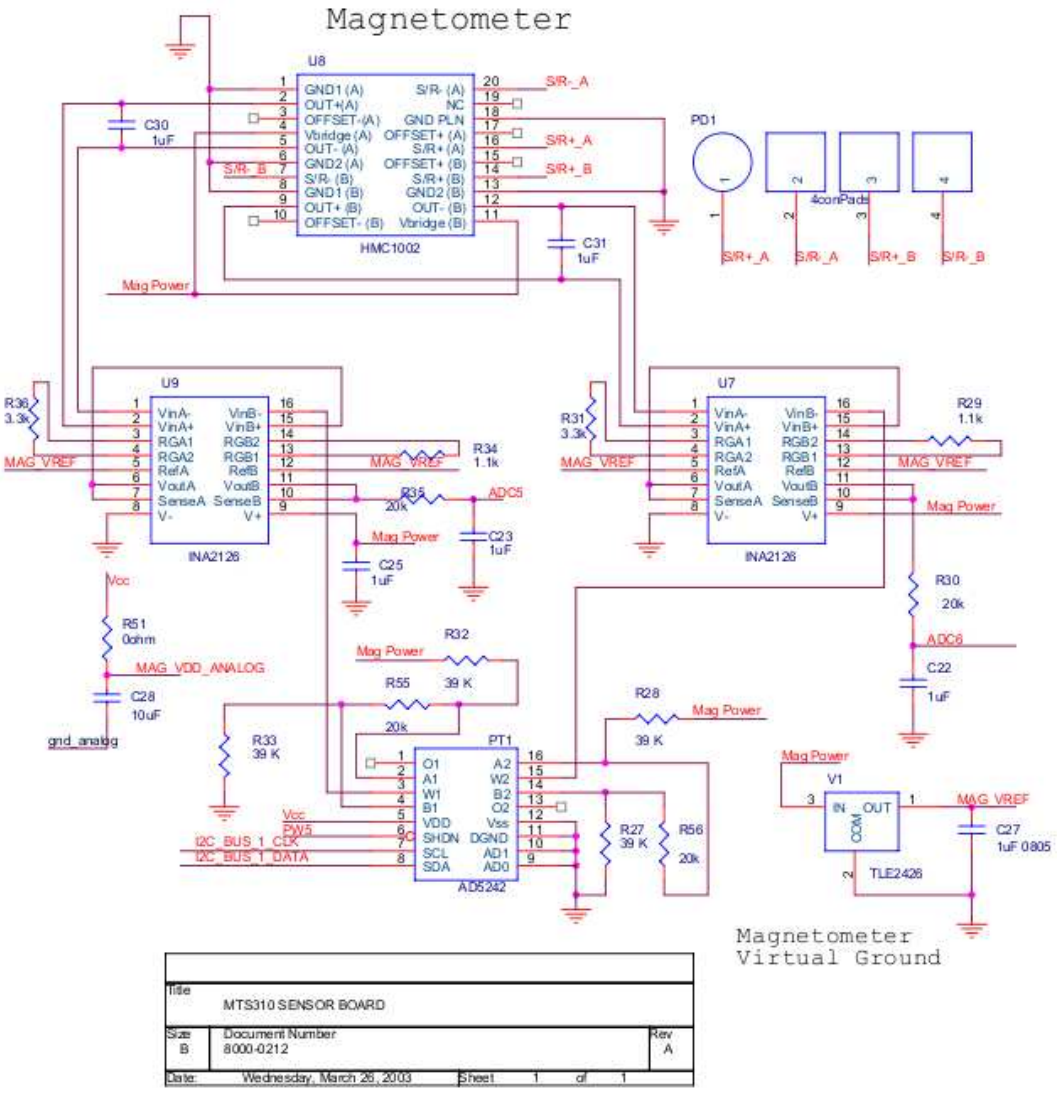


Figure 2.3: MTS310CB Magnetometer Circuit Diagram

The differential output voltage V_{diff} is then mapped to the full span of the ADC using additional hardware on the MTS310CB sensor board as shown in Fig.2.3. In particular, V_{diff} is amplified over two stages by a micropower instrumentation amplifier dual version (INA2126) and a dual-channel, I^2C (Inter-Integrated Circuit) compatible, 256 position, digital potentiometer (AD5242) is used in a voltage divider configuration to ensure that the amplification output is kept within the limit of the ADC span [3]. Our experiments show that one unit of change in potentiometer value offsets the ADC reading by 54 ADC counts, a result that was also analytically validated as follows. The gain of each amplifier stage is given as [19]

$$G_1 = 5 + \frac{80k\Omega}{R_{G_1}} = 5 + \frac{80k\Omega}{3.3k\Omega} = 29.242 \tag{2.2}$$

$$G_2 = 5 + \frac{80k\Omega}{R_{G_2}} = 5 + \frac{80k\Omega}{1.1k\Omega} = 77.727 \tag{2.3}$$

where R_{G_1} and R_{G_2} are the external resistors on the magnetometer circuit for adjusting the gain of each amplifier.

Voltages at the A and B terminals of the potentiometer are (Nominal resistance between the A-B terminals of AD5242 is 100kΩ)

$$V_A = \frac{39k\Omega + (20k\Omega||100k\Omega)}{39k\Omega + (20k\Omega||100k\Omega) + 39k\Omega} \times 3V = 1.7641 \text{ V} \tag{2.4}$$

$$V_B = \frac{39k\Omega}{39k\Omega + (20k\Omega||100k\Omega) + 39k\Omega} \times 3V = 1.2359 \text{ V}. \tag{2.5}$$

Then, V_{AB} is calcuated as

$$V_{AB} = V_A - V_B = 1.7641 - 1.2359 = 0.5282 \text{ V}. \tag{2.6}$$

V_{AB} is spanned in 256 steps. Hence, one unit of potentiometer change results in $V_{AB}/256$ = 2.0633mV voltage change at the wiper terminal. This voltage is amplified at the corresponding 2^{nd} stage amplifier. Then the voltage difference at the input of the ADC is

$$V_{ADC} = \frac{V_{AB}}{256} \times G_2 = 2.0633 \times 77.727 = 160.37 \text{ mV}. \tag{2.7}$$

Since the full-scale ADC output is $2^{10} - 1 = 1023$, spanned by a 3V supply,

$$\frac{V_{ADC} \times 1023}{3000} \text{ ADC counts}. \tag{2.8}$$

[3] During our experiments with HMC1002, we noticed that magnetic sensor was not responding to any change in the ambient magnetic field even we moved huge ferromagnetic bodies in the vicinity of the sensor. Our investigation showed that I^2C addressing was wrong in the magnetic sensor driver code in TinyOS-2.x CVS repository and we updated the code in our tinyos tree. The code in the repository is updated by May 2010.

Therefore, for each magnetometer axis, the actual magnetic sensor reading can be calculated by offsetting the ADC reading by the change in ADC units due to the potentiometer as

$$ADC_{total} = ADC + 54 \times PotentiometerBias \tag{2.9}$$

which is then converted to Gauss by the following

$$ADC_{total} = V_{diff} \times G_{total} \times \frac{ADC_{fullscale}}{V_{supply}} \tag{2.10}$$

Here, V_{supply}=3000mV, V_b=3V, $ADC_{fullscale}$=1023, and $G_{total}=G_1 \times G_2 \simeq 29 \times 78 = 2262$. Substituting the expressions for V_{diff} and ADC_{total} into (2.10), the magnetic flux density (henceforth, we will refer to the magnetic flux density in short as *magnetic field*) for either magnetometer axis may be found to be

$$B_s = \frac{(ADC + 54 \times PotentiometerBias)}{0.001 \times G_{total} \times ADC_{fullscale} * S} - \frac{V_{offset}}{V_b \times S} \tag{2.11}$$

Each magnetometer in the HMC1002 package has a different sensitivity and bridge offset voltage. Accurate magnetic field calculation in each axis requires those parameters to be known. However, it is not necessary to measure the ambient, but the differential magnetic field measurements caused by the ferromagnetic target accurately to make ferromagnetic target detection and localization. During its calculation, the constant term in (2.11) due to V_{offset} cancels out and the magnetic field caused by the target in mGauss is calculated as

$$\Delta B_s = \frac{(\Delta ADC + 54 \times \Delta PotentiometerBias)}{0.001 \times G_{total} * ADC_{fullscale} \times S} \tag{2.12}$$

Hence, the only unknown parameter is the sensitivity whose estimation is explained in Section 2.3.2.

2.2.2 Magnetic Sensor Axis Orientations

From Fig.2.3, ADC5 and ADC6 pins are connected to DieA and DieB on the HMC1002 shown in Fig.2.4. In the TinyOS-2.1.0 MTS310CB magnetic sensor driver code, ADC5 and ADC6 are interpreted as Y and X axis, respectively. Hence, DieA shows the Y axis, and DieB shows the X axis.

In the course of our experiments, it was observed that the magnetic sensor axis orientations differed across sensors. Several experiments were conducted to determine the orientation for each sensor, the results of which are shown in Fig. 2.5.

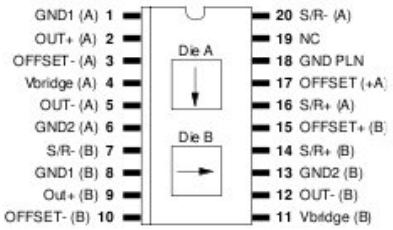


Figure 2.4: HMC1002 pinout specifications

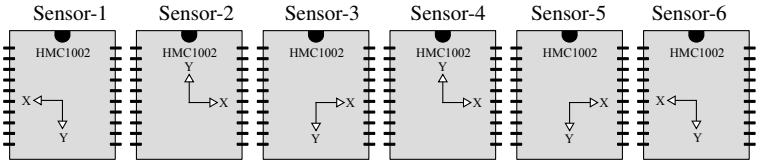


Figure 2.5: HMC1002 sensor axis orientations for different sensors

2.2.3 Magnetic Sensor Calibration

The INA2126 instrumentation amplifiers used in the magnetometer circuitry are not rail-to-rail, limited by the specifications shown in Table 2.1. This causes the signals to be clipped by the instrumentation amplifier and inhibits use of the full span of the ADC. For example, for a meaningful reading from Sensor 5, the minimum and maximum ADC counts were observed to vary between 203 and 774, with magnetometer saturation occurring beyond these limits as shown in Fig.2.6.

To obtain reliable readings, we calibrated the magnetometer by adaptively adjusting the potentiometers of the X and Y axis so that the mid-scale readings remain around 512 ADC counts, and only considered readings within the range of 250 to 750 as being reliable. For this purpose, we adapted a magnetometer calibration algorithm [20] developed in TinyOS-1.x to our application.

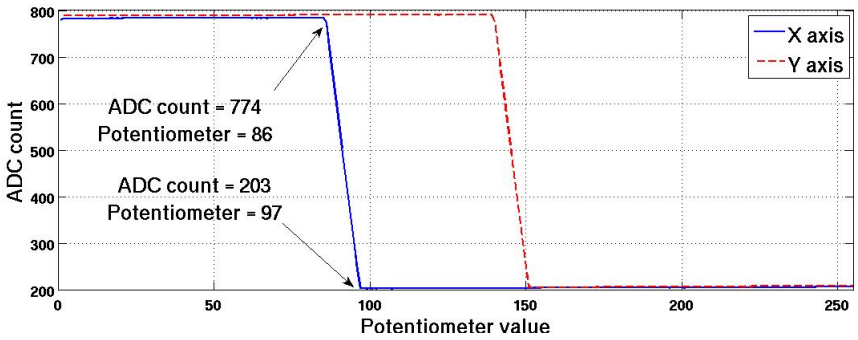


Figure 2.6: Change in ADC count in X and Y axis of Sensor 5 with the full span of potentiometer values (256 steps).

Table 2.1: INA2126 Output Specifications

Output Voltage (V)	Condition	Min	Typ	Max
Positive	$R_L = 25k\Omega$	(V_+)-0.9	(V_+)-0.75	
Negative	$R_L = 25k\Omega$	(V_-)+0.95	(V_-)+0.8	

2.3 TARGET MODELING

2.3.1 Magnetic Field Model

The iron bar (20 cm in height and 3.5 cm in diameter) used as a test target in this study is a ferromagnetic target that causes a disturbance in the Earth's magnetic field at any distance in space. This disturbance was modeled using a magnetic dipole moment model [21], which describes the magnetic field B generated by a point dipole of moment m at a distance r from the target as

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi}\frac{1}{r^3}\left[3(\mathbf{m}.\hat{\mathbf{r}})\hat{\mathbf{r}} - \mathbf{m}\right] = \frac{\mu_0}{4\pi}\left[\frac{3(\mathbf{m}.\mathbf{r})\mathbf{r}}{r^5} - \frac{\mathbf{m}}{r^3}\right] \tag{2.13}$$

where r is the L-2 norm of the vector $\mathbf{r}$, $\hat{\mathbf{r}}$ is the unit vector in the $\mathbf{r}$ direction,and μ_0 is the permeability of free space. (2.13) can be rewritten in Cartesian coordinates as

$$B_x(x,y,z) = \frac{\mu_0}{4\pi}\left[\frac{3(m_x x + m_y y + m_z z)x}{(x^2+y^2+z^2)^{5/2}} - \frac{m_x}{(x^2+y^2+z^2)^{3/2}}\right] \tag{2.14}$$

$$B_y(x,y,z) = \frac{\mu_0}{4\pi}\left[\frac{3(m_x x + m_y y + m_z z)y}{(x^2+y^2+z^2)^{5/2}} - \frac{m_y}{(x^2+y^2+z^2)^{3/2}}\right] \tag{2.15}$$

$$B_z(x,y,z) = \frac{\mu_0}{4\pi}\left[\frac{3(m_x x + m_y y + m_z z)z}{(x^2+y^2+z^2)^{5/2}} - \frac{m_z}{(x^2+y^2+z^2)^{3/2}}\right] \tag{2.16}$$

2.3.2 Model Validation

The magnetic target model can be validated by comparing magnetic sensor readings with the dipole field formulation results. However, calculation of the experimental magnetic signal strength requires estimation of sensor sensitivity. Perhaps the first approach that comes to mind for estimating the sensitivity is to take measurements under a known magnetic field, and estimate sensor sensitivity jointly with the magnetometer bridge offset voltage. However, since an absolute magnetic flux density is difficult to reliably create, we took the following alternative approach: the change in the ambient magnetic field reading by the sensor in the presence of a specific target are recorded. Sensor sensitivity is estimated jointly with the magnetic dipole parameters of that specific target.

Table 2.2 shows the experimental and analytical parameters required. The parameters shown in bold must be estimated, while the remaining are known values. For the experimental data and analytical model to match, the parameter vector [Sensitivity, m_x, m_y, m_z, z] must be estimated using either the X or Y axis data, known locations of the target relative to the sensor, and the magnetic dipole model.

Table 2.2: Parameters required for signal model validation

Experimental Inputs	*Analytical Inputs*
Sensitivity	x
Amplifier Gain	y
ADC Full Scale	$\mathbf{m_x}$
Potentiometer bias	$\mathbf{m_y}$
ADC count	$\mathbf{m_z}$
	z
	B_{dip} formula

Since the model dependence on the unknown parameters is nonlinear, a nonlinear least squares approach was used to estimate the unknowns. The surface fitting tool in MATLAB, which allows interactive surface fitting using linear or nonlinear regression, was used to automate this procedure. Nonlinear least squares iteratively converges to the optimal value given a good

initial estimate. This initial estimate was found by conducting an exhaustive grid search using the mean square error (MSE) criterion.

The surface fitting tool offers two non-linear least squares algorithms: Trust-Region, and Levenberg-Marquardt. Although the Levenberg-Marquardt is a more popular algorithm, the Trust-Region algorithm can solve complex non-linear problems more efficiently, and places constraints on the algorithm coefficients. Thus, the Trust-Region option was preferred for this study. The resulting estimate of the unknown parameter vector is shown in Table 2.3. The magnitude and orthogonal components of the magnetic dipole field produced by the test target are shown in Fig. 2.7. The plots in Fig. 2.8 show a good match between analytical and experimental range profiles for the estimates obtained.

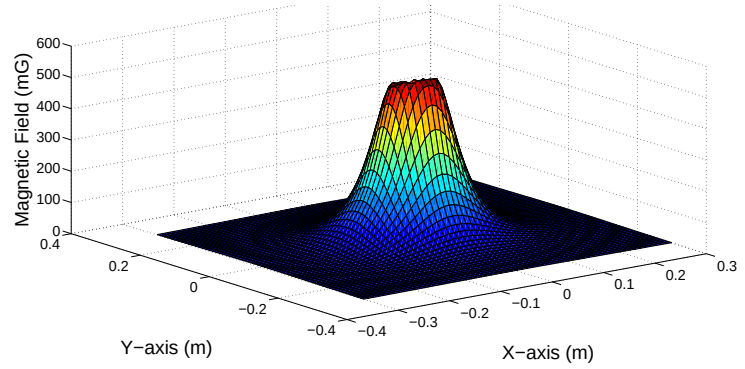
Table 2.3: Target parameter estimates

Parameter vector estimate using collected Sensor 5 magnetometer X axis data over the ranges in Fig. 2.9
$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} -0.002791 \\ -0.01634 \\ 0.1954 \end{bmatrix} Am^2$
X Axis Sensitivity: 3.243 $mV/V_{ex}/G$
z: -0.0692 m
Goodness of Fit Metric: RMSE=10.07

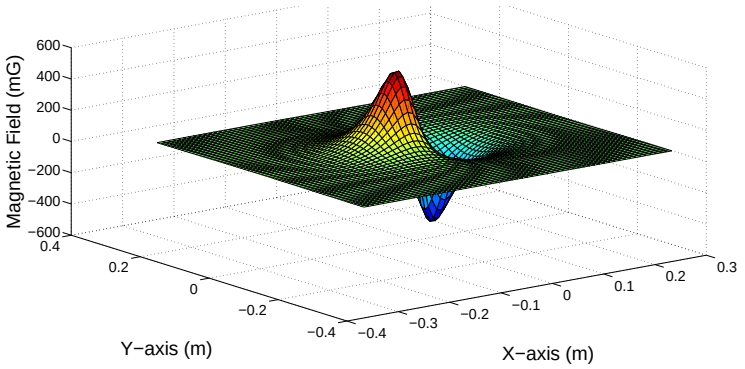
Note that the sensitivity varies across sensors; thus, to ensure correct measurements throughout the network, the sensitivity of each sensor must be individually estimated. Leaving sensitivity as the only unknown, the problem is reduced to a linear least squares parameter estimation problem, whose results are shown in Table 2.4.

Table 2.4: Estimated sensor sensitivities ($mV/V_{ex}/G$)

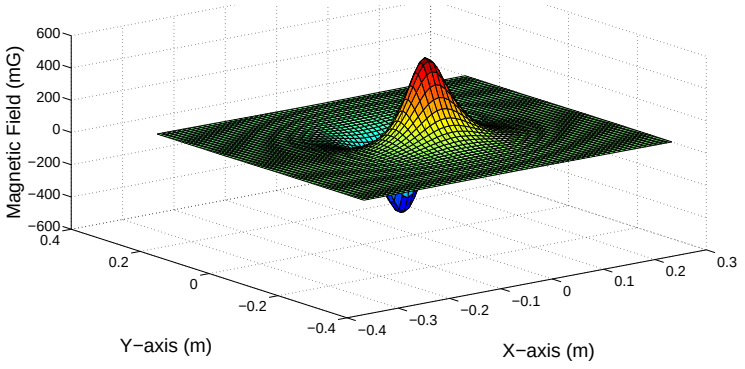
Sensor No	1	2	3	4	5	6
X	3.6564	3.2893	3.3912	3.243	2.9396	3.5688
Y	2.9226	2.7956	2.8927	3.388	2.6116	2.7236



(a)

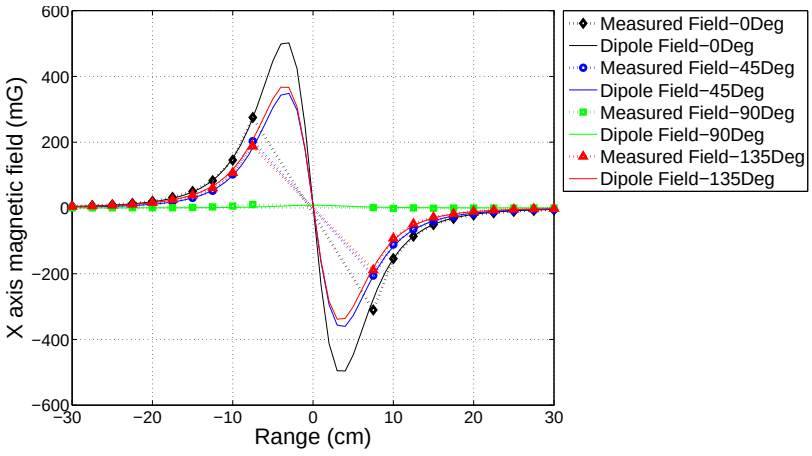


(b)

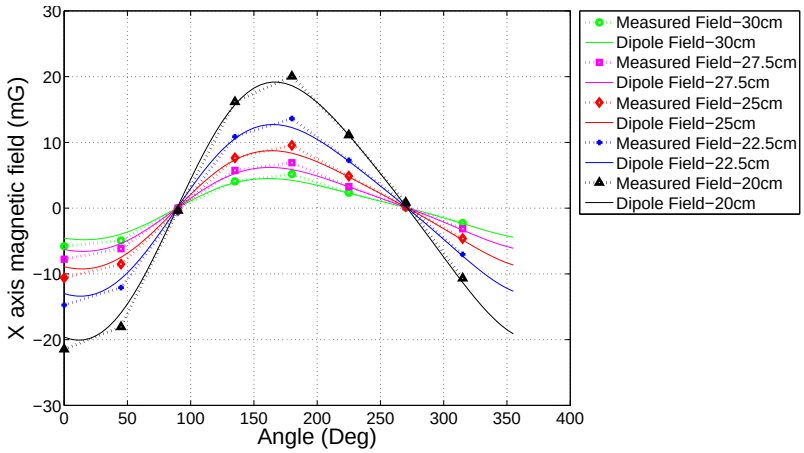


(c)

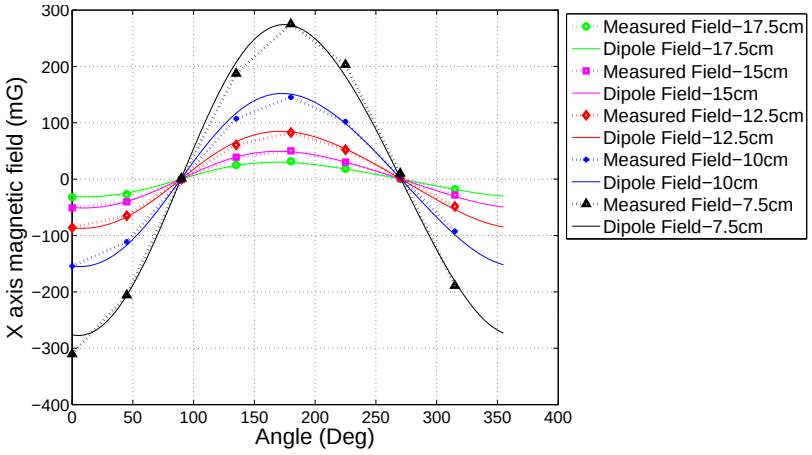
Figure 2.7: Iron bar target magnetic dipole field. (a) magnitude, (b) x component, and (c) y component



(a)



(b)



(c)

Figure 2.8: Signal model validation for sensor 5, which compares the calculated magnetic field in the x-axis with measurements. (a) X axis vs angles at 0, 45, 90, and 135 degrees, (b) X axis vs. ranges at 30, 27.5, 25, 22.5, and 20cm, and (c) X axis vs ranges at 17.5, 15, 12.5, 10, and 7.5cm

2.4 TARGET DETECTION

2.4.1 Magnetic Sensor Coverage

The magnetic sensor coverage is highly correlated with the target characteristics. To assess the variation of signal strength with range and angle for the iron bar test target, the test setup shown in Fig. 2.9 was constructed. The sensor was placed at each grid point surrounding the test target located at a fixed position. In this way, the spatial magnetic field variation was measured and used to study target characteristics. Results showed that for the test target in question, the field disturbance is almost undetectable beyond a distance of 30 centimeters. Thus, the coverage for each sensor was determined to be a circle of radius 30 cm, and sensor placement for the target localization phase of this study was designed accordingly.

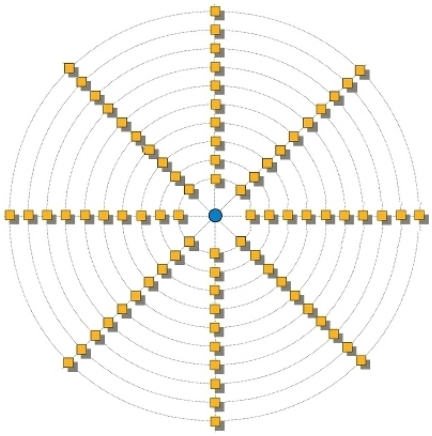


Figure 2.9: Experimental configuration for taking measurements and studying sensor coverage. Blue circle at the center and the surrounding yellow squares represent the target, and sensors, respectively. We took measurements at distances 7.5, 10, 12.5, 15, 17.5, 20, 22.5, 25, 27.5, 30 cm for each angular position from the target position.

2.4.2 Detection Algorithm

As the target moves within the 30 cm sensing radius, the magnetic field reading also increases. If, however, the sensor is exposed to too large a magnetic field, the HMC1002 may saturate, triggering a calibration requirement. Depending on sensor sensitivity, when the target moves at a range of about 5-15 cm, the magnetometer saturates and calibration is required. We emphasize that calibration does not depend on sensor network topology, but on the relative

distance between the target and sensor. Hence, to prevent long term inhibition of detection and tracking due to calibration intervals, and to sustain continual operation, sensors are sparsely deployed and the target is allowed to move in such a way that sensors read feasible values most of the time. An important issue with the MTS310 board is that there is no automatic saturation recovery circuit on the board. Hence, sensor calibration is performed in software adaptively during run-time.

After calibration, the MICAz motes collect magnetic sensor readings both along the X and Y axis. Conversion of the ADC count reading to mGauss is performed according to (2.12), then the magnetic energy is calculated as follows

$$MagneticEnergy^{(i)} = (r_x^{(i)} - A_x)^2 + (r_y^{(i)} - A_y)^2 \tag{2.17}$$

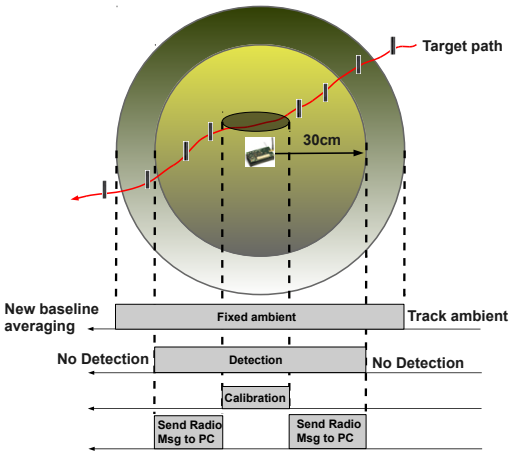
where for the i^{th} sample, r_x^i is the observed magnetic field in the X-axis, r_y^i is the observed magnetic field in Y-axis, A_x is the ambient magnetic field along the X-axis, and A_y is the ambient magnetic filed along the Y-axis.

The ambient magnetic field value must be stored to cancel the offset created by environmental, as opposed to target-related, factors. Under normal conditions, the ambient magnetic field of the Earth measured at a certain location varies throughout the day and is affected by other external variables such as temperature. Thus, ambient magnetic readings are noisy and variable even over the course of data collection for a single point. In this work, the ambient is tracked by calculating a moving average baseline when there is no target. Although the baseline cannot be tracked while performing target detection, it is updated immediately after the target moves outside the sensing radius, after which the system continues to track the ambient field.

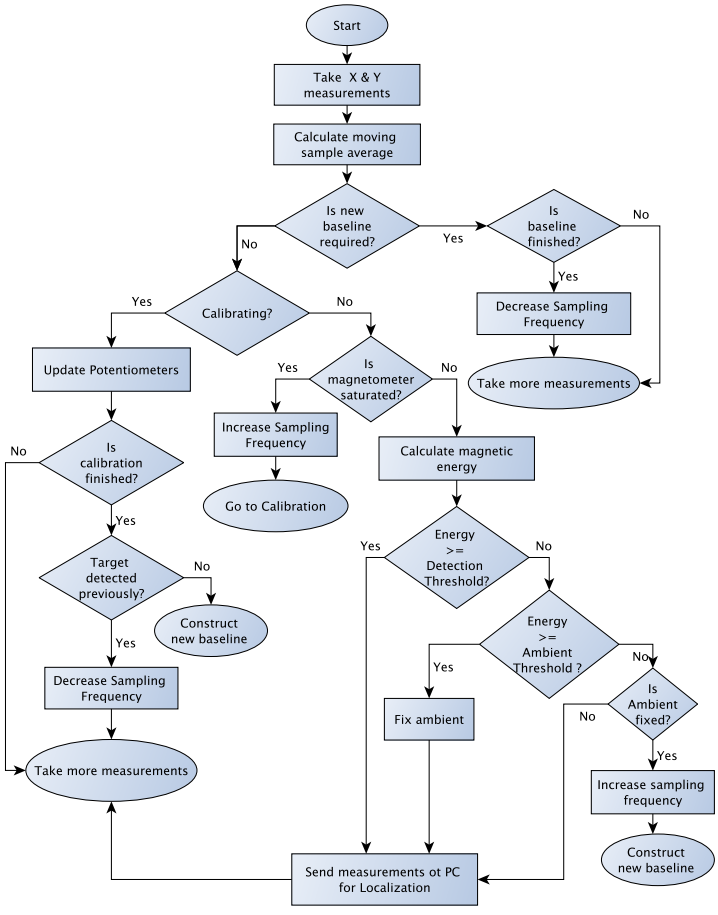
When the change in measured magnetic field is greater than the detection threshold, the ambient utilized for subsequent detection decisions is fixed at the most recent moving average value computed that excludes measurements containing any rapid changes due to the target. During our experiments, we utilized a 1 mGauss energy threshold for updating the ambient, and a 17 mGauss energy threshold for target detection.

In summary, target detection is performed in two steps, as shown in Fig. 2.10(a). The first step is detecting a change in the ambient magnetic field that requires updating the stored ambient value and the second one is detecting the entry of the test target into the sensing radius. Fig.

2.10(b) shows the flowchart illustrating the main steps within the detection scheme.



(a)



(b)

Figure 2.10: Illustration of target detection and algorithm steps taken by MICAz. (a) Target detection schematic, (b) Target detection algorithm flowchart.

2.5 CENTRALIZED TARGET LOCALIZATION

Magnetic sensors can be used not just to detect the presence of ferrous targets, but also to extract locational information. The HMC1002 is a two-axis magnetometer that takes vector measurements. Consequentially, we could in principle localize the target using just a single sensor. This potential is demonstrated for the test target using a localization algorithm that uses a maximum-likelihood formulation to find unknown position parameters.

2.5.1 Maximum-Likelihood Estimation

The magnetic field measurements have been found to be well modeled by the Gaussian distribution (using the distribution fit tool of MATLAB), albeit with different variances for different sensors and locations. We can thus construct a probabilistic model for sensor observations, assuming that the measurement of each of the N sensors is conditionally independent of the others, given the disturbance. More specifically,

$$r_x^{(i)} = B_x(x^{(i)}, y^{(i)}, z) + \eta_x^{(i)}, r_y^{(i)} = B_y(x^{(i)}, y^{(i)}, z) + \eta_y^{(i)}$$

where $i = 1, 2...N$, x, y, and $z = -0.0692m$ are relative distances between the target and each sensor in the respective directions, $n_x^{(i)}$ and $n_y^{(i)}$ are the noise in X and Y axes that will be modeled as zero mean Gaussian with variance $\sigma_x^{(i)}$ and $\sigma_y^{(i)}$, respectively. All noise terms are assumed independent.

Since sensor locations are known, the distance between each sensor and the target can be expressed relative to the distance between sensor 1 and the target. Then,

$$x^{(i)} = x^{(1)} + \triangle x^{(i)}, y^{(i)} = y^{(1)} + \triangle y^{(i)}$$

$\triangle x^{(i)}$ and $\triangle y^{(i)}$ are the relative distances of sensor i ($i = 2, 3, ..N$) to sensor 1 in the x and y coordinates. Therefore, the target localization problem turns into the estimation of $x^{(1)}$ and $y^{(1)}$. The likelihood function can be written as

$$p(r_x^1, r_y^1, r_x^2, r_y^2, ..., r_x^N, r_y^N | x^{(1)}, y^{(1)}) = \prod_{i=1}^{N} \left\{ \frac{1}{2\pi\sigma_x^{(i)}\sigma_y^{(i)}} \exp\left\{ -\frac{(r_x^{(i)} - B_x^{(i)})^2}{2\sigma_x^{(i)^2}} - \frac{(r_y^{(i)} - B_y^{(i)})^2}{2\sigma_y^{(i)^2}} \right\} \right\}$$

After maximizing the log-likelihood functions with respect to $x^{(1)}$ & $y^{(1)}$, we have the follow-

ing

$$\sum_{i=1}^{N}(\frac{\partial B_x^{(i)}}{\partial x^{(1)}}\frac{1}{\sigma_x^{(i)^2}})(r_x^{(i)}-B_x^{(i)})+\sum_{i=1}^{N}(\frac{\partial B_y^{(i)}}{\partial x^{(1)}}\frac{1}{\sigma_y^{(i)^2}})(r_y^{(i)}-B_y^{(i)})=0$$

$$\sum_{i=1}^{N}(\frac{\partial B_x^{(i)}}{\partial y^{(1)}}\frac{1}{\sigma_x^{(i)^2}})(r_x^{(i)}-B_x^{(i)})+\sum_{i=1}^{N}(\frac{\partial B_y^{(i)}}{\partial y^{(1)}}\frac{1}{\sigma_y^{(i)^2}})(r_y^{(i)}-B_y^{(i)})=0$$

The ML estimates $\hat{x}^{(1)}$ and $\hat{y}^{(1)}$ are values that satisfy each of the following $2N$ equations:

$$r_x^{(i)}-B_x^{(i)}=0, r_y^{(i)}-B_y^{(i)}=0$$

As expected, this system is over-determined and these equations will be inconsistent. A least-squares solution [22] can be found, however, as the equations are non-linear, there is no simple closed-form expression for the least-squares solution. If the problem was linear, the least-squares solution would have the noise covariance matrix as a weighting factor, but from the above it can be seen that the likelihood ratios lose the information about the data points having different noise variance. Hence, instead of pursuing the ML approach, we proceed to directly address this problem as a least squares problem.

2.5.2 Least-Squares Estimation

For least-squares estimation, rather than constructing a probabilistic model for the data, we shall make use of the magnetic signal model. More specifically, our non-linear least-squares estimator will choose the parameter vector $\mathbf{P} = [x, y]$ such that the difference between the magnetic signal model data and measurements is minimized. Closeness will be measured by the following mean-squared error objective function:

$$J(\mathbf{P})=\frac{1}{N}\sum_{i=1}^{N}\left\{(r_x^{(i)}-B_x^{(i)}(\mathbf{P}))^2+(r_y^{(i)}-B_y^{(i)}(\mathbf{P})))^2\right\} \tag{2.18}$$

$$\mathbf{P_{est}}=\underset{P\in TargetQuadrant}{\operatorname{argmin}}\{J(\mathbf{P})\} \tag{2.19}$$

The parameter vector $\mathbf{P_{est}}$ that minimizes the objective function J($\mathbf{P}$) subject to being in the same quadrant as target, gives us the least-squares estimate.

Simulated annealing [23] is one of several possible least-squares methods that can be used to solve global optimization problems which arise in target localization. It is a meta-heuristic algorithm inspired by physical annealing in metallurgy, which is a thermal process for obtaining

low energy states of a solid material. For the algorithm to respond as fast as possible, the parameter vector should possess a limited search space. In this work, we selected a localization resolution of 1cm, resulting in 30x30=900 possible target locations. Algorithm parameters were selected empirically. The starting temperature was set to 100 and cooling rate to 0.99. Algorithm terminated after reducing the temperature for 300 successive iterations or when the magnetic energy change from one iteration to the next was below 0.5. At each iteration, 10 neighbouring states were examined and the objective function in (2.18) was evaluated. It is observed that the algorithm terminated after a few tens of milliseconds.

Localization performance is obtained via mean localization error E_{loc}

$$E_{loc} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(X_{est}^{(i)} - X_{actual})^2 + (Y_{est}^{(i)} - Y_{actual})^2} \tag{2.20}$$

We calculated E_{loc} using four sensors to localize the target at the ranges 10, 15, 20, 25, and 30 cm and at angles 0, 45, 90, 135, 180, 225, 270, and 315 degrees. As shown in Fig. 2.11, the mean localization error decreases with decreasing range, which can be considered as increasing SNR.

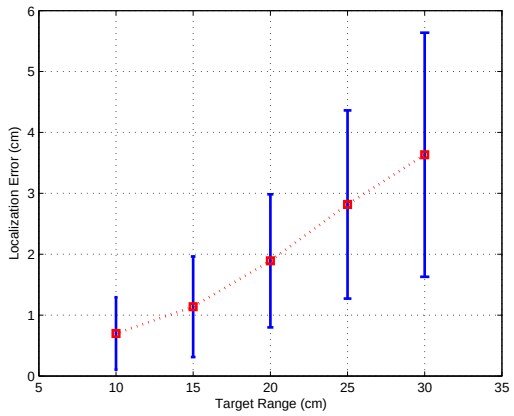


Figure 2.11: Mean localization error. The bars represent half the uniform distribution standard deviation

2.6 SEQUENTIAL LOCALIZATION

To test the target model, localization algorithm, sensor sampling frequency and network functionality when the target is moving, a test setup comprised of four sensor nodes and a gateway

node was connected to the PC as shown in Fig. 2.12. Then, the localization was applied sequentially at fixed time intervals, while the test target was slowly moved through the network coverage region. To track the motion as closely as possible, the sensor calibration and baseline averaging required must be performed as quickly as possible. Towards this aim, the sampling rate was set at 128 Hz.

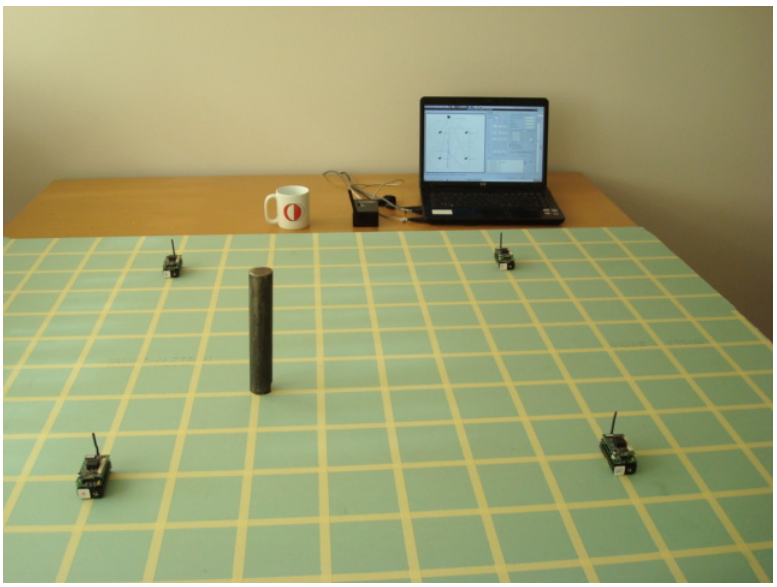


Figure 2.12: Wireless sensor network test setup

Moreover, we assume that the target moves smoothly within the coverage region, and that the target may start and stop, but does not spend long amounts of time in a single location. For a target velocity of about 5 cm/s, we set the sampling rate to 4 Hz during the normal data acquisition period. In Fig. 2.13, a sample run of sequential localization of a moving target is shown.

Thus, for slow-moving targets with a speeds less than 5 cm/s, the magnetic sensor network is able to successfully localize mobile targets with a mean error of 3.6 cm for targets at the detection limit.

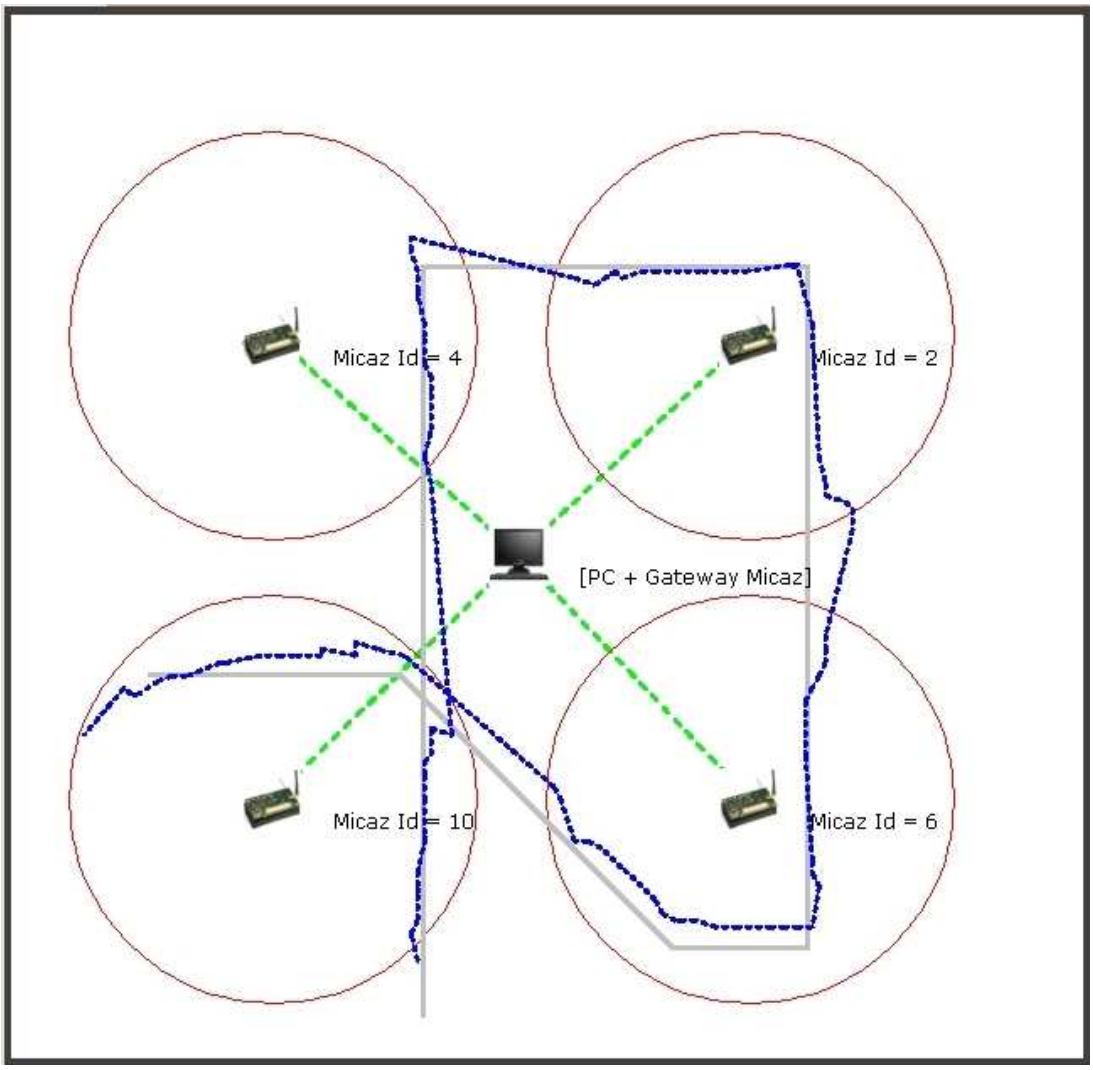


Figure 2.13: Target tracking with sparsely deployed wireless sensor network via sequential localization. Sensors are sparsely deployed as each one can localize target individually. Purple circles represent 30cm magnetic coverage, solid light gray lines are the actual target path whereas dashed blue lines are the estimated target path. See Fig. 2.11 for the mean and std. in localization error which varies with the relative distance between the target and sensor.

2.7 CONCLUSIONS

In this chapter, we presented a comprehensive approach for the modeling of ferromagnetic targets, and summarized important issues related to the practical implementation of magnetic WSNs. A small iron bar is used as a test target to examine the modeling of the target's magnetic signature. Performance of the WSN was examined in the context of the detection and localization of ferromagnetic targets. Despite the ease in saturation of magnetic sensors and hence preclusion of taking feasible magnetic readings, and sensor's being highly susceptible to ambient magnetic noise, we proposed an efficient ferromagnetic target detection scheme. Centralized localization is performed centrally via the Simulated Annealing algorithm, and performance results show approximately 10% localization error with respect to the real distance between target and sensor.

CHAPTER 3

PACKET SCHEDULING ON AN ENERGY HARVESTING BROADCAST LINK

3.1 INTRODUCTION

Since its formulation a decade ago [25], the problem of energy-efficient packet transmission scheduling has drawn considerable interest from the research community [26, 27, 28]. The basic offline problem is to assign transmission durations (equivalently, code rates) to a set of packets whose arrival times are known beforehand, so that they are all transmitted within a given time window with minimum total energy. Recently, the problem has been recast with a formulation where the goal is to minimize the *time* by which all packets are transmitted, given that energy is harvested at certain known instants [30].

In this chapter, we extend the formulation in [30] to a multiuser scenario with one sender and multiple receivers. In particular, we assume an AWGN Broadcast Channel where the sender gets replenished with arbitrary amounts of energy at arbitrary points in time. The harvested energy becomes instantly available for use, and the transmission power can be changed at any time by the sender. The choices of power level and the rates to individual receivers across time is called a *schedule*. The sender needs to transmit a certain number of bits to each receiver. We consider the case that these bits are available at the beginning of transmission. The goal is to find a schedule that minimizes the time by which the data of all users has been transmitted. Throughout the chapter, we focus on the *offline* problem, where the energy harvesting times as well as packet arrival times are known in advance. The online version of the problem in which the times of energy harvests are not known a priori and decisions need to be made in real-time as the harvests occur, is interesting yet analytically less tractable and left outside the

scope of this chapter.

It is well known that both with optimal and practical coding schemes, the energy per bit increases with the transmission rate, in other words, transmitting fast is inefficient in terms of energy [31]. This is the root of the sender's dilemma: it will pay off for the sender to slow down, yet it needs to minimize the overall transmission duration. Interestingly, it turns out that even if all packets were available in the beginning, the optimal schedule starts slowly, deferring some of the harvested energy for future use. More precisely, we will show that in the optimal schedule the transmission power is non-decreasing in time, similarly to the point-to-point schedule [30].

In the point-to-point problem, determining power levels determines the schedule, as transmission rate is a function of average power. In the broadcast problem, however, there is no one-to-one correspondence between the transmission power and the rate point. For example, with optimal coding, there is a continuum of rates on the boundary of the capacity region corresponding to a certain average power constraint. Hence, the rates and the power have to be determined together. We observe that in the optimal schedule, the average rates used by the users are proportional to their numbers of bits, i.e. the schedule always continuously transmits to all users at the same time and finishes transmission to all users at the same time. Having made this observation, we can exploit the mathematical similarities between this problem and the problem in [29], and show that the solution is found by the algorithm FlowRight, defined in [29] and adapted here to work with different parameters.

In the next section, we make observations about the two-user AWGN broadcast channel capacity region. The statement of the problem as a cost minimization problem, as well as its solution will use certain structural properties of the AWGN capacity region, such as the monotonicity and convexity of the average power with respect to the rate pair. Of course, this specific rate region can only be approached under optimal coding as blocklengths and the number of information bits go to infinity. For example, in the single user AWGN channel the numerical value of the minimum energy per bit corresponding to a given reliability monotonically decreases with the number of information bits [32]. However, the basic structural properties of the rate region will be satisfied by the achievable rate regions of many suboptimal practical coding schemes as well as finite blocklength optimal coding schemes.

We define the problem in Section 3.3 and explore the properties of the optimal solution. This is

followed in Section 3.4 by the description of the modified FlowRight algorithm, and the proof of its convergence and optimality of the resulting schedule. The complexity of the iterative algorithm is analyzed in Section 3.5. The implementation of this algorithm is discussed, followed by a numerical example in Section 3.6. Section 3.7 summarizes our conclusions.

3.2 BROADCAST CHANNEL

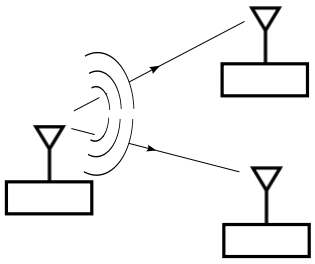


Figure 3.1: Two-user broadcast channel

Consider a discrete-time AWGN broadcast channel with one sender and two receivers as shown in Fig. 3.1. The signal received by the i^{th} user at time k is given by

$$Y_i[k] = \sqrt{s_i}X[k] + Z_i[k], \tag{3.1}$$

where $X[k]$ is the transmitted signal with average power constraint P, $\sqrt{s_i}$'s are the channel gains and the $Z_i[k]$'s are i.i.d. zero-mean Gaussian noise with variance σ^2. The capacity region of the channel assuming $s_1 > s_2 > 0$ are real constants, is the set of rate pairs (r_1, r_2) such that [34]

$$r_1 \leq \frac{1}{2}\log_2\left(1 + \frac{\alpha s_1 P}{\sigma^2}\right) \tag{3.2}$$

$$r_2 \leq \frac{1}{2}\log_2\left(1 + \frac{(1-\alpha)s_2 P}{\alpha s_2 P + \sigma^2}\right) \tag{3.3}$$

for some $0 \leq \alpha \leq 1$.

It is straightforward to show that, given s_1 and s_2, for any $P_1 > P_2$, the capacity region corresponding to an average power constraint P_1 dominates the one corresponding to P_2. Therefore, given a rate pair (r_1, r_2), there is a unique $P = g(r_1, r_2)$ (see [29]) such that (r_1, r_2) lies on the boundary of the rate region with average power constraint equal to P. After replacing the

inequalities in (3.2) and (3.3) by equalities, the function $g(r_1, r_2)$ is written as follows. (3.2) can be written as as $\alpha s_1 P/\sigma^2 = 2^{2r_1} - 1$. Hence, $\alpha = (\sigma^2/Ps_1)(2^{2r_1} - 1)$. After substituting into (3.2) and rearranging the terms, we obtain

$$\begin{aligned} g(r_1, r_2) &= \sigma^2\left(\frac{(2^{2r_1}-1)}{s_1} + \frac{(2^{2r_2}-1)}{s_2} + \frac{(2^{2r_1}-1)(2^{2r_2}-1)}{s_1}\right) \\ &= \sigma^2\left(\frac{(2^{2r_2}-1)}{s_2} + \frac{(2^{2r_1}-1)2^{2r_2}}{s_1}\right). \end{aligned} \tag{3.4}$$

The function $g(r_1, r_2)$ is twice continuously differentiable and strictly convex in r_1 and r_2 (proven in Appendix A). Throughout the chapter, it will be useful to express the r_1 and r_2 as a function of each other and the minimum average power P. By algebraic manipulation of (3.2) and (3.3), we obtain the following:

$$r_1 = h_1(P, r_2) = \frac{1}{2}\log_2\left(\frac{s_1(s_2P + \sigma^2)}{s_2\sigma^2 2^{2r_2}} - \frac{s_1 - s_2}{s_2}\right) \tag{3.5}$$

$$r_2 = h_2(P, r_1) = \frac{1}{2}\log_2\left(\frac{\frac{s_2P}{\sigma^2} + 1}{\frac{s_2}{s_1}(2^{2r_1} - 1) + 1}\right). \tag{3.6}$$

The properties satisfied by these rate functions for the AWGN BC capacity region with $s_1 > s_2$ summarized in the following proposition will be used in the rest of the chapter.

Proposition 1 *The functions h_1 and h_2, defined in* (3.5),(3.6) *on $\mathfrak{R}^+ \times \mathfrak{R}^+$ satisfy the following properties:*

1. *Nonnegativity: $h_1(P, r) \geq 0$, $h_2(P, r) \geq 0$.*

2. *Monotonicity: $h_1(P, r)$, $h_2(P, r)$ are both monotone decreasing in r, and monotone increasing in P.*

3. *Concavity: $h_1(P, r)$ and $h_2(P, r)$ are concave in P and r.*

4. *The rate of the user with the weaker channel satisfies the following: $\frac{\partial^2 h_2(P,r)}{\partial r \partial P} = 0$, $\frac{\partial^2 h_2(P,r)}{\partial P \partial r} = 0$.*

Proof. See Appendix B

3.3 PROBLEM DEFINITION

Consider the broadcast link as described in the previous section, with a sender who needs to transmit $B_1 < \infty$ and $B_2 < \infty$ bits with a certain degree of reliability to users 1 and 2, respectively [1]. Also assume that at time t_1=0, sender has $E_1 > 0$ units of energy available and at times $t_2, ..., t_{k+1}$, energies are harvested with amounts $E_2, ..., E_{K+1}$, respectively, as depicted in Fig 3.2. Inter-arrival times of energy harvests are named as *epochs*, and marked with $\xi_i, i = 1, ..., k$.

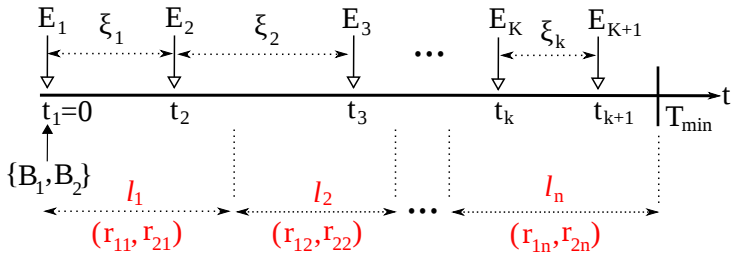


Figure 3.2: System model with all of the bits of the two users are available at the beginning. Energies arrive at times t_i where $i \in \{t_1, t_2, ..., t_{k+1}\}$

It will be assumed that the sender has the ability to change its rate pair at any time, according to the available energy and remaining number of bits. Such ideal adaptation, which has been used in previous literature (e.g., [30], and references therein), may be approximated by using adaptive coding and modulation in a practical system.

Starting at $t = 0$, let $\{(r_{11}, r_{21}), (r_{12}, r_{22}), ..., (r_{1n}, r_{2n}), \ldots\}$, be the successive pairs of rates used by the sender, and $\{l_1, l_2, ..., l_n, \ldots\}$ be the respective durations for which these pairs are used. Here, $r_{ij} \geq 0$ is user i's rate in the j^{th} rate pair. By definition, at least one user's rate changes from one rate pair to the next one. We will refer to the sequence of rate pairs and durations as a *schedule*. The problem of interest is to find an optimal *offline* schedule, that is, a schedule that minimizes the overall transmission completion time of the B_1 and B_2 bits to their respective destinations, with complete knowledge of future energy harvesting instants and the amounts to be harvested.

It will also be assumed that the problem is feasible; that is, sufficient energy will be harvested to transmit the given $B_1 < \infty$ and $B_2 < \infty$ bits in arbitrarily large but finite total time, T. Note

[1] Throughout the chapter, two receivers will be considered for ease of exposition. However, the results can be generalized to more than two receivers.

that for any given E, there is a small enough rate (equivalently, long enough transmission duration) such that B_1 and B_2 bits can be transmitted with energy E, provided that the minimum energy per bit required for communication on the broadcast channel for the given finite amount of bits is satisfied [33]. In the point-to-point case with infinite blocklengths, the well known limit for energy per bit is -1.59 dB. For sending finite amounts of data, the minimum energy per bit is higher even at nonvanishing values of error probability. However, the upper and lower bounds in [32] on energy per bit come very close to the ideal limit at $B = 10^3$ bits, and even at smaller numbers of bits.

In order to define the two-user broadcast channel offline scheduling problem as an optimization problem, we will use the set of observations stated in Lemmas 3.3.1-3.3.5. Lemma 3.3.5 will establish that in an optimal schedule the transmission to both users ends at the same time. Lemma 3.3.2 will establish that in an optimal schedule the rates and power level do not change between energy harvests that are used. As a consequence of these two results, the problem reduces to Problem 1.

We start by proving a more general result than Lemma 3.3.2 which will be used in the proof of Lemma 3.3.2 as well as Theorem 1 in Section 3.3. Specifically, we take a finite time window which is divided into two slots such that different power levels are used in each. We show that by using a more even distribution of power (reducing the difference of the power levels) as much as energy causality permits, at least the same number of bits can be transmitted in the same amount of time using the same amount of energy. In the special case when this time window is within (or all of) one epoch, all the energy that is used is available in the beginning hence the powers can be completely equalized.

Lemma 3.3.1 *Suppose that within a time window (τ_1, τ_2), the sender changes its transmit power at point τ^* such that $\tau_1 < \tau^* < \tau_2$. Keeping the total consumed energy in (τ_1, τ_2) constant, the sender can send at least the same number of bits to the users within the same duration by bringing power levels closer to each other, if feasible (i.e., unless such a change requires energy to be used before its harvested.)*

Proof. Let the total duration be $t = \tau_2 - \tau_1$, and the lengths of the two slots βt and $(1 - \beta)t$, with power levels in the two slots P_1 and P_2, as illustrated in Fig.3.3. Denote the rate pairs in

the 1^{st} and 2^{nd} slots as (r_{11}, r_{21}) and (r_{12}, r_{22}), respectively.

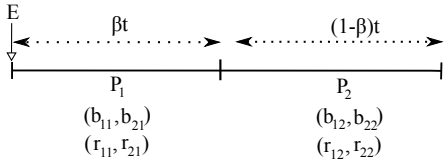


Figure 3.3: Illustration of the transmission scheme used in Lemma 3.3.1.

First, consider the case where the power level used in the first slot is smaller: $P_1 < P_2$. When $P_1\beta t$ is equal to the total energy available for use in (t_1, τ^*), transferring energy from the second slot to the first is not feasible, and we stop. However, if it is possible to transfer some positive amount of energy ΔE from the second slot to the first, we shall show that we can only improve the allocation.

Let us denote the average rates for the 1^{st} and 2^{nd} users as $\bar{r_1} \triangleq \beta r_{11} + (1-\beta) r_{12}$ and $\bar{r_2} \triangleq \beta r_{21} + (1-\beta) r_{22}$, respectively. We will show that keeping the total consumed energy and $\bar{r_1}$ constant, the sender can achieve an average rate $\bar{\bar{r_2}}$ for the second user such that $\bar{\bar{r_2}} \geq \bar{r_2}$ by changing P_1 to $P_1^{'}$ and P_2 to $P_2^{'}$ satisfying

$$P_1 \leq P_1^{'} \leq P_2^{'} \leq P_2. \tag{3.7}$$

While keeping the total energy constant, a certain amount of energy should be transferred from the 2^{nd} slot to the 1^{st} one in order to satisfy (3.7). In this case, we have the following

$$P_1^{'} = P_1 + (1-\beta)\Delta P \ , \ P_2^{'} = P_2 - \beta \Delta P. \tag{3.8}$$

Average rate belonging to the 2^{nd} user over the whole duration t is given by

$$\begin{aligned} \bar{\bar{r_2}} &= h_2(P_1^{'}, \bar{r_1})\beta + h_2(P_2^{'}, \bar{r_1})(1-\beta) \\ &\geq h_2(P_1, r_{11})\beta + h_2(P_2, r_{12})(1-\beta) \\ &= \bar{r_2} \end{aligned} \tag{3.9}$$

(3.9) follows from the fact that

$$h_2(P_1^{'}, \bar{r_1})\beta + h_2(P_2^{'}, \bar{r_1})(1-\beta) - h_2(P_1, r_{11})\beta - h_2(P_2, r_{12})(1-\beta) \geq 0 \tag{3.10}$$

for all $\beta = \{0, 1\}$ with equality achieved at $\beta = 0, 1$. This is a consequence of the properties listed in Section 3.2, and proved in Appendix C.

In the remaining case, $P_1 > P_2$, a similar argument holds where $P_2 \leq P_2^{'} \leq P_1^{'} \leq P_1$. Note that in this case it is always possible to improve the allocation as it is always feasible to transfer a positive amount of energy from the first slot to the second- energy can be deferred for future use.

We conclude that keeping the total consumed energy constant, one can find rate pairs such that at least the same number of bits can be transmitted to the users within the same duration by reallocating power levels closer to each other. ■

Corollary 1 *In an optimal schedule, transmission power remains constant between two energy harvests that are used.*

Proof. The claim is that the power does not change within epochs, with the exception of the last epoch. (In the last epoch that is used, the transmission ends and the power is reduced to zero at some point within the epoch.) As, by definition, no new energy or data is added during an epoch, it is quite intuitive that the decision on power allocation does not change at a point during an epoch. To reach contradiction, suppose that the sender changes its power allocation during an epoch. From Lemma 3.3.1, the power levels can be allocated closer to each other so that at least the same number of bits can be transmitted to the users. Since this case is not limited by causality, this procedure can be continued until the power levels within the epoch are equalized, strictly improving the schedule, contradicting the optimality of the original schedule. ■

Lemma 3.3.2 *In an optimal schedule, the rate pair remains constant between energy harvests that are used.*

Proof. From Corollary 1, we know that power level stays constant during epochs in an optimal schedule. Now, suppose the sender changes its rate pair at some point during an epoch, while the power is constant at P. Let the lengths of the two slots as βt and $(1-\beta)t$ and the rate pairs in the 1^{st} and 2^{nd} slots as (r_{11}, r_{21}) and (r_{12}, r_{22}). Due to the concavity of $h_2(P, r)$ in r, setting r_1 to the average rate only improves r_2

$$h_2(P, \beta r_{11} + (1-\beta) r_{12}) \geq \beta h_2(P, r_{11}) + (1-\beta) h_2(P, r_{12}).$$

Hence, by equating the rate pair, at least the same number of bits can be transmitted at the same time. ■

The next result is an observation of the structure of the basic solution when there is *only one energy harvest* (the one at $t = 0$).

Lemma 3.3.3 *Suppose that E units of energy is available at the beginning for the sender to transmit B_1 and B_2 bits to users 1 and 2, respectively. To minimize the overall transmission duration, the sender finishes transmission to both users at the same time.*

Proof. To reach contradiction, suppose that in an optimal solution, the sender finishes transmission to one of the users before the other. This means that the rate pair changes at some point (when the transmission of one of the users ends before the other), although no new energy has been harvested. By Lemma 3.3.2, averaging the power levels and rates and using one rate pair continuously would enable us to send at least the same number of bits during the same time. This contradicts the optimality of the original solution. ■

Lemma 3.3.3 tells us that the ratio of the rates r_1/r_2 is equal to the ratio of the bits B_1/B_2, hence for the AWGN case from (3.2) and (3.3) the ratio of powers, α, can be found by setting:

$$r_1 = \frac{1}{2}\log_2(1 + \frac{s_1 p_1}{\sigma^2})$$
$$r_2 = \frac{1}{2}\log_2(1 + \frac{s_2 p_2}{s_2 p_1 + \sigma^2})$$

where $p_1 = P\alpha(B_1, B_2)$, $p_2 = P(1 - \alpha(B_1, B_2))$. Using $\frac{r_1}{r_2} = \frac{B_1}{B_2}$, one can obtain

$$\frac{B_1}{B_2} = \frac{\frac{1}{2}\log_2(1 + \frac{\alpha s_1 P}{\sigma^2})}{\frac{1}{2}\log_2(1 + \frac{(1-\alpha)s_2 P}{\alpha s_2 P + \sigma^2})}$$

Solving

$$\left(1 + \frac{\alpha s_1 P}{\sigma^2}\right)^{B_2} = \left(1 + \frac{(1-\alpha)s_2 P}{\alpha s_2 P + \sigma^2}\right)^{B_1}$$

for α and substituting into (3.2) and (3.3) yields a rate pair (r_1, r_2), for a given value of P.

Before discussing how to find the right value of P, it will be illustrative to present an alternative proof for Lemma 3.3.3. Suppose the sender has an average power level P to use. The question is to obtain the minimum termination time for all the bits, $T_{\min}$, given by the following:

$$T_{\min} = \min(\max(\frac{B_1}{r_1}, \frac{B_2}{r_2})) = f(r_1, r_2)$$
$$= \begin{cases} \frac{B_1}{r_1} & if\ \frac{r_1}{r_2} < \frac{B_1}{B_2} \\ \frac{B_2}{r_2} & if\ \frac{r_1}{r_2} > \frac{B_1}{B_2} \end{cases}$$

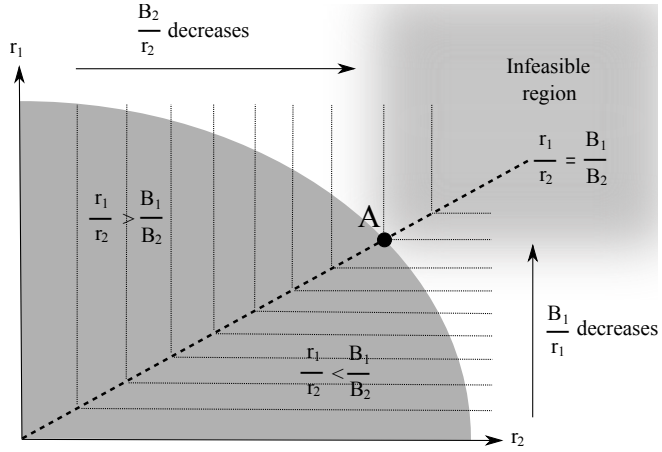


Figure 3.4: Illustration of Lemma 3.3.3: The rate pair (r_1, r_2) that minimizes overall transmission time for the sender to transmit B_1 and B_2 bits to each user is at point A.

The contours of constant $f(r_1, r_2)$ are shown in Figure 3.4. The value of f gets smaller as we move outward from the rate region. The last contour that still touches the region, touches it at the point labelled A. Therefore, $T_{\min}$ is obtained by the rate pair at point A on the boundary of the rate region, which satisfies $\frac{r_1}{r_2} = \frac{B_1}{B_2}$.

Given E, B_1, and B_2, determining the right value for P entails solving a nonlinear equation which can be done iteratively: start with a guess for the average power P, and find the point A on the rate region defined by the value P. The resulting $P.T_{\min}$ may be greater (or smaller) than E, which means the initial guess overshot (or undershot) the optimal power level. Next, reset P to $E/T_{\min}$, and repeat the procedure. The value of $T_{\min}$ will strictly decrease with each iteration if the original guess for P was too low, and strictly increase with each iteration if the initial guess was too high. As there is a unique [2] optimal $T_{\min}$, this procedure will converge. This iterative method has been used in generating the numerical examples given later in the chapter where its complexity is also discussed.

The following result extends the result of Lemma 3.3.3 to the case with two energy harvests.

Lemma 3.3.4 *Let there be a total of two energy harvests, whose times $t_1 = 0$ and $t_2 > 0$ are known by the sender as shown in Fig 3.5. For any given $B_1 > 0$ and $B_2 > 0$, the sender can transmit B_1 and B_2 bits to users 1 and 2, respectively, in the minimum total time, $T^{\text{opt}}(2)$, by*

[2] Note that the total energy used to transmit B_1 and B_2 bits, given by $T.g(r_1, r_2) = T.g(B_1/T, B_2/T)$ is convex, monotonically decreasing in T. Combining this with our initial assumption about E being large enough to satisfy the minimum energy per bit requirement ($E > \lim_{T\to\infty} T.g(B_1/T, B_2/T)$), there is always a unique smallest value T for which $T.g(r_1, r_2)$ is just below E.

encoding them such that the transmission to each user ends at the same time.

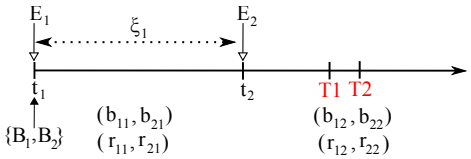


Figure 3.5: The setting of Lemma 3.3.4: Given two energy harvests, to minimize the overall transmission duration of B_1 and B_2 bits to each user, the sender finishes the transmission to both users at the same time, i.e. $T_1 = T_2$.

Proof. The proof will make use of Lemmas 3.3.2 and 3.3.3. Take any rate pair allocation (r_{11}, r_{21}) for the 1^{st} epoch, of length ξ_1, and suppose that this results in b_{11} and b_{12} bits being transmitted to the two users, respectively, where $b_{11} = r_{11}\xi_1$, $b_{21} = r_{21}\xi_1$. This leaves the following numbers of bits to be sent in the next epoch $b_{12} = B_1 - r_{12}\xi_1$, $b_{22} = B_2 - r_{22}\xi_1$.

By Lemma 3.3.3, the total time to finish these remaining bits, which is $T^{\text{opt}}(2) = T_{\min} = \min(\max(T_1, T_2))$ will be minimized by setting $T_1 = T_2$. More explicitly,

$$
\begin{aligned}
T_{\min} &= \min(\max(T_1, T_2)) = \min(\max(\frac{b_{12}}{r_{12}}, \frac{b_{22}}{r_{22}})) \\
&= \min(\max(\frac{B_1 - r_{11}\xi_1}{r_{12}}, \frac{B_2 - r_{21}\xi_i}{r_{22}})) \\
&= \begin{cases} \frac{B_1 - r_{11}\xi_1}{r_{12}} & if\ \frac{r_{12}}{r_{22}} < \frac{B_1 - r_{11}\xi_1}{B_2 - r_{21}\xi_1} \\ \frac{B_2 - r_{21}\xi_1}{r_{22}} & if\ \frac{r_{12}}{r_{22}} > \frac{B_1 - r_{11}\xi_1}{B_2 - r_{21}\xi_1} \end{cases}
\end{aligned}
$$

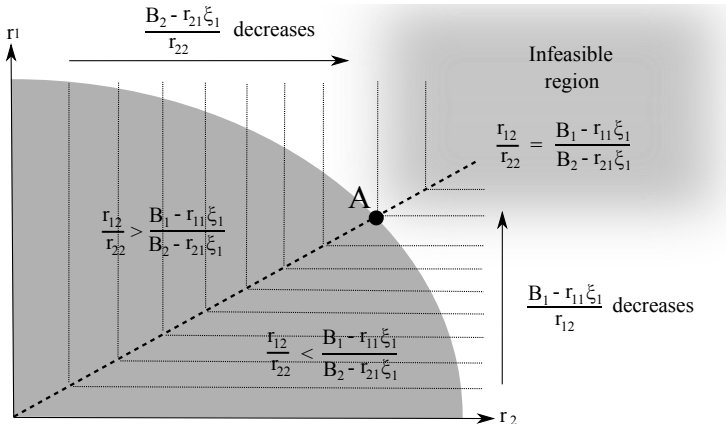


Figure 3.6: Illustration of the proof of Lemma 3.3.4.

$T^{\text{opt}}(2)$ is obtained by the rate pair satisfying $\frac{r_{12}}{r_{22}} = \frac{B_1 - r_{11}\xi_i}{B_2 - r_{21}\xi_i}$ as point A on achievable rate region shown in Fig. 3.6. By Lemma 3.3.2, for the resulting two-epoch schedule to be optimal, a constant power and rate pair must have been used in the first epoch, and we have just proved that $T_1 = T_2$ for any constant choice in the first epoch.

Finally, we generalize the first result of Lemma 3.3.4, to a general number of energy harvests.

Lemma 3.3.5 C*onsider the system model with an arbitrary number of energy harvests described in Fig. 3.2. In a schedule that minimizes overall transmission duration, the sender finishes transmission to both users at the same time.*

Proof. The claim has been proved for $k = 2$ energy harvests, in Lemma 3.3.4. We will prove the general case by induction. Suppose that there are k energy harvests with the k^{th} one at time t_k, and the induction hypothesis holds, such that the optimal scheduler finishes transmission to both users at $T^{\text{opt}}(k)$. Now, consider adding a new energy harvest at time t_{k+1}. We have the following possible cases:

1. $t_{k+1} \geq T^{\text{opt}}(k)$: By the time the $(k+1)^{st}$ energy harvest arrives, the transmission has been completed, so by causality this energy harvest cannot help, and $T^{\text{opt}}(k+1) = T^{\text{opt}}(k)$.

2. $t_{k+1} < T^{\text{opt}}(k)$: In this case, the $(k+1)^{st}$ harvest will be used, to reduce the completion time. Starting at time t_k, the sequence of rate pairs will change from $\{(r_{1i}, r_{2i})\}$ for $i = 1, \ldots, k$ to $\{(r'_{1i}, r'_{2i})\}$ for $i = 1, \ldots, k$ and $(r'_{1(k+1)}, r'_{2(k+1)})$ for the newly added epoch. As in the proof of Lemma 3.3.4, whatever the number of bits allocated to the new epoch is, this rate pair will have a slope equal to the ratio of the number of bits remaining for this epoch. Hence, the bits will be terminated at some time $T^{\text{opt}}(k+1) \leq T^{\text{opt}}(k)$. ■

We are now ready to state the broadcast transmission scheduling problem as an optimization problem. From Section 3.2, for a given rate pair, there corresponds a unique average power level P given by $g(r_1, r_2)$ such that this rate pair is on the boundary of the rate region with average power constraint P. The function $g(r_1, r_2)$ is strictly convex and continuously differentiable in r_1 and r_2. Using Lemmas 3.3.2 and 3.3.5, the problem can be written in terms of epoch rates.

Given B_1, B_2, and the sequence $\{E_i\}$, supposing the problem is feasible (the total amount of energy is sufficient for transmitting the total number of bits), one can find an upperbound for the transmission completion time, T^{up} in several ways. A simple one (which is possible when E_1 is sufficient for transmitting the total number of bits), is to set the power so low such that only the first harvest is used to transmit all the bits. A much better upperbound will be obtained by the procedure that will be described within the initialization step of the Flowright algorithm, in Section 3.4.

Given an upperbound for completion time, T^{up}, we set k^{up} equal to the index of the last energy harvest before this time, that is, $k^{\text{up}} = \max\{i : \sum_{j=1}^{i} \xi_j \leq T^{\text{up}}\}$. We know that the optimal solution will use at most k^{up} harvests, and WLOG, remaining harvests can be ignored. Hence the problem reduces to finding $T^{\text{opt}}(k^{\text{up}})$:

Problem 1 Transmission Time Minimization of Data Available at the Beginning on an Energy Harvesting Broadcast Channel:

$$\textit{Minimize: } T = T(\{(r_{1i}, r_{2i})\}_{1 \leq i \leq k^{\text{up}}})$$

$$\textit{subject to: } r_{1i}, r_{2i} \geq 0, \quad 1 \leq i \leq k^{\text{up}}$$

$$0 < T \leq T^{\text{up}}$$

$$\sum_{i=1}^{k} g(r_{1i}, r_{2i})\xi_i \leq \sum_{i=1}^{k} E_i \tag{3.11}$$

$$for\ k = 1, 2, ..., k^* = \max\{i : \sum_{j=1}^{i} \xi_j \leq T\}$$

$$\sum_{i=1}^{k^*} g(r_{1i}, r_{2i})\xi_i + g\left(\frac{(B_1 - \sum_{i=1}^{k^*} r_{1i}\xi_i)^+}{\left(T - \sum_{i=1}^{k^*} \xi_i\right)}, \frac{(B_2 - \sum_{i=1}^{k^*} r_{2i}\xi_i)^+}{\left(T - \sum_{i=1}^{k^*} \xi_i\right)}\right)\left(T - \sum_{i=1}^{k} \xi_i\right) = \sum_{i=1}^{k+1} E_i \tag{3.12}$$

The set of constraints in (3.11) ensure that energy causality is respected. At any time during transmission, the sender should have consumed *at most* the energy harvested up to that point, whereas by the end of transmission, it should have consumed *all* the harvested energies up to that instant. The constraint in (3.12) ensures that all the bits of each user have been transmitted by the time T. Note that, by assigning nonzero values to all k^{up} rates, one obtains a continuum of values of T that satisfy the constraints (note the $(\ )^+$ used in the last constraint which sets the result to zero whenever the argument is negative), but the infimum of these is the solution of the problem.

This is not a standard convex optimization problem due to T appearing in the final equality constraint. Yet, we will establish that this minimization problem can be solved iteratively using an adaptation of the FlowRight algorithm [29]. Before moving on to the solution, we present our final observations in the optimal schedule in Theorem 3.3.6.

Theorem 3.3.6 In the optimal schedule,

1. Average powers assigned to epochs are monotonically nondecreasing, i.e., $P_1 \leq P_2 \leq ... \leq P_n$.

2. Energy consumed during any whole constant power allocation band equals the total energy harvested in that band.

Proof.

1. To reach contradiction, suppose that in an optimal solution, powers do not increase monotonically, i.e., that we can find two powers such that $P_i > P_{i+1}$. From Corollary 1, we know that splitting average power can only increase overall transmission duration, hence by equalizing average power for both epochs (this never violates energy causality, hence it is feasible), we could find a rate pair such that more number of bits would be transmitted to each user. This contradicts the optimality of the original solution.

2. To reach contradiction, suppose that we have a constant power allocation region with average power P_1 and energy consumed is less than the energy harvested in that period. Then, there is a certain amount of energy deferred for later use increasing average power in some later constant power allocation region. Let us call the power in this region as P_2. From part one, we know that $P_1 < P_2$. Without violating energy causality, we can always have a small amount of increase in the former power allocation region to have a feasible average power $P_1^{'}$ and a decrease in the latter to have a feasible average power $P_2^{'}$. Since, $P_1 < P_1^{'} \leq P_2^{'} < P_2$, from Lemma 3.3.1, we can find rate pairs for the two new constant power regions to transmit at least the same number of bits to the users as in the original allocation. Now, keeping the rates (and hence powers) over the rest of the transmission duration the same, we have a new schedule. Under this new schedule, the same number of bits can be transmitted to the users at most by the end of the same

time duration as in the original allocation. This conflicts the optimality of the original allocation.

■

The next section is devoted to the solution of Problem 1.

3.4 OPTIMAL OFFLINE SCHEDULING WITH THE FLOWRIGHT ALGORITHM

FlowRight is an iterative algorithm proposed in the earlier literature [29] to solve minimum-energy scheduling problems over multiple-access channels, broadcast channels, and channels with fading when packets of all users need to be transmitted before a deadline T. Through a number of steps, it is adapted here to solve Problem 1. FlowRight performs simple iterations starting from a feasible initial schedule. Each iteration strictly improves the schedule (decreases T), which ultimately converges to the unique optimal T.

Initialization: The consumed energy in each epoch is set precisely equal to the energy harvested in the beginning of that epoch. This schedule is feasible in terms of energy. Therefore, the average power consumed at each epoch is $P_i = \frac{E_i}{\xi_i}$. Given this average power, one can assign rate pairs (r_{1i}, r_{2i}) on the achievable rate region boundary such that $\frac{r_{1i}}{r_{2i}} = \frac{B_1}{B_2}$ and after $n^{\text{up}} = \underset{i}{\text{argmin}} \{r_{1i} = 0, r_{2i} = 0\}$ epochs, B_1 and B_2 bits are transmitted to each user. The initialization phase is explained in the following pseudo-code.

```
//Initialization
i=0;
while (B_1 ≠ 0 or B_2 ≠ 0)
{
    i++;
    Select (r^0_1i, r^0_2i) such that:
    g(r^0_1i, r^0_2i) = E_i/ξ_i and r^0_1i/r^0_2i = B_1/B_2
    // Update remaining bits of each user
    B_1 = B_1 − r^0_1i ξ_i;
    B_2 = B_2 − r^0_2i ξ_i;
```

}

$n^{\text{up}} = i$; //Set the initial number of epochs to be considered

// which is an upperbound on the number of epochs used

// by optimal schedule

FlowRight performs local optimizations on pairs of epochs sequentially, i.e., on epochs $(1, 2)$, $(2, 3)$, $(3, 4)$, ... , until all epoch pairs are processed. This completes one iteration of the algorithm. Then, it continues with the next iteration, again performing local optimization on pairs of epochs at a time. The algorithm terminates after K iterations such that $K = \min\{k : T^k = T^{k-1}, i = 1, ..., n^k, j = 1, 2\}$, where T^k is the transmission completion time and $n^k \leq n$ is the number of epochs used at the end of k^{th} iteration.

The local optimizations are done in the following way: Let E_i^k be the energy consumed at the i^{th} epoch, at the end of the k^{th} iteration. Then, $E_i^0 = E_i$, $i = 1, 2, ..., n$. Also, let b_{ji}^k be the number of bits transmitted to j^{th} user at i^{th} epoch at the end of k^{th} iteration.

Now, consider the first two epochs (Fig.3.2). The total amount of energy used in these two epochs is

$$E_{total} = E_1^0 + E_2^0$$

and numbers of bits to transmit to each user are

$$b_1 = r_{11}^0 \xi_1 + r_{12}^0 \xi_2 \, , \, b_2 = r_{21}^0 \xi_1 + r_{22}^0 \xi_2.$$

Keeping total amount of energy and total numbers of bits constant, by respecting energy causality, we update (r_{11}^0, r_{21}^0) to locally optimal values (r_{11}^1, r_{21}^1) as described within the proof of Lemma 3.3.4 and illustrated in Figure 3.6. We then update the number of bits transmitted to each user and the energy consumed in the first epoch so that at the end of the first iteration the values of these are:

$$b_{11}^1 = r_{11}^1 \xi_1 \, , \, b_{21}^1 = r_{21}^1 \xi_1$$

$$E_1^1 = g(r_{11}^1, r_{21}^1) \xi_1$$

and reset the number of bits and energy at 2^{nd} epoch to new values as

$$b_{12}^0 = b_1 - b_{11}^1 \, , \, b_{22}^0 = b_2 - b_{21}^1$$

$$E_2^0 = E_{total} - E_1^1.$$

Then, we continue with local optimization on epochs $(2, 3)$. The total amount of energy is

$$E_{total} = E_2^0 + E_3^0$$

and the numbers of bits to transmit to each user are

$$b_1 = b_{12}^0 + r_{13}^0 \xi_3 \ , b_2 = b_{22}^0 + r_{23}^0 \xi_3.$$

Again, keeping E_{total}, b_1, and b_2 constant, we update rate pairs to locally optimal values (r_{12}^1, r_{22}^1) as described in Lemma 3.3.4. The next local optimization is on epochs $(3, 4)$. We proceed in this way to obtain (r_{1i}^1, r_{2i}^1) for $i = 1, 2, ..., n$. This completes the first iteration of the algorithm.

When the first iteration is finished, we start from the beginning and update rates two epochs at a time similar to the above. Using the observation described later in part one of Theorem 3.4.1, transmission completion time strictly decreases after each iteration, and the number of epochs used, n, is non-increasing from iteration to iteration. Hence, n is updated at the end of each iteration. We terminate after K iterations, where $K = \min\{k : T^k = T^{k-1}, i = 1, ..., n^k, j = 1, 2\}$. A pseudo-code for the algorithm is as follows.

```
passes=0; T^0 = T^up;
do{
    passes++;
    for (i=1:n-1){
        if(i==1){
            // Total bits of 1^st user in the epoch pair (1,2)
            b_1 = r_11^{k-1} ξ_1 + r_12^{k-1} ξ_2;
            // Total bits of 2^nd user in the epoch pair (1,2)
            b_2 = r_21^{k-1} ξ_1 + r_22^{k-1} ξ_2;
            // Energy used in the 1^st epoch at k^th iteration
            E_1^k = g(r_11^{k-1}, r_21^{k-1}) ξ_1;
            // Energy used in the 2^nd epoch at k^th iteration
            E_2^k = g(r_12^{k-1}, r_22^{k-1}) ξ_2;
            // Total energy used in the epoch pair (1,2)
            E_total = E_1^k + E_2^k;
```

// Set the feasible energy amount in the 1^{st} epoch at
// k^{th} iteration to the first energy harvest amount
$E_1^{k(max)} = E_1^0$;
}else{
// Total bits of 1^{st} user in epoch pair $(i, i+1)$
$b_1 = b_1 - r^k_{1(i-1)}\xi_{(i-1)} + r^{k-1}_{1(i+1)}\xi_{(i+1)}$;
// Total bits of 2^{nd} user in epoch pair $(i, i+1)$
$b_2 = b_2 - r^k_{2(i-1)}\xi_{(i-1)} + r^{k-1}_{2(i+1)}\xi_{(i+1)}$;
// Energy for the i^{th} epoch at k^{th} iteration
$E_i^k = E_{total} - g(r^k_{1(i-1)}, r^k_{2(i-1)})\xi_i$;
// Energy for the $(i+1)^{th}$ epoch at k^{th} iteration
$E^k_{i+1} = g(r^{k-1}_{1(i+1)}, r^{k-1}_{2(i+1)})\xi_{(i+1)}$;
// Total energy used in i^{th} and $(i+1)^{th}$ epoch
$E_{total} = E_i^k + E^k_{i+1}$;
// Feasible energy for i^{th} epoch at k^{th} iteration
Calculate_Emax(&$E_i^{k(max)}$);
}
[r^k_{1i} $r^{k-1}_{1(i+1)}$ r^k_{2i} $r^{k-1}_{2(i+1)}$] = update($E_i^{k(max)}$, E_i^k, E^k_{i+1}, b_1, b_2);
}
Update_n(); // Update the number of epochs
Calculate_T(&T^k); // Current transmission completion time
} while($T^k \neq T^{k-1}$)

Next, we shall prove that **FlowRight** solves Problem 1, through the following steps: We first establish that the cost function (that is, the completion time T) strictly decreases after each iteration of FlowRight, until it stops. We then show that the algorithm always stops. Finally, we prove that it cannot stop before achieving $T = T^{\text{opt}}$, which proves our claim. These claims are made precise in the following theorem.

Theorem 3.4.1 T*he following statements hold:*

1. As Flowright runs, the objective function T of Problem 1 strictly decreases *after each iteration. Conversely, if T did not change after an iteration, then FlowRight has stopped*

at the previous iteration.

2. FlowRight stops, and returns a sequence $\{r_{1i}^{\infty}, r_{2i}^{\infty}\}$.

3. When FlowRight stops, average powers corresponding to the resultant rate assignments at each epoch increase monotonically, i.e., $P_1 \leq P_2 \leq ... \leq P_n$.

Proof.

1. Suppose that we are on the k^{th} iteration of the algorithm, and so far `update` has operated on all epoch pairs up to the pair $(i-1, i)$. During the $(k-1)^{st}$ iteration, `update` has performed a local optimization on epochs $(i+1, i+2)$ and determined the rate pair to be used within the $(i+1)^{st}$ epoch. Let us call the number of bits to be transmitted in that epoch as $(\hat{b}_{1(i+1)}^{(k-1)}, \hat{b}_{2(i+1)}^{(k-1)})$. As the k^{th} iteration progress, `update` performs a local optimization on the epoch pair $(i-1, i)$. This operation determines the rate pair for the $(i-1)^{st}$ epoch at k^{th} iteration and temporarily updates the rate pair in the i^{th}. Suppose that this local optimization results in the minimum T_{local} before the end of the i^{th} epoch. Hence in the rest of the i^{th} epoch, rate pair changes to $(0, 0)$ i.e., a gap occurs. Let us call the number of bits to be transmitted in the i^{th} epoch as (b_{1i}^{k}, b_{2i}^{k}). Now define the total number of bits to be transmitted within epochs $(i, i+1)$ as $b_1 \triangleq b_{1i}^{(k-1)} + \hat{b}_{1(i+1)}^{(k-1)}$, and $b_2 \triangleq b_{2i}^{(k-1)} + \hat{b}_{2(i+1)}^{(k-1)}$. From Lemma 3.3.2, we know that by using a constant rate pair within the i^{th} epoch i.e., filling the gap within the epoch, at least the same number of bits can be transmitted to the users as in the original, slotted allocation. As the k^{th} iteration progress, `update` performs a local optimization on the epoch pair $(i, i+1)$. This operation determines the rate pair for the i^{th} epoch at k^{th} iteration and temporarily updates the rate pair in the $(i+1)^{th}$ epoch resulting in the minimum T_{local}. During this operation `update` at least tries to assign a constant rate pair in the i^{th} epoch using the same amount of energy since this reduces the number of bits to be transmitted in the $(i+1)^{th}$ epoch with the same amount of energy and hence, transmission surely ends before the end of the $(i+1)^{th}$ epoch i.e., the gap moves from i^{th} to the $(i+1)^{th}$ epoch. Therefore, after `update` operation, transmission completion time to transmit b_1 and b_2 bits within the epoch pair $(i, i+1)$ reduces. Continuing this way, the gap propagates to the end of the k^{th} iteration, hence the initial B_1 and B_2 bits are transmitted before the transmission completion time achieved at the end of $(k-1)^{st}$ iteration.

To prove the converse claim, suppose that after k^{th} iteration, T did not change. Then, there has not occurred any gap during local optimizations, otherwise it would have propagated to the last epoch used and hence, reduced T. Therefore, performing further iterations can not create any gaps during local optimizations meaning that algorithm has indeed stopped at $(k-1)^{st}$ iteration.

2. FlowRight initially starts from a feasible $T \geq T^{\text{opt}}$, which is obviously lower bounded by T^{opt}, the unique smallest completion time. From part one, indexed by iteration number k, $T(\{r_{1i}^k, r_{2i}^k\})$ is a strictly decreasing real, bounded below by a real number T^{opt}, hence the iterations eventually stop. Hence, the rate pairs $\{r_{1i}^k, r_{2i}^k\}$ converge to some final value $\{r_{1i}^\infty, r_{2i}^\infty\}$.

3. Suppose that after the algorithm has stopped, we can find two epochs i and $i+1$ with average powers such that $P_i > P_{i+1}$. From Lemma 3.3.2, we know that splitting average power can only increase overall transmission duration, hence by equalizing average power for both epochs (this never violates energy causality, hence it is feasible), we could find a rate pair such that more number of bits would be transmitted to each user. Therefore, a local optimization will result in a shorter transmission time i.e., a gap will occur triggering the next pair of epochs for a new local optimization as well, and so on. This contradicts with the claim that the algorithm has stopped. Therefore, when the algorithm has stopped, the resulting average power allocations corresponding to $\{(r_{1i}^\infty, r_{2i}^\infty)\}$ must be non-decreasing. ■

Theorem 3.4.2 The schedule returned by FlowRight is optimal, i.e., $T(\{r_{1i}^\infty, r_{2i}^\infty\}) = T^{\text{opt}}$.

Proof. Suppose that FlowRight stops and returns a schedule $\{r_{1i}^\infty, r_{2i}^\infty\} \triangleq S^{\text{fr}}$, with completion time $T(\{r_{1i}^\infty, r_{2i}^\infty\}) \triangleq T^{\text{fr}}$. Since FlowRight respects feasibility, T^{fr} can not be smaller than T^{opt}. Hence, $T^{\text{fr}} \geq T^{\text{opt}}$. Suppose $T^{\text{fr}} > T^{\text{opt}}$. We will show that this will contradict the assumption that FlowRight has stopped.

Consider the case that T^{opt} is in the m^{th} epoch and T^{fr} is in the n^{th} epoch with $n \geq m$ as shown in Fig. 3.7.

There must be a schedule $\{r_{1i}^{\text{opt}}, r_{2i}^{\text{opt}}\} \triangleq S^{\text{opt}}$ that achieves T^{opt}. Suppose that both S^{opt} and S^{fr} are equal up to epoch s, which is the *first* time they differ either in terms of power level or

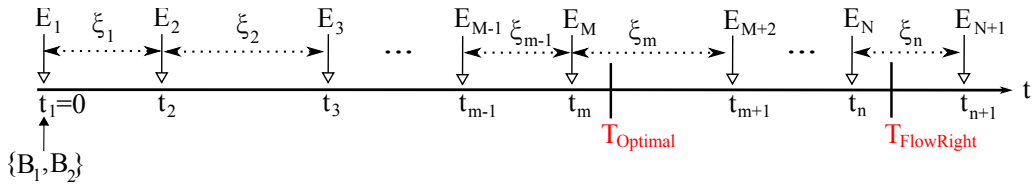


Figure 3.7: Illustration of the case $T^{\text{fr}} > T^{\text{opt}}$ in the proof of Theorem 3.4.2.

rates, or both. Let us denote the power allocated in epoch s in S^{opt} as P_s^{opt} and in S^{fr} as P_s^{fr}. There are three possibilities for epoch s:

1. $P_s^{\text{fr}} > P_s^{\text{opt}}$: We will show that this case is not possible. From part two of Theorem 3.3.6, we know that all the harvested energies are consumed within any constant power allocation region in S^{opt}. We also know from Theorem 3.4.1 that FlowRight cannot reduce its power level after this point. Hence, starting from epoch s when S^{opt} consumes all the energy at the end of that constant power region, S^{fr} would have consumed more energy than S^{opt}, which violates energy causality. This contradicts the fact that FlowRight always respects causality.

2. $P_s^{\text{fr}} < P_s^{\text{opt}}$: In this case, the power allocation in S^{fr} cannot stay at P_s^{fr} till the end of T^{opt}. To prove this claim, suppose that T^{opt} and T^{fr} are both in the same epoch. Since within the last epoch of S^{opt}, power allocation changes to zero between T^{opt} and T^{fr}, the power allocation in S^{opt} is not constant up to time T^{fr}. Also, we know that both schedules consume the total energy harvested up to time T^{fr}. In this case using Corollary 1, starting from the epoch s, S^{fr} should have transmitted more bits than S^{opt} using the same amount of energy, otherwise FlowRight can improve the schedule in the next iteration violating the assumption the algorithm has stopped. Hence, FlowRight has transmitted more than the initial B_1 and B_2 number of bits, contradicting the fact that it respects feasibility. For the general case where T^{fr} is not in the same epoch with T^{opt}, total energy consumed by S^{fr} is greater than that consumed by S^{opt} and this schedule certainly violates bit feasibility, as well. Therefore, the power allocation in S^{fr} should increase at some epoch before T^{opt}.

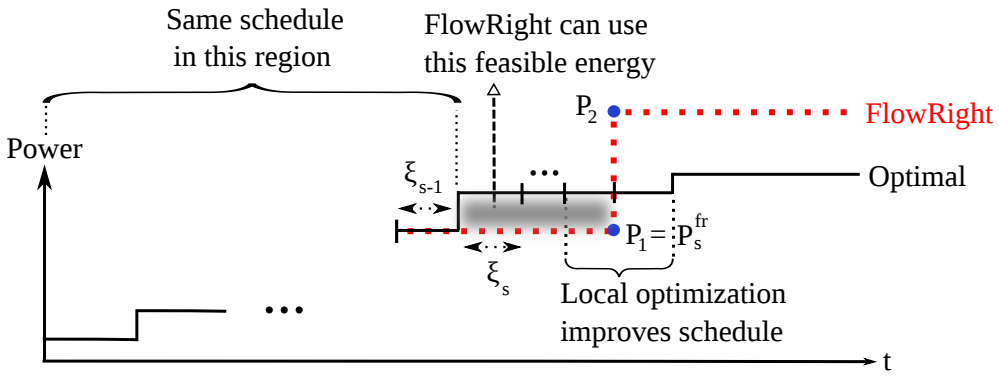


Figure 3.8: Illustration of the case (in the proof of Theorem 3.4.2) that at the first change between the optimal schedule and the schedule returned by FlowRight, average power in the optimal schedule is greater than the one in the schedule returned by FlowRight.

Now, we are left with the power allocation scheme [3] shown in Fig. 3.8. Suppose we now run FlowRight again on this schedule. Obviously, up to and including the epoch pair $(s-1, s)$, any local optimization does not the improve the schedule since energy flow is limited by causality. Since power is constant from the epoch pair $(s, s+1)$ up to the one in which power changes from P_1 to P_2 (Fig. 3.8), local optimizations on these epoch do not make any change in the schedule, as well. However, consider the epoch pair in which power changes from P_1 to P_2. A certain amount of additional feasible energy can be used in the former epoch to increase average power level to $P_1^{'}$. Then, keeping the total consumed energy the same within these two epochs, the average power in the latter epoch reduces to a power level $P_2^{'}$. In this case we have $P_1 < P_1^{'} \leq P_2^{'} < P_2$ and from Lemma 3.3.1 we can find rate pairs for the two epochs to transmit at least the same number of bits to the users as in the original allocation. Hence, after a local optimization in this epoch pair, transmission of bits within this epoch pair certainly ends before the end of latter epoch i.e., a gap occurs contradicting with the initial assumption that the algorithm has stopped.

3. $P_s^{\text{fr}} = P_s^{\text{opt}}$: In this case, the power allocation in S^{fr} can not be greater than the one in S^{opt} at the *second* change, otherwise energy causality is violated. Also, the power allocation in S^{fr} cannot be smaller than the one in S^{opt} at the *second* change, otherwise Flowright has not stopped. These two claims can be proven with similar arguments

[3] At the *second* change, the power allocation in S^{fr} greater than the one in S^{opt} is shown, yet the proof holds for the other cases, as well.

as in the proofs of cases (a) and (b). Hence, power allocation should be the same in both schedules at the *second* change. By the same argument, power allocation in S^{fr} follows that of S^{opt} until the beginning of the last constant power band of S^{fr} before T^{opt}. But in this case FlowRight has chosen different rate pairs in at least two epochs for the same power levels. This means, S^{fr} has violated bit feasibility by the time T^{opt}, since changing rate pairs affects the transmitted number of bits in favor of one user at the expense of the other. This contradicts the fact that FlowRight respects feasibility.

Therefore, $T^{\text{fr}} = T^{\text{opt}}$. ■

3.5 ALGORITHM COMPLEXITY

The FlowRight algorithm has polynomial time complexity in the number of epochs, n, that are used.

The core computational step in the algorithm is the calculation of T_{local} for two epochs given the total E units of energy, the causality constraint (how much of the total energy is available for use in the first epoch), and B_1 and B_2 bits to transmit to each user as in Lemma 3.3.3. This entails the solution of a nonlinear equation, which in our numerical computations has been done by making an initial guess for the power level, solving for rates, and then recomputing power as described in Section 3.3. The exact number of iterations depend on the initial guess for the power level P, but in our experiments this computation typically converge after a few steps. Let c_1 be the worst case computation time for calculating T_{min} for one epoch. On the local optimization of each epoch pair, one of the following two cases will occur:

1. Causality constraint is inactive: In this case, the power allocation will be constant and the local optimization reduces to finding T_{local} as if all the energy is harvested at the beginning of the epoch pair. The worst case complexity is therefore c_1.

2. Causality constraint is active: In this case, there is no energy flow. Therefore, given the maximum feasible average power for the first epoch, the algorithm computes the minimum transmission time to transmit all the remaining bits of two users on the second epoch as in Lemma 3.3.4. For each choice of rate pair for the first epoch, an optimiza-

tion must be made for the second. Let c_2 be the worst case number of times the first epoch rates need to be updated. So the worst case complexity of the local optimization is $c_1 \times c_2$.

For n epochs, $n-1$ local optimizations are performed at each iteration. In the worst case, causality is active for all pairs, corresponding to a computational complexity of $C_{iter} = c_1 \times c_2 \times (n-1)$.

While FlowRight theoretically terminates when T does not change from one iteration to the next, in terms of implementation it would make sense to stop the iterations when the change is, say, within some ϵ of the average epoch size. The choice of ϵ will determine the number of iterations and hence the linear scaling coefficient of complexity. In our test runs of FlowRight, we have observed that the number of iterations for achieving convergence sufficient for all practical purposes is on the order of the number of epochs. The actual number of computations will depend highly on optimizing the solution of the nonlinear equation in the local optimization stage of the implementation.

3.6 A NUMERICAL EXAMPLE

Consider a two-user AWGN broadcast channel with bandwidth 100 KHz and noise power spectral density $N_0 = 10^{-14}$ Watts/Hz. Suppose that the path loss from sender to the 1^{st} and 2^{nd} users are 70dB and 75dB, respectively. The sender needs to transmit 7.5Mbits to the 1^{st} user and 5Mbits to the 2^{nd} user. Energy harvests of amounts [4, 4, 3, 10, 3, 3, 8, 4, 3, 5] Joules arrive at [0, 1, 3, 7, 8, 10, 14, 15, 17, 20] seconds.

FlowRight computes the final schedule shown in Fig. 3.9. The stopping criterion used in this example was successive iterations being within $\frac{\epsilon}{n}\sum_{i=1}^{n}\xi_i$ where $\epsilon = 10^{-6}$. The algorithm stopped after 15 iterations and in the resulting optimal schedule, the last two energy harvests are not used. Note that in the final schedule, transmit powers remain constant during epochs, and are non-decreasing in time. Also note that bursts of energy harvests that arrive close together are combined and transmit power is kept constant as much as possible. Whenever there is a long gap between harvests, that is when the algorithm switches to a higher power level.

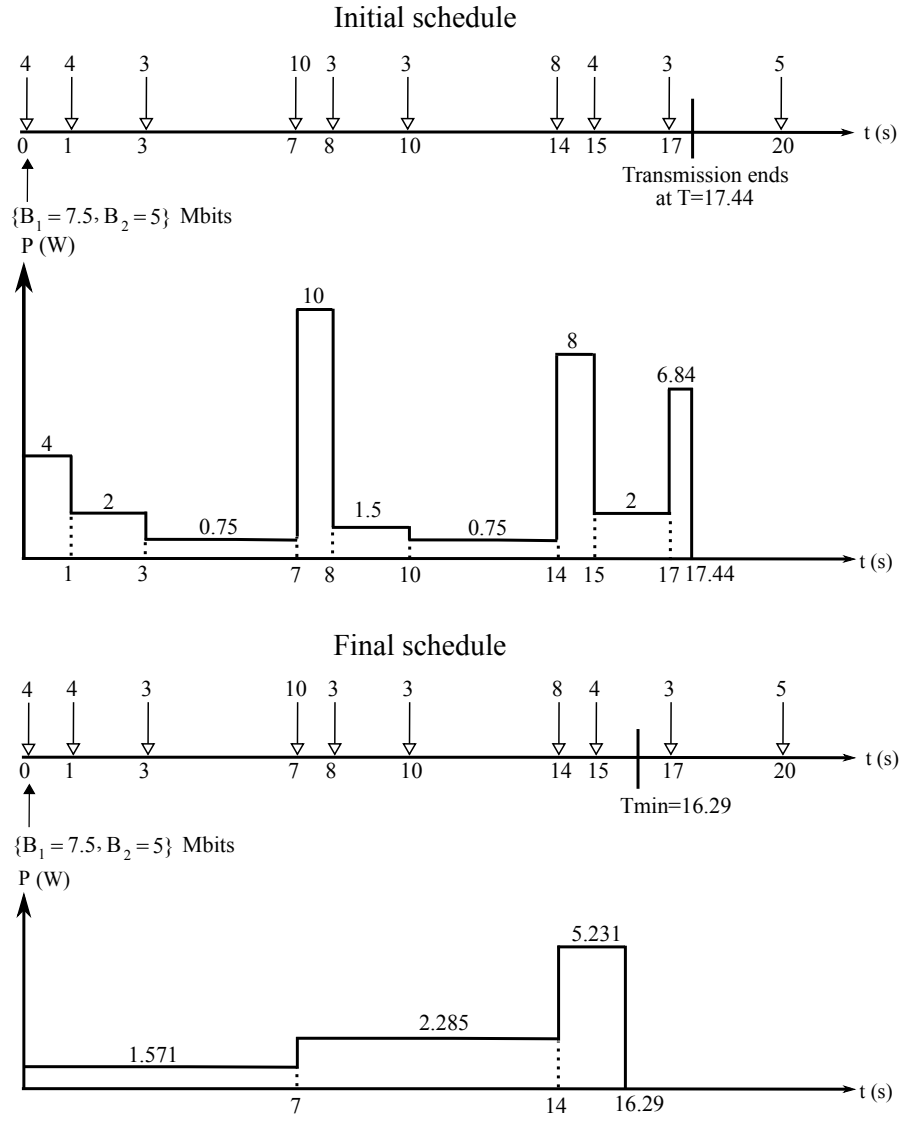


Figure 3.9: A numerical example for the execution of FlowRight algorithm. The top figure represents the transmission completion time, T=17.44s, after the initialization phase of the algorithm to transmit B_1=7.5Mbits and B_2=5Mbits for the given energy harvest instants with the corresponding energy amounts. The initial schedule is $\{(r_{1i}, r_{2i})\}$=[(514, 343),(452, 301),(363, 242),(595, 397),(426, 284),(363, 242),(576, 384),(452, 301),(562, 375)]Kbps with durations $\{\xi_i\}$ as shown in the top figure. Power allocations at the end of initialization phase are shown in the second figure. In the third figure, final transmission completion time, $T_{\min}$=16.29s, is shown after the termination of FlowRight. In the last figure, transmit powers are shown to be [1.571, 2.285, 5.231]W for the durations [7, 7, 2.29]s in the final schedule. Changes in transmit powers occur at the end of epochs, while they remain constant during epochs. The final schedule is $\{(r_{1i}^{\infty}, r_{2i}^{\infty})\}$=[(458, 262),(458, 262),(458, 262),(460, 313),(460, 313),(460, 313),(464, 428),(464, 428)]Kbps with durations $\{\xi_i\}$ as shown in the last figure.

3.7 CONCLUSIONS

In this chapter, we formulated and solved the offline minimization of transmission completion time problem for an energy harvesting broadcast link where all packets are available at the beginning. We showed that the sender need not change its transmission power during an epoch and should end the transmission to each user at the same time in the optimal schedule. Moreover, transmit powers exhibit non-decreasing behaviour in the optimal schedule meaning that energy harvests may not necessarily be depleted at the end of each epoch, and could be deferred for later use i.e., optimal scheduler extends the transmission completion duration. This is an interesting result since the aim of optimal scheduler is to minimize overall transmission completion time. After making these observations, the FlowRight algorithm is adapted to solve the problem. Starting with an upper bound on the optimal transmission completion time T, the algorithm strictly decreases T after each iteration and converges to the optimal value.

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

In this thesis, we investigated efficient wireless network design in the context of a practical problem i.e., the implementation of a WSN testbed comprised of nodes with finite batteries and capable of RF communication to perform ferromagnetic target detection and localization and, a theoretical problem i.e., data scheduling in a multi-user communication system capable of energy harvesting instead of using finite batteries.

Challenges in building a WSN performing ferromagnetic target detection and localization are addressed in Chapter 2. A test target is modeled as a magnetic dipole, and the model is validated through experiments. Magnetic dipole parameters are estimated jointly with sensor sensitivity which varies across sensors. Even though test targets with more complicated geometries could be selected to work with, we selected an iron bar to accurately model the target, and validate dipole moment model, while knowing that targets with complicated geometries can be modeled as the superposition of magnetic dipole moments. An efficient detection scheme is proposed coping with highly noisy magnetic measurements by pre-filtering noise using a moving average filter and track the smoothed ambient magnetic signal, and the preclusion of saturation of magnetic sensors by performing soft-calibration relatively much faster than the reliable sensor data sampling. Localization is performed on the PC side via the Simulated Annealing algorithm, for which a performance of approximately 10% localization error with respect to the real distance between target and sensor is obtained. Performance of the WSN is examined through operating the network to sequentially localize the moving test target. There is inevitably some amount of random delay in the network operation due to processing delay caused by TinyOS, random back-off in the MAC layer, transmission of reliable magnetic readings to PC via the physical layer, and delay caused by Simulated Annealing

algorithm, but yet the network responds with acceptable delay.

In future work, we intend to explore tracking performance and energy efficiency of more sophisticated tracking algorithms, such as Kalman filter based methods and particle filtering. In particular, we intend to use the magnetic sensor network developed in this thesis as a experimental test bed for investigation energy-efficient tracking algorithms for randomly distributed sensor networks, addressing the practical considerations of sensor fusion and overcoming the challenges presented by sparsity. The results of this thesis can also be extended to applications involving vehicle tracking and target classification.

In Chapter 3, the offline transmission completion time minimization problem on an energy harvesting broadcast link is formulated and solved. It is observed that, in the optimal solution, energy harvests may not necessarily be depleted at the end of each epoch, and could be deferred for later use. The schedule tries to "hurry up and be lazy" at the same time. The sender picks rates from the broadcast capacity region judiciously, such that it completes transmission to both users at the same time, T. In the optimal schedule, the powers are nondecreasing in time, so that transmission rate is highest toward the end. It is shown that the problem can be solved efficiently with a modification of the FlowRight algorithm. The proposed algorithm starts with an upperbound on T and strictly improves it after every iteration or "pass" through the schedule, and stops when T converges to the optimal value.

There are a number of directions for further work related to the problem presented in Chapter 3. One of these is solving the offline minimization problem when data arrive during transmission, rather than being available in the beginning. Our preliminary work on this modification of the problem indicates that its solution has similar structural properties to the first problem, such as the powers being nondecreasing in time, and rates not changing between data arrival or energy harvest instants. Here, the optimal solution has more reason to be "lazy" in terms of transmission rate, as data will continue to come and it may be wise to save energy for future data arrivals. We believe that a further modified version of the iterative algorithm described in Chapter 3 solves this version of the problem.

A second direction for further work is addressing the multiple-access version of this problem. There, energy harvests will be occurring at the senders, possibly at different points in time. Finding a distributed solution for that case may be a difficult yet interesting problem.

Finally, another issue of interest is time-varying channel gain. The case of time-varying channel gain is interesting, and perhaps more meaningful to be setup as an online problem, rather than an offline problem, as channel gain variation is often difficult to predict (whereas energy harvesting times or packet formation times may be known ahead of time in some applications.) As the offline problem formulation has facilitated the analysis of the problem, going to an online formulation is arguably the most important challenge. While there are different ways to formulate the online problem, for example, as a dynamic control problem, our intuition is that approximate methods that leverage the offline formulation may be more tractable and insightful.

REFERENCES

[1] Mehmet A. Antepli, Sevgi Z. Gurbuz and Elif Uysal-Biyikoglu. Ferromagnetic Target Detection and Localization with a Wireless Sensor Network. *Military Communications Conference*, 2010, accepted.

[2] V. K. Boda, A. Nasipuri, I. Howit. Design Considerations for a Wireless Sensor Network for Locating Parking Spaces. *Proceedings of IEEE SoutheastCon*, pages 698-703, Richmond, VA, Mar 2007.

[3] S. Y. Cheung , S. Coleri , B. Dundar, S. Ganesh, C.W. Tan, P. Varaiya. Traffic Measurement and Vehicle Classification with Single Magnetic Sensor. *Transportation Research Record: Journal of the Transportation Research Board*, volume 1917, pages 173-181, 2005.

[4] K. Dimitropoulos, N. Grammaldis, I. Gragopoulos, H. Gao, Th. Heuer, M. Weinmann, S.Voit, C. Stockhammer, U. Hartmann and N. Pavlidou. Detection, Tracking and Classification of Vehicles and Aircraft based on Magnetic Sensing Technology. *Trans. on Engineering, Computing and Technology*, volume 14, pages 161-166, 2006.

[5] B. Ginzburg, L. Frumkis, B.Z. Kaplan. Processing of magnetic scalar gradiometer signals using orthonormalized functions. *Sensors and Actuators A: Physical*, volume 102 (1-2), pages 67-75, December 2002.

[6] B. Ginzburg, L. Frumkis, B.Z. Kaplan. Investigation of advanced data processing technique in magnetic anomaly detection systems. *International Journal on Smart Sensing and Intelligent Ststems*, volume 1, no.1, pages 110-122, March 2008

[7] R. Farrell , R.o Garcia, D. Lucarelli, A. Terzis , I. J. Wang. Localization in Multi-Modal Sensor Networks. *Proceedings of The Third International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2007.

[8] Memsic Inc. MTS/MDA Sensor Board Users Manual. Revision A, June 2007. http://www.memsic.com/support/documentation/wireless-sensor-networks/category/6-user-manuals.html (Last visited on August 2010)

[9] Honeywell Inc. 1- and 2- Axis Magnetic Sensors. Magnetic Sensors Literature. http://www.ssec.honeywell.com/magnetic/datasheets.html#datasheets. (Last visited on August 2010)

[10] Yeti2 - TinyOS 2 Plugin for Eclipse. ETH Zurich. http://tos-ide.ethz.ch/wiki/index.php (Last visited on 20 August 2010)

[11] Deluge T2. http://docs.tinyos.net/index.php/Deluge_T2 (Last visited on 20 August 2010)

[12] Stanford Informatin Networks Group. Collection Tree Protocol. http://sing.stanford.edu/pubs/sing-09-01.pdf. (Last visited on 20 August 2010)

[13] UC Berkeley. The Collection Tree Protocol (CTP). http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html. (Last visited on 20 August 2010)

[14] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis. Four bit wireless link estimation. *In Proceedings of the Sixth Workshop on Hot Topics in Networks (HotNets VI)*, 2007.

[15] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. *MobiCom 2003: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134-146, September 2003.

[16] UC Berkeley. Dissemination of Small Values. http://www.tinyos.net/tinyos-2.x/doc/html/tep118.html. (Last visited on 20 August 2010)

[17] UCD Dublin. Octopus: A Dashboard for Sensor Networks Visual Control. http://www.csi.ucd.ie/content/octopus-dashboard-sensor-networks-visual-control (Last visited on August 2010)

[18] Honeywell Inc. Application Note - AN218 Vehicle Detection Using AMR Sensors. Magnetic Sensors Literature. http://www.ssec.honeywell.com/magnetic/datasheets.html #appnotes. (Last visited on 20 August 2010)

[19] Texas Instruments. INA2126 Micropower Instrumentation Amplifier Single and Dual Versions. http://focus.ti.com/docs/prod/folders/print/ina2126.html. (Last visited on 20 August 2010)

[20] Geeknet Inc. SCM Repositories - tinyos. http://tinyos.cvs.sourceforge.net/viewvc/tinyos/tinyos-1.x/apps/MicaSBVerify/MicaSBTest1. (Last visited on August 2010)

[21] D. J. Griffiths. *Introduction to Electrodynamics*, Prentice Hall, 1999.

[22] T. Kailath, A.H.Sayed and B. Hassibi. *Linear Estimation*, Prentice Hall, 2000

[23] Wikipedia. Simulated annealing. http://en.wikipedia.org/wiki/Simulated_annealing. (Last visited on August 2010)

[24] E. Uysal-Biyikoglu. Design of minimum-energy high performance wireless communication networks: Inter-layer optimization and algorithms. project sponsored by TUBITAK, No:106E119, 2006.

[25] B. Prabhakar, E. Uysal-Biyikoglu, and A. El Gamal. Energy-efficient Transmission over a Wireless Link via Lazy Packet Scheduling. *IEEE INFOCOM*, pages 386-394, 2001.

[26] R. A. Berry and R. G. Gallager. Communication over fading channels with delay constraints. *IEEE Transactions on Information Theory*, volume 48, pages 1135-1149, May 2002.

[27] P. Nuggehalli, V. Srinivashan, and R. R. Rao. Delay constrained energy efficient transmission strategies for wireless devices. *in Proc.IEEE INFOCOM*, volume 3, pages 1765-1772, New York, June 2002.

[28] M. A. Zafer and E. Modiano. A calculus approach to energy-efficient data transmission with quality of service constraints. *IEEE/ACM Transactions on Networking*, volume 17, pages 898-911, June 2009.

[29] Elif Uysal-Biyikoglu and A. El Gamal. On adaptive transmission for energy efficiency in wireless data networks. *IEEE Transsactions on Information Theory*, volume 50, pages 3081-3094, Dec 2004.

[30] J. Yang and S. Ulukus. Optimal Packet Scheduling in an Energy Harvesting Communication System. Working paper, 2010.

[31] Elif Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal. Energy-efficient Transmission over a Wireless Link via Lazy Packet Scheduling. *IEEE Trans. Networking*, pp. 487-499, Aug. 2002

[32] Y. Polyanskiy, H. V. Poor and S. Verdu. Minimum energy to send k bits with and without feedback. *2010 IEEE Int. Symposium on Information Theory*. Austin, Texas, June 13-18, 2010.

[33] Aman Jain, Sanjeev R. Kulkarni and Sergio Verdu. Minimum Energy per Bit for Gaussian Broadcast Channels with Cooperating Receivers and Common Message. *Proc. Forty-Seventh Annual Allerton Conference on Communication, Control, and Computing*, Monticello, USA, Sep. 2009.

[34] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, John Wiley & Sons, Inc., 1991.

APPENDIX A

PROOF OF STRICT CONVEXITY OF $g(r_1, r_2)$

$g(r_1, r_2)$ is strictly convex in **dom** f if and only if its Hessian matrix is positive definite for all $(r_1, r_2) \in$ **dom** g, i.e., [1]

$$H = \begin{bmatrix} g_{xx} & g_{xy} \\ g_{yx} & g_{yy} \end{bmatrix} > 0.$$

1^{st} and 2^{nd} order derivatives of $g(r_1, r_2)$ can be written as

$$g_x(r_1, r_2) = \frac{\sigma^2(2\ln 2)2^{2r_1}2^{2r_2}}{s_1} \tag{A.1}$$

$$g_y(r_1, r_2) = \sigma^2(2\ln 2)2^{2r_2}\left(\frac{1}{s_2} + \frac{(2^{2r_1}-1)}{s_1}\right) \tag{A.2}$$

$$g_{xx}(r_1, r_2) = \frac{\sigma^2(2\ln 2)^2 2^{2r_1}2^{2r_2}}{s_1} = (2\ln 2)g_x(r_1, r_2) \tag{A.3}$$

$$g_{yx}(r_1, r_2) = g_{xy}(r_1, r_2) = \sigma^2\frac{2^{2r_1}2^{2r_2}(2\ln 2)^2}{s_1} = (2\ln 2)g_x(r_1, r_2) \tag{A.4}$$

$$g_{yy}(r_1, r_2) = \sigma^2(2\ln 2)^2 2^{2r_2}\left(\frac{1}{s_2} + \frac{(2^{2r_1}-1)}{s_1}\right) = (2\ln 2)g_y(r_1, r_2). \tag{A.5}$$

Note that $g_{xy} = g_{yx}$, and H is symmetric. Therefore, H positive definite if and only if

$$g_{xx} > 0$$

$$g_{xx}g_{yy} - g_{xy}g_{yx} > 0.$$

Obviously, $g_{xx} > 0$ and $g_{yy} > 0$. Since, $s_1 > s_2$,

$$\begin{aligned} g_{xx}g_{yy} - g_{xy}g_{yx} &= (2\ln 2)^2 g_x(r_1, r_2)\left(\underbrace{g_y(r_1, r_2) - g_x(r_1, r_2)}_{>0}\right) \\ &> 0 \end{aligned}$$

[1] g_x and g_y represent the first order partial derivatives of $g(r_1, r_2)$ with respect to r_1 and r_2, respectively. Second order partial derivatives of $g(r_1, r_2)$ are represented by g_{xx}, g_{xy}, g_{yx} and g_{yy}

APPENDIX B

PROOF OF PROPOSITION 1

1^{st} and 2^{nd} order derivatives of $h_1(P,r)$ and $h_2(P,r)$ for the AWGN BC are as follows [1]

$$h_{1_x}(P,r) = \frac{1}{2}(\log_2 e)\frac{s_1 s_2}{s_1 s_2 P + s_1\sigma^2 - (s_1 - s_2)\sigma^2 2^{2r}} \geq 0 \tag{B.1}$$

$$h_{1_y}(P,r) = -\frac{s_1 s_2 P + s_1\sigma^2}{s_1 s_2 P + s_1\sigma^2 - (s_1 - s_2)\sigma^2 2^{2r}} \leq 0 \tag{B.2}$$

$$h_{1_{xx}}(P,r) = -\frac{1}{2}(\log_2 e)\frac{(s_1 s_2)^2}{(s_1 s_2 P + s_1\sigma^2 - (s_1 - s_2)\sigma^2 2^{2r})^2} \leq 0 \tag{B.3}$$

$$h_{1_{yy}}(P,r) = -\frac{(2\ln 2)(s_1 - s_2)(s_1 s_2 P + s_1\sigma^2)\sigma^2 2^{2r}}{(s_1 s_2 P + s_1\sigma^2 - (s_1 - s_2)\sigma^2 2^{2r})^2} \leq 0 \tag{B.4}$$

$$h_{1_{xy}}(P,r) = -\frac{s_1 s_2(s_1 - s_2)\sigma^2 2^{2r}}{(s_1 s_2 P + s_1\sigma^2 - (s_1 - s_2)\sigma^2 2^{2r})^2} \leq 0$$

$$h_{1_{yx}}(P,r) = \frac{(s_1 - s_2)(s_1 - s_2)\sigma^2 2^{2r}}{(s_1 s_2 P + s_1\sigma^2 - (s_1 - s_2)\sigma^2 2^{2r})^2} \geq 0$$

$$h_{2_x}(P,r) = \frac{1}{2}(\log_2 e)\frac{s_2}{s_2 P + \sigma^2} \geq 0 \tag{B.5}$$

$$h_{2_y}(P,r) = -\frac{2^{2r_1}}{(2^{2r_1} - 1) + \frac{s_1}{s_2}} \leq 0 \tag{B.6}$$

$$h_{2_{xx}}(P,r) = -\frac{1}{2}(\log_2 e)\frac{{s_2}^2}{(s_2 P + \sigma^2)^2} \leq 0 \tag{B.7}$$

$$h_{2_{yy}}(P,r) = -\frac{(2\ln 2)2^{2r_1}\frac{s_1 - s_2}{s_2}}{((2^{2r_1} - 1) + \frac{s_1}{s_2})^2} \leq 0 \tag{B.8}$$

$$h_{2_{xy}}(P,r) = h_{2_{yx}}(P,r) = 0 \tag{B.9}$$

From (3.2) and (3.3), $h_1(P,r)$ and $h_2(P,r)$ are nonnegative. Monotonicity follows from (B.1), (B.2), (B.5) and (B.6) as the signs of the first order partial derivatives of $h_1(P,r)$ and $h_2(P,r)$ in their respective domains are always fixed. From (B.3), (B.4), (B.7) and (B.8), $h_1(P,r)$ and $h_2(P,r)$ are concave in power and, respectively rate when the other parameter is held constant. The last property follows from (B.9).

[1] h_{i_x} and h_{i_y} represent the first order partial derivatives of h_i with respect to P and r, respectively (i $\in \{1,2\}$). Second order partial derivatives of h_i are represented by $h_{i_{xx}}$, $h_{i_{xy}}$, $h_{i_{yx}}$ and $h_{i_{yy}}$

APPENDIX C

PROOF OF LEMMA 3.3.1

Using (3.8), (3.10) can be written as

$$f(\beta) = h_2(P_1+(1-\beta)\Delta P, \bar{r_1})\beta+h_2(P_2-\beta\Delta P, \bar{r_1})(1-\beta)-h_2(P_1, r_{11})\beta-h_2(P_2, r_{12})(1-\beta). \quad \text{(C.1)}$$

The 1^{st} and 2^{nd} order derivatives of f with respect to β are as follows [1]

$$\begin{aligned}
\frac{\partial f}{\partial \beta} &= h_2(P_1 + (1-\beta)\Delta P, \bar{r_1}) - h_2(P_2 - \beta\Delta P, \bar{r_1}) - h_2(P_1, r_{11}) + h_2(P_2, r_{12}) \\
&\quad +\beta\left\{h_{2_x}(P_1 + (1-\beta)\Delta P, \bar{r_1})(-\Delta P) + h_{2_y}(P_1 + (1-\beta)\Delta P, \bar{r_1})(r_{11} - r_{12})\right\} \\
&\quad +(1-\beta)\left\{h_{2_x}(P_2 - \beta\Delta P, \bar{r_1})(-\Delta P) + h_{2_y}(P_2 - \beta\Delta P, \bar{r_1})(r_{11} - r_{12})\right\} \qquad \text{(C.2)} \\
\frac{\partial^2 f}{\partial^2 \beta} &= 2(\underbrace{h_{2_x}(P_1 + (1-\beta)\Delta P, \bar{r_1})(-\Delta P) - h_{2_x}(P_2 - \beta\Delta P, \bar{r_1})(-\Delta P)}_{\leq 0}) \\
&\quad +2(\underbrace{h_{2_y}(P_1 + (1-\beta)\Delta P, \bar{r_1})(r_{11} - r_{12}) - h_{2_y}(P_2 - \beta\Delta P, \bar{r_1})(r_{11} - r_{12})}_{=0}) \\
&\quad +\beta\left\{\underbrace{h_{2_{xx}}(P_1 + (1-\beta)\Delta P, \bar{r_1})(-\Delta P)^2}_{\leq 0} + \underbrace{h_{2_{xy}}(P_1 + (1-\beta)\Delta P, \bar{r_1})(-\Delta P)(r_{11} - r_{12})}_{=0}\right\} \\
&\quad +\beta\left\{\underbrace{h_{2_{yx}}(P_1 + (1-\beta)\Delta P, \bar{r_1})(-\Delta P)(r_{11} - r_{12})}_{=0} + \underbrace{h_{2_{yy}}(P_1 + (1-\beta)\Delta P, \bar{r_1})(r_{11} - r_{12})^2}_{\leq 0}\right\} \\
&\quad +(1-\beta)\left\{\underbrace{h_{2_{xx}}(P_2 - \beta\Delta P, \bar{r_1})(-\Delta P)^2}_{\leq 0} + \underbrace{h_{2_{xy}}(P_2 - \beta\Delta P, \bar{r_1})(-\Delta P)(r_{11} - r_{12})}_{=0}\right\} \\
&\quad +(1-\beta)\left\{\underbrace{h_{2_{yx}}(P_2 - \beta\Delta P, \bar{r_1})(-\Delta P)(r_{11} - r_{12})}_{=0} + \underbrace{h_{2_{yy}}(P_2 - \beta\Delta P, \bar{r_1})(r_{11} - r_{12})^2}_{\leq 0}\right\} \\
&\quad \leq 0 \qquad \text{(C.3)}
\end{aligned}$$

(C.3) always holds according to the properties in Proposion 1. Hence f is concave in β.

[1] h_{2_x} and h_{2_y} represent the first order partial derivatives of h_2 with respect to P and r, respectively. Second order partial derivatives of h_2 are represented by $h_{2_{xx}}$, $h_{2_{xy}}$, $h_{2_{yx}}$ and $h_{2_{yy}}$