High Performance Adaptive Transmit Beamforming for Wireless Networks using Binary CSIT

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

BALASUBRAMANIAN GOPALAKRISHNAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Professor NIKOS D. SIDIROPOULOS

May, 2015

Acknowledgements

I would like to start by expressing my sincere gratitude towards my research advisor Prof. Nikos Sidiropoulos who has been a great person to work with and I thank him for the inspiration, encouragement and the endless source of ideas he has given me, that has helped to strive harder to achieve better results and successfully complete my PhD at the University of Minnesota. The experience with Prof. Nikos is something that I will always cherish as it has helped me to grow professionally and intellectually.

I would also like to thank Dr. Nihar Jindal, who was my initial graduate research advisor. Dr. Jindal helped me to ease through the transition process into graduate school and provided valuable academic and research guidance which was very important at the initial phase of my doctoral research.

I would like to extend my gratitude towards the Head of Department, Electrical Engineering Prof. David Lilja and Director of Graduate Studies (DGS) for Electrical Engineering, Prof. Randall Victora, for their extensive support during my stint in the University.

I would like to express my gratitude towards Graduate School at the University of Minnesota, Department of Electrical Engineering for awarding me the Graduate School Fellowship for the first three years of my academic research. I would also like to thank the National Science Foundation which has been responsible for funding the research projects that I was working on during the course of my PhD.

Over, the course of my PhD, I have had the opportunity to interact with incredible labmates (Dr. Omar Mehanna, Kejun Huang, Dr. Niranjay Ravindran, John Marcos, Artem Mosesov, Aritra Konar, John Tranter, Dr. Xiao Fu, Dr. Peng Wu, Joe Blomer, Dr. Amogh Rajanna, Dr. Juyul Lee, Bo Yang, Charilaos Kanatsoulis and Dr. Thimios Tsakonas). The discussions that I had with them on my research problems were very fruitful and enabled me to view the problem

of interest with a different perspective which further resulted in better ideas and research directions. I would also like to thank my colleagues in the Digital Technology Center (DTC) and Department of Electrical Engineering at the University of Minnesota. I would like to sincerely thank my friends whom I have met during the course of my stay in Minnesota who have created a friendly atmosphere around me and have helped me on numerous occasions when I needed support.

I would also like to thank my Father, Mother and my close relatives, from the bottom of my heart, for their blessings and the constant moral support and motivation they have provided throughout my project which has helped in the successful completion of my doctoral thesis. Finally I would like to thank God for being the guiding light throughout my research odyssey, giving me the strength, will-power and determination in order to successfully complete my PhD.

Dedication

Dedicated to

My grandparents and parents who have been my pillars of support throughout my life

Abstract

Transmit beamforming is a characteristic feature of the modern wireless communication standards like 4G-LTE and 802.11 ac because of the increasing demand for higher data rates and better Quality of Service at the user-end. Transmit beamforming uses multiple transmit antennas and channel state information (CSI) at the transmitter (Tx) to steer the radiated power towards the intended receiver (Rx) while limiting the leakage caused in other directions. In the absence of channel reciprocity, this channel information is acquired at the transmitter by channel estimation at the intended Rx and subsequent feedback of the quantized channel information back to the Tx. This conventional training method requires a complex Rx design and high communication overhead, which could be a burden when the receivers operate on battery power and have limited computational resources and restricted communication capabilities. Obtaining CSI for limiting interference caused due to leakage, can be much more challenging especially when the Rx affected by the spatial leakage interference (sidelobes) is not cooperating with the Tx (as in secondary transmit beamforming in underlay cognitive radio networks, for achieving a high Quality of Service at the secondary Rx while limiting the interference to the primary Rx).

This thesis proposes various algorithms that enable the Tx, which has no initial CSI, learn to beamform on-the-fly and asymptotically attain the performance achievable using perfect CSI at the Tx, using 1-bit direct or implicit periodic feedbacks from the receivers of interest. The receivers are assumed to have limited computational capability. This thesis starts by considering long-term transmit beamforming for point-to-point Multiple-Input Single-Output (MISO) links and proposes an online beamforming and learning algorithm using the analytic center cutting plane method (ACCPM) which is shown to asymptotically attain optimal performance. A robust maximum-likelihood formulation is next developed to combat feedback errors and correlation drift.

The setup is then extended to an underlay cognitive radio network for designing secondary transmit beamforming vectors that maximize the average Signal-to-Noise Ratio (SNR) at the secondary Rx while limiting the interference to the primary Rx, using direct binary channel quality indicator feedback bits from the secondary Rx and indirect ACK-NACK feedback from the primary Rx. When the primary interference threshold is known at the secondary Tx, it

is analytically shown that the proposed algorithm converges to maximum average SNR at the secondary Rx achieved using perfect CSI at the Tx.

Subsequently, the thesis considers max-min fair transmit beamforming for single group multicast networks (which is NP-hard in general) and introduces a new class of adaptive beamforming algorithms that features guaranteed convergence and state-of-the-art performance at low complexity, when perfect CSI is available at the Tx. Each beamforming vector update takes a step in the direction of an inverse SNR-weighted linear combination of the SNR-gradient vectors of all the users. Convergence of this update to a Karush-Kuhn-Tucker (KKT) point of a related proportionally fair beamforming is established. Simulations show that the proposed approach outperforms the prior state-of-art in terms of multicast rate, at considerably lower complexity. For cases where there is no initial channel state information at the transmitter, an extension of the online algorithm developed for point to point MISO links is developed that simultaneously learns the user channel correlation matrices and adapts the beamforming vector to maximize the minimum (long-term average) SNR among the users, using only periodic binary SNR feedback from each receiver.

The design methodology for the multicast beamforming problem is finally extended in a novel fashion to address the problem of obtaining feasible solutions to non-convex Quadratically Constrained Quadratic Programs (QCQP) with two-sided constraints when the associated matrices are positive semi-definite. In this context, the proposed algorithm starts with a infeasible solution which is iteratively updated using a gradient of the log-barrier function of the non-convex constraints followed by projection onto the intersection of the set of convex constraints (which is accomplished using a low-complexity Cyclic Dykstra's Projection algorithm) and a refining step using successive linear approximation. Convergence of the algorithm is established using the Descent lemma and simulations show that the algorithm obtains feasible solutions with a high probability at a much lower complexity compared to the state-of-the-art.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Transmit beamforming is an integral part of the modern wireless communication standards in cellular systems (4G-LTE) and Wi-Fi networks (802.11 ac/ah) mainly because of the increasing usage of multiple transmit antennas in smartphones, wireless routers etc., and increased demands for higher data rates and Quality of Service at the user-end. Scarcity of the resources in the time and frequency domain has resulted in utilizing the spatial degrees of freedom using multiple transmit antennas to attain high data rates in the downlink. Transmit beamforming uses multiple antennas at the transmitter (Tx) to steer the radiated power in the direction of the intended receiver while limiting the interference caused to other neighboring communication systems, thereby facilitating co-existence. However, for efficiently using the spatial degrees of freedom and designing good transmit beamforming vectors, channel state information is required *a priori*.

In the absence of channel reciprocity, this channel state information is acquired by periodic transmission of pilot symbols by the Tx which are subsequently used to estimate the channel coefficients at the receiver (Rx), which are quantized (using scalar or vector quantization) and fed back to the Tx. This requires a sophisticated Rx design and a high communication overhead, which can be a burden when the Rx operates on battery power and / or has limited computational resources and / or restricted communication capability. In addition to the CSI about the channel to the intended Rx, it might also be necessary to obtain the CSI about the interference channel from the Tx to the Rx of other co-existing systems, in order to limit the interference at the corresponding Rx. This might be a very challenging task especially if the co-existing systems are non-cooperative (as in the primary network (licensed user) in a cognitive radio network setup

does not cooperate with the secondary network (unlicensed user)). In this context, the important question that needs to be addressed is as follows: Starting from no initial CSIT, is it possible for the Tx to learn the channel information accurately and design transmit beamforming vectors that attain a high received SNR using a simple Rx design and limited feedback requirement for different types of wireless networks ?

Interestingly, the answer to this question is affirmative and this is the underlying theme of this thesis. We start by considering a long-term transmit beamforming problem for an isolated point to point MISO link, where the transmitter has no CSI about the channel statistics and the Rx has limited computational resources and restricted communication capability. We explore how the Tx can learn to beamform on-the-fly from very low-rate channel quality indicator bits fed back from the Rx (average received SNR $\gtreqless$ pre-determined threshold), while transmitting payload simultaneously. In this context, an online beamforming and learning algorithm is developed using the analytic center cutting plane method (ACCPM), and is shown to asymptotically attain optimal performance using very limited feedback requirements when there are no SNR measurement errors and feedback communication errors. A robust maximum likelihood based formulation is also proposed to address the practical scenario where there are errors and bit-flips. Simulations are used to show that the robust formulation is successful in combating these errors and enables the Tx to asymptotically converge to the optimal beamforming vector that achieves the maximum average received SNR. A variation of the robust formulation is also proposed that successfully track the variations in the channel correlation matrix.

The techniques proposed for an isolated MISO link are used as a foundation for designing joint cognitive beamforming and primary interference avoidance algorithms in underlay cognitive radio networks (CRNs). The proposed algorithms utilize the 1-bit feedback in the secondary system and the ACK/NACK feedback on the reverse primary link for designing secondary transmit beamforming vectors that maximize the SNR at the secondary Rx while limiting the interference caused at the primary Rx. Convergence of the beamforming vector to the optimal one (obtained with perfect CSIT) is proven in the case when the secondary transmitter knows the primary interference threshold. When the primary interference threshold is unknown, a power back-off mechanism is proposed to enable the secondary Tx to learn the unknown primary interference threshold. The main novelty is the ability to gradually acquire CSI and design optimal transmit beam patterns from rudimentary CSI feedback from the intended secondary Rx and indirect feedback from the primary Rx without assuming reciprocity or altering the primary's

signaling protocol, while enabling asymptotically optimal performance from only binary CSI.

The focus subsequently shifts to single group multicast beamforming which features in the Evolved Multimedia Broadcast Multicast Service (eMBMS) as a part of LTE standard. The beamformer design that maximizes the minimum SNR (determines the maximum downlink rate) is a non-convex optimization max-min fair problem which is NP-hard in general. Here, we start by assuming perfect CSIT and propose several adaptive transmit beamforming algorithms namely Additive Update (AU), Multiplicative Update (MU) and MU with Successive Linear Approximation (MU-SLA) that guarantee convergence and state-of-the-art performance at low complexity. In every iteration of the AU algorithm, the beamforming vector is updated by taking a step along the inverse-SNR weighted SNR-gradient direction of all the users, as computed using the previous iterate. This is followed by a scaling step to satisfy a transmit power constraint, and the whole procedure is repeated until the iterates converge. Convergence of the AU algorithm is derived by establishing that the corresponding update is a gradient projection of a related non-convex proportionally fair beamforming problem. Simulations show that MU-SLA outperforms the state of the art, while the AU and MU operate close to the state of the art and outperform all the other algorithms, at an order of magnitude lower complexity, thereby achieving a favorable performance-complexity trade-off.

In the absence of initial CSIT, we propose an online *cognitive multiplicative update* (CMU) algorithm for designing long-term beamforming vectors using binary channel quality feedback from every user. Two threshold selection techniques at the Tx, namely i) *multiple threshold selection* and ii) *common threshold selection*, are proposed for effectively reducing the uncertainty in the channel correlation matrices of the users in each slot. It is shown that the former reduces the uncertainty faster and converges to the true channel correlation matrices at a faster rate than the latter, at the cost of higher communication overhead. A simple modification is also proposed to completely eliminate the communication overhead in ii) by varying the transmit power. Simulations show that the CMU algorithm using the aforementioned threshold selection methods converges to the performance achieved with perfect CSIT.

The approaches used for multicast transmit beamforming are then used to tackle a much broader and interesting class of problems, namely, for obtaining feasible solutions of non-convex QCQPs with two-sided constraints when the associated matrices are positive semi-definite (NP-hard problem). Here, we propose a low-complexity algorithm named Projected

AU-SLA (Additive Update with Successive Linear Approximation) which starts with a random infeasible vector and at every step, the previous iterate is updated along the gradient of the objective function (log-barrier function comprising the non-convex constraints) evaluated using the previous iterate. The resultant vector is projected onto the set of all the convex constraints using a Cyclic Dykstra's projection algorithm which has a relatively low computational complexity. The convergence of the Gradient Projection algorithm and the Descent Lemma are used to prove the establish the convergence of the Projected AU-SLA algorithm. Simulations are used to show that the Projected AU-SLA algorithm is successful is obtaining feasible solutions with high probability at low complexity.

The fact that the proposed algorithms are able to attain the perfect CSIT performance, starting from no initial CSIT at low complexity and limited feedback requirements, making them ideal for practical implementation As radio research and development inches closer to deployment, the proposed algorithms which can work under realistic channel and feedback conditions are likely to have a big impact in terms of practical transceiver and network engineering.

Chapter 2

Transmit Beamforming for Point to Point Links using Binary CSIT

2.1 Introduction

Transmit beamforming uses multiple antennas and CSIT to steer radiated power towards directions of interest, limiting leakage in other directions [1]. The direction(s) of interest correspond to line-of-sight and specular multipath components of the propagation channel to the desired receiver(s), while limiting leakage controls interference to nearby co-channel systems. Transmit beamforming improves the QoS at the Rx while efficiently use of the available power at the Tx. The price paid is the need for accurate channel estimation at the Rx, and channel state feedback to the Tx (without channel reciprocity). In order to mitigate the communication overhead involved in feeding back instantaneous channels, an alternative is to work with the channel correlation matrix, which enables the Tx to optimize the *average* received signal to noise ratio (SNR). This is known as *long-term* transmit beamforming.

2.1.1 Prior Work and Motivation

The transmit beamforming vector that maximizes the average received SNR for a MISO link is the principal eigenvector of the channel correlation matrix [2]. This channel information is typically obtained by estimation of the channel correlation matrix at the Rx using pilot sequences transmitted by the Tx. This information is then quantized by the Rx and fed back to the Tx.

However, when channel reciprocity is assumed (i.e., the forward channel and reverse channel are conjugates of each other), then these matrices can be estimated at the Tx using the uplink transmissions of the Rx. Channel reciprocity can only be assumed in time-division duplex systems, but even there reciprocity can be a very coarse approximation, e.g., due to differences in local scattering, or when nodes use different transmit and receive beam patterns

For most types of systems, CSI is acquired at the Rx-side and fed back to the corresponding Tx. In our context, this means that each Rx is responsible for estimating the channel correlation matrix for its own link. Instead of feeding back the correlation matrix, each Rx may compute the optimal beamforming vector (i.e., its principal eigenvector, e.g., via the power method) and send this back to the Tx. Either way, scalar or vector quantization is needed to limit the feedback rate. The beamforming vector can be quantized using a custom codebook, see [3] where bounds on the codebook size needed for a given SNR loss are given. The approach in [3] was developed for instantaneous feedback, but it can be extended [3] to long-term feedback. A rate-distortion analysis of vector quantization performance in this context can be found in [4], which quantifies the loss in the SNR performance as a function of the size of the codebook. In [3] and [4], the codebook is assumed to be designed off-line and shared between the Tx and Rx beforehand; this is not suitable when the Tx and Rx are opportunistically paired. Furthermore, the codebook is usually optimized for a particular probability distribution. Therefore, the codebook needs to be re-designed when the channel distribution changes.

What if the receivers cannot perform correlation matrix estimation, summarization, and feedback? This can happen if they have limited computation and communication capabilities (low cost, small size and battery), when they are hard-wired for legacy protocols, or when they are opportunistically paired with a transmitter, with limited negotiation before payload transmission. Is it possible to acquire accurate CSI at the transmitter with only rudimentary feedback from the receivers, e.g., acknowledgment / negative-acknowledgement (ACK/NACK)-type? Is it possible to design effective transmit beamforming solutions this way and still match the performance in the case when the Tx has perfect CSI?

For a point-to-point MISO channel, Mudumbai *et al.* [5] and Banister *et al.* [6] have proposed transmit weight vector adaptation algorithms using a series of 1-bit feedbacks from the Rx. In both the cases, the Tx periodically updates and stores the beamforming vector $\mathbf{w_{best}}(t)$, that resulted in the maximum SNR at the Rx until the current time slot t and the Rx updates and stores the maximum received SNR till t denoted by $SNR_{best}(t)$. At the start of time slot $(t+1)$,

the Tx perturbs $\mathbf{w_{best}}(t)$ to obtain the beamforming vector $\mathbf{w}(t+1)$ and uses it to send data to the Rx. The Rx measures the corresponding received SNR, compares it with $SNR_{best}(t)$ and reports a '1' or a '0' to the Tx depending on whether the SNR at current time slot is $\geq$ or $< SNR_{best}$. Upon receiving a '1', the Tx updates $\mathbf{w_{best}}(t+1)$ with $\mathbf{w}(t+1)$ or with $\mathbf{w_{best}}(t)$ otherwise. The difference between [5] and [6] is in the way $\mathbf{w_{best}}(t)$ is perturbed using the 1-bit feedback at time t to obtain $\mathbf{w}(t+1)$. In [5], all entries of the beamforming vector have fixed magnitude and equal to $\sqrt{\frac{1}{N}}$, (N = number of Tx antennas) and the phase of every entry is randomly perturbed each time; while in [6], the 1-bit feedback from the Rx at time t is used by the Tx to update a coarse estimate of the gradient of the received SNR, which is subsequently used for perturbing $\mathbf{w_{best}}(t)$ and then scaled to the available Tx power. It is assumed that the channel remains fixed or changes very slowly during this adaptation process. Asymptotic convergence to the optimal beamforming vector has been shown for both algorithms when the channel remains fixed. Simulations show that these random exploration algorithms result in a slow convergence rate.

Xu *et al.* propose an approach [7] similar to the algorithm proposed in this chapter, where transmit beamforming for wireless energy transfer (vs. communication) is considered, and a cutting plane method to learn the channel correlation matrix from one-bit feedback. While several design choices are naturally different, these being independent pieces of work, the core idea is the same in both papers. Summarizing the differences, [7] assumes separate learning and 'bulk transfer' phases, uses higher-rank precoding instead of beamforming during the learning stage, and does not communicate thresholds to the receiver - at the cost of not controlling the transmission power during the learning phase. Using a higher-rank precoder may enable faster exploration and learning, but requires separate up-conversion chains that are not needed during show time (the payload phase) where beamforming is used.

2.1.2 Contributions

We begin by considering the long-term transmit beamforming problem for a MISO link in the case when the Rx has limited computational capabilities, and/or is paired up opportunistically with the Tx. We explore how the Tx can learn to beamform on-the-fly from very low-rate channel quality indicator bits fed back from the Rx (average received SNR $\gtrless$ pre-determined threshold), while transmitting payload simultaneously. The beamforming vectors are designed such that they not only exploit the acquired information gathered in the past to maximize a Tx-side

estimate of the average received SNR, but are also diverse enough to explore the channel correlation space efficiently and learn the channel correlation matrix accurately over time. Towards this end, the analytic center cutting plane method (ACCPM) from optimization is leveraged to develop an online channel correlation matrix learning algorithm based on one-bit SNR feedback. In the absence of binary measurement or feedback communication errors, the proposed algorithm restricts the channel correlation estimate to a ball of radius r centered around the true value, within $\mathcal{O}\left(\frac{N^2}{r^2}\right)$ iterations (where N is the number of transmit antennas) and the average received SNR converges asymptotically to the maximum achievable SNR value (obtained with perfect knowledge of the channel correlation matrix at the Tx) [8].

For the practically important case where there are occasional SNR measurement errors at the receiver, or binary feedback errors in the reverse link from the receiver to the transmitter, a robust maximum likelihood formulation is proposed and shown to be effective in dealing with such errors. A discounted maximum likelihood formulation is also proposed to enable CSI tracking and adaptation of the transmit beamforming vector in cases where the channel correlation matrix itself changes slowly, as time goes by.

2.2 System Model and Problem Formulation

Consider a point-to-point MISO link comprising a Tx with N antennas and a Rx with a single antenna. Time is divided into transmission rounds or *slots* of length T seconds, with each slot comprising enough symbols for the Rx to perform relatively accurate power estimation. Initially, the Tx starts transmitting data using an arbitrary beamforming vector $\mathbf{w}_0$. At time $tT+\tau$, where t is a 'slow time' (slot index) and τ is 'fast time', the channel from the Tx to the Rx is modeled as a complex random $N \times 1$ vector $\mathbf{h}(tT+\tau)$, with $E\left[\mathbf{h}(tT+\tau)\mathbf{h}^H(tT+\tau)\right] = \mathbf{R_h}$, $\forall t$, $\forall \tau$. At the same time, the Tx sends the complex zero-mean unit-variance symbol $x(tT+\tau)$ times a complex beamforming vector $\mathbf{w}_t$, and the Rx measures

$$y(tT+\tau) = \mathbf{w}_t^H \mathbf{h}(tT+\tau)x(tT+\tau) + z(tT+\tau), \tag{2.1}$$

where the additive noise $z(\cdot)$ is a complex random variable zero mean, variance σ^2, and is independent of $x(\cdot)$ and $\mathbf{h}(\cdot)$. In order to decode the data, the Rx should at least estimate $\mathbf{w}_t^H \mathbf{h}(tT+\tau)$. This can be accomplished using a few pilot symbols per slot (or differential modulation/demodulation), and it is far simpler and less complex than estimating the vector

$\mathbf{h}(tT+\tau)$. The average received SNR for slot t is given by

$$E\left(\frac{|\mathbf{w}_t^H\mathbf{h}|^2}{\sigma^2}\right) = \frac{\mathbf{w}_t^H\mathbf{R}_\mathbf{h}\mathbf{w}_t}{\sigma^2} \tag{2.2}$$

The beamforming vector that maximizes the average received SNR in (2.2) is the principal eigenvector of $\mathbf{R}_\mathbf{h}$ scaled according to the available transmit power. It is assumed that the Tx does not have any initial CSI and its objective is to learn $\mathbf{R}_\mathbf{h}$ and maximize the average received SNR based on binary CSIT - that is, binary slot-average SNR feedback from the Rx. More specifically, in each time slot t, the Rx estimates the average SNR and compares it with a threshold γ_t. A '1' is fed back to the Tx if the average SNR is $\geq \gamma_t$ and a '0' is fed back otherwise. It is initially assumed that there are no errors while measuring the average SNR at the Rx or during communication of the 1-bit feedback from the Rx to the Tx. Based on the 1-bit feedback at time t, the Tx infers the following information about $\mathbf{R}_\mathbf{h}$

$$\begin{cases} \mathbf{w}_t^H\mathbf{R}_\mathbf{h}\mathbf{w}_t \geq \gamma_t, & \text{when } s_t = 1; \text{ or} \\ \mathbf{w}_t^H\mathbf{R}_\mathbf{h}\mathbf{w}_t < \gamma_t, & \text{when } s_t = 0, \end{cases} \tag{2.3}$$

where s_t is the 1-bit feedback at time t. It should be noted that the inference at the Tx at time t is a linear inequality in $\mathbf{R}_\mathbf{h}$. For every feedback bit, the Tx learns an additional linear inequality, which can reduce the uncertainty about $\mathbf{R}_\mathbf{h}$. This naturally raises the question whether an appropriate choice of $\{\mathbf{w}_t, \gamma_t\}$ can quickly shrink the feasible region for $\mathbf{R}_\mathbf{h}$, and even yield a sequence of estimates $\hat{\mathbf{R}_\mathbf{h}}(t) \to \mathbf{R}_\mathbf{h}$, as $t \to \infty$. More importantly, is it possible to approach the SNR attained with full CSIT, i.e., using the principal eigenvector of $\mathbf{R}_\mathbf{h}$? While this may seem an ambitious goal with only rudimentary CSIT, we will see that the answer is on the affirmative, and in fact relatively few feedback bits suffice to approach optimum performance, for practical purposes.

Simultaneous exploration - exploitation: For every slot t, the Tx can choose $\mathbf{w}_t$ in such a way that it not only gathers a significant amount of information about $\mathbf{R}_\mathbf{h}$ (from the 1-bit feedback), but also tries to deliver a high average received SNR, thus enabling channel learning in parallel with payload transmission. To accomplish the former objective, the beamforming vectors chosen at each instant should be as diverse as possible relative to the previously chosen weight vectors, so that, over time, the Tx will learn about $\mathbf{R}_\mathbf{h}$ from as many different directions as possible and gain significantly new information at every time instant t as compared to all the previous time slots. For the latter, the best that the Tx can do to deliver a high average

received SNR is to assume that $\hat{\mathbf{R}}_\mathbf{h}$ is close to $\mathbf{R}_\mathbf{h}$ and choose the beamforming weight vector along the direction of the principal eigenvector of $\hat{\mathbf{R}}_\mathbf{h}$. Since the Tx does not have any CSI to start with, initially it has to give preference towards choosing weight vectors that aggressively explore the channel correlation space to improve the accuracy of $\hat{\mathbf{R}}_\mathbf{h}$; and as time progresses, slowly shift emphasis towards beamforming vectors in the direction of the principal eigenvector of $\hat{\mathbf{R}}_\mathbf{h}$. This ensures that as $\hat{\mathbf{R}}_\mathbf{h}$ approaches $\mathbf{R}_\mathbf{h}$ (as will be shown later), $\mathbf{w}_t$ will approach the direction of the principal eigenvector of $\mathbf{R}_\mathbf{h}$, thus attaining the maximum average received SNR that is achievable with perfect knowledge of $\mathbf{R}_\mathbf{h}$ at the Tx.

At the end of slot t, the Tx has learned the following set of linear inequalities about $\mathbf{R}_\mathbf{h}$ from the t received feedback bits from the Rx.

$$\mathbf{w}_i{}^H \mathbf{R}_\mathbf{h} \mathbf{w}_i \geq \gamma_i, \quad \forall i \in \mathcal{G}_1 \tag{2.4}$$

$$\mathbf{w}_i{}^H \mathbf{R}_\mathbf{h} \mathbf{w}_i < \gamma_i, \quad \forall i \in \mathcal{G}_2 \tag{2.5}$$

where $\mathcal{G}_1 = \{i \ : 1 \leq i \leq t, \ s_i = 1\}$, $\mathcal{G}_2 = \{i \ : 1 \leq i \leq t, \ s_i = 0\}$, $\mathcal{G}_1 \bigcup \mathcal{G}_2 = \{1, 2, \ldots, t\}$ and t is the number of elapsed time slots. These inequalities form a linear polytope with $\mathbf{R}_\mathbf{h}$ as an interior point. The volume of this polytope is a measure of uncertainty about $\mathbf{R}_\mathbf{h}$ at the Tx at time t.

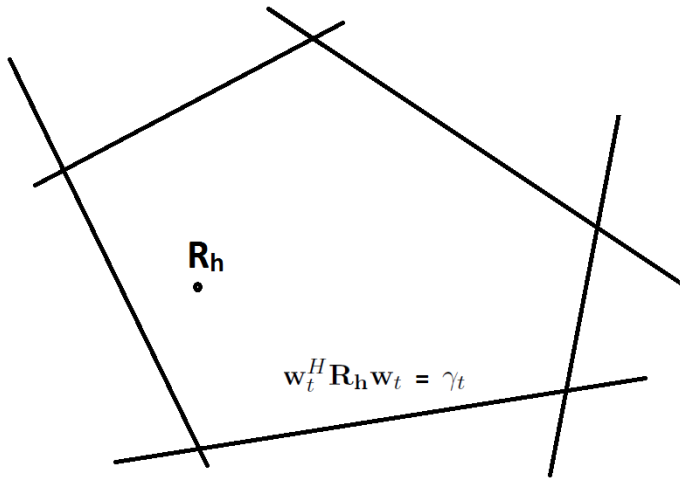

Figure 2.1: **Uncertainty region of $\mathbf{R}_\mathbf{h}$ at time t formed by the linear inequalities inferred by the Tx**

The proposed algorithm has two steps for every time slot. Consider the time slot (t). First of all, the Tx updates its estimate $\hat{\mathbf{R}}_{\mathbf{h}}(t-1)$ (the Tx-side estimate of $\mathbf{R_h}$ at time $t-1$) using the inequality inferred from the the feedback bit s_t to obtain the updated estimate $\hat{\mathbf{R}}_{\mathbf{h}}(t)$. Subsequently, it uses the estimate to design the beamforming vector $\mathbf{w}_{t+1}$ and the threshold γ_{t+1} for the next time slot.

2.2.1 $\hat{\mathbf{R}}_{\mathbf{h}}(t)$ update

We propose to update $\hat{\mathbf{R}}_{\mathbf{h}}(t)$ as follows.

$\mathbf{\Pi_1}$

$$\hat{\mathbf{R}}_{\mathbf{h}}(t) = \arg\max_{\mathbf{R_h}} \sum_{i\in\mathcal{G}_1} \log\left(\text{Tr}\left(\mathbf{W}_i\mathbf{R_h}\right) - \gamma_i\right) + \sum_{j\in\mathcal{G}_2} \log\left(\gamma_j - \text{Tr}\left(\mathbf{W}_j\mathbf{R_h}\right)\right) + \log\det\mathbf{R_h}$$

where $\mathbf{W}_i = \mathbf{w}_i\mathbf{w}_i^H$ and the term ${\mathbf{w}_i}^H\mathbf{R_h}\mathbf{w}_i$ has been rewritten as Tr $(\mathbf{W}_i\mathbf{R_h})$. $\mathbf{\Pi_1}$ is a convex optimization problem (maximization of a concave objective function) which obtains the analytic center of the feasible region at time slot t formed by the linear inequalities (2.4)-(2.5) and the positive semi-definite cone [9] [10]. The $\log\ \det$ term in the objective ensures that the solution of $\mathbf{\Pi_1}$ has positive semi-definite (i.e., positive eigenvalues). $\mathbf{\Pi_1}$ can be solved efficiently using interior point methods with worst case complexity $\mathcal{O}(N^7)$.

2.2.2 Design of beamforming vector $\mathbf{w}_{t+1}$ and threshold γ_{t+1}

• *Design of beamforming vector* $\mathbf{w}_{t+1}$: After updating $\hat{\mathbf{R}}_{\mathbf{h}}(t)$, we formulate an optimization problem to select $\mathbf{w}_{t+1}$ as shown.

$$\mathbf{\Pi_2}: \quad \mathbf{w}_{t+1} = \arg\max_{\|\mathbf{w}\|=1} \mathbf{w}^H\hat{\mathbf{R}}_{\mathbf{h}}(t)\mathbf{w} - \lambda_t\mathbf{w}^H\mathbf{V}_{w,t}\mathbf{V}_{w,t}^H\mathbf{w}$$

where $\mathbf{V}_{w,t} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t]$, and λ_t is a non-increasing function of t e.g., $\lambda_t = \frac{\lambda}{\lceil 0.1t\rceil}$, with $\lambda_1 \gg 1$. The solution of $\mathbf{\Pi_2}$ can be obtained in closed form i.e., $\mathbf{w}_{t+1}$ is the unit vector along the principal eigenvector (eigenvector corresponding to the eigenvalue of largest magnitude) of the matrix $\left(\hat{\mathbf{R}}_{\mathbf{h}}(t) - \lambda_t\mathbf{V}_{w,t}\mathbf{V}_{w,t}^H\right)$. The objective function in $\mathbf{\Pi_2}$ consists of two terms, the first one is proportional to the Tx-side estimate of the average received SNR (which is close to the actual average received SNR if the Tx has estimated $\hat{\mathbf{R}}_{\mathbf{h}}(t)$ close to $\mathbf{R_h}$), and the second one is the squared norm of the vector $\mathbf{V}_{w,t}^H\mathbf{w}$ whose i^{th} entry is the scalar dot-product of $\mathbf{w}$ with

$\mathbf{w}_i,\ i = 1, 2, ..., t$. The relative importance of each term towards the decision of $\mathbf{w}_{t+1}$ is determined by the scalar λ_t (varies with t). Maximization of this objective function gives a weight vector that strikes a balance between maximizing the estimated average received SNR and minimizing similarity to the weight vectors chosen in previous time slots. For small t, $\lambda_t \gg 1$ and the choice of weight vector is dictated by $\mathbf{w}^H \mathbf{V}_{w,t} \mathbf{V}_{w,t}^H \mathbf{w}$, yielding diverse weight vectors that explore different directions, gathering new information about $\mathbf{R_h}$ for every t; for large t, $\lambda_t \ll 1$ and preference shifts to the first term $\mathbf{w}^H \hat{\mathbf{R}}_\mathbf{h}(t) \mathbf{w}$, resulting in weight vectors aligned with the principal eigenvector of $\hat{\mathbf{R}}_\mathbf{h}(t)$. Hence the choice of λ_t is a non-increasing function of t with $\lambda_1 \gg 1$ allows the Tx to switch from exploration initially (to learn $\mathbf{R}_h$ when it has not initial CSI) to exploitation as time progresses (when the Tx gathers sufficient information to estimate $\hat{\mathbf{R}}_\mathbf{h}(t)$ accurately close to $\mathbf{R_h}$). Therefore, if $\hat{\mathbf{R}}_\mathbf{h}(t) \to \mathbf{R_h}$ as $t \to \infty$ and $\lambda_t \to 0$, the beamforming vector chosen by the Tx will asymptotically align itself with the principal eigenvector of $\mathbf{R_h}$, thus attaining the maximum average received SNR.

• *Design of threshold γ_{t+1} using Analytic Center Cutting Plane Method (ACCPM)*: After choosing $\mathbf{w}_{t+1}$, the Tx selects an appropriate SNR threshold γ_{t+1} such that the subsequent inequality constraint for $\mathbf{R_h}$ obtained from the 1-bit feedback at time slot $t+1$ considerably reduces the uncertainty region of $\mathbf{R_h}$ at time t denoted by $\mathcal{P}_t$, where $\mathcal{P}_t = \{\mathbf{R} : \mathbf{R} \succeq 0, {\mathbf{w}_i}^H \mathbf{R} \mathbf{w}_i \geq \gamma_i,\ \forall i \in \mathcal{G}_1, {\mathbf{w}_i}^H \mathbf{R} \mathbf{w}_i < \gamma_i,\ \forall i \in \mathcal{G}_2,\ \mathcal{G}_1 \bigcup \mathcal{G}_2 = \{1, 2, \ldots, t\}\}$. This is crucial for the convergence of $\hat{\mathbf{R}}_\mathbf{h}(t)$ to $\mathbf{R_h}$. The threshold should be communicated to the Rx so that it can compare the SNR and sent the feedback bit to the Tx which subsequently improves the estimation accuracy of $\hat{\mathbf{R}}_\mathbf{h}(t)$ at the Tx. Since the Tx already communicates payload information to the Rx in parallel to learning to beamform, the new threshold can 'piggyback' on the payload transmission at limited overhead - unlike the Rx feedback on the reverse link, which is more severely limited in terms of rate. (The basic method still works without having the Tx dictate thresholds to the Rx, albeit convergence to the true channel correlation matrix cannot be guaranteed in this case). Later in this chapter, we also propose an alternative to eliminate the threshold communication at the cost of dynamic transmit power variation at the Tx.

One way to ensure that the feasible region is reduced at each time step is to choose $\mathbf{w}_{t+1}$ and γ_{t+1}, such that the resulting hyperplane ${\mathbf{w}_{t+1}}^H \mathbf{R} \mathbf{w}_{t+1} = \gamma_{t+1}$ passes through an interior point of $\mathcal{P}_t$. Here, we propose to design the beamforming vector $\mathbf{w}_{t+1}$ and the threshold γ_{t+1} such that the resulting hyperplane passes through the analytic center of $\mathcal{P}_t$ (Analytic Center Cutting

Plane Method - ACCPM), which is $\hat{\mathbf{R}}_\mathbf{h}$. Since the analytic center is the point that maximizes the product of distances to the defining hyperplanes and the positive semi-definite cone, it gives the deepest interior point of $\mathcal{P}_t$. This choice ensures that there is a significant reduction of the uncertainty of $\mathbf{R_h}$ irrespective of the orientation of the beamforming vector $\mathbf{w}_{t+1}$. Hence for a given $\mathbf{w}_{t+1}$, we choose

$$\gamma_{t+1} = {\mathbf{w}_{t+1}}^H \hat{\mathbf{R}}_\mathbf{h}(t)\mathbf{w}_{t+1}. \tag{2.6}$$

This ensures that the resulting cutting plane ${\mathbf{w}_{t+1}}^H \mathbf{R}\mathbf{w}_{t+1} = \gamma_{t+1}$ will pass through $\hat{\mathbf{R}}_\mathbf{h}(t)$ and cut off a significant part of the current uncertainty region $\mathcal{P}_t$.

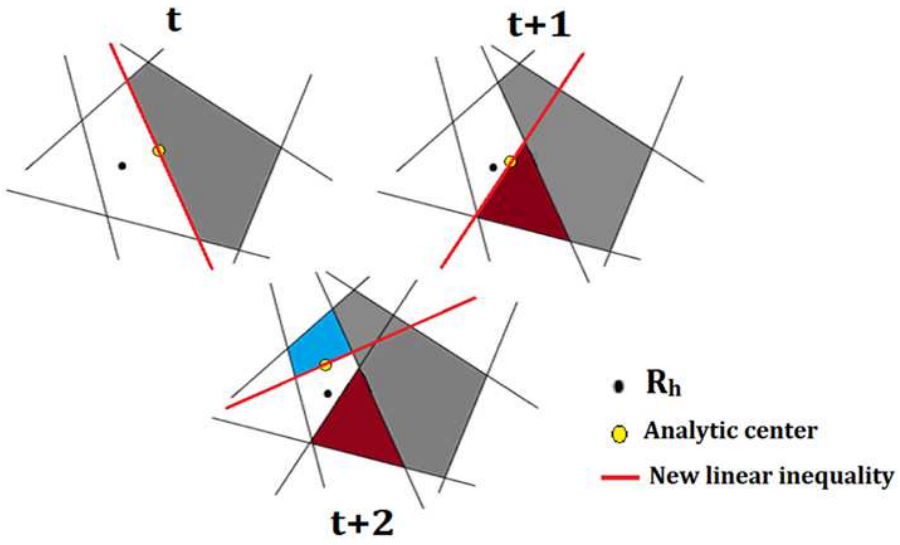

Figure 2.2: **Illustration of the reduction of the uncertainty region of R_h using ACCPM**

From known convergence results for ACCPM [10],[11], it follows that $\hat{\mathbf{R}}_\mathbf{h}(t)$ (updated as the analytic center of the current feasible region) is restricted to a ball of radius r around $\mathbf{R_h}$ within $\mathcal{O}\left(\frac{N^2}{r^2}\right)$ iterations. Therefore, if λ_t is designed so that it becomes negligible by $\lceil \frac{N^2}{r^2} \rceil$ iterations, then the objective function of $\mathbf{\Pi_2}$ can be approximated as $\mathbf{w}^H \hat{\mathbf{R}}_\mathbf{h}(t)\mathbf{w}$. Hence asymptotically, as $\hat{\mathbf{R}}_\mathbf{h}(t) \rightarrow \mathbf{R_h}$, the beamforming weight vector will converge to the principal eigenvector of $\mathbf{R_h}$ and the average received SNR will approach the maximum achievable average SNR (obtained with perfect *a priori* knowledge of $\mathbf{R_h}$).

One can avoid the threshold communication overhead by fixing the threshold at the receiver,

and scaling the transmit beamforming vector instead, as shown below.

$$\mathbf{w}_t{}^H \mathbf{R_h} \mathbf{w}_t \gtreqless \gamma_t \rightarrow \tilde{\mathbf{w}}_t^H \mathbf{R_h} \tilde{\mathbf{w}}_t \gtreqless 1 \tag{2.7}$$

where $\tilde{\mathbf{w}}_\mathbf{t} = \frac{\mathbf{1}}{\gamma_\mathbf{t}} \mathbf{w_t}$. This enables the Rx to always compare the average SNR against a fixed threshold, thereby eliminating the threshold communication overhead. However, this relinquishes control of transmission power and requires the power amplifier at the Tx to operate in the linear region over a wider range of transmit power values. Threshold communication appears appealing from a practical point of view, taking into account the limited dynamic range of the receiver front-end, and power amplifier nonlinearities.

2.3 Maximum Likelihood Formulation

In practice, bits may be flipped due to inaccurate SNR estimation at the Rx or communication errors in the reverse link. Assuming a memoryless feedback link, these errors are independent from slot to slot. We propose modeling both types of errors using an additive measurement noise model that is equivalent *from the point of view of the Tx*. Including the effect of this noise in our model, the inequalities become

$$\begin{cases} \mathbf{w}_t^H \mathbf{R_h} \mathbf{w}_t + n_t \geq \gamma_t, & \text{when } s_t = 1, \text{ or} \\ \mathbf{w}_t^H \mathbf{R_h} \mathbf{w}_t + n_t < \gamma_t, & \text{when } s_t = 0, \end{cases} \tag{2.8}$$

where $n_t \sim \mathcal{N}(0, \sigma_n^2)$ is the equivalent noise at time t. The measurements received at the Tx are the bits $s_t = \left(\text{sign}\left(\mathbf{w}_t^H \mathbf{R_h} \mathbf{w}_t + n_t - \gamma_t\right) + 1\right)/2$. The conditional likelihood of these bits $s_1, s_2, \ldots, s_t$ conditioned on the unknown parameter $\mathbf{R_h}$ can be written as

$$\begin{aligned} f(\mathbf{s}_t|\mathbf{R_h}) &= \prod_{i \in \mathcal{G}_1} \Pr\left[\text{Tr}(\mathbf{W}_i \mathbf{R_h}) + n_i \geq \gamma_i\right] \prod_{i \in \mathcal{G}_2} \Pr\left[\text{Tr}(\mathbf{W}_i \mathbf{R_h}) + n_i < \gamma_i\right] \\ &= \prod_{i \in \mathcal{G}_1} \Phi\left(\frac{\text{Tr}(\mathbf{W}_i \mathbf{R_h}) - \gamma_i}{\sigma_n}\right) \prod_{i \in \mathcal{G}_2} \Phi\left(\frac{\gamma_i - \text{Tr}(\mathbf{W}_i \mathbf{R_h})}{\sigma_n}\right), \end{aligned}$$

where $\mathbf{s}_t = [s_1, s_2, \ldots, s_t]^T$, $\mathcal{G}_1$ and $\mathcal{G}_2$ are defined as before, and $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{z^2}{2}} dz$ is the standard Gaussian c.d.f. At time slot $t+1$, $\hat{\mathbf{R}}_\mathbf{h}(t)$ is updated as the maximum likelihood estimate (MLE) $\hat{\mathbf{R}}_\mathbf{h}^{MLE}(t)$ (as compared to the analytic center in the error-free case) obtained

from $\mathbf{\Pi_3}$ which maximizes the log-likelihood function $\log\left(f(\mathbf{s}_t|\mathbf{R_h})\right)$ with a positive semi-definite constraint.

$\mathbf{\Pi_3}$

$$\hat{\mathbf{R}}_{\mathbf{h}}^{MLE}(t) = \arg\max_{\mathbf{R_h}\succeq 0} \sum_{i\in\mathcal{G}_1} \log\Phi\left(\frac{\mathrm{Tr}(\mathbf{W}_i\mathbf{R_h})-\gamma_i}{\sigma_n}\right) + \sum_{i\in\mathcal{G}_2} \log\Phi\left(\frac{\gamma_i-\mathrm{Tr}(\mathbf{W}_i\mathbf{R_h})}{\sigma_n}\right)$$

$\mathbf{\Pi_3}$ is a convex optimization problem since it involves the maximization of the logarithm of the c.d.f. of a Gaussian distribution which is concave, with a positive semi-definite constraint which is convex. Once the channel correlation matrix estimate is updated, $\mathbf{w}_{t+1}$ is chosen as the principal eigenvector of $\hat{\mathbf{R}}_{\mathbf{h}}^{MLE}(t) - \lambda_t \mathbf{V}_{w,t}\mathbf{V}_{w,t}^H$ and $\gamma_{t+1} = \mathrm{Tr}\left(\mathbf{W}_{t+1}\hat{\mathbf{R}}_{\mathbf{h}}^{MLE}(t)\right)$, where $\mathbf{V}_{w,t} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t]$ and λ_t is a non-increasing function of t.

Our 1-bit/slot measurement model is a special case of what is known as a *probit model*. Statistical identifiability and MLE consistency conditions and proof for the probit model can be found in [12], and a more compact proof for a generalized model can be found in [13]. The basic idea is that, by the law of large numbers, the normalized log-likelihood function will converge to its expectation, and by the information inequality this will have a unique maximum at the true parameter when this is identifiable. However, the proof assumes that the regressors $\mathrm{vec}(\mathbf{W}_t^T)$ are independently drawn from a distribution with nonsingular $E[\mathrm{vec}(\mathbf{W}_t^T)\mathrm{vec}(\mathbf{W}_t^T)^H]$. A random model for $\mathrm{vec}(\mathbf{W}_t^T)$ is needed to invoke the information inequality. In our context, however, the $\mathrm{vec}(\mathbf{W}_t^T)$'s are iteratively generated - in fact, judiciously designed - based on interim ML estimates of the sought channel correlation matrix, according to the proposed exploration - exploitation trade-off schedule. In certain cases, one can prove consistency of the MLE designed for independent and identically distributed (i.i.d.) data, but operating on non-i.i.d. data [14, 15]. We do not have proof of convergence and consistency of the MLE in our context, but our experiments indicate that the MLE approaches the true $\mathbf{R_h}$ as the number of feedback bits increases.

2.3.1 Tracking changes in $\mathbf{R_h}$

When the channel correlation matrix $\mathbf{R_h}$ changes over time due to mobility, the Tx should be capable of tracking these changes and adapting its beamforming vector to maintain high average SNR at the Rx. Assuming that $\mathbf{R_h}$ changes slowly with time, it is natural to consider

the following 'discounted' modification of the MLE in $\mathbf{\Pi_3}$.

$\mathbf{\Pi_{3'}}$

$$\hat{\mathbf{R}}_{\mathbf{h}}^{dMLE}(t) = \arg\max_{\mathbf{R_h}\succeq 0} \sum_{i\in\mathcal{G}_1} \beta^{t-i} \log\Phi\left(\frac{\text{Tr}(\mathbf{W}_i\mathbf{R_h}) - \gamma_i}{\sigma_n}\right) + \sum_{j\in\mathcal{G}_2} \beta^{t-j} \log\Phi\left(\frac{\gamma_j - \text{Tr}(\mathbf{W}_j\mathbf{R_h})}{\sigma_n}\right)$$

where $0 < \beta < 1$ and $\mathcal{G}_1 \bigcup \mathcal{G}_2 = \{1, 2, \ldots, t\}$. Each term inside the summation of the objective function is weighted by a forgetting factor that decays exponentially with time. As a result, the terms corresponding to inequalities obtained from the recent past are given higher weight. Therefore, after $\mathbf{R_h}$ changes, the Tx will learn about the new channel correlation matrix since it will give more weight to the inequalities from the recent past (i.e., from the new $\mathbf{R_h}$) and will give less consideration to the inequalities resulting from the old $\mathbf{R_h}$ (as a result of using the forgetting factor).

2.4 Simulation Results

For simulation purposes, the channel vector $\mathbf{h}$ is drawn independently during each time slot from a complex normal distribution with zero mean and covariance matrix $\mathbf{R_h}$ and $\lambda_t = \frac{5}{\lceil 0.1t \rceil}$. The channel correlation matrix $\mathbf{R_h}$ was obtained by generating a random orthonormal matrix $\mathbf{U}$, a random diagonal matrix $\mathbf{D}$ with positive real numbers along the main diagonal, and setting $\mathbf{R_h} = \mathbf{UDU}^H$.

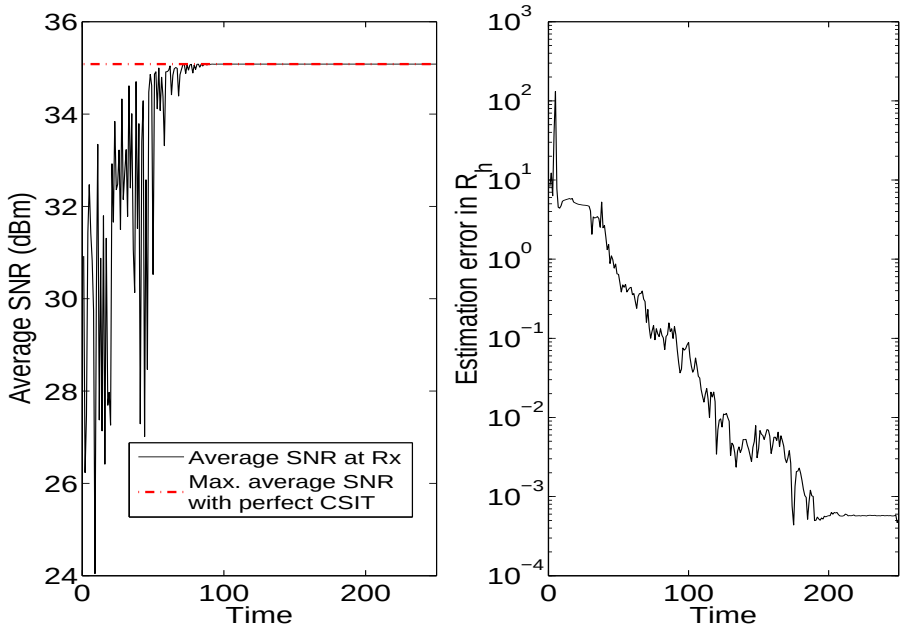

Figure 2.3: Evolution of average link SNR (left) and $\|\mathbf{R_h} - \hat{\mathbf{R}}_\mathbf{h}\|_F$ (right) for ACCPM, $N = 5$.

Fig. 2.3 shows simulation results for the average link SNR and the estimation error $\|\mathbf{R_h} - \hat{\mathbf{R}}_\mathbf{h}\|_F$ for a point to point MISO link using $N = 5$ transmit antennas, using ACCPM for a single channel realization. The x-axis is labeled (slow) 'time', i.e., the slot index.The dotted line in the figure on the left represents the maximum achievable average SNR with perfect knowledge of $\mathbf{R_h}$ at the Tx. It can be seen The solid line represents the average received SNR at each time t.It can be seen that the received SNR attained using the proposed algorithm starts from no initial CSIT and converges to the maximum SNR obtained using perfect CSIT. It takes approximately 80 time slots (or (80/(5(5+1)/2)) $\approx$ 6 feedback bits per complex entry or $\approx$ 3 bits per real entry

of $\mathbf{R_h}$) for the algorithm to converge to the maximum achievable SNR.

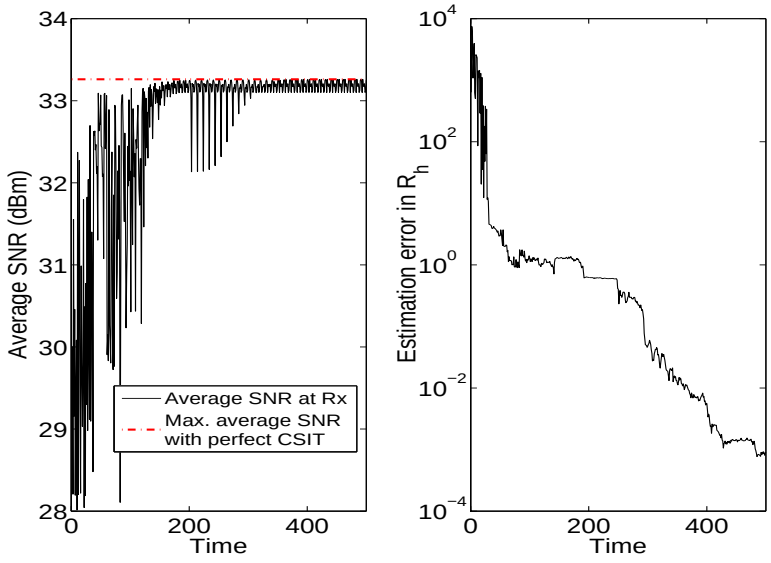

Figure 2.4: Evolution of average link SNR (left) and $\|\mathbf{R_h} - \hat{\mathbf{R}}_\mathbf{h}\|_F$ (right) for ACCPM, $N = 10$.

Fig. 2.4 shows corresponding results for $N = 10$ transmit antennas. Again, it can be clearly seen that the proposed algorithm converges to the maximum achievable SNR. However, the time taken by the algorithm to converge to the maximum achievable SNR at the Rx increases as N increases ($\approx$ 200 time slots for N = 10, or 7 bits per complex entry of $\mathbf{R_h}$). Furthermore, it can be seen from the figures on the right that the channel correlation matrix estimation error also decreases with t. Fig. 2.5 plots the Monte-Carlo simulation of the average link SNR for $N = 5$ by averaging over 100 random realizations of $\mathbf{R_h}$.

Fig. 2.6 compares the average SNR performance of the a) proposed ACCPM algorithm with b) the distributed beamforming algorithm in [5], b) the gradient sign algorithm in [6], and the one-bit null space learning algorithm proposed by Noam *et al.* in [16], for an isolated MISO link with $N = 5$ transmit antennas. [16] proposed an algorithm that enables the secondary Tx to learn the *fixed* interference channel to the primary Rx, by measuring a monotonic function of the interference to the primary Rx or overhearing the ACK/NACK feedback [16] in the reverse primary link. They proposed to vary the secondary transmit precoding matrix to probe the primary Rx, gradually collecting information on what it can tolerate. They proposed using a

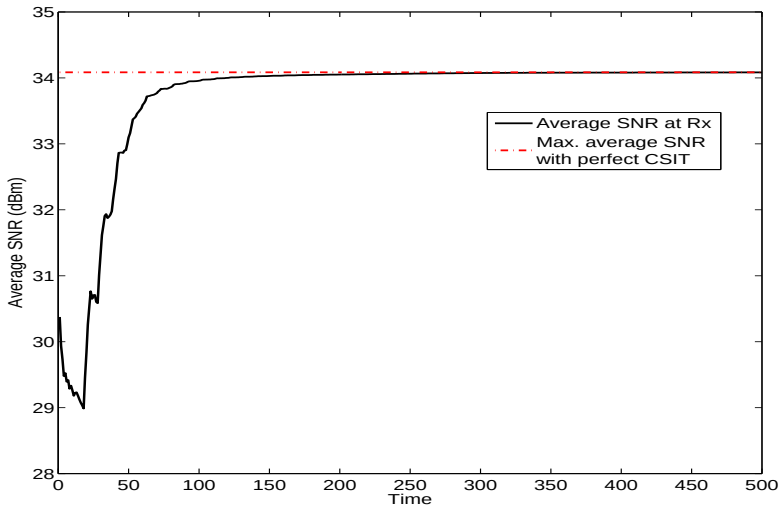

Figure 2.5: Monte Carlo simulation for evolution of average link SNR in isolated MISO link, $N = 5$.

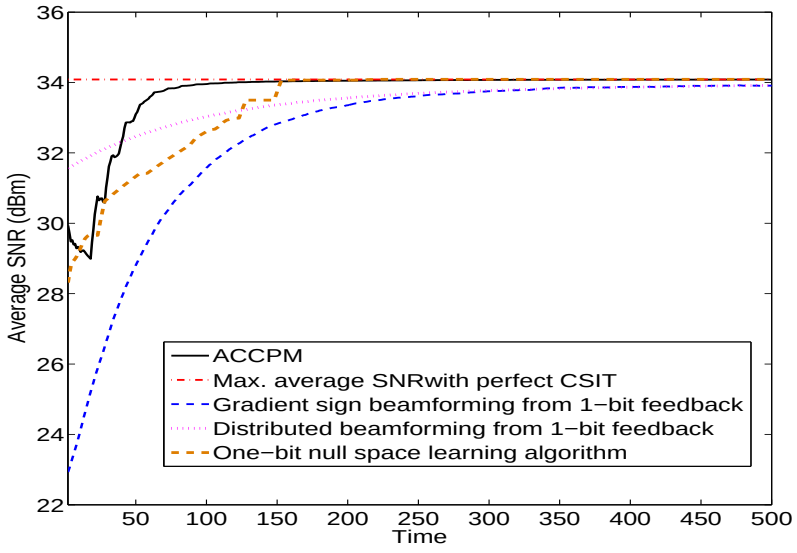

Figure 2.6: Performance comparison of average link SNR in isolated MISO link for a) proposed ACCPM algorithm, b) distributed beamforming from 1-bit feedback, c) gradient sign algorithm, and d) one-bit null space learning algorithm, $N = 5$.

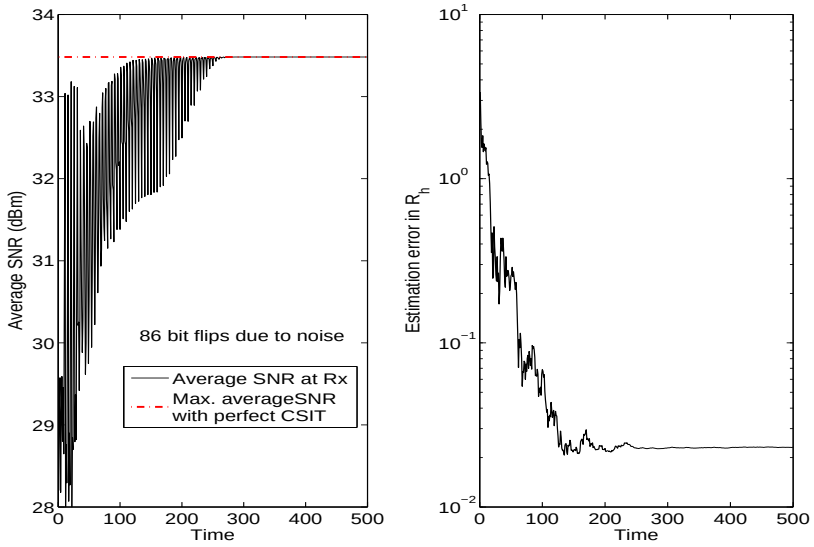

Figure 2.7: Evolution of average link SNR (left) and $\|\mathbf{R_h} - \hat{\mathbf{R}}_\mathbf{h}\|_F$ (right) for MLE, $N = 5$.

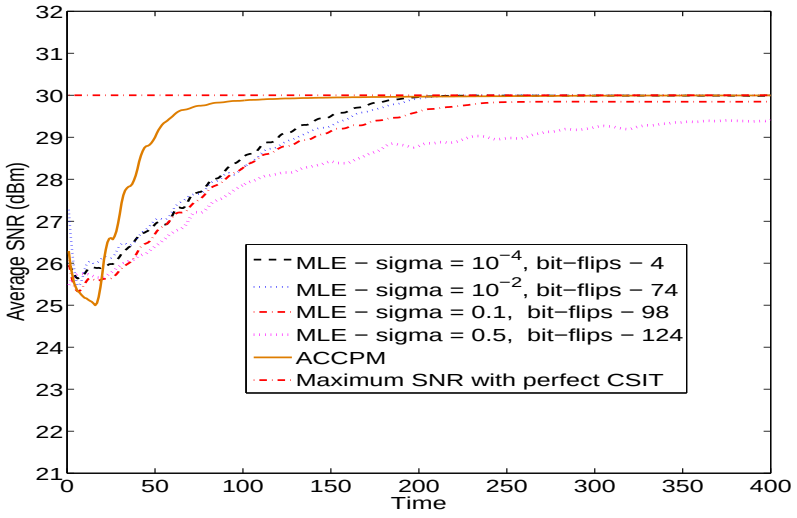

Figure 2.8: Performance comparison of average SNR using MLE update for different σ_n, $N = 5$.

cyclic Jacobi subspace estimation algorithm, and proved that it converges to the interference channel between the secondary Tx and primary Rx. The secondary Tx can indirectly learn the best signaling subspace this way, without assuming channel reciprocity, and without altering the primary communication protocol – an exciting development.

The figure reports the results averaged over 100 Monte Carlo draws of $\mathbf{R_h}$. It can be seen that the proposed algorithm converges much faster to the maximum SNR than the other algorithms (100 versus 500, 500 and 150). The performance of the proposed algorithm is superior to algorithms b) and c) because it uses a better exploration technique as compared to exploration using random perturbation [5] or a coarse estimate of the SNR gradient [6]. It is also interesting to note that the random perturbation technique [5] converges faster than the gradient sign algorithm [6]. This can be attributed to the fact that the beamforming vector could be way off compared to the principal eigenvector of $\mathbf{R}_h$ (optimal) if the estimate of the SNR gradient is not accurate enough. The performance gain of algorithm a) over d) can be attributed to the faster convergence of the ACCPM in comparison to the Cyclic-Jacobi algorithm ($\mathcal{O}(N^2)$ versus $\mathcal{O}(N^2 \log N)$ [16]). Furthermore, it can also be seen that there is a slight gap in the maximum SNR and the maximum value attained by the algorithms in [5] and [6], even after 500 iterations.

The average received SNR and the estimation error for $\mathbf{R_h}$ using the MLE formulation is plotted in Fig. 2.7 for $N = 5$ and $\sigma_n = 0.01$. As mentioned in the captions, there were 86 bit flips among the 500 feedback bits (17%). It can be seen that even in the presence of bit flips, $\hat{\mathbf{R}}_{\mathbf{h}}^{MLE}(t)$ approaches $\mathbf{R_h}$, resulting in the average received SNR at the user's side approaching the maximum achievable SNR with perfect knowledge of $\mathbf{R_h}$. However, the time taken for convergence of $\hat{\mathbf{R}}_{\mathbf{h}}^{MLE}(t)$ to $\mathbf{R_h}$ is higher than in the case without errors (Fig. **??**), i.e., 250 with errors versus 80 without errors. One reason for the slower convergence rate of the MLE is the bit flips occurring as a result of the noise. Another reason is that, unlike the ACCPM, the MLE is not explicitly designed to quickly cut down the uncertainty region. Fig. 2.8 shows the average SNR obtained using the MLE update for different σ_n values. The performance of the ACCPM algorithm assuming perfect data (no bit flips) is also plotted as a baseline for comparison. The plots have been averaged over 100 Monte Carlo runs. It can be seen that as σ_n increases, the time taken for convergence also increases. This is because of the increased number of bit flips due to higher noise variance. However, it is interesting to note that even with a small number of bit flips (i.e., when $\sigma_n = 10^{-4}$, average bit-flips = 4), the convergence of the MLE algorithm is slower compared to the ACCPM algorithm. On the other hand, it should

also be noted that the noise can render the set of inequalities infeasible - in which case the ACCPM is no longer applicable, but MLE still works and manages to approach the optimum solution, without explicitly rejecting the conflicting inequalities. This is pretty remarkable, as it addresses an important practical concern in our context. Fig. 2.9 plots a Monte-Carlo average link SNR using the MLE for $N = 5$, averaging over multiple random realizations of $\mathbf{R_h}$.

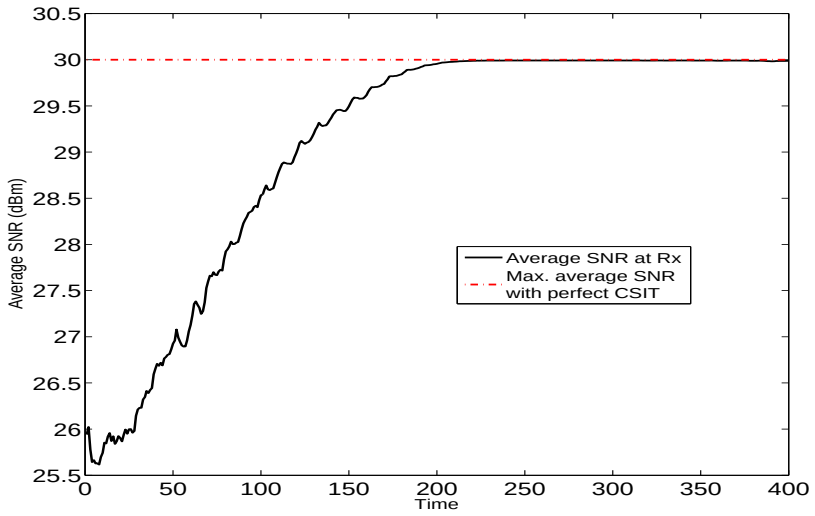

Figure 2.9: Monte Carlo simulation for evolution of average link SNR using MLE formulation in isolated MISO link, $N = 5$.

Fig. 2.10 shows the average SNR performance for $N = 3$ when $\mathbf{R_h}$ changes with time and the discounted MLE formulation is used. For simulation purposes, $\mathbf{R_h}$ is changed to a new correlation matrix at $t = 400$. The channel correlation matrices for the simulation were generated in the same fashion as in the point to point MISO case. For the plots shown in Fig. 2.10, the value of $\beta = 0.95$. It can be seen from Fig. 2.10 that the discounted MLE formulation is able to track the changes in $\mathbf{R_h}$ and adapt the beamforming vectors to achieve the maximum SNR at the receiver. The reason for the perturbation in the SNR performance immediately after $t = 400$ is due to the fact that the inequalities obtained at the Tx during this interval comprises mainly from the old channel correlation matrix. However, the glitches disappear quickly as time progresses, when the effect of the forgetting factor β sets in and the MLE estimate of the

channel correlation is mainly influenced by the new channel correlation matrix.

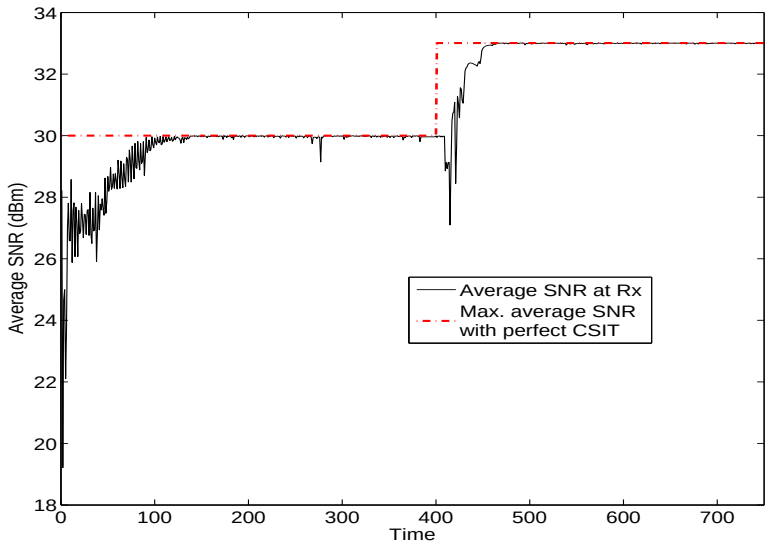

Figure 2.10: Channel correlation tracking performance using the discounted MLE, $N = 3$

2.5 Summary

In this chapter, we have proposed an efficient way to accurately estimate the channel correlation matrix at the Tx of a MISO link based on binary feedback from the Rx, obtained by comparing the average received SNR with a threshold that is varied adaptively by the Tx and communicated to the Rx. This algorithm is used for designing transmit beamforming vectors. The proposed technique is promising because the Tx starts without any CSI, and as time progresses, not only does it obtain an accurate estimate of the correlation matrix and the maximum-SNR beamformer without dedicated training, but it does so while transmitting payload in parallel with the learning process. A maximum likelihood formulation was proposed to accommodate measurement/feedback communication errors that can produce inconsistent inequalities, in which case the ACCPM is no longer applicable. A discounted maximum likelihood formulation was also proposed for tracking changes in the channel correlation matrix.

Chapter 3

Transmit Beamforming for Cognitive Radios using Binary CSIT

3.1 Introduction

Secondary transmit beamforming in a cognitive radio system enables the secondary Tx to steer the radiated power using multiple antennas and attain a high throughput at the secondary Rx without affecting the QoS of the primary network [17],[18]. This facilitates the co-existence of the primary (licensed user) and secondary (unlicensed user), which is crucial for dynamic spectrum access in cognitive radio networks. Designing long-term transmit beamforming vectors for the secondary system in cognitive radio networks without any CSIT, presents the secondary Tx with the challenge of learning the correlation matrix of the channel to the non-cooperative primary Rx, in addition to the correlation matrix of the channel to the intended secondary Rx. For a cognitive radio underlay network, where the secondary and the primary system share the same frequency band for communication, this knowledge is essential for optimizing the average received SNR at the secondary Rx, while limiting interference to the primary network, thereby maintaining the Quality of Service (QoS) at the primary Rx. It is difficult for the secondary Tx to learn the correlation matrix of the channel to the primary Rx versus the secondary Rx, due to lack of cooperation from the primary Rx [19, 20] and this task is even more challenging when there is no channel reciprocity (i.e., the channel from the secondary Tx to the primary Rx is not the conjugate of the channel from primary Rx to secondary Tx). This is especially true when the primary is a legacy system, that is not only unwilling, but also unable to cooperate.

3.1.1 Prior Work and Motivation

In a cognitive underlay scenario [17] comprising a primary and a secondary link, the optimal transmit beamforming vector that maximizes the average SNR at the secondary Rx while limiting the interference caused to the primary Rx can be obtained by solving a convex optimization problem [21], [22], provided the secondary Tx knows the correlation matrices of the channel to the secondary Rx and the crosstalk channel to the primary Rx. Information about these correlation matrices can only be obtained through the respective receivers, except in cases where channel reciprocity can be assumed. Channel reciprocity can only be assumed in time-division duplex systems, but even there reciprocity can be a very coarse approximation, e.g., due to differences in local scattering, or when nodes use different transmit and receive beampatterns.

For designing secondary transmit beamforming vectors in the absence of CSI at the secondary Tx, almost all methods in the literature assume crosstalk channel reciprocity and suggest that the secondary Tx can learn the channel to the primary Rx by overhearing the transmissions of the primary Rx to the primary Tx [19, 20, 23, 24] for learning the interference channel to the primary Rx. For learning the channel to the secondary Rx at the secondary Tx, the state of the art techniques suggest using either secondary channel reciprocity or the channel estimation at the secondary Rx using pilot symbols which is subsequently fed back to the secondary Tx. The main drawback here is two-fold. First of all, the popular communication standards like LTE for cellular systems and 802.11 ac / ah for Wi-Fi systems use different frequency bands for uplink and downink transmission which nullifies the channel reciprocity assumption. Even if the channel reciprocity assumption is valid, it is difficult for secondary Tx to overhear the primary transmissions and learn the interference channel without any synchronization and knowledge of the modulation and rate-adaptation schemes used in the primary system. So it is important to design techniques for the secondary Tx to indirectly learn the interference channel to the primary Rx when there is no channel reciprocity.

For the cognitive radio underlay scenario, notable exceptions to the pervasive use of the crosstalk channel reciprocity assumption are recent works by R. Zhang [25] and Noam *et al.* [16],[18]. A primary and a secondary single antenna Tx-Rx link were considered in [25], where it was assumed that the communication protocol in the primary network and the transmission rate and power adaptations by the primary Tx are known to the secondary Tx. The communication is split into two phases: a) active learning, and b) supervised transmission. During

the active learning phase, the secondary Tx probes the primary Rx with interfering signals, observes the corresponding transmit rate/power adaptations in the primary network and uses this as indirect feedback from the primary network to estimate the interference channel gain from the secondary Tx to the primary Rx. During the supervised transmission phase, the secondary Tx uses the interference channel estimate obtained through active learning and transmits data in such a way that the secondary Rx receives it with high SNR and the interference to the primary Rx is below its interference threshold, which is also assumed known at the secondary Tx. Overall, [25] requires inside knowledge and tight monitoring of the primary system, which may not be possible in ad-hoc deployments.

Noam *et al.* [16],[18] proposed an algorithm that enables the secondary Tx to learn the *fixed* interference channel to the primary Rx, by measuring a monotonic function of the interference to the primary Rx [18] or overhearing the ACK/NACK feedback [16] in the reverse primary link. They proposed to vary the secondary transmit precoding matrix to probe the primary Rx, gradually collecting information on what it can tolerate. They proposed using a cyclic Jacobi subspace estimation algorithm for estimating the eigen-decomposition of the interference channel, and proved that it converges to the interference channel between the secondary Tx and primary Rx. The secondary Tx can indirectly learn the best signaling subspace this way, without assuming channel reciprocity, and without altering the primary communication protocol – an exciting development. Ideally though, such primary Rx 'probing' should be done in parallel with secondary Rx channel exploration (and possibly also payload transmission); and acquisition speed is of essence.

In this chapter, we consider cognitive underlay networks and investigate the ability of the secondary Tx to acquire accurate CSI about the channels to the secondary Rx and primary Rx using rudimentary feedback from the receivers(e.g., acknowledgment / negative-acknowledgement (ACK/NACK)-type) and effectively utilize it for designing transmit beamforming vectors that attain a high SNR at the secondary Rx without comprising the QoS at the primary Rx. Additionally, we explore the scenario when the receivers cannot perform correlation matrix estimation, summarization, and feedback? This can happen if they have limited computation and communication capabilities (low cost, small size and battery), when they are hard-wired for legacy protocols, or when they are opportunistically paired with a transmitter, with limited negotiation before payload transmission. Is it possible to design effective transmit beamforming solutions this way for cognitive underlay networks ?

3.1.2 Contributions

The techniques proposed for an isolated MISO link in chapter 2 are subsequently used as a foundation for designing secondary transmit beamforming vectors for cognitive radio networks (CRNs). Learning to beamform at the secondary transmitter in a cognitive underlay setting is far more challenging, as the primary receiver cannot be assumed to cooperate in 'teaching' the secondary transmitter how to avoid causing interference. Based on novel formulations that exploit the binary feedback in the secondary system *and* the possibility of 'overhearing' the usual ACK/NACK feedback on the reverse primary link, joint cognitive beamforming and primary interference avoidance algorithms are developed. Two distinct scenarios are considered, depending on whether or not the secondary transmitter knows the primary interference threshold. When it does, convergence of the beamforming vector to the optimal one (obtained with perfect CSIT) is proven; otherwise a power back-off mechanism is proposed to enable the secondary transmitter to learn the unknown primary interference threshold. Interestingly, simulations show that it is possible to learn the primary interference threshold and approach optimal secondary link performance this way, as if perfect knowledge of the interference threshold and CSIT were available - albeit we do not have proof of convergence in this case.

The main novelty is the ability to gradually acquire CSI and design optimal transmit beam patterns from rudimentary CSI feedback. This is the first solution to jointly tackle secondary SNR maximization *and* primary interference avoidance, without assuming reciprocity or altering the primary's signaling protocol, while enabling asymptotically optimal performance from only binary CSI. Some of the proposed approaches are very simple, making them ideal for practical implementation. As cognitive radio research and development inches closer to deployment, algorithms that can work under realistic channel and feedback conditions are likely to have a big impact in terms of practical transceiver and network engineering.

3.2 System Model

Consider a cognitive radio underlay scenario where the secondary and the primary Tx share the same frequency spectrum for communicating with the corresponding receivers. The constraint for the secondary Tx in this scenario is that it should transmit in such a way that the interference caused due to its transmission at the primary Rx is restricted below a primary interference threshold. In our cognitive radio setup, the secondary system consists of a multi-antenna Tx

serving a single antenna Rx, coexisting with a primary system comprising a Tx-Rx pair, see Fig. 3.1. Let $\mathbf{h}_{ij}$ denote the random complex channel from the Tx of system i to the Rx of system j and $\mathbf{R}_{ij}$ be the auto-correlation matrix of $\mathbf{h}_{ij}$, $\{i, j\} \in \{1, 2\}$, where index 1 refers to the primary system, and index 2 to the secondary system. Initially, the secondary Tx does not know $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$, the correlation matrices of the channels to the secondary Rx and to the primary Rx, respectively. Its goal is to design beamforming vectors for learning these matrices while sending payload data to the secondary Rx, with the ultimate objective of maximizing the average SNR at the secondary Rx without seriously degrading the QoS of the primary Rx. The challenge here for the secondary Tx is that the primary Rx cannot be assumed to cooperate in the learning process. Note that we do not assume channel reciprocity, as this does not hold with frequency-division duplex or when nodes employ different transmit and receive beam patterns. The secondary Rx (sRx) is assumed to have limited computational resources and restricted

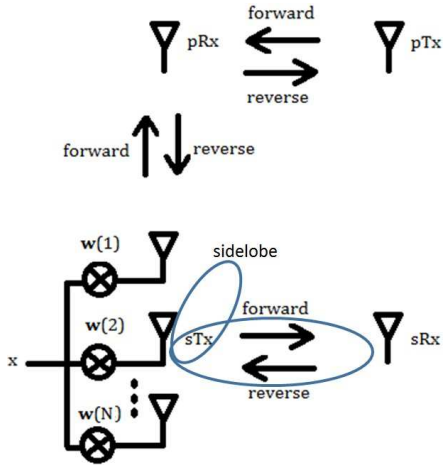

Figure 3.1: **Secondary beamforming schematic: primary system (#1, top) and secondary system (#2, bottom).**

communication capability. Every time, when the secondary transmitter sends the data using a beamforming vector, the secondary receiver measures the average SNR, compares it with a pre-determined threshold and feeds back a '1' or a '0' based on whether the average SNR is greater than or less than the threshold. In addition to the feedback bits from the sRx, the

secondary Tx (sTx) can overhear the regular ACK-NACK feedback that the primary Rx (pRx) sends to the primary Tx (pTx) over its reverse link. We consider two scenarios in what follows, depending on whether or not the sTx knows the primary interference threshold τ_p. Note that, in the absence of third-party interference, τ_p primarily depends on the modulation and coding scheme employed over the primary link, which are often fixed in legacy systems. On the other hand, if the primary system uses adaptive modulation and coding, assuming that τ_p is known to the secondary system is far less realistic. It is assumed that the adaptation procedure by the sTx is much quicker than that of the primary Tx.

Case 1 - sTx knows primary interference threshold τ_p

At time slot t, the sTx employs beamforming vector $\mathbf{w}_t$. Let $\mathcal{G}_{p1}$ and $\mathcal{G}_{p0}$ be the set of time slots where the pRx sends an ACK and NACK feedback, respectively to the pTx and $\mathcal{G}_{p0} \bigcup \mathcal{G}_{p1} = \{1, 2, \ldots, t\}$. Listening in to the primary's reverse link, the sTx infers that

$$\begin{aligned} \mathbf{w}_i^H \mathbf{R}_{21} \mathbf{w}_i &< \tau_p \, \forall i \in \mathcal{G}_{p1} \\ \mathbf{w}_j^H \mathbf{R}_{21} \mathbf{w}_j &\geq \tau_p \, \forall j \in \mathcal{G}_{p0} \end{aligned}$$

$$\begin{aligned} \iff \quad & \text{Tr}(\mathbf{W}_i \mathbf{R}_{21}) < \tau_p \, \forall i \in \mathcal{G}_{p1} \\ & \text{Tr}(\mathbf{W}_j \mathbf{R}_{21}) \geq \tau_p \, \forall j \in \mathcal{G}_{p0} \end{aligned} \tag{3.1}$$

On the other hand, the sTx also receives 1-bit feedback from the sRx, yielding the following inequalities.

$$\begin{aligned} \text{Tr}\left(\mathbf{W}_i \mathbf{R}_{22}\right) &\geq \gamma_i, \quad \forall i \in \mathcal{G}_{s1} \\ \text{Tr}\left(\mathbf{W}_i \mathbf{R}_{22}\right) &< \gamma_i, \quad \forall i \in \mathcal{G}_{s2} \end{aligned} \tag{3.2}$$

where $\mathcal{G}_{s1} = \{i : i \in \{1, 2, \ldots, t\}, s_i = 1\}$ and $\mathcal{G}_{s2} = \{i : i \in \{1, 2, \ldots, t\}, s_i = 0\}$ and $\mathcal{G}_{s1} \bigcup \mathcal{G}_{s2} = \{1, 2, \ldots, t\}$. For every time slot t, the secondary Tx has to update its estimate of $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$ and design the beamforming vector $\mathbf{w}_t$ such that the average received signal power of the sRx is maximized without causing excessive interference to the pRx on average.

Update of $\hat{\mathbf{R}}_{\mathbf{22}}(t)$ and $\hat{\mathbf{R}}_{\mathbf{21}}(t)$

At time $t+1$, the sTx updates $\hat{\mathbf{R}}_{22}(t+1)$ and $\hat{\mathbf{R}}_{21}(t+1)$ as follows

$\mathbf{\Pi_4}$

$$\hat{\mathbf{R}}_{22}(t+1) = \arg\max_{\mathbf{R}_{22}} \sum_{i\in\mathcal{G}_{s1}} \log\left(\text{Tr}\left(\mathbf{W}_i\mathbf{R}_{22}\right)-\gamma_i\right) + \sum_{j\in\mathcal{G}_{s2}} \log\left(\gamma_j - \text{Tr}\left(\mathbf{W}_j\mathbf{R}_{22}\right)\right) + \log\det\mathbf{R}_{22} \quad (3.3)$$

$\mathbf{\Pi_5}$

$$\hat{\mathbf{R}}_{21}(t+1) = \arg\max_{\mathbf{R}_{21}} \sum_{i\in\mathcal{G}_{p0}} \log\left(\text{Tr}\left(\mathbf{W}_i\mathbf{R}_{21}\right)-\tau_p\right) + \sum_{j\in\mathcal{G}_{p1}} \log\left(\tau_p - \text{Tr}\left(\mathbf{W}_j\mathbf{R}_{21}\right)\right) + \log\det\mathbf{R}_{21} \quad (3.4)$$

where $\hat{\mathbf{R}}_{22}(t+1)$ and $\hat{\mathbf{R}}_{21}(t+1)$ are the estimates of $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$ at time $t+1$. From (3.3) and (3.4), it can be seen that $\hat{\mathbf{R}}_{22}(t+1)$ and $\hat{\mathbf{R}}_{21}(t+1)$ are the solution of convex optimization problems (maximization of a concave objective function) which obtain the analytic centers of the convex region at time slot t formed by the associated linear inequalities in (3.2) and (3.1) and the positive semi-definite cone [9, 10]. $\mathbf{\Pi_4}$ and $\mathbf{\Pi_5}$ can be solved efficiently using interior point methods with worst case complexity $\mathcal{O}(N^7)$.

Design of beamforming vector $\mathbf{w}_{t+1}$ and threshold γ_{t+1}

We propose the following steps to design the beamforming vector $\mathbf{w}_{t+1}$. First we solve

$\mathbf{\Pi_6}$

$$\tilde{\mathbf{w}}_{t+1} = \arg\max_{\mathbf{w}} \left[\mathbf{w}^H\hat{\mathbf{R}}_{22}(t+1)\mathbf{w} - \lambda_t\mathbf{w}^H\mathbf{V}_{w,t}\mathbf{V}_{w,t}^H\mathbf{w}\right]$$

$$s.t. \quad \mathbf{w}^H\hat{\mathbf{R}}_{21}(t+1)\mathbf{w} \le \tau_p \quad (3.5)$$

$$\|\mathbf{w}\|^2 \le P_w \quad (3.6)$$

where $\mathbf{V}_{w,t} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t]$, P_w is the maximum available transmit power at the sTx and λ_t is a non-increasing function of t e.g., $\lambda_t = \frac{\lambda}{\lceil 0.1t \rceil}$, with $\lambda_1 \gg 1$. The objective function in $\mathbf{\Pi_6}$ consists of two terms, the first one is proportional to the Tx-side estimate of the average received SNR at the sRx and the second one is the squared norm of the vector $\mathbf{V}_{w,t}^H\mathbf{w}$ whose i^{th}

entry is the scalar dot-product of $\mathbf{w}$ with $\mathbf{w}_i$, $i = 1, 2, ..., t$. The relative importance of each term towards the decision of $\mathbf{w}_{t+1}$ is determined by the scalar λ_t (varies with t). The first constraint is introduced to limit the estimated interference power at the pRx due to transmissions in the secondary network. The second constraint restricts the norm of the beamforming vector to the maximum available transmit power at the sTx. By solving $\mathbf{\Pi_6}$, we get a $\tilde{\mathbf{w}}_{t+1}$ that not only maximizes the sTx-side estimate of average received signal power at sRx, but also is diverse enough to explore the channel correlation space, gain more information about $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$ and eventually improve the accuracy of their estimates, while limiting the average interference to pRx below τ_p. $\mathbf{\Pi_6}$ involves the maximization of an indefinite quadratic objective function subject to two convex quadratic constraints. This problem belongs to a special class of non-convex optimization problems and from the results in [22], the optimal solution to $\mathbf{\Pi_6}$ can be obtained by using semi-definite relaxation (SDR) and solving $\mathbf{\Pi_{6a}}$.

$\mathbf{\Pi_{6a}}$

$$\tilde{\mathbf{W}}_{t+1} = \arg\max_{\mathbf{W}\succeq\mathbf{0}} \left[\text{Tr}\left(\mathbf{W}\hat{\mathbf{R}}_{22}\right) - \lambda_t \text{Tr}\left(\mathbf{W}\mathbf{V}_{w,t}\mathbf{V}_{w,t}^H\right) \right]$$

$$s.t. \quad \text{Tr}\left(\mathbf{W}\hat{\mathbf{R}}_{21}\right) \leq \tau_p \tag{3.7}$$

$$\text{Tr}\left(\mathbf{W}\right) \leq P_w \tag{3.8}$$

where $\mathbf{W} = \mathbf{w}\mathbf{w}^H$ and $\tilde{\mathbf{w}}_{t+1}$ can be obtained as the principal eigenvector of $\tilde{\mathbf{W}}_{t+1}$. Notice that semi-definite relaxation incurs no loss of optimality here, as shown in [22]. Once $\tilde{\mathbf{w}}_{t+1}$ is obtained, $\mathbf{w}_{t+1}$ is designed as follows.

$$\mathbf{w}_{t+1} = \left(\sqrt{\frac{\tau_p}{\tilde{\mathbf{w}}_{t+1}^H \hat{\mathbf{R}}_{21}(t+1)\tilde{\mathbf{w}}_{t+1}}} \right) \tilde{\mathbf{w}}_{t+1} \tag{3.9}$$

This scaling step is necessary because of the following reason.

Remark 1 *The sTx learns $\mathbf{R}_{21}$ by using the ACCPM to reduce the uncertainty region formed by the set of linear inequalities in* (3.1) *and the positive semi-definite cone. For the uncertainty region of $\mathbf{R}_{21}$ to reduce significantly during each time slot $t+1$, the beamforming vector $\mathbf{w}_{t+1}$ should be designed in such a way that the hyperplane $\mathbf{w}_{t+1}^H \mathbf{R} \mathbf{w}_{t+1} = \tau_p$ passes through $\mathbf{R} = \hat{\mathbf{R}}_{21}(t+1)$. It should be noted here that the primary interference threshold τ_p is fixed and cannot be varied by the sTx (since the primary network does not cooperate with sTx). Therefore, the beamforming vector should be scaled instead, to satisfy the equation of the hyperplane. The*

drawback here is that the transmission power will vary at every time slot which would require the power amplifiers at the sTx to operate linearly over a wide dynamic range. In contrast to the case of an point to point MISO link that we considered in chapter 2, this is necessary here because we wish to drive two parallel cutting plane iterations (one each for reducing the uncertainty of $\mathbf{R}_{21}$ *and* $\mathbf{R}_{22}$*) with a common beamforming vector and threshold, thus we must use all degrees of freedom we have.*

After designing $\mathbf{w}_{t+1}$, the threshold for the sRx γ_{t+1} is designed as follows

$$\gamma_{t+1} = \mathbf{w}_{t+1}^H \hat{\mathbf{R}}_{22}(t+1)\mathbf{w}_{t+1} \tag{3.10}$$

This ensures that the hyperplane $\mathbf{w}_{t+1}^H \mathbf{R}\mathbf{w}_{t+1} = \gamma_{t+1}$ passes through $\mathbf{R} = \hat{\mathbf{R}}_{22}(t+1)$.

Convergence of $\mathbf{w}_{t+1}$ to the optimal beamforming vector

The design of $\mathbf{w}_{t+1}$ and γ_{t+1} ensures that the hyperplanes corresponding to the inequalities inferred by the sTx upon receiving the feedback bit and overhearing the primary ACK-NACK feedback for time $t+1$ pass through the analytic centers of the feasible regions of $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$, i.e., $\hat{\mathbf{R}}_{22}(t+1)$ and $\hat{\mathbf{R}}_{21}(t+1)$, respectively. Hence convergence of ACCPM [11] can be invoked. As a result, $\hat{\mathbf{R}}_{22}(t+1)$ and $\hat{\mathbf{R}}_{21}(t+1)$ are confined to a ball of radius r centered around $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$ respectively, within $\mathcal{O}\left(\frac{N^2}{r^2}\right)$ iterations. Furthermore, λ_t decreases to zero as $t \to \infty$. Therefore, as $\hat{\mathbf{R}}_{22}(t+1) \to \mathbf{R}_{22}$, $\hat{\mathbf{R}}_{21}(t+1) \to \mathbf{R}_{21}$ and $\lambda_t \to 0$, $\mathbf{\Pi_6}$ becomes

$$\begin{aligned} &\mathbf{\Pi_{6b}} \\ &\quad \tilde{\mathbf{w}} = \arg \max_{\|\mathbf{w}\|^2 \le P_w} \quad \mathbf{w}^H \mathbf{R}_{22} \mathbf{w} \\ &\quad s.t. \quad \mathbf{w}^H \mathbf{R}_{21} \mathbf{w} \le \tau_p \end{aligned}$$

which is the optimal secondary beamforming vector design when sTx has perfect knowledge of $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$. Note that this is a special kind of non-convex problem ('one constraint away' from the Rayleigh quotient) that can be solved exactly [22].

Case 2 - sTx does not know τ_p

Here we assume that the sTx does not have any knowledge about the primary interference threshold τ_p. In this case, by overhearing the primary ACK-NACKs, the sTx infers the following

inequalities

$$\begin{aligned} &\mathbf{w}_i^H \mathbf{R}_{21} \mathbf{w}_i \leq \mathbf{w}_j^H \mathbf{R}_{21} \mathbf{w}_j \\ \Longleftrightarrow &\text{Tr}(\mathbf{W}_i \mathbf{R}_{21}) \leq \text{Tr}(\mathbf{W}_j \mathbf{R}_{21}),\ \forall i \in \mathcal{G}_{p1}, \forall j \in \mathcal{G}_{p0} \end{aligned} \tag{3.11}$$

These intuition behind this inference is that the interference caused in the time slots by the sTx corresponding to an ACK transmission by the pRx is lesser than the interference caused when a NACK is sent. However, this relies on the underlying assumption that the pTx does not change its transmission strategy (i.e. transmit power or transmission rate) when the secondary is learning the channel information (or in essence; the secondary adaptation interval is much faster than the primary adaptation interval).

From the feedback bits sent by the sRx to the sTx, the sTx infers the inequalities mentioned in (3.2). As before, the sTx updates $\hat{\mathbf{R}}_{22}(t+1)$ as the analytic center of the associated linear inequalities and the positive semidefinite cone as shown in (3.3). For updating $\hat{\mathbf{R}}_{21}(t+1)$, we do not use the analytic center update because the inequalities inferred by the sTx corresponding to $\mathbf{R}_{21}$ in (3.11) are homogeneous and pass through the origin, which may cause numerical instability while using the log-barrier function for solving the convex optimization problem. Therefore, we propose a formulation for jointly designing the beamforming vector $\mathbf{w}_{t+1}$ and updating $\hat{\mathbf{R}}_{21}(t+1)$.

$$\begin{aligned} &\boldsymbol{\Pi_7}: \\ &\max_{\mathbf{w}, \mathbf{R}_{21} \succeq 0} \left[\mathbf{w}^H \hat{\mathbf{R}}_{22}(t+1) \mathbf{w} - \mu \mathbf{w}^H \mathbf{R}_{21} \mathbf{w} - \lambda_t \mathbf{w}^H \mathbf{V}_{w,t} \mathbf{V}_{w,t}^H \mathbf{w} \right] \\ &\qquad s.t. \qquad \|\mathbf{w}\|^2 \leq P_w \qquad (3.12) \\ &\qquad \text{Tr}(\mathbf{W}_i \mathbf{R}_{21}) \leq \text{Tr}(\mathbf{W}_j \mathbf{R}_{21}), \forall i \in \mathcal{G}_{p1}, \forall j \in \mathcal{G}_{p0} \qquad (3.13) \end{aligned}$$

where $\mu \in \mathcal{R}_+$ and λ_t is a non-increasing function of t. The motivation for the objective function in $\boldsymbol{\Pi_7}$ is as follows. The first term is the sTx-side estimate of the signal power received at the sRx. For a given $\mathbf{w}$, the second term in the objective of $\boldsymbol{\Pi_7}$ selects from the admissible $\hat{\mathbf{R}}_{21}(t+1)$ the one that is most favorable from the sTx point of view; that is, the one that is annoyed the least by $\mathbf{w}_{t+1}$. The third term diversifies the choice of $\mathbf{w}$ to explore the channel correlations space from as many directions as possible to gather information about $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$, and eventually improve the accuracy of the estimates.

Since $\mathbf{\Pi_7}$ is not jointly convex in $\mathbf{w}$ and $\mathbf{R}_{21}$, it can be tackled using alternating optimization. For a fixed $\mathbf{w}$, the update of $\mathbf{R}_{21}$ is semi-definite programming (SDP). For a fixed $\mathbf{R}_{21} = \hat{\mathbf{R}}_{21}(t+1)$, the update of $\mathbf{w}$ is simply the unit vector along the eigenvector corresponding to the maximum magnitude eigenvalue of $\hat{\mathbf{R}}_{22}(t) - \mu\hat{\mathbf{R}}_{21}(t+1) - \lambda_t \mathbf{V}_{w,t}\mathbf{V}_{w,t}^H$. The threshold γ_{t+1} for the sRx at time $t+1$ can be obtained using ACCPM, namely $\gamma_{t+1} = \text{Tr}\,(\mathbf{W}_{t+1}\hat{\mathbf{R}}_{22}(t+1))$, where $\mathbf{W}_{t+1} = \mathbf{w}_{t+1}\mathbf{w}_{t+1}^H$.

Since the update of $\hat{\mathbf{R}}_{22}(t)$ is independent of $\hat{\mathbf{R}}_{21}(t)$ and there is no dependency in terms of the constraints, convergence of $\hat{\mathbf{R}}_{22}(t)$ follows from the ACCPM, as before. On the other hand, the constraints determining the uncertainty region for $\mathbf{R}_{21}$ are all homogeneous, $\text{Tr}(\mathbf{W}_m\mathbf{R}_{21}) \leq \text{Tr}(\mathbf{W}_n\mathbf{R}_{21})$, $\forall m \in \mathcal{G}_{p1}, \forall n \in \mathcal{G}_{p0}$ and $\mathbf{R}_{21} \succeq 0$. The hyperplanes corresponding to the linear inequalities all pass through the origin, so the uncertainty region remains unbounded for all t. As a result, there is no hope that $\hat{\mathbf{R}}_{21}(t)$ will converge to $\mathbf{R}_{21}$, or that the interference to the pRx will converge below its (unknown) interference tolerance level. On the other hand, with appropriate choice of the sequence of the $\mathbf{W}_i$'s ($\Leftrightarrow$ the $\mathbf{w}_i$'s), there is hope that the *direction* of vec$(\hat{\mathbf{R}}_{21}(t))$ will align with that of vec$(\mathbf{R}_{21})$, despite the fact that the scale cannot be recovered. This suggests using an additional interference management mechanism to limit sTx to pRx interference, when needed. Towards this end, we may: (a) *Fix μ and vary P* at slot $t+1$ based on whether an ACK or NACK was heard from the pRx at slot t; i.e., $\mathbf{w}_{t+1}$ is scaled by $\sqrt{\alpha_{t+1}}$, where $\alpha_{t+1} = \alpha_t\alpha$, if a NACK was heard, else $\alpha_{t+1} = \max(\alpha_t/\alpha, P_w)$, with back-off parameter $\alpha < 1$. Alternatively, we may (b) *Fix P and vary μ* in $\mathbf{\Pi_5}$, thereby changing the relative preference to directions that cause lower estimated average interference power to the pRx. Whenever the sTx hears a NACK from the pRx, it sets $\mu_{t+1} = \mu_t\delta$, while an ACK results in $\mu_{t+1} = \mu_t/\delta$, with $\delta > 1$.

As we will see, simulations show that the fix μ vary P power back-off scheme can learn the primary interference threshold and approach optimal secondary link performance - albeit we do not have proof of convergence in this case.

3.3 Simulation Results

3.3.1 Cognitive radio underlay simulations

For simulation purposes, $\mathbf{h}_{ij} \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}_{ij}),\ i,j \in \{1,2\}$. $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$ were obtained by generating random orthonormal matrices $\mathbf{U}_{22}$ and $\mathbf{U}_{21}$, random diagonal matrices $\mathbf{D}_{22}$ and $\mathbf{D}_{21}$ with positive diagonals, and setting $\mathbf{R}_{22} = \mathbf{U}_{22}\mathbf{D}_{22}\mathbf{U}_{22}^H$, $\mathbf{R}_{21} = \mathbf{U}_{21}\mathbf{D}_{21}\mathbf{U}_{21}^H$.

Fig. 3.2 shows the simulation results for the case when τ_p is known to the sTx. The top plot shows the average received SNR at the sRx and the bottom plot shows the average interference power at the pRx for $N = 5$, $\lambda_t = \frac{5}{\lceil 0.1t \rceil}$, τ_p = 0.5 and $P_w = 5$. The dotted horizontal line in the top plot is the maximum achievable average received SNR at the sRx (with perfect knowledge of $\mathbf{R}_{22}$, $\mathbf{R}_{21}$ and τ_p at the sTx). The horizontal line in the bottom in Fig. 3.2 represents the primary interference power threshold τ_p. It can be seen that the average SNR at sRx converges to the maximum achievable SNR value (obtained with perfect knowledge of $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$ at sTx) and the average interference power at the pRx converges to τ_p. Fig. 3.3 plots the Monte-Carlo simulation of the average SNR at sRx for $N = 5,\ \tau_p = 0.1$ by averaging over multiple random realizations of $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$. It can be seen from Fig. 3.3 that the average SNR at the sRx attains the maximum average SNR and the average interference power at the pRx is limited to τ_p for every random realization of $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$.

Fig. 3.4 shows simulation results for the average received SNR at the sRx and the average interference power at the pRx when τ_p is unknown to the sTx with $N = 5$, $\lambda_t = \frac{5}{\lceil 0.1t \rceil}$, τ_p = 0.5, $P_w = 5$, and back-off parameters α = 0.8 and δ = 2. The dotted straight line in the top plot is the maximum achievable average received SNR at sRx (with perfect knowledge of $\mathbf{R}_{22}$, $\mathbf{R}_{21}$ and τ_p at the sTx). The solid horizontal line in the two lower plots in Fig. 3.4 represents the primary interference power threshold which is not known to the sTx. It is very pleasing and intriguing to see that the proposed power back-off mechanism (a) approaches optimal performance in terms of sRx SNR, while the interference it causes to the pRx converges to the primary interference threshold τ_p, which is unknown! At the same time, the indirect back-off mechanism (b) clearly fails in this case - which speaks for the importance of choosing the right back-off scheme. Fig. 3.5 plots the Monte-Carlo simulation of average SNR at sRx and average interference power at pRx using power back-off mechanism (a) for $N = 5,\ \tau_p = 0.1$ by averaging over multiple random realizations of $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$. Note that there is a small gap relative to optimal performance in this case (32 versus 33.5 dBm).

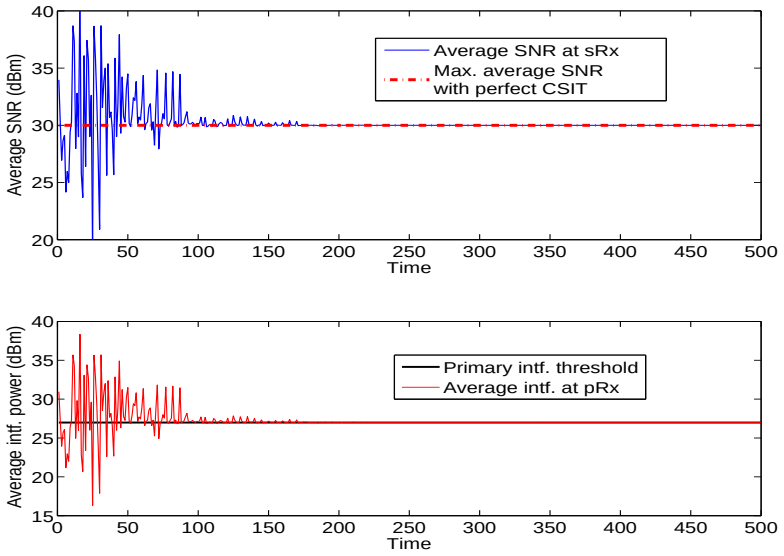

Figure 3.2: τ_p is known at sTx - Avg. SNR at sRx (top) and avg. interference power at pRx (bottom), $N = 5$, $\tau_p = 0.5$

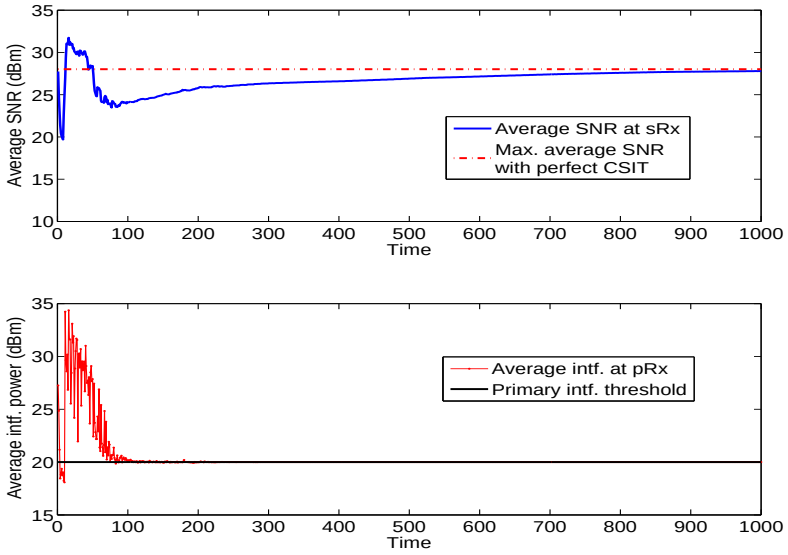

Figure 3.3: τ_p is known at sTx - Monte Carlo simulation for Avg. SNR at sRx (top) and avg. interference power at pRx (bottom), $N = 5$, $\tau_p = 0.1$

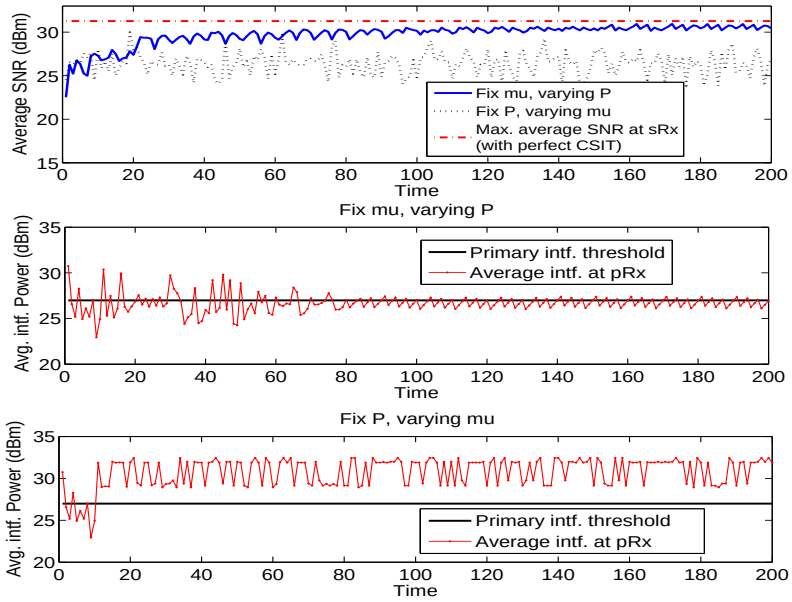

Figure 3.4: τ_p is unknown at sTx - Avg. SNR at sRx (top) and interference power at pRx for two candidate back-off schemes, $N = 5$, $\tau_p = 0.5$

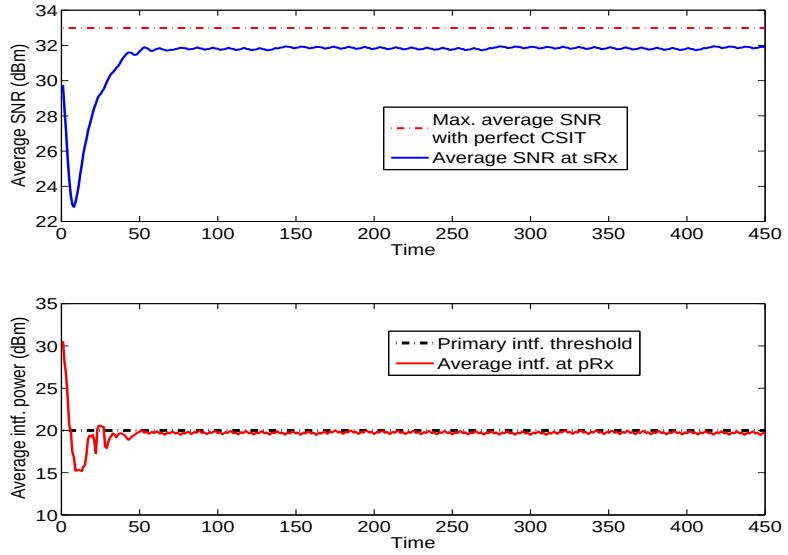

Figure 3.5: τ_p is unknown at sTx - Monte Carlo simulation for Avg. SNR at sRx (top) and avg. interference power at pRx (bottom), $N = 5$, $\tau_p = 0.1$

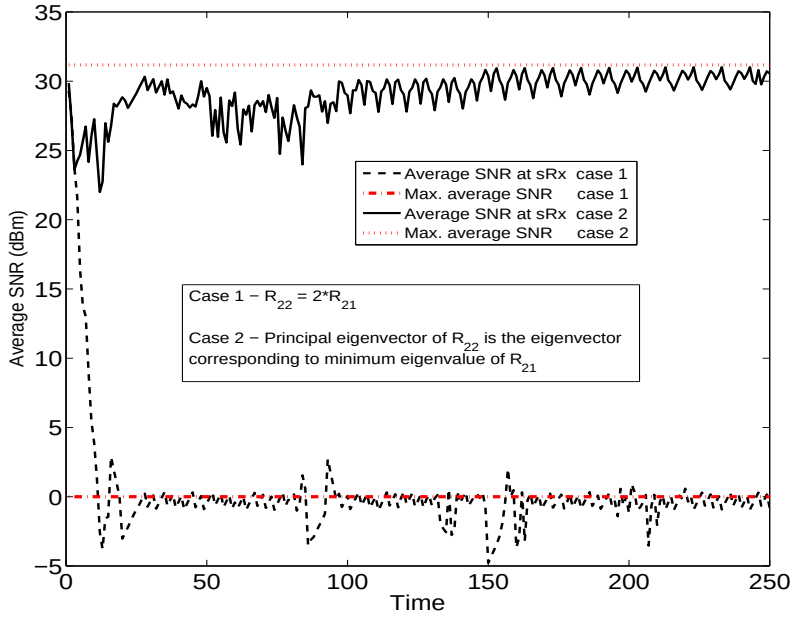

Figure 3.6: Comparison of average SNR performance at sRx for $\tau_p = 5.10^{-4}$

It should be noted that in all the cases, the interference caused by the sTx at the pRx exceeds the threshold initially before damping down quickly to the interference threshold τ_p. This could be seen as a nuisance to the primary network. But this has to be done by the sTx since it does not have any CSI about the interference channels and it has to rely only on the indirect ACK - NACK feedback by the pRx to infer information about the interference channel.

Fig. 3.6 highlights the average SNR performance at the secondary Rx of a cognitive radio using interference management scheme (a) for two extreme cases. For case 1, $\mathbf{R}_{22}$ has been chosen proportional to $\mathbf{R}_{21}$. This can occur when the secondary Rx and the primary Rx are aligned to each other when viewed from the secondary Tx. In this case it is generally difficult for the secondary Tx to design transmit beamforming vectors for providing high average SNR at the secondary Rx without causing excessive interference to the primary Rx. For case 2, $\mathbf{R}_{22}$ and $\mathbf{R}_{21}$ have been designed such that the principal eigenvector of $\mathbf{R}_{22}$ is aligned with the minor eigenvector of $\mathbf{R}_{21}$, i.e., the direction of the eigenvector corresponding to the minimum eigenvalue of $\mathbf{R}_{21}$. This is a very desirable scenario for the secondary Tx because if it aligns the transmit beamforming vector along the principal eigenvector of $\mathbf{R}_{22}$, then it can achieve high

average SNR at the secondary Rx as well as cause the least possible interference to the primary Rx simultaneously. For this simulation, the correlation matrix $\mathbf{R}_{22}$ was generated in the same fashion as mentioned in the previous paragraph and $\mathbf{R}_{21}$ was generated based on the conditions required for cases 1 and 2 mentioned here. It can be seen that there is approximately 30 dB difference in the maximum average SNR between these two cases, and that the secondary Tx achieves the maximum possible average SNR in both cases.

3.4 Summary

In this chapter, we have proposed pertinent extensions of the algorithms used for point-to-point MISO links (in chapter 2) to an underlay cognitive radio network setup for designing beamforming vectors at the secondary Tx to maximize the average received SNR at the secondary Rx without causing excessive interference to the primary Rx. Through relevant simulations, it was shown that the proposed algorithms for cognitive radio networks enable the secondary Tx to learn the relevant channel correlation matrices, starting from no CSI,and design beamformers to attain the maximum achievable SNR value at the secondary Rx, obtained when the secondary Tx has perfect knowledge of the primary interference threshold and channel correlation matrices to the secondary Rx and the primary Rx.

Chapter 4

Adaptive Algorithms for Single-Group Multicast Beamforming

4.1 Introduction

Multicast beamforming is a part of the Evolved Multimedia Broadcast Multicast Service (eMBMS) in the Long-Term Evolution (LTE) standard for efficient audio and video streaming. Multicast beamforming utilizes multiple transmit antennas and channel state information at the transmitter (CSIT) to steer transmitted power towards a group of subscribers while limiting interference to other users and systems [26]. In single group multicasting, where all users are interested in the same information stream from the transmitter (Tx), the maximum common data rate is determined by the minimum received signal to noise ratio (SNR). Hence the objective is to maximize the minimum received SNR subject to transmit power constraints (max-min-fair multicast beamforming). An alternative is to minimize the transmit power subject to appropriate quality-of-service (QoS) guarantees formulated in terms of the minimum SNR for each user (QoS multicast beamforming). The two formulations are essentially equivalent from an optimization point of view [26] in the sense that, the solutions of both the optimization problems are the same up to a scaling factor.

All work till date on multicast beamforming has assumed that some grade of CSIT (instantaneous or statistical, perfect or inexact) is available. In practice CSI has to be acquired, and that can be a serious burden – especially when the number of users and/or antennas is large. CSIT can be acquired before beamforming optimization, but in reality channels change over

time and users may drop in or out of the multicast, so it is appealing to consider joint online CSIT acquisition and beamformer adaptation. This is a challenging problem, especially when channel reciprocity cannot be assumed, and the receiver (Rx) equipment is limited in terms of computation and communication capabilities.

4.1.1 Related Work

Sidiropoulos *et al.* [26] considered max-min-fair and QoS multicast beamforming for a multi-antenna Tx serving multiple users, each with a single antenna Rx. It was shown in [26] that these two formulations are essentially equivalent NP-hard optimization problems, which can be expressed as a non-convex quadratically constrained quadratic program (QCQP). Semi-definite relaxation (SDR) followed by Gaussian randomization (SDR-G) was proposed in [26] to obtain an upper bound on the attainable minimum SNR and a good sub-optimal solution, respectively.

For a large number of antennas or users, the quality of the approximation obtained using SDR-G deteriorates considerably (i.e., the gap between the minimum SNR attained using the sub-optimal beamforming vectors obtained from SDR-G algorithm and the upper bound calculated from the SDR increases significantly). This prompted a search for better approximations of the multicast beamforming problem over the past decade. The best approach so far is the successive linear approximation (SLA) algorithm proposed by Tran *et al.* for the QoS version of the problem [27]. The SLA algorithm starts with a feasible vector, say $\mathbf{w}_0$. The non-convex QoS constraints are linearized about $\mathbf{w}_0$ using first-order Taylor series expansion, and the resulting convex problem is solved to obtain the next iterate $\mathbf{w}_1$, which is subsequently used for linearization in the next iteration. It was shown in [27] that the SLA algorithm converges to a KKT point of the QoS problem formulation, and attains a higher minimum SNR than SDR-G in simulations. SLA has lower worst-case complexity per iteration than SDR-G ($\mathcal{O}(N+K)^{3.5}$ for one SLA iteration vs. $\mathcal{O}(N^2+K)^{3.5}$ overall for SDR-G, where N is the number of antennas at the Tx and K is the number of users). However, the overall complexity of SLA can be greater than that of SDR-G, because of the outer linearization iterations required for convergence of the SLA algorithm.

A different approximation of the max-min-fair formulation was recently proposed by Demir *et al.* [28]. In [28], the non-convex part of the problem is isolated to a rank-one constraint, which is replaced with an equivalent non-convex bilinear trace constraint. The interesting aspect of Demir's approach is that the resulting reformulation is naturally amenable to alternating

maximization (AM). Simulations in [28] showed that the AM attained a higher minimum SNR compared to SDR-G. However, the computational complexity of the AM algorithm is higher than SDR-G, since AM involves solving an SDP (of the same size as SDR-G) at each AM step. Furthermore, our simulations indicate that SLA outperforms AM, and SLA has lower complexity than AM.

SDR-G, SLA and AM involve solving one or more convex optimization problems to obtain a good transmit beamforming vector that attains a high minimum SNR. For large N and K, the computational cost of solving these convex optimization problems becomes prohibitive, and low-complexity algorithms are needed. The first low-complexity adaptive algorithm for (max-min-fair) multicast beamforming was proposed by Lozano [29]. In every iteration of Lozano's algorithm, the new iterate is obtained by updating the previous iterate with a fixed step along the SNR gradient direction of the user with the least SNR in the previous iteration. This is followed by a scaling step to satisfy the transmit power constraint. Simulations showed that Lozano's algorithm can achieve a higher minimum SNR than the SDR-G approach when $K \gg N$. The computational complexity of Lozano's algorithm is $\mathcal{O}(KN)$ for instantaneous rank-one CSIT and $\mathcal{O}(KN^2)$ for long-term higher-rank CSIT, which is much lower than SDR-G and SLA. Matskani *et al.* [30] observed that Lozano's algorithm can exhibit limit cycle behavior (resulting in oscillations), and proposed a variation called (damped) LLI (Lozano with Lopez Initialization). This employs a diminishing step size and a warm-start using a more sophisticated initialization using the weight vector that maximizes average SNR [31]. Simulations showed that the LLI algorithm obtains a higher minimum SNR than Lozano's algorithm at the same complexity.

Abdelkader *et al.* [1] proposed a low-complexity algorithm based on channel orthogonalization using QR decomposition, to approximate the QoS problem when $K \geq N$. For every run of the QR algorithm, a set of N out of K channel vectors are chosen randomly and stacked into a $N \times N$ matrix $\mathbf{H}$, and the QR decomposition $\mathbf{H} = \mathbf{QR}$ is obtained. The columns of the orthogonal matrix $\mathbf{Q}$ are used as basis vectors of the N-dimensional space and the beamforming vector is modeled as a weighted linear combination of these basis vectors. The corresponding weights are obtained in closed form [1], followed by a scaling step to satisfy the QoS constraints. This procedure is repeated multiple times for different random draws of the N channels and the final beamforming vector is the best obtained among these random draws. Simulations showed that when $K \gg N$, the QR algorithm performs better than the SDR-G approach, at

$\mathcal{O}(N^2)$ complexity - which is much lower than SDR-G. However for smaller values of K, the performance of the QR algorithm is inferior to the SDR-G algorithm.

Multicast beamforming in the case of only $K = 2$ users was specially considered in [32], which derived the optimal solution for this case (note that the NP-hardness proof in [26] does not apply when $K < N$). Motivated by the optimal solution for $K = 2$, [32] proposed an orthogonalization-based successive beamforming (SB) algorithm for general K. In the SB algorithm, the beamforming vector is successively constructed by orthogonalizing the subspace spanned by the channel vector of every user until a maximum iteration count is reached (equal to the minimum value of number of transmit antennas and number of users). As we will see in the simulations section, the performance of SB quickly becomes inferior to SDR-G as K increases, so it is not competitive to the state-of-the-art.

When only imperfect CSIT is available, a robust multicast beamforming formulation (which further includes interference constraints) has been considered in [33]. The problem is formulated as a non-convex QCQP and two randomization algorithms are proposed to obtain sub-optimal solutions. Furthermore, a specific case of the problem was identified for which the optimal solution can be obtained in polynomial time via SDR [33].

QR, Lozano, and LLI feature low complexity, but also a relatively large gap to the SDR performance bound. Perhaps more importantly (since the SDR bound is generally not attainable), our simulations show that the minimum SNR attained by these algorithms is still significantly lower than that of SLA. Another drawback is that QR, Lozano, and LLI require tuning of parameters through trial and error.

Summarizing, no algorithm offers state-of-the-art performance (SLA) at low-enough complexity (QR/Lozano/LLI). One of our original goals was to fill this gap; as we will see, our new algorithms come close to SLA in terms of performance, at QR/Lozano/LLI complexity. Even better, the proposed algorithms can be used to warm-start a single iteration of SLA, and this turns out to outperform (iterative) SLA, as we will see.

Our second goal was to come up with a multicast beamforming algorithm that gradually learns the required CSI as it adapts the beamformer weights. Online algorithms for designing transmit beamforming weights for unicast transmission without initial CSIT have been developed in [34]-[35], using binary feedback from the Rx. Reference [34] proposed a variation of the Cyclic Jacobi subspace estimation algorithm to learn the eigen-decomposition of the instantaneous channel / channel correlation matrix ($\mathbf{H}$ / $\mathbf{R}$) of a Multiple-Input Multiple-Output

(MIMO) link. The 1-bit feedback was assumed to be based on a monotonic function of the instantaneous / average received signal power. It was shown that this algorithm asymptotically converges to the eigen-decomposition of $\mathbf{H}$ / $\mathbf{R}$.

An adaptive thresholding algorithm was proposed in [36] to simultaneously transmit data and learn the optimal long-term beamforming vector (i.e., the principal eigenvector of $\mathbf{R}$), using binary feedback from the Rx of a Multiple-Input Single-Output (MISO) unicast channel link. For every new transmit beamforming vector, a '1' or a '0' is fed back by the Rx, based on whether the measured average SNR is $\geq$ or $<$ a pre-determined threshold. From every feedback bit, a new linear inequality involving $\mathbf{R}$ is inferred at the Tx, and $\hat{\mathbf{R}}$ is updated as the analytic center of the region formed by the positive semi-definite (p.s.d.) cone and all the linear inequalities inferred until that point. The new beamforming vector is designed to create a balance between gathering new information about $\mathbf{R}$ and attaining a high average SNR using the knowledge acquired from all the feedback bits; while the new threshold is designed in order to reduce the existing uncertainty regarding $\mathbf{R}$. Asymptotic convergence to the maximum SNR attained with perfect CSIT was established in [36].

A similar algorithm using essentially the same analytic center cutting plane method (ACCPM) as [36] was independently and simultaneously proposed in [35] for maximizing the instantaneous energy harvested at the Rx of a MIMO link. In [35], the 1-bit Rx feedback at time slot t is based on whether the energy harvested at time t is $\geq$ or $<$ that at time $t-1$. Simulations in [36] and [35] showed the faster convergence rate of the respective algorithms to the optimal value (obtained with perfect CSIT) in comparison with [34]. The algorithms in [34] - [35] can be used to learn the user channels in a multicast setup, by considering every user of the multicast individually.

4.1.2 Contributions

In this chapter, we consider a single-group multicast cell with N antennas at the Tx serving K single antenna users. We consider two scenarios: a) the Tx has perfect CSI for all K users, and b) the Tx has no initial CSI for any user. When perfect CSIT is available, we propose a new class of adaptive multicast beamforming algorithms comprising *Additive Update* (AU), *Multiplicative Update* (MU), and *Multiplicative Update - Successive Linear Approximation* (MU-SLA) algorithms, with guaranteed convergence and state-of-the-art performance at low complexity. In every iteration of the AU algorithm, the beamforming vector is updated by taking a step

along the inverse-SNR weighted SNR-gradient direction of all the users, as computed using the previous iterate. This is followed by a scaling step to satisfy a transmit power constraint, and the whole procedure is repeated until the iterates converge. The fixed point equation of this algorithm is analyzed for a simple but insightful example, and convergence is established by interpreting the AU as successive concave approximation of (or projected gradient update for) proportionally fair beamforming [37]. The MU algorithm, which is a limiting case of the AU algorithm, attains the same minimum SNR as AU, but has faster convergence and it also eliminates the need for step-size selection. We currently have proof of convergence only for the AU - the analysis does not carry over verbatim to the MU for technical reasons. The MU-SLA algorithm uses the solution provided by the MU algorithm as an initialization for a *single SLA iteration*. Simulations show that MU-SLA outperforms SLA, while the AU and MU operate close to SLA and outperform all the other algorithms, at an order of magnitude lower complexity. The performance-complexity trade off is analyzed for the proposed algorithms and the previous state-of-art using relevant simulations.

In the absence of initial CSIT, if the receivers do not have sufficient computational and energy resources to estimate, quantize and feed back accurate CSI to the Tx, we propose an online *cognitive multiplicative update* (CMU) algorithm for designing long-term beamforming vectors using binary channel quality user feedback. In the CMU algorithm, every user only feeds back a '1' or a '0' in each time slot, depending on whether its average received SNR is $\geq$ or $<$ a pre-determined threshold. Using the feedback bits from every user, the Tx learns new linear inequalities about the channel matrices of the users and updates its estimate using the ACCPM. The new beamforming vector is designed to gather useful information about the channel and also use the accumulated knowledge to attain a high minimum SNR among the users. Two threshold selection techniques at the Tx, namely i) *multiple threshold selection* and ii) *common threshold selection*, are proposed for effectively reducing the uncertainty in the channel correlation matrices of the users in each slot. It is shown that the former reduces the uncertainty faster and converges to the true channel correlation matrices at a faster rate than the latter, at the cost of higher communication overhead. A simple modification is also proposed to completely eliminate the communication overhead in ii) by varying the transmit power. Simulations show that the CMU algorithm using the aforementioned threshold selection methods converges to the performance achieved with perfect CSIT.

4.2 System Model and Problem Formulation

We consider a single-group multicast cell consisting of a Tx with N antennas and K single antenna receivers. The Tx transmits the common data x which has zero-mean and unit-variance, to all the K receivers using a unit-norm beamforming vector $\mathbf{w}$. The corresponding received signal at the k^{th} Rx is given by

$$y_k = \mathbf{w}^H \mathbf{h}_k x + z_k, \forall k \in \{1, 2, \ldots, K\} \tag{4.1}$$

where $\mathbf{h}_k$ is the channel between the Tx and the k^{th} Rx which is modeled as a $N \times 1$ zero-mean complex random vector. z_k is wide-sense stationary additive noise at the k^{th} Rx, assumed independent of x and $\mathbf{h}_k$, with zero-mean and variance σ_k^2. The received SNR at the k^{th} Rx is given by

$$\frac{|\mathbf{w}^H \mathbf{h}_k|^2}{\sigma_k^2} = \frac{\mathbf{w}^H \mathbf{R}_k \mathbf{w}}{\sigma_k^2} \tag{4.2}$$

where $\mathbf{R}_k = \mathbf{h}_k \mathbf{h}_k^H \succeq \mathbf{0}, \forall k \in \{1, 2, ..., K\}$. We can absorb σ_k into $\mathbf{h}_k$, and thereafter work with the scaled channels $\tilde{\mathbf{h}}_k = \mathbf{h}_k / \sigma_k$ (assuming that the k^{th} Rx can estimate σ_k beforehand and inform the Tx, or scale $\mathbf{h}_k$ before sending it to the Tx). We will assume that this has already been done, and drop the $\tilde{}$ for brevity. The objective of the Tx is to design unit norm transmit beamforming vectors that maximize the minimum SNR among the users. This can be formulated as follows.

$$\begin{aligned} \mathbf{\Pi_8} \qquad & \arg \max_{\mathbf{w} \in \mathbb{C}^N} \min_{k \in \{1,2,..,K\}} \mathbf{w}^H \mathbf{R}_k \mathbf{w} \\ & s.t. \quad \|\mathbf{w}\|^2 = 1 \end{aligned}$$

where $\forall k \in \{1, 2, \ldots, K\}$. $\mathbf{\Pi_8}$ is NP-hard, as shown in [26].

4.3 Tx has perfect CSI about $\{\mathbf{R}_k\}_{k=1}^K$

When the Tx has perfect CSI, the SLA algorithm [27] is the state-of-the-art in terms of attaining the highest possible minimum SNR / multicast rate. However, it has a relatively high worst-case complexity. The first low-complexity adaptive algorithm for multicast beamforming was Lozano's [29]. In every iteration of Lozano's algorithm, the new iterate is obtained by updating the previous iterate with a fixed step along the SNR gradient direction of the user with the least

SNR in the previous iteration. This is followed by a scaling step to satisfy the transmit power constraint. Lozano's algorithm focuses only on the weakest user in each iteration and ignores all other users. In certain cases [30], this strategy results in fluctuations in the minimum SNR due to limit cycles, as improving the SNR of one user may reduce the SNR of another and vice-versa. When there are multiple users experiencing low SNR, it makes intuitive sense that we should take all the user-channels into account while taking the next step. Furthermore, users experiencing different SNR 'grades' should be appropriately weighted in the computation of the new direction. This intuition naturally suggests the following *Additive Update* (AU) algorithm, which we first introduce below in the context of a simplified, two-user scenario.

Example - Consider a scenario with $K = 2$ users. The initial unit norm beamforming vector is chosen randomly and is denoted by $\mathbf{w}_1$. At every iteration $n \geq 1$, the new beamforming vector iterate is obtained as follows.

$$\begin{aligned} \tilde{\mathbf{w}}_{n+1} &= \mathbf{w}_n + \alpha \left[\frac{\mathbf{h}_1 \mathbf{h}_1^H \mathbf{w}_n}{|\mathbf{h}_1^H \mathbf{w}_n|^2} + \frac{\mathbf{h}_2 \mathbf{h}_2^H \mathbf{w}_n}{|\mathbf{h}_2^H \mathbf{w}_n|^2} \right] \\ &= \mathbf{w}_n + \alpha \left[\frac{\mathbf{h}_1}{\mathbf{w}_n^H \mathbf{h}_1} + \frac{\mathbf{h}_2}{\mathbf{w}_n^H \mathbf{h}_2} \right] \qquad (4.3) \\ \mathbf{w}_{n+1} &= \frac{\tilde{\mathbf{w}}_{n+1}}{\|\tilde{\mathbf{w}}_{n+1}\|} \qquad (4.4) \end{aligned}$$

where α is the fixed positive step-size for every iteration. From (4.3), $\tilde{\mathbf{w}}_{n+1}$ is a sum of $\mathbf{w}_n$ and a fixed step times a direction vector, which is an inverse SNR weighted sum of the SNR gradients, evaluated at $\mathbf{w}_n$. Therefore, in the $(n+1)^{th}$ iteration, $\mathbf{w}_{n+1}$ is updated along a direction that favors the user with the least SNR in the n^{th} iteration, but also takes into account, the other user. More generally, for $K > 2$, using an inverse SNR weighted sum of the SNR gradients of all users will favor those experiencing lower SNRs in the n^{th} iteration. This should be contrasted with [29], [30], which only focus on the weakest user. From (4.3), it is easy to see that the fixed point equation [1] is given by (4.5).

$$\mathbf{w}_{fp} = \frac{1}{c} \left[\frac{\mathbf{h}_1}{\mathbf{w}_{fp}^H \mathbf{h}_1} + \frac{\mathbf{h}_2}{\mathbf{w}_{fp}^H \mathbf{h}_2} \right] \qquad (4.5)$$

where $c \in \mathbb{R}^1$ is a constant introduced to scale the magnitude of $\mathbf{w}_{fp}$ to unity.

Proposition 1 *For $K = 2$, the multicast rate attained by a fixed point of the proposed AU*

[1] A fixed point of a mapping $\mathbf{f}(\cdot) : \mathbb{R}^N \to \mathbb{R}^N$ is any $\mathbf{x} \in \mathbb{R}^N$ satisfying $\mathbf{f}(\mathbf{x}) = \mathbf{x}$.

algorithm is $r_{min} = \log_2(1 + SNR_{\min,fp})$*, where*

$$SNR_{\min,fp} = \frac{1}{2}\Bigg[\min\left(\|\mathbf{h}_1\|^2, \|\mathbf{h}_2\|^2\right) + \min\left(\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}, \frac{\|\mathbf{h}_2\|}{\|\mathbf{h}_1\|}\right)|\mathbf{h}_1^H\mathbf{h}_2|\Bigg] \tag{4.6}$$

Proof 1 *See Appendix A1.*

It is instructive to compare this result to the max-min SNR in two special cases. The best situation for multicast beamforming is when $\mathbf{h}_2 = s\mathbf{h}_1$, for some $s \in \mathbb{C}$, i.e., the two user channel vectors are collinear. Then $\text{SNR}_{\min,fp} = \min(||\mathbf{h}_1||^2, ||\mathbf{h}_2||^2)$, which is equal to the optimum max-min SNR. The worst situation is when the two channel vectors are orthogonal, $\mathbf{h}_1^H\mathbf{h}_2 = 0$, in which case $\text{SNR}_{\min,fp} = \frac{1}{2}\min(||\mathbf{h}_1||^2, ||\mathbf{h}_2||^2)$, while the optimum max-min SNR can be easily shown to be $\frac{||\mathbf{h}_1||^2||\mathbf{h}_2||^2}{||\mathbf{h}_1||^2+||\mathbf{h}_2||^2}$. Without loss of generality, assume $||\mathbf{h}_1||^2 \leq ||\mathbf{h}_2||^2$. Then $\text{SNR}_{\min,fp} = \frac{1}{2}||\mathbf{h}_1||^2$, while the optimum max-min SNR is $\frac{||\mathbf{h}_1||^2}{\frac{||\mathbf{h}_1||^2}{||\mathbf{h}_2||^2}+1}$ [32], and satisfies $\frac{||\mathbf{h}_1||^2}{2} \leq \frac{||\mathbf{h}_1||^2}{\frac{||\mathbf{h}_1||^2}{||\mathbf{h}_2||^2}+1} \leq ||\mathbf{h}_1||^2$. We see that the AU fixed point is optimum in the case of balanced channel norms, and no worse than 3 dB off the optimum even in the worst (near-far) case. While this is clearly a toy example (e.g., the NP-hardness proof in [26] does not apply when $K = 2$), it is still satisfying to see that the simple AU iteration is so close to optimum in these two extreme cases. Motivated by these preliminary observations, we next consider the AU algorithm for general K.

4.3.1 Additive Update algorithm

In this section, we consider the case when there are $K \geq 2$ users and the matrices $\{\mathbf{R}_k\}_{k=1}^K \succeq \mathbf{0}$ have rank ≥ 1. An example of higher-rank scenario is when the objective is to maximize the minimum average SNR (instead of instantaneous SNR) among the users. In this case, $\{\mathbf{R}_k\}_{k=1}^K$ are the channel correlation matrices, which are full rank with probability one if the channel vectors are drawn from a continuous distribution. The motivation to consider average SNR is that instantaneous channel estimation and feedback requires much higher computation and communication overhead relative to infrequent channel correlation feedback.

For general K, the AU weight vector update is

$$\begin{aligned} \tilde{\mathbf{w}}_{n+1} &= \mathbf{w}_n + \alpha \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right) \mathbf{w}_n \\ \mathbf{w}_{n+1} &= \frac{\tilde{\mathbf{w}}_{n+1}}{\|\tilde{\mathbf{w}}_{n+1}\|} \end{aligned} \tag{4.7}$$

where α is a positive constant step size, and ε is a positive constant that is introduced for numerical stability. It can be seen here that the update direction in the $(n+1)^{th}$ iteration is the summation of SNR-gradient of k^{th} user (i.e., $\mathbf{R}_k \mathbf{w}_n$) weighted by the SNR attained by the k^{th} user in the n^{th} iteration (i.e., $\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n$), $\forall k = 1, 2, ..., K$. The AU update takes all the user channels into consideration, favoring weaker users more than stronger ones (i.e., those with lower SNR over those with higher SNR attained by $\mathbf{w}_n$ in the previous iteration). The fixed point equation of the AU algorithm is:

$$\mathbf{w}_{fp} = \frac{1}{c} \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k}{\mathbf{w}_{fp}^H \mathbf{R}_k \mathbf{w}_{fp} + \varepsilon} \right) \mathbf{w}_{fp} \tag{4.8}$$

Whereas the AU update has been intuitively developed and motivated up to this point, the following proposition reveals that it can be viewed as an approximation of a problem that is related to (but different from) the max-min-fair formulation $\mathbf{\Pi_1}$.

Proposition 2 *The beamforming vector obtained at the $(n+1)^{th}$ iteration of the AU algorithm is the solution of a strongly concave approximation (cf.* (4.9) *and* (4.10)*) of the proportionally fair [38] multicast beamforming problem $\mathbf{\Pi_2}$ at $\mathbf{w} = \mathbf{w}_n$.*

$$\mathbf{\Pi_9} \quad \mathbf{w}^* = \arg \max_{\|\mathbf{w}\|^2 = 1} \frac{1}{2} \sum_{k=1}^{K} \log \left(\mathbf{w}^H \mathbf{R}_k \mathbf{w} + \varepsilon \right)$$

Proof 2 *It can be shown that $\mathbf{\Pi_9}$ is a non-convex optimization problem [38] which is difficult to solve in general. Denote $f(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^{K} \log \left(\mathbf{w}^H \mathbf{R}_k \mathbf{w} + \varepsilon \right)$, the objective function of $\mathbf{\Pi_9}$. Consider a strongly concave approximation of $f(\mathbf{w})$ at the point $\mathbf{w} = \mathbf{w}_n$.*

$$f(\mathbf{w}) \approx f(\mathbf{w}_n) + \overbrace{\left(\sum_{k=1}^{K} \frac{(\mathbf{R}_k \mathbf{w}_n)^H (\mathbf{w} - \mathbf{w}_n)}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right)}^{\nabla f(\mathbf{w}_n)^H (\mathbf{w} - \mathbf{w}_n)} - \frac{\|\mathbf{w} - \mathbf{w}_n\|^2}{2\alpha} \tag{4.9}$$

Denote the right hand side of (4.9) *as $u(\mathbf{w}, \mathbf{w}_n)$. The sum of the first two terms in $u(\mathbf{w}, \mathbf{w}_n)$ is the first order Taylor series approximation of $f(\mathbf{w})$ at $\mathbf{w} = \mathbf{w}_n$. The last term in $u(\mathbf{w}, \mathbf{w}_n)$ is a*

proximal regularizer which is included to make $u(\mathbf{w}, \mathbf{w}_n)$ strongly concave ($\alpha > 0$). Instead of solving $\mathbf{\Pi_9}$, suppose that we iteratively solve $\mathbf{\Pi_{9r}}$ to obtain $\mathbf{w}_{n+1}$ from $\mathbf{w}_n$.

$$\mathbf{\Pi_{9r}} \quad \mathbf{w}_{n+1} = \arg \max_{\|\mathbf{w}\|^2=1} u(\mathbf{w}, \mathbf{w}_n)$$

From the definition of $u(\mathbf{w}, \mathbf{w}_n)$, it can be seen that the solution of $\mathbf{\Pi_{9r}}$ can be obtained in closed form as shown below.

$$\mathbf{w}_{n+1} = \frac{\mathbf{w}_n + \alpha \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right) \mathbf{w}_n}{\left\| \mathbf{w}_n + \alpha \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right) \mathbf{w}_n \right\|} \tag{4.10}$$

It can be seen from (4.7) *and* (4.10) *that the $(n+1)^{th}$ iterate of the AU algorithm is the solution of $\mathbf{\Pi_{9r}}$. Hence the AU algorithm obtains a beamforming vector that promotes proportional fairness in terms of the received SNR among the users.*

The natural next question is whether the AU algorithm converges. The following result shows that it does.

Theorem 1 *The iterates obtained from the AU algorithm converge to a KKT point of $\mathbf{\Pi_9}$, provided $0 < \alpha \leq \frac{2}{L_{\nabla_f}}$, where $L_{\nabla_f} = \sum_{k=1}^{K} \left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda_{\max}^2(\mathbf{R}_k)}{\varepsilon^2} \right) = \sum_{k=1}^{K} \left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\|\mathbf{h}_k\|^2}{\varepsilon^2} \right)$ (when $\mathbf{R}_k = \mathbf{h}_k \mathbf{h}_k^H$) is the Lipschitz constant of the gradient of $f(\mathbf{w})$ i.e., ∇_f and $\lambda_{\max}(\mathbf{R}_k) = \|\mathbf{h}_k\|^2$ (when $\mathbf{R}_k = \mathbf{h}_k \mathbf{h}_k^H$) is the maximum eigenvalue of $\mathbf{R}_k$.*

Proof 3 *See Appendix A2.*

4.3.2 Multiplicative Update algorithm

The proof of Theorem 1 in Appendix B requires the technical condition $0 < \alpha \leq \frac{2}{L_{\nabla_f}}$, but our experiments indicate that AU converges even for $\alpha > \frac{2}{L_{\nabla_f}}$. This motivates the following limiting version of the AU algorithm, which we will call the *Multiplicative Update* (MU) algorithm. The update step in the $(n+1)^{th}$ iteration is given by:

$$\tilde{\mathbf{w}}_{n+1} = \left(\sum_{k=1}^{K} \frac{\mathbf{R}_k \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right), \quad \mathbf{w}_{n+1} = \frac{\tilde{\mathbf{w}}_{n+1}}{\|\tilde{\mathbf{w}}_{n+1}\|} \tag{4.11}$$

The new iterate is the unit vector along the inverse SNR weighted SNR gradient direction of all the K users (i.e., only the direction vector of AU). It can be seen from (4.7) that the AU

update approaches the MU update as α increases. From (4.8) and (4.11) it can also be seen that the MU algorithm has the same fixed point condition as the AU algorithm. Simulations indicate that the MU algorithm always converges faster than and to the same fixed point as AU (the convergence rate of the AU approached that of MU as α increases from 0.01 to 1), without requiring any parameter tuning. The technical difficulty of using Theorem 1 for proving convergence of the MU algorithm at this point is that the proof in Theorem 1 places an upper bound on the step-size value of the gradient update, for the iterates to converge.

To gain more insight about the MU algorithm, consider the proportionally fair multicast beamforming problem $\mathbf{\Pi_9}$. Since the objective is non-concave, consider the maximization of its first order Taylor series ($\mathbf{\Pi_{9m}}$) about $\mathbf{w} = \mathbf{w}_n$.

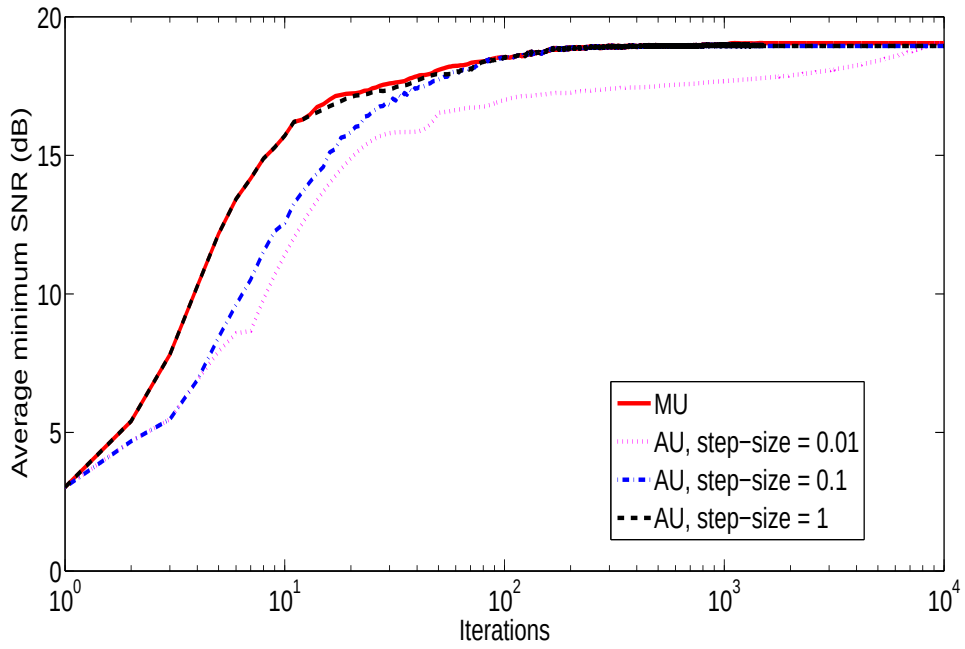

Figure 4.1: Comparison of convergence rate of MU and AU algorithm for $N = 25$, $K = 500$

$$\mathbf{\Pi_{9m}} \quad \arg \max_{\|\mathbf{w}\|^2=1} f(\mathbf{w}_n) + \left(\sum_{k=1}^{K} \frac{(\mathbf{R}_k \mathbf{w}_n)^H (\mathbf{w} - \mathbf{w}_n)}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \right)$$

where $f(\mathbf{w})$ is the objective function in $\mathbf{\Pi_9}$. It is straightforward to see that the solution of $\mathbf{\Pi_{9m}}$ can be obtained in closed form and is equal to the update in (4.11). Therefore the $(n+1)^{th}$

iterate of the MU algorithm is the solution of the linear approximation $\mathbf{\Pi_{9m}}$ of $\mathbf{\Pi_9}$ at $\mathbf{w} = \mathbf{w}_n$. The difference with the AU is that in $\mathbf{\Pi_{9m}}$ we do not have the proximal regularization term that we had in $\mathbf{\Pi_{9r}}$.

Fig. 4.1 compares the minimum SNR obtained from the MU with that of the AU algorithm for $N = 25,\ K = 500$ and for various step-sizes α when $\mathbf{h}_k \sim \mathcal{C}N(\mathbf{0}, \mathbf{I}), \forall k$ and $\mathbf{R}_k = \mathbf{h}_k \mathbf{h}_k^H$. The plot has been obtained after averaging over 100 Monte-Carlo simulations. It can be seen that MU and the AU algorithms converge to the same minimum SNR and the convergence rate of the AU approaches the MU algorithm as α increases from 0.01 to 1.

4.3.3 MU-SLA algorithm

An iterative successive linear approximation (SLA) algorithm has been proposed by Tran *et al.* [27] to approximately solve the following NP-hard QoS multicast beamforming problem $\mathbf{\Pi_{10}}$ [26].

$$\begin{aligned} \mathbf{\Pi_{10}} \qquad & \min_{\mathbf{w} \in \mathbb{C}^N} \|\mathbf{w}\|^2 \\ & \text{s.t.} \quad |\mathbf{w}^H \mathbf{h}_k|^2 \geq 1, \quad \forall k \in \{1, 2, ..K\} \end{aligned}$$

The SLA algorithm starts with a feasible initialization $\mathbf{w}_0$. The non-convex SNR constraints for all the K users are linearized around $\mathbf{w}_0$ using Taylor series expansion and the resulting convex quadratic program with linear constraints $\mathbf{\Pi_{10r}}$ is solved to obtain $\mathbf{w}_1$, which is subsequently used as the linearization point for the next iteration. The quadratic program solved in the $(n+1)^{th}$ SLA iteration is:

$$\begin{aligned} \mathbf{\Pi_{10r}} \qquad & \min_{\mathbf{w} \in \mathbb{C}^N} \|\mathbf{w}\|^2 \\ & \text{s.t.} \quad \|\mathbf{p}_{n,k}\|^2 + 2\mathbf{p}_{n,k}^T (\mathbf{v}_k - \mathbf{p}_{n,k}) \geq 1 \\ \mathbf{v}_k = [\Re(\mathbf{w}^H \mathbf{h}_k), \Im(\mathbf{w}^H \mathbf{h}_k)]^T, \forall k \in \{1, 2, ...K\} \end{aligned}$$

where $\Re(\cdot)$ ($\Im(\cdot)$) takes the real (imaginary) part of its argument, and $\mathbf{p}_{n,k} = [\Re(\mathbf{w}_n^H \mathbf{h}_k), \Im(\mathbf{w}_n^H \mathbf{h}_k)]^T$. Note that SLA was developed for the QoS rather than the max-min-fair formulation, but the two are equivalent up to normalization [26].

SLA solves a relatively complex quadratic program in each iteration, and the final result depends on the initialization. MU iterations, on the other hand, are very fast; but MU is geared towards proportional fairness, not max-min fairness, so the final result of MU can be refined to

improve its max-min fairness. This naturally motivates a two-step *MU-SLA* algorithm which can take advantage of the high-quality solutions obtained quickly via the MU algorithm, and the ability of the SLA algorithm to perform accurate 'last mile' minimum SNR refinement.

In more detail, the MU-SLA algorithm takes the solution obtained from the MU algorithm, denoted by $\mathbf{w}_{MU}$, scales it by the inverse square root of the minimum SNR attained using $\mathbf{w}_{MU}$ (to maintain feasibility for $\mathbf{\Pi_{10}}$) and then uses this vector to initialize and solve a *single* SLA iteration. The resulting vector determines the transmit beamforming vector direction, which is then scaled to the desired transmit power. Our experiments (presented in the simulation results section) indicate that, in terms of minimum SNR and hence multicast rate, MU-SLA is the new state-of-art, as it outperforms all other multicast beamforming algorithms available as of this writing.

4.4 Tx has no initial CSI about $\{\mathbf{R}_k\}_{k=1}^K$

Here, we assume that the matrices $\mathbf{R}_k,\ k \in \{1, 2, ..., K\}$ are channel correlation matrices and have rank ≥ 1. It is also assumed that all the receivers have limited computational / energy resources. As a result, the conventional training method of channel correlation matrix estimation at every Rx followed by quantization and feedback to the Tx cannot be used. In this section, we propose an online transmit beamforming algorithm for a single group multicast network where the Tx uses binary feedback from every Rx to simultaneously learn $\{\mathbf{R}_k\}_{k=1}^K$ and design beamforming vectors that attain a high minimum average SNR.

Time is divided into slots of length T seconds, with the duration of each slot long enough for every Rx to perform accurate power estimation. At time $tT + \tau$, where $t \in \mathbb{Z}$ is an integer slot index and $\tau \in [0, T)$ is 'fast time', the channel from the Tx to the k^{th} Rx is modeled as a zero-mean $N \times 1$ complex random vector $\mathbf{h}_k(tT + \tau)$, with a correlation matrix $\mathbf{R}_k = E\left(\mathbf{h}_k(tT + \tau)\mathbf{h}_k(tT + \tau)^H\right) \succeq \mathbf{0},\ \forall k \in \{1, 2, \ldots, K\}$ and $\forall t \in \mathbb{Z},\ \forall \tau \in [0, T)$. At every time slot t, the Tx sends a zero-mean unit-variance common message $x(tT + \tau)$ times a unit-norm complex beamforming vector $\mathbf{w}_t$ to all the receivers in the downlink. The received signal at the k^{th} Rx is

$$y_k(tT + \tau) = \mathbf{w}_t^H \mathbf{h}_k(tT + \tau)x(tT + \tau) + z_k(tT + \tau) \tag{4.12}$$

$\forall k \in \{1, 2, \ldots, K\}$, where $z_k(tT + \tau)$ is the additive noise at Rx k (assumed to be wide-sense stationary) with zero-mean, variance σ_k^2, and independent of $x(tT + \tau)$ and $\mathbf{h}_k(tT + \tau)\ \forall t$ and

$\forall \tau$. In the sequel, we assume that the received signal has been multiplied by $\frac{1}{\sigma_k}$ and absorb this factor into $\mathbf{h}_k(tT+\tau)$, for convenience. This makes the noise power equal to 1, and SNR equal to signal power.

In order to decode $x(tT+\tau)$, every Rx should estimate the complex scalar $\mathbf{w}_t^H \mathbf{h}_k(tT+\tau)$. One way to accomplish this task is to transmit pilot symbols at the start of every time slot to aid every Rx in this estimation process. An alternative is to use differential modulation. However, this estimation is less complex since it is a scalar estimation problem at every Rx. During every time slot t, the k^{th} Rx measures the average SNR $\mathbf{w}_t^H \mathbf{R}_k \mathbf{w}_t$ and compares it with a pre-determined threshold $\gamma_k(t)$. A '1' or a '0' is fed back by the k^{th} Rx when its average SNR is $\geq$ or $<$ $\gamma_k(t)$. It is assumed that there are no significant measurement errors (inequality flips) at the Rx or in the communication of the 1-bit feedback to the Tx. Based on the 1-bit feedback from the k^{th} Rx at time slot t, the Tx learns that

$$\begin{cases} \mathbf{w}_t^H \mathbf{R}_k \mathbf{w}_t \geq \gamma_k(t), & \text{when } s_k(t) = 1; \text{ or} \\ \mathbf{w}_t^H \mathbf{R}_k \mathbf{w}_t < \gamma_k(t), & \text{when } s_k(t) = 0, \end{cases} \tag{4.13}$$

where $k \in \{1, 2, ..., K\}$ and $s_k(t)$ is the 1-bit feedback from the k^{th} Rx at time slot t. Here, we propose an online Cognitive Multiplicative Update (CMU) algorithm to appropriately design a sequence of $\{\mathbf{w}_t, \{\gamma_k(t)\}_{k=1}^K\}_t$ that enables the Tx to learn $\{\mathbf{R}_k\}_{k=1}^K$ using binary feedback from all the receivers and attain a high value for the minimum average SNR among the users.

4.4.1 Cognitive Multiplicative Update Algorithm

Exploration-Exploitation Tradeoff - At every time slot, the Tx has to design the beamforming vector in such a way that it can not only infer new information about $\{\mathbf{R}_k\}_{k=1}^K$ (exploration), but also use the knowledge accumulated from the feedback bits in previous time slots, to attain a high value of minimum average SNR (exploitation) among all the receivers. Since the Tx does not have any initial CSI, it is desirable to focus on exploration initially. As time progresses (number of feedback bits from each Rx increases), and the Tx is progressively able to accurately estimate the $\{\mathbf{R}_k\}_{k=1}^K$ matrices, preference can be shifted to exploitation of the estimates of the channel correlation matrices of the users to attain a high minimum SNR. At the end of time slot t, the Tx has learned the following inequalities about $\{\mathbf{R}_k\}_{k=1}^K$ using the feedback bits from

every Rx.

$$\begin{aligned} \mathbf{w}_i{}^H \mathbf{R}_k \mathbf{w}_i &\geq \gamma_k(i), \quad \forall i \in \mathcal{G}_{k1}; \\ \mathbf{w}_j{}^H \mathbf{R}_k \mathbf{w}_j &< \gamma_k(j), \quad \forall j \in \mathcal{G}_{k2} \end{aligned} \tag{4.14}$$

where $\mathcal{G}_{k1} = \{i \ : 1 \leq i \leq t, \ s_k(i) = 1\}$, $\mathcal{G}_{k2} = \{j \ : 1 \leq j \leq t, \ s_k(j) = 0\}$, $\mathcal{G}_{k1} \bigcup \mathcal{G}_{k2} = \{1, 2, \ldots, t\}$ and $k \in \{1, 2, ..., K\}$.

Channel correlation matrix estimation

We propose to update $\hat{\mathbf{R}}_k(t), \ \forall k \in \{1, 2, ..., K\}$ (the Tx-side estimate of $\mathbf{R}_k$ at time t) as the solution of an optimization problem in $\mathbf{\Pi_{11}}$.

$$\begin{aligned} &\mathbf{\Pi_{11}} \\ &\hat{\mathbf{R}}_k(t) = \arg\max_{\mathbf{R}} \sum_{i \in \mathcal{G}_{k1}} \log\left(\text{Tr}\left(\mathbf{W}_i \mathbf{R}\right) - \gamma_k(i)\right) + \\ &\qquad\qquad \sum_{j \in \mathcal{G}_{k2}} \log\left(\gamma_k(j) - \text{Tr}\left(\mathbf{W}_j \mathbf{R}\right)\right) + \log\det \mathbf{R} \end{aligned}$$

where $\mathbf{W}_i = \mathbf{w}_i \mathbf{w}_i^H$ and the term $\mathbf{w}_i{}^H \mathbf{R} \mathbf{w}_i$ has been rewritten as $\text{Tr}(\mathbf{W}_i \mathbf{R})$. $\mathbf{\Pi_{11}}$ is a convex optimization problem which obtains the analytic center of the region formed by the linear inequalities till time slot t (4.14) for a particular k and the positive semi-definite cone [9, 10]. It can be solved efficiently using interior point methods.

Design of beamforming vector $\mathbf{w}_{t+1}$

Once the Tx updates $\{\hat{\mathbf{R}}_k(t)\}_{k=1}^K$, we formulate $\mathbf{\Pi_{12}}$ to design $\mathbf{w}_{t+1}$.

$$\begin{aligned} &\mathbf{\Pi_{12}} \\ &\mathbf{w}_{t+1} = \arg\max_{\|\mathbf{w}\|^2 = 1} \frac{1}{2} \sum_{k=1}^{K} \log\left(\mathbf{w}^H \hat{\mathbf{R}}_k(t) \mathbf{w} + \varepsilon\right) - \frac{\lambda_t}{2} \mathbf{w}^H \mathbf{V}_{p,t} \mathbf{w} \end{aligned}$$

where $\mathbf{V}_{p,t} = \mathbf{V}_{w,t} \mathbf{V}_{w,t}^H$, $\mathbf{V}_{w,t} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t]$, $\varepsilon > 0$ and λ_t is a non-increasing function of t, e.g., $\lambda_t = \frac{\lambda}{\lceil 0.1t \rceil}$, with $\lambda_1 \gg 1$. The objective function of $\mathbf{\Pi_{12}}$ comprises two terms. The first term promotes proportional fairness of SNR among users (if $\hat{\mathbf{R}}_k(t)$ is close to $\mathbf{R}_k$, $\forall k$) and the second term promotes the choice of a vector that is least similar to the beamforming vectors in all the previous time slots (since it minimizes the norm of the vector $\mathbf{V}_{w,t}^H \mathbf{w}$ whose i^{th} entry

is $\mathbf{w}^H\mathbf{w}_i, \forall i \leq t$). It has been shown that the proportional fairness function is non-concave. We replace the non-concave objective function of $\mathbf{\Pi_{12}}$ with the first-order Taylor series approximation, resulting in an optimization problem $\mathbf{\Pi_{12r}}$ which is obtained after removing the constant terms that are irrelevant to the optimization.

$$\mathbf{\Pi_{12r}}$$
$$\mathbf{w}_{t+1} = \arg \max_{\|\mathbf{w}\|^2=1} \left(\sum_{k=1}^{K} \frac{\hat{\mathbf{R}}_k(t)\mathbf{w}_t}{\mathbf{w}_t^H \hat{\mathbf{R}}_k(t)\mathbf{w}_t + \varepsilon} - \lambda_t \mathbf{V}_{p,t}\mathbf{w}_t \right)^H \mathbf{w}$$

The closed form solution of $\mathbf{\Pi_{12r}}$ is given by:

$$\begin{aligned} \tilde{\mathbf{w}}_{t+1} &= \left(\sum_{k=1}^{K} \frac{\hat{\mathbf{R}}_k(t)}{\mathbf{w}_t^H \hat{\mathbf{R}}_k(t)\mathbf{w}_t + \varepsilon} \right) \mathbf{w}_t - \lambda_t \mathbf{V}_{p,t}\mathbf{w}_t, \\ \mathbf{w}_{t+1} &= \frac{\tilde{\mathbf{w}}_{t+1}}{\|\tilde{\mathbf{w}}_{t+1}\|} \end{aligned} \tag{4.15}$$

The possibility of obtaining a beamforming vector update in closed form is the main motivation behind approximating the objective function of $\mathbf{\Pi_{12}}$ with the first-order Taylor series of the whole function as opposed to a concave approximation of the proportional fairness term alone (non-concave part), which would result in solving an optimization problem, thereby increasing the overall complexity significantly in every step.

The weight λ_t in $\mathbf{\Pi_{12}}$ decides the extent of preference given to the proportional fairness term (exploitation) in comparison with the diversity promoting term $\mathbf{w}^H\mathbf{V}_{p,t}\mathbf{w}$ (exploration). We propose to choose λ_t as a non-increasing function of t with $\lambda_1 \gg 1$ (for e.g. $\lambda_t = \frac{\lambda_1}{\lceil 0.1t \rceil}$). For small t, since $\lambda_t \gg 1$, the choice of weight vector is dictated by $\mathbf{w}^H\mathbf{V}_{p,t}\mathbf{w}$, thereby yielding diverse weight vectors that explore different directions for gathering information about $\{\mathbf{R}_k\}_{k=1}^K$. For large t, $\lambda_t \ll 1$, the Tx would have obtained sufficient information to accurately estimate $\{\mathbf{R}_k\}_{k=1}^K$ and preference shifts to the proportional fairness term, resulting in weight vectors that attempt to achieve a high minimum average SNR value among all the receivers. As $t \to \infty$, $\lambda_t \to 0$, the performance of the CMU algorithm asymptotically approaches that of the MU algorithm (where Tx has perfect CSI) if $\hat{\mathbf{R}}_k(t) \to \mathbf{R}_k$, $\forall k \in \{1, 2, ..., K\}$ (which is accomplished by appropriately designing the thresholds).

Design of thresholds $\{\gamma_k(t+1)\}_{k=1}^K$

Once $\mathbf{w}_{t+1}$ is designed, the Tx has to choose the thresholds $\{\gamma_k(t+1)\}_{k=1}^K$ for the K receivers. $\{\gamma_k(t+1)\}_{k=1}^K$ has to be chosen in such a way that the subsequent inequality inferred by the Tx from $\{s_k(t+1)\}_{k=1}^K$ significantly reduces the uncertainty about $\{\mathbf{R}_k\}_{k=1}^K$ at time slot t denoted by the region $\mathcal{P}_k(t) = \{\mathbf{R} : \mathbf{R} \succeq 0, {\mathbf{w}_i}^H \mathbf{R} \mathbf{w}_i \geq \gamma_k(i),\ \forall i \in \mathcal{G}_{k1}, {\mathbf{w}_i}^H \mathbf{R} \mathbf{w}_i < \gamma_k(i),\ \forall i \in \mathcal{G}_{k2},\ \mathcal{G}_{k1} \bigcup \mathcal{G}_{k2} = \{1, 2, \ldots, t\}\},\ \forall k \in \{1, 2, ..., K\}$. In this regard, we propose two threshold selection techniques.

Multiple threshold selection - Here, the Tx selects a unique threshold for every Rx at time slot $(t+1)$ which is given by

$$\gamma_k(t+1) = \mathbf{w}_{t+1}^H \hat{\mathbf{R}}_k(t) \mathbf{w}_{t+1}, \quad \forall k \in \{1, 2, ..., K\} \tag{4.16}$$

This selection (inspired by the ACCPM in convex optimization) ensures that the hyperplane corresponding to the inequality inferred about $\mathbf{R}_k$ at the Tx from $s_k(t+1)$, i.e., $\mathbf{w}_{t+1}^H \mathbf{R} \mathbf{w}_{t+1} = \gamma_k(t+1),\ \mathbf{R} \in \mathcal{C}^{N \times N},\ \mathbf{R} \succeq \mathbf{0}$ passes through the analytic center of $\mathcal{P}_k(t)$ i.e., $\hat{\mathbf{R}}_k(t)$, $\forall k \in \{1, 2, ..., K\}$. The analytic center $\hat{\mathbf{R}}_k(t)$ maximizes the product of distances to the defining hyperplanes and the p.s.d. cone in $\mathcal{P}_k(t)$ and gives the deepest interior point of $\mathcal{P}_k(t)$. Hence for a given $\mathbf{w}_{t+1}$, this choice of $\{\gamma_k(t+1)\}_{k=1}^K$ ensure that each of the K inequalities inferred by the Tx from $\{s_k(t+1)\}_{k=1}^K$ significantly reduces the uncertainty about $\mathbf{R}_k, \forall k \in \{1, 2, ..., K\}$ (irrespective of the orientation of the beamforming vector $\mathbf{w}_{t+1}$, since the analytic center is the deepest interior point of the uncertainty region). Since separate thresholds are chosen for every user, the reduction of the uncertainty region for $\mathbf{R}_k,\ k = 1, 2, ..., K$ is independent of each other. Therefore, using the convergence analysis of ACCPM [11], it can be shown that $\hat{\mathbf{R}}_k(t)$ is confined to a ball of radius r around $\mathbf{R}_k, \forall k \in \{1, 2, ..., K\}$ within $\mathcal{O}\left(\frac{N^2}{r^2}\right)$ iterations. As $t \to \infty, \lambda_t \to 0$, $\hat{\mathbf{R}}_k(t) \to \mathbf{R}_k, \forall k$ and the performance of the CMU algorithm asymptotically approaches that of the MU algorithm with perfect knowledge of $\{\mathbf{R}_k\}_{k=1}^K$ at the Tx, even though CMU starts with no CSIT. However, the downlink signaling overhead is very high, because the Tx has to communicate $\gamma_k(t+1)$ to the k^{th} Rx, $\forall k \in \{1, 2, .., K\}$ for every time slot. This overhead increases linearly with K.

Common threshold selection - Here, the Tx selects a common threshold $\gamma(t+1)$ for all the users

at time slot $(t+1)$.

$$\gamma(t+1) = \mathbf{w}_{t+1}^H \hat{\mathbf{R}}_{k_t}(t)\mathbf{w}_{t+1}, \text{ with } k_t := \text{mod}(t, K) + 1 \tag{4.17}$$

From (4.17), it can be seen that the common threshold at each time slot t is selected in a round-robin fashion. From the linear inequalities inferred by the Tx at time $t+1$, this selection ensures guaranteed reduction in the uncertainty region of user k_t only. Therefore, the uncertainty region of the channel correlation matrix of every user is certainly reduced at least once every K time slots. The convergence proof of ACCPM [11] can be used to show that $\hat{\mathbf{R}}_k(t)$ is confined to a ball of radius r around $\mathbf{R}_k(t), \forall k \in \{1, 2, ..., K\}$ within $\mathcal{O}\left(\frac{KN^2}{r^2}\right)$ iterations. In the worst-case, the convergence rate of common threshold selection will be K times slower than that of multiple threshold selection; but in practice, inequalities designed to reduce the uncertainty for one user will also reduce the uncertainty for other users. On the other hand, the per-slot communication overhead for the common threshold selection technique remains fixed even as K increases, since a single threshold is communicated to all K users. It should also be noted that the threshold communication can be avoided completely in this case, by keeping the thresholds at the users-side fixed for all t and scaling the transmit power instead. The set of linear inequalities inferred by the Tx at time $t+1$ can be modified as shown below.

$$\tilde{\mathbf{w}}_{t+1}^H \mathbf{R}_k \tilde{\mathbf{w}}_{t+1} \gtreqless 1, \forall k \ \in \ \{1, 2, ..., K\} \tag{4.18}$$

where $\tilde{\mathbf{w}}_{t+1} = \sqrt{\frac{1}{\gamma_{t+1}}}\mathbf{w}_{t+1}$ and $\gtreqless$ means that the Tx will choose the inequality as $\geq$ or $<$ based on whether the 1-bit feedback is a '1' or a '0' respectively. In order to account for the variation of transmit power, the power amplifiers should have a much wider linear operating region to avoid non-linearities in the measurement of the received signal power and average SNR.

4.5 Simulation Results

Tx has perfect CSI

The average minimum SNR of the AU / MU and the MU-SLA algorithms are compared with the SDR upper bound and other state-of-the-art algorithms, namely SDR-G [26] (10^5 randomizations), SLA [27], AM [28], LLI [30], QR algorithm [1] and the SB algorithm [32]. For the AU algorithm, the step-size is selected to satisfy the Lipschitz continuity condition. For Fig. 4.2 - Fig. 4.3 and Fig. 4.6 - Fig. 4.7, the channel vectors $\mathbf{h}_k$ are drawn from an i.i.d. $\mathcal{CN}(\mathbf{0}, \mathbf{I})$ distribution. The codes are executed using CVX [39] as the modeling language. The plots are obtained after averaging over 100 Monte-Carlo (MC) runs. For each run, the AU and the MU algorithms are executed until $\|\mathbf{w}_{n+1} - \mathbf{w}_n\| \leq 10^{-4}$ or a maximum of 1000 iterations, whichever comes first.

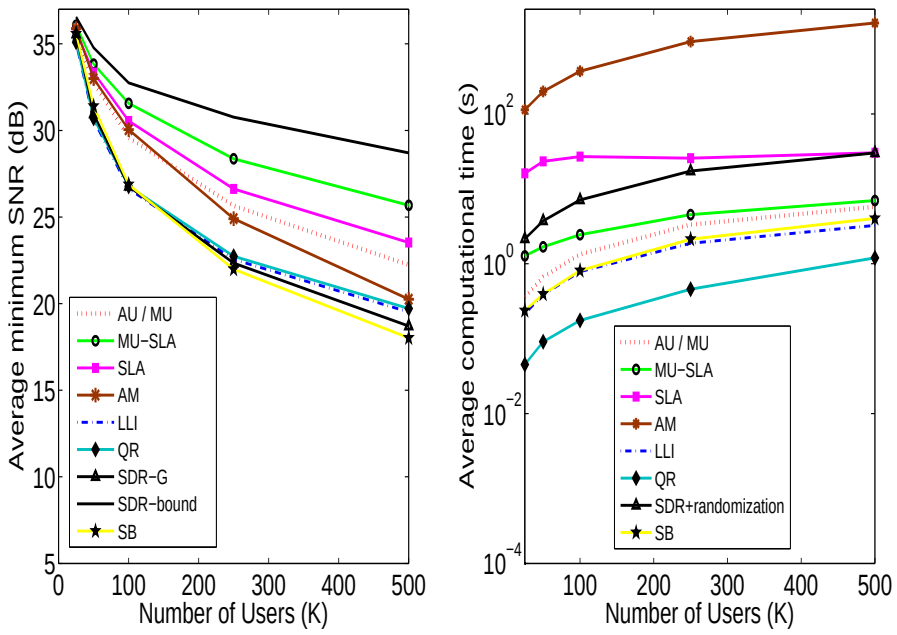

Figure 4.2: Comparison of average minimum SNR and computation time versus K for $N = 20$ antennas when the user channels are drawn from an i.i.d. Gaussian distribution.

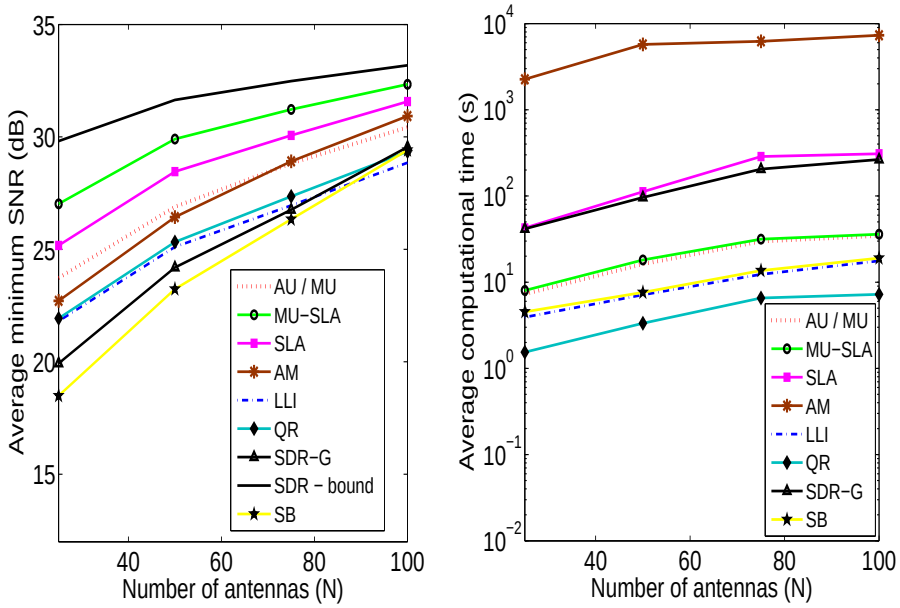

Figure 4.3: Comparison of average minimum SNR and computation time versus N for $K = 450$ users when the user channels are drawn from an i.i.d. Gaussian distribution.

Fig. 4.2 compares the average minimum SNR and the average computation time [2] (averaged over MC runs) of all the algorithms versus the number of users (K) for a fixed $N = 20$ transmit antennas. Fig. 4.3 compares the variation of the same metrics with the number of transmit antennas N for $K = 450$. It can be seen that the MU-SLA algorithm attains the highest minimum SNR among all the algorithms (≈ 0.8 dB above SLA for $N = 20, K = 500$); whereas the minimum SNR attained by the MU / AU algorithm is close but inferior to the SLA algorithm (≈ 1 dB below SLA for $N = 20, K = 500$). It is also interesting to note that the minimum SNR of the AU / MU algorithm is slightly lower than the AM algorithm when K is small, but it outperforms the latter when $K > 150$ (≈ 2 dB above AM for $N = 20, K = 500$). A similar trend is seen in Fig. 4.3 also in terms of minimum SNR performance and average computational time. The average minimum SNR increases with N for a fixed K because the Tx has higher degrees of freedom for a larger value of N which enables it beamform efficiently in space to attain a higher minimum SNR among the users.

[2] The computation time depends on software, hardware, coding quality and other implementation issues. We used standard / author-supplied codes where possible and carefully coded our implementations to ensure that the results are fair and indicative of the *relative* complexity of the different algorithms.

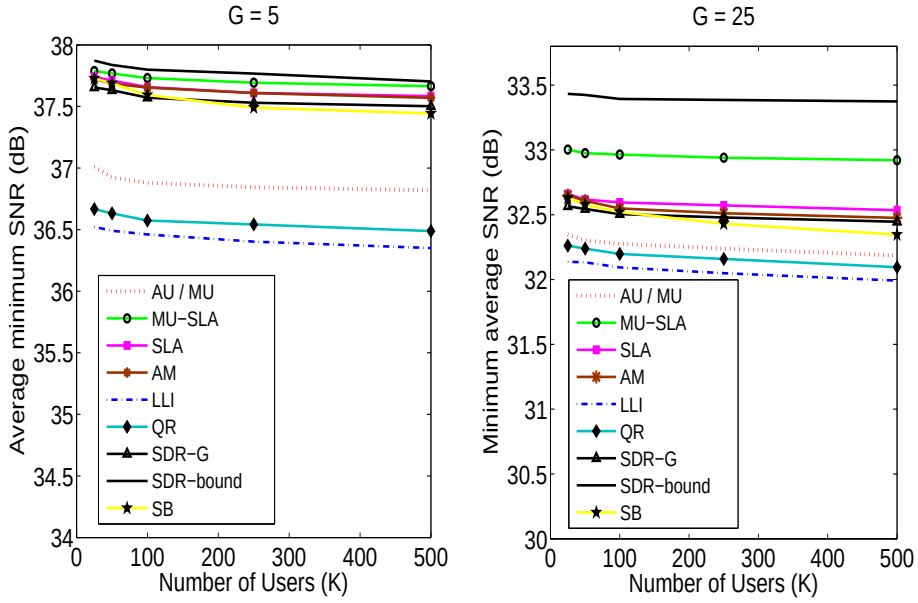

Figure 4.4: Comparison of average minimum SNR versus K for N = 20 antennas when channels to all users are drawn from a mixture of $G = 5$ (left) and $G = 25$ (right) Rician distributions.

The average computation time of the MU-SLA algorithm is very close to the AU / MU algorithm ($\mathcal{O}(10^1)$s for $K = 450$), both of which are significantly less than the SLA, the SDR-G ($\mathcal{O}(10^2)$s) and the AM ($\mathcal{O}(10^3)$s) algorithms. The MU-SLA algorithm outperforms the state-of-the-art by attaining the highest minimum SNR at a computational complexity similar to or much lower than all the high-performance algorithms. The gap between the SDR upper bound and the average minimum SNR achieved by the algorithms increases with $\frac{K}{N}$ ($\approx$ 0.4 dB for $K = 25, N = 20$ to $\approx$ 3.2 dB for $K = 500, N = 20$ for the MU-SLA algorithm in Fig. 4.2 and $\approx$ 0.5 dB for $K = 450, N = 100$ to $\approx$ 3 dB for $K = 450, N = 10$ for the MU-SLA algorithm in Fig. 4.3). This is in concurrence with the results on multicast capacity in [40]: it is difficult to attain a high minimum SNR among the users as $\frac{K}{N}$ increases when the channels are drawn from an i.i.d. zero-mean complex Gaussian distribution.

Fig. 4.4 and Fig. 4.5 compare the average minimum SNR when the channels are drawn from a Rician distribution for $N = 20$. This simulation models a practical scenario, where the users are clustered into multiple spatial groups in a given area (e.g., University campus), and the channel vectors of the users in a group are correlated. For this simulation, the users are

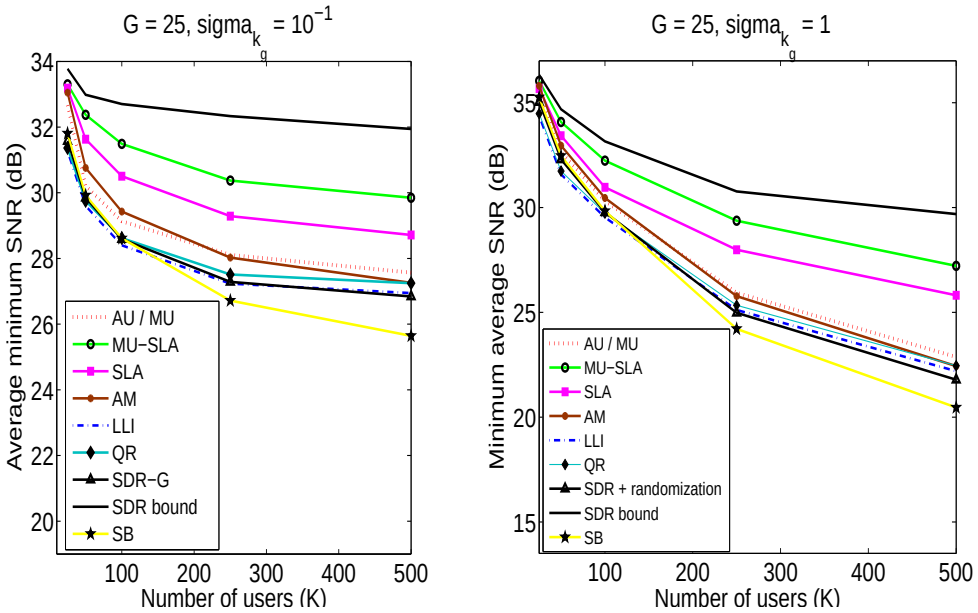

Figure 4.5: Comparison of average minimum SNR versus K for N = 20 antennas when channels to all users are drawn from a mixture of $G = 25$ Rician distributions with $\sigma_{k_g} = 10^{-1}$ (left) and $\sigma_{k_g} = 1$ (right).

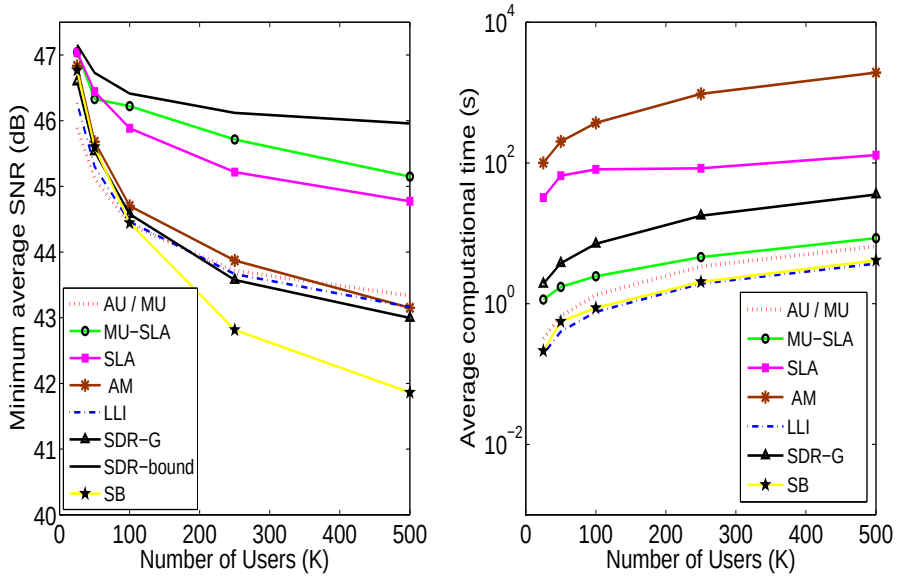

Figure 4.6: Comparison of average minimum SNR and computation time versus K for $N = 20$ antennas, when $\{\mathbf{R}_k\}_{k=1}^{K}$ are full rank.

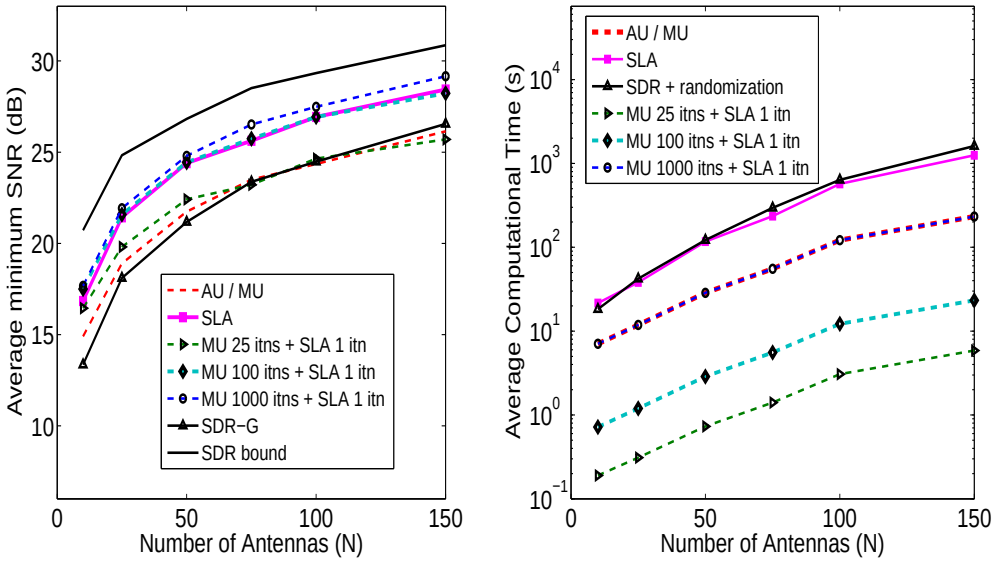

Figure 4.7: Comparison of minimum average SNR and computation time versus N for $K = 500$ users, when channels to all users are drawn from an i.i.d. complex Gaussian distribution.

clustered into G spatial groups and the channel to every user is modeled as follows:

$$\mathbf{h}_k = \mathbf{h}_{LOS,k_g} + \sigma_{k_g} * \left(\text{randn}(N,1) + \sqrt{-1} * \text{randn}(N,1)\right)$$

where $k_g = \text{mod}(k, G)$ and $\mathbf{h}_{LOS,k_g}, k_g \in \{0, 1, ...G-1\}$ are the line-of-sight channels (common to all users in a group) from the Tx to the various clusters. If $\sigma_{k_g} \ll 1$, then the common line-of-sight component dominates the channels of the users in group k_g, thereby making them highly correlated. In Fig. 4.4, $\sigma_{k_g} = 10^{-3}, \forall k_g \in \{0, 1, 2, .., G-1\}$ and $G \in \{5, 25\}$. Since $\sigma_{k_g} \ll 1$, the correlation of user-channels in a group is very high and the whole system can be approximated as single Tx with N antennas serving G users (instead of K).

As expected, the variation in the average minimum SNR of all the algorithms with K is rather small, because G is fixed. Also, it can be seen from Fig. 4.4 that the gap between the average minimum SNR for all the algorithms and the SDR bound is much less than that in Fig. 4.2 (1.3 dB for $G = 5, K = 450$ and 1.5 dB for $G = 25, K = 450$ vs. 9 dB in Fig. 4.2 for SDR-G at $K = 450$). For $G = 5$, all the algorithms perform close to optimal SDR bound because the Tx has more degrees of freedom than number of groups ($N > G$) and the gap increases as G increases to 25 and $N < G$.

In Fig. 4.5, the value of G is fixed at 25 and $\sigma_{k_g} \in \{0.1, 1\}$. In Fig. 4.5, the variation in the minimum SNR with respect to K increases as σ_{k_g} increases. As σ_{k_g} increases, the LOS component becomes less dominant, the correlation between different channels to users in a particular group decreases, and the performance approaches the case where the channels to all the K users are drawn from an i.i.d. complex Gaussian distribution with zero mean (in Fig. 4.2).

Fig. 4.6 compares the minimum average SNR and the average computation time when $\{\mathbf{R}_k\}_{k=1}^K$ are full-rank channel correlation matrices. For each Monte-Carlo run, $\mathbf{R}_k = \mathbf{M}_k\mathbf{M}_k^H$, $\forall k \in \{1, 2, ..., K\}$, where the entries of $\mathbf{M}_k$ are drawn from an i.i.d. $\mathcal{CN}(0, 1)$ distribution. The QR algorithm is not used for comparison because it is not applicable when the matrices $\{\mathbf{R}_k\}_{k=1}^K$ have rank > 1. The SLA algorithm has been modified appropriately to work in this case (see Appendix D). It can be seen that the minimum SNR attained by MU-SLA algorithm is higher than all the other algorithms (0.5 dB above SLA at $K = 450$) and the minimum SNR attained by the AU / MU algorithm is lesser than the SLA algorithm (1.8 dB below SLA), but still higher than SDR-G (for $K > 150$) and AM for $K = 450$. The average computation time for the MU-SLA and the AU / MU algorithms ($\mathcal{O}(10^0)s$ at $K = 450$) are order(s) of magnitude lower than SDR-G ($\mathcal{O}(10^1)s$), SLA ($\mathcal{O}(10^2)s$) and AM ($\mathcal{O}(10^3)s$).

In Fig. 4.7, we explore the SNR performance-computation time trade off for the MU-SLA algorithm. For this simulation, the MU algorithm is run for 25, 100 and 1000 iterations. The beamforming vectors at the end of these iterations are scaled as mentioned in section III-C and used as initialization for one SLA iteration. The resultant beamforming vector direction is scaled to the required transmit power. From Fig. 4.5, it can be seen that the minimum SNR of the MU-SLA algorithm improves with more iterations of the MU algorithm (e.g., at $N = 150$, 1000 MU iterations with 1 SLA iteration attains $\approx$ 3.5 dB higher minimum SNR than 25 MU iterations with 1 SLA iteration). This means that the SLA algorithm is quite sensitive to the quality of the initialization. On the other hand, the average computation time required for the MU-SLA algorithm increases with the number of MU iterations. From the plots, the best trade off seems to be the MU-SLA algorithm with 100 MU iterations and 1 SLA iteration, since its performance is $\approx$ 0.5 dB below the MU-SLA algorithm with 1000 MU iterations and 1 SLA iteration, while the average computation time is $\mathcal{O}(10^0)$ seconds at $N = 150$.

Tx has no initial CSI

Fig. 4.8 and Fig. 4.9 compare the minimum average SNR of the CMU algorithm with that of MU (perfect CSIT) for ($N = 5$, $K = 20$), and ($N = 20$, $K = 50$) respectively. The value of λ_t is chosen as $\lambda_t = \frac{1}{\lceil 0.1t \rceil}$. For each MC run, $\mathbf{R}_k = \mathbf{M}_k \mathbf{M}_k^H$, $\forall k \in \{1, 2, ..., K\}$, where the entries of $\mathbf{M}_k$ are drawn from an i.i.d. $\mathcal{CN}(0, 1)$ distribution. The dotted horizontal red line in each of the figures is the minimum average SNR attained by the MU algorithm with perfect CSIT.

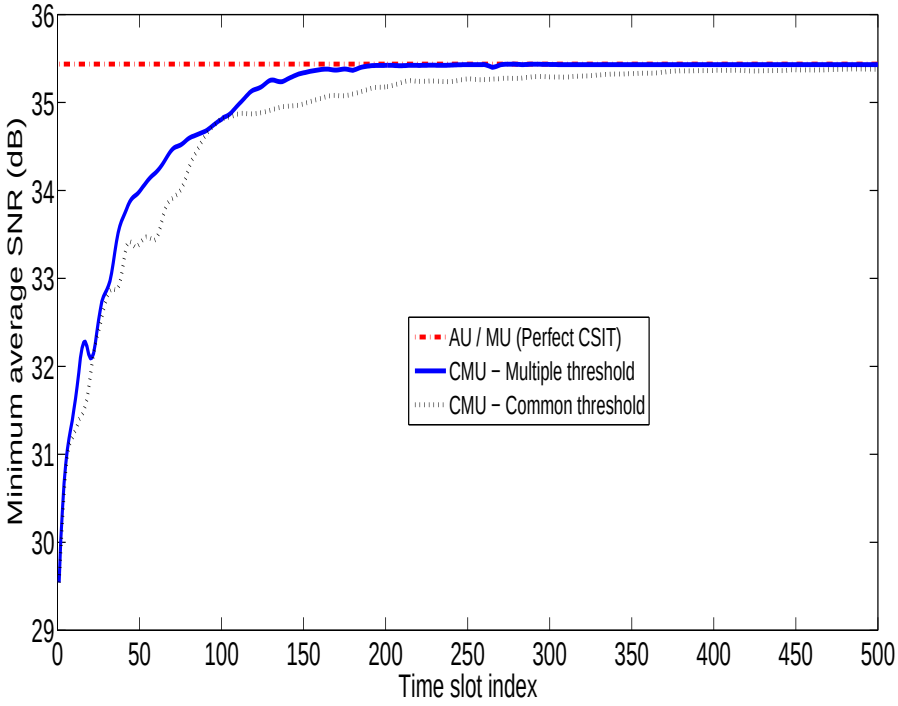

Figure 4.8: Average minimum SNR of CMU algorithm for $N = 5$, $K = 20$

It can be seen from Fig. 4.8 and Fig. 4.9 that the performance of the CMU algorithm with both threshold selection methods converges to the performance attained with perfect CSIT. It is evident that the CMU with multiple threshold selection converges faster than the common threshold selection (≈ 200 vs. 450 time slots when $N = 5, K = 20$ and ≈ 2500 vs. 4800 time slots when $N = 20, K = 50$).

It is interesting to note that the common threshold method is only ≈ 2.5 times slower for

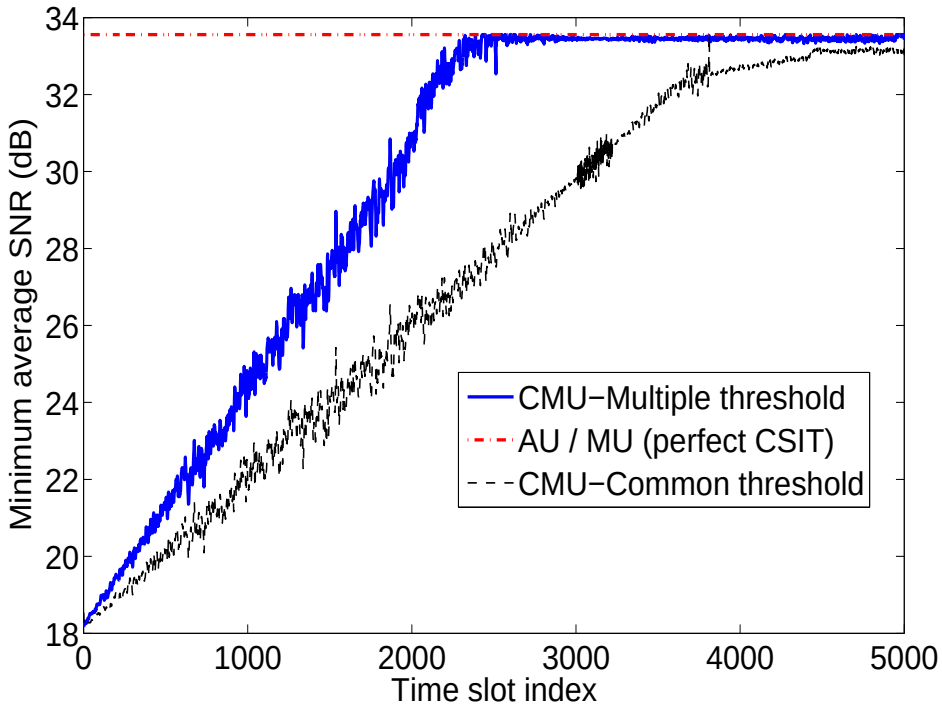

Figure 4.9: Average minimum SNR of CMU algorithm for $N = 20$, $K = 50$

$N = 5$ and ≈ 2 times slower for $N = 20$ than the multiple threshold method (the worst-case scenario is $K = 20$ times slower in the first case and $K = 50$ times slower in the second). As alluded to earlier, this is due to the fact that the common threshold selection not only reduces the uncertainty in the channel correlation matrix of the user chosen via round-robin selection, but also decreases the uncertainty of the other users as well, although not to the extent accomplished by multiple threshold selection in every time slot.

4.6 Summary

We considered the single group multicast network beamforming problem and proposed novel adaptive algorithms of low complexity, namely the AU, MU and MU-SLA, that obtain transmit beamforming vectors which attain a high minimum SNR when the Tx has perfect CSI. The fixed point equation of AU and MU was studied, and proof of convergence of the AU algorithm to a KKT point of proportionally fair beamforming was established. Extensive simulations were used to show that the MU-SLA algorithm outperforms the prior state-of-art (SLA) in terms of minimum SNR (and therefore multicast rate), whereas the AU / MU algorithm attains minimum SNR close to SLA, at far lower complexity.

When the Tx does not have any initial CSI and the receivers have limited computational resources, an online CMU algorithm based on ACCPM and appropriate threshold selection techniques were proposed to enable the Tx to learn the channel correlation matrices and design long-term transmit beamforming vectors simultaneously, using binary feedback from every Rx. Asymptotic convergence of the CMU algorithm to perfect-CSIT performance was established by invoking convergence results for the ACCPM from convex optimization, and verified in pertinent simulations. A variation of the common threshold selection technique was proposed to eliminate the threshold communication overhead at the cost of varying the transmit power (and the associated difficulties this imposes on power amplifiers). It was interesting to see that the convergence rate of the common threshold selection technique was much faster than the worst-case rate, thereby making it a strong contender for multicasting scenarios with only limited on-line feedback from the user terminals.

Last but not least, it is interesting to note that the proposed algorithms appear to be useful in the context of wireless power transfer - a concept that has gained traction recently, as a means of charging electrical devices without using power cables [41]. In this regard, the problem of maximizing the total sum-power harvested at all the receivers of a multi-user MISO system subject to a transmit power constraint was considered in [41]. The authors approximated the non-convex optimization problem using SDR and proved optimality of the SDR solution. Maximizing the sum-power harvested at all the receivers may lead to non-uniform power harvesting across different receivers. A 'fair' alternative is to maximize the minimum power harvested at every Rx subject to transmit power constraints. This latter formulation is exactly same as $\mathbf{\Pi_1}$, where $\mathbf{w}$ is the transmit beamforming vector, $\mathbf{R}_k$ is the channel correlation matrix of the k^{th} Rx

and $\mathbf{w}^H\mathbf{R}_k\mathbf{w}$ represents the average power harvested at the k^{th} Rx. Therefore, the proposed algorithms can be used verbatim for this application as well.

Chapter 5

Fast Feasibility Pursuit of Non-Convex QCQPs with Applications to Cognitive Radio Multicast Beamforming

5.1 Introduction

Quadratically constrained quadratic programs (QCQPs) have a wide range of applications in signal processing and wireless communications [42]. Non-convex QCQPs are NP-hard in general. Existing approaches relax the non-convexity using semidefinite relaxation (SDR) or linearize the non-convex part and solve the resulting convex problem. However, these techniques are seldom successful in even obtaining a feasible solution when there are non-convex constraints. In this chapter we consider the feasibility problem for a special class of non-convex QCQPs with two-sided constraints where the associated matrices are positive semi-definite (p.s.d.). This problem can be formulated as shown in $\mathbf{\Pi_{15}}$.

$$
\begin{aligned}
\mathbf{\Pi_{15}} \quad & \arg \max_{\mathbf{w} \in \mathbb{C}^N} 1 \\
s.t. \quad & \mathbf{w}^H \tilde{\mathbf{R}}_k \mathbf{w} \geq \gamma_k, \; \forall k = 1, 2, ..., K \\
& \mathbf{w}^H \tilde{\mathbf{G}}_j \mathbf{w} \leq \tau_j, \; \forall j = 1, 2, ..., J \\
& \|\mathbf{w}\|^2 \leq 1
\end{aligned}
$$

where $\tilde{\mathbf{R}}_k \succeq \mathbf{0}$ and $\tilde{\mathbf{G}}_j \succeq \mathbf{0}$, $\gamma_k > 0, \tau_j > 0 \; \forall k = 1, 2, ..., K; \; j = 1, 2, ..., J$. This can be reformulated as shown in $\mathbf{\Pi_{15a}}$.

$$\begin{aligned} \mathbf{\Pi_{15a}} \qquad & \arg \max_{\mathbf{w} \in \mathbb{C}^N} 1 \\ s.t. \qquad & \mathbf{w}^H \mathbf{R}_k \mathbf{w} \geq 1, \; \forall k = 1, 2, ..., K \\ & \mathbf{w}^H \mathbf{G}_j \mathbf{w} \leq 1, \; \forall j = 1, 2, ..., J \\ & \|\mathbf{w}\|^2 \leq 1 \end{aligned}$$

where $\mathbf{R}_k = \tilde{\mathbf{R}}_k / \gamma_k \succeq \mathbf{0}$, $\mathbf{G}_j = \tilde{\mathbf{G}}_j / \tau_j \succeq \mathbf{0}$, $\forall k = 1, 2, ..., K; \; j = 1, 2, ..., J$. Assume that there exists at least one feasible solution to the problem $\mathbf{\Pi_{15a}}$. It has been shown in [43] that $\mathbf{\Pi_{15a}}$ is NP-hard in general (except when $K + J < 3$, when $\mathbf{\Pi_{15a}}$ can be solved in polynomial time using the semi-definite relaxation (SDR) technique [21, 22]). A method using SDR followed by a Gaussian randomization procedure (SDR-G) has been proposed by [10, 43] that attempts to obtain feasible solutions of $\mathbf{\Pi_{15a}}$. However, our simulations show that the probability of obtaining a feasible solution using the SDR-G algorithm deteriorates considerably as the problem dimension (N) and the number of constraints ($K + J$) increases. Furthermore, the worst-case complexity of the SDR-G algorithm is also very high ($\mathcal{O}(N^2 + K + J)^{3.5}$ [10, 44]), which makes it a computationally taxing algorithm for large N, K and J.

The state-of-the-art for $\mathbf{\Pi_{15a}}$ is the Feasible Point Pursuit using Successive Convex Approximation (FPP-SCA) algorithm [45]. The FPP-SCA algorithm starts with an infeasible solution, say $\mathbf{x}_0$ and the non-convex part of all the constraints are linearized about this point. This is followed by additional of non-negative slack variables to the constraints in order to maintain feasibility about $\mathbf{x}_0$. The resulting convex optimization problem is solved to obtain $\mathbf{x}_1$ and the above procedure is repeated till convergence of the sum of slack variables. Simulations show that the FPP-SCA is successful in obtaining a feasible solution in polynomial time (worst case complexity $\mathcal{O}(N + K + 3J)^{3.5}$ per iteration), with high probability which is remarkable considering the fact the feasibility problem is NP-hard. Even though the worst-case complexity per iteration of FPP-SCA is lower than SDR-G, it still involves solving multiple convex optimization problems until convergence, which makes it a computationally involved algorithm for large N, K and J.

5.1.1 Contributions

In this chapter, we propose a low-complexity algorithm named Projected AU-SLA (Additive Update with Successive Linear Approximation) for obtaining feasible solutions of $\mathbf{\Pi_{15a}}$ with high probability (similar to FPP-SCA). The Projected AU-SLA algorithm considers a non-convex optimization problem where the objective function is a variation of the log-barrier of the non-convex constraints in $\mathbf{\Pi_{15a}}$ subject to the convex constraints in $\mathbf{\Pi_{15a}}$. The Projected AU-SLA algorithm starts with a random infeasible vector and at every step, the previous iterate is updated along the gradient of the objective function evaluated using the previous iterate. The resultant vector is projected onto the set of all the convex constraints using a Cyclic Dykstra's projection algorithm which has a lower computational complexity than solving a convex optimization problem to obtain the optimal projection. It will be shown that the iterates converge to a KKT point of the non-convex optimization considered by Projected AU-SLA. Once the iterates converge, the resulting vector is further refined using *a single SLA iteration* [27]. Simulations show that the Projected AU-SLA outperforms SDR-G and performs very close to the FPP-SCA algorithm in terms of attaining feasibility with a very high probability. Furthermore, the computational complexity of Projected AU-SLA is much lower than the FPP-SCA and SDR-G algorithms, which is very remarkable considering the fact that the feasibility problem for non-convex QCQPs itself is NP-hard.

The Projected AU-SLA algorithm is further extended to the application of cognitive multicast beamforming, where it is used for designing beamforming vectors at the secondary transmitter (Tx) of a single-group multicast network to maximize the downlink multicast rate in the secondary network while restricting the interference to all the receivers in the primary network. It is shown that the cognitive multicast beamforming problem is a special case of $\mathbf{\Pi_{15a}}$ where the problem is NP-hard; but it is very easy to always find a feasible solution. Simulations are used to compare the multicast rate achieved by the proposed algorithm with the state-of-the-art and it is shown that the proposed algorithm achieves a favorable performance complexity trade off as compared to other existing related algorithms (i.e., high multicast rate at low complexity).

5.2 Problem Formulation

Consider the following optimization problem formulated in $\mathbf{\Pi_{16}}$

$$\begin{aligned} \mathbf{\Pi_{16}} \qquad & \arg \max_{\mathbf{w}\in\mathbb{C}^N} \min_{k=1,2,...,K} \mathbf{w}^H \mathbf{R}_k \mathbf{w} \\ s.t. \qquad & \mathbf{w}^H \mathbf{G}_j \mathbf{w} \leq 1,\ \forall j = 1,2,...,J \\ & \|\mathbf{w}\|^2 \leq 1 \end{aligned}$$

where $\{\mathbf{R}_k\}_{k=1}^K$ and $\{\mathbf{G}_j\}_{j=1}^J$ are defined in $\mathbf{\Pi_{15a}}$. This problem can be reformulated as shown in $\mathbf{\Pi_{16a}}$.

$$\begin{aligned} \mathbf{\Pi_{16a}} \qquad & \arg \max_{\mathbf{w}\in\mathbb{C}^N, t\in\mathbb{R}_+} t \\ s.t. \qquad & \mathbf{w}^H \mathbf{R}_k \mathbf{w} \geq t,\ \forall k = 1,2,...,K \\ & \mathbf{w}^H \mathbf{G}_j \mathbf{w} \leq 1,\ \forall j = 1,2,...,J \\ & \|\mathbf{w}\|^2 \leq 1 \end{aligned}$$

Now suppose that the optimal solution of $\mathbf{\Pi_{16a}}$ (i.e. $t^*, \mathbf{w}^*$) is available from an oracle and that $t^* \geq 1$, then from $\mathbf{\Pi_{16a}}$ and $\mathbf{\Pi_{15a}}$, it can be seen that there is at least one feasible solution to the non-convex QCQP in $\mathbf{\Pi_{15a}}$ i.e., $\mathbf{w}^*$. $\mathbf{\Pi_{16a}}$ has been considered in the context of cognitive multicast beamforming by Phan *et al.*[46] for designing beamforming vectors that maximize the minimum SNR among the users of a single-group multicast network in the secondary system of a cognitive radio network, while restricting the interference caused at every primary receiver (Rx) below a known threshold (i.e., 1 in $\mathbf{\Pi_{16a}}$). In this context, the matrices $\mathbf{R}_k$ and $\mathbf{G}_j$, $\forall k = 1,2,...,K;\ j = 1,2,...,J$ are the correlation matrices / outer products of the channels from the secondary Tx to the k^{th} secondary Rx and j^{th} primary Rx respectively and are by definition p.s.d. . However, it has been shown that $\mathbf{\Pi_{16a}}$ is NP-hard in general (except when $K + J < 3$, [21, 22]). Therefore, we consider efficient approximations to obtain sub-optimal solutions (say $\mathbf{w}_{so}^*$) of $\mathbf{\Pi_{16a}}$. If the objective function value t_{so}^* attained by $\mathbf{w}_{so}^*$ is such that $t_{so}^* \geq 1$, then we have successfully found a feasible solution of $\mathbf{\Pi_{15a}}$ in polynomial time.

5.2.1 Successive Linear Approximation approach

An effective approximation of $\mathbf{\Pi_{16a}}$ is a modified version of the Successive Linear Approximation (SLA) algorithm proposed by Tran *et al.* [27]. The SLA approach starts with a random

initial vector $\tilde{\mathbf{w}}_1$ which is scaled by a factor β to obtain $\mathbf{w}_1 = \beta\tilde{\mathbf{w}}_1$ which satisfies the constraints $\mathbf{w}^H\mathbf{G}_j\mathbf{w} \leq 1,\ \forall j = 1, 2, ..., J$ and $\|\mathbf{w}\|^2 \leq 1$. The vector $\mathbf{w}_1$ is a feasible vector because, it is always possible to find a corresponding $t > 0$ such that $(\mathbf{w}_1, t)$ satisfies all the constraints in $\mathbf{\Pi_{16a}}$. For $n \geq 1$, $\mathbf{w}_{n+1}$ is obtained from $\mathbf{w}_n$ by the solving the following optimization problem in an iterative fashion until the iterates converge.

$\mathbf{\Pi_{17}}$

$$\begin{aligned} & \mathbf{w}_{n+1} = \arg \max_{\mathbf{w}\in\mathbb{C}^N,\, t\in\mathbb{R}_+} t \\ s.t. \quad & \mathbf{w}_n^H\mathbf{R}_k\mathbf{w}_n + 2\Re\left[(\mathbf{R}_k\mathbf{w}_n)^H(\mathbf{w}-\mathbf{w}_n)\right] \geq t, \quad \forall k \in \{1, 2, ...K\} \\ & \mathbf{w}^H\mathbf{G}_j\mathbf{w} \leq 1,\ \forall j = 1, 2, ..., J \\ & \|\mathbf{w}\|^2 \leq 1 \end{aligned}$$

It has been shown in [27] that the iterates obtained by successively solving $\mathbf{\Pi_{17}}$ converge to a Karush-Kuhn-Tucker (KKT) point of $\mathbf{\Pi_{16a}}$. The worst case complexity per iteration of $\mathbf{\Pi_{17}}$ is $\mathcal{O}(N + K + J)^{3.5}$ which is slightly lesser than that of the FPP-SCA algorithm [45]. However, the focus here is to design low complexity algorithms for finding feasible solutions of $\mathbf{\Pi_{16a}}$ and in this regard we propose the following Projected Additive Update with Successive Linear Approximation (Projected AU-SLA) algorithm which is described in detail below.

5.3 Projected AU-SLA algorithm

The algorithm starts with a random beamforming vector $\tilde{\mathbf{w}}_1$ and is scaled by a factor $\sqrt{\gamma}$ to obtain $\mathbf{w}_1$, where

$$\gamma = \min\left(\frac{1}{\|\tilde{\mathbf{w}}_1\|^2}, \max\left[\frac{1}{\tilde{\mathbf{w}}_1^H\mathbf{G}_j\tilde{\mathbf{w}}_1}\right]_{j=1}^{J}\right) \tag{5.1}$$

The scaling is done in order to ensure that $\mathbf{w}_1$ is a feasible vector for the optimization problem $\mathbf{\Pi_{16}}$. For the n^{th} iteration ($n \geq 1$), $\mathbf{w}_{n+1}$ is obtained from $\mathbf{w}_n$ as follows.

$$\tilde{\mathbf{w}}_{n+1} = \mathbf{w}_n + \alpha\left(\sum_{k=1}^{K}\frac{\mathbf{R}_k}{\mathbf{w}_n^H\mathbf{R}_k\mathbf{w}_n + \varepsilon}\right)\mathbf{w}_n \tag{5.2}$$

$$\mathbf{w}_{n+1} = \mathcal{P}_{\mathcal{W}}\left[\tilde{\mathbf{w}}_{n+1}\right] \tag{5.3}$$

where α is a positive constant step size, ε is a positive constant that is introduced for numerical stability and $\mathcal{P}_{\mathcal{W}}[\mathbf{w}]$ is the projection of $\mathbf{w}$ onto the feasible set $\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\|^2 \leq$

$1, \mathbf{w}^H\mathbf{G}_j\mathbf{w} \leq 1; \forall j = 1, 2, ..., J\}$. From (5.2), it can be seen that $\tilde{\mathbf{w}}_{n+1}$ is a sum of $\mathbf{w}_n$ and a fixed step times a direction vector, which is a weighted sum of the gradients of the terms $\mathbf{w}^H\mathbf{R}_k\mathbf{w}, \forall k$ evaluated at $\mathbf{w} = \mathbf{w}_n$ i.e., $\mathbf{R}_k\mathbf{w}_n$ where the weight is $\frac{1}{\mathbf{w}_n^H\mathbf{R}_k\mathbf{w}_n+\varepsilon}$. The projection step obtains the closest vector in the set $\mathcal{W}$ (in terms of Euclidean distance) to $\tilde{\mathbf{w}}_{n+1}$.

The weight update step in (5.2) - (5.3) can be interpreted as a projected gradient update of the problem shown in $\mathbf{\Pi_{18}}$ i.e., $\mathbf{w}_{n+1} = \mathcal{P}_{\mathcal{W}}\left[\tilde{\mathbf{w}}_{n+1}\right]$, $\tilde{\mathbf{w}}_{n+1} = \mathbf{w}_n + \alpha\nabla f(\mathbf{w}_n)$ where $f(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{K}\log\left(\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon\right)$.

$$\begin{aligned}\mathbf{\Pi_{18}} \quad \mathbf{w}^* &= \arg\max_{\|\mathbf{w}\|^2\leq 1} \frac{1}{2}\sum_{k=1}^{K}\log\left(\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon\right) \\ &\quad s.t. \quad \mathbf{w}^H\mathbf{G}_j\mathbf{w} \leq 1, \ \forall j = 1, 2, ..., J\end{aligned}$$

In the context of beamformer design, $\mathbf{\Pi_{18}}$ is a non-convex proportionally fair beamforming problem [38] and so the weight vector update can be considered as using the proportional fairness design criterion instead of the max-min fairness metric used in $\mathbf{\Pi_{16a}}$ for obtaining sub-optimal solutions of $\mathbf{\Pi_{16a}}$.

Proposition 3 *The objective function $f(\mathbf{w})$ in $\mathbf{\Pi_{18}}$ is Lipschitz continuous with a Lipschitz constant $L_{\nabla_f} = \sum_{k=1}^{K}\left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda_{\max}^2(\mathbf{R}_k)}{\varepsilon^2}\right)$, where $\lambda_{\max}(\mathbf{R}_k)$ is the maximum eigenvalue of $\mathbf{R}_k$.*

Proof 4 *It can be seen from $\mathbf{\Pi_{18}}$ that*

$$\begin{aligned}\nabla^2 f(\mathbf{w}) &= \sum_{k=1}^{K}\frac{(\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon)\mathbf{R}_k - 2\mathbf{R}_k\mathbf{w}\mathbf{w}^H\mathbf{R}_k}{\left(\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon\right)^2} \\ \|\nabla^2 f(\mathbf{w})\| &\leq \sum_{k=1}^{K}\left\|\frac{\mathbf{R}_k}{\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon}\right\| + 2\left\|\frac{\mathbf{R}_k\mathbf{W}\mathbf{R}_k}{\left(\mathbf{w}^H\mathbf{R}_k\mathbf{w} + \varepsilon\right)^2}\right\| \\ \|\nabla^2 f(\mathbf{w})\| &\leq \sum_{k=1}^{K}\left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda_{\max}^2(\mathbf{R}_k)}{\varepsilon^2}\right) \qquad (5.4)\end{aligned}$$

where $\mathbf{W} = \mathbf{w}\mathbf{w}^H$ and the last two inequalities use the fact that $\|\mathbf{w}\| \leq 1$, $\mathbf{w}^H\mathbf{R}_k\mathbf{w} \geq 0, \forall\mathbf{w}$, and $\mathbf{w}^H\mathbf{R}_k\mathbf{w} \leq \lambda_{\max}(\mathbf{R}_k), \forall\mathbf{w} : \|\mathbf{w}\| \leq 1$. From (A.11)*, it can be seen that $\nabla^2 f(\mathbf{w})$ can be universally bounded over the feasible region. Furthermore it can also be seen that $\nabla f(\mathbf{w})$ is continuously differentiable. Hence, it can be seen that $\nabla f(\mathbf{w})$ is Lipschitz continuous in $\mathbf{w}$.*

Theorem 2 *The weight update step in* (5.2) - (5.3) *converges to a KKT point of* $\mathbf{\Pi_{18}}$ *provided* $0 < \alpha \leq \frac{2}{L_{\nabla f}}$, *where* $L_{\nabla f} = \sum_{k=1}^{K} \left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda_{\max}^2(\mathbf{R}_k)}{\varepsilon^2} \right)$ *and* $\lambda_{\max}(\mathbf{R}_k)$ *is the maximum eigenvalue of* $\mathbf{R}_k$.

Proof 5 *From* (5.2) - (5.3), *the* $(n+1)^{th}$ *iteration can be viewed as a projected gradient update as shown below*

$$\mathbf{w}_{n+1} = \mathcal{P}_{\mathcal{W}}[\mathbf{w}_n + \alpha \nabla f(\mathbf{w}_n)] \tag{5.5}$$

where $f(\mathbf{w})$ *is the objective function of* $\mathbf{\Pi_{18}}$ *and* $\mathcal{P}_{\mathcal{W}}[\mathbf{w}]$ *is the projection of* $\mathbf{w}$ *onto the feasible set* $\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\|^2 \leq 1, \mathbf{w}^H \mathbf{G}_j \mathbf{w} \leq I_p; \forall j = 1, 2, ..., J\}$. *Therefore, the update step is a projected gradient step in each iteration. Using the convergence results of the projected gradient method in [47, Chapter 2, p. 240] it can be shown that iterates obtained from* (5.2) - (5.3) *converge to a KKT point of* $\mathbf{\Pi_{18}}$ *if* $0 < \alpha \leq \frac{2}{L_{\nabla f}}$.

5.3.1 Cyclic Dykstra's Projection Algorithm

Every iteration of the Projected AU-SLA algorithm involves a projection step onto an intersection region of many convex sets i.e., $\mathbf{w}_{n+1} = \mathcal{P}_{\mathcal{W}}[\mathbf{w}_n + \alpha \nabla f(\mathbf{w}_n)]$. One way to obtain the projection is to solve a convex QCQP which has a worst case complexity of $\mathcal{O}(N^{3.5})$ per iteration. For large N, this will be a bottleneck, since the gradient update step has a per iteration worst case complexity of $\mathcal{O}(N^2)$ only. Here, we propose a low complexity alternative for obtaining $\mathbf{w}_{n+1}$ from $\tilde{\mathbf{w}}_{n+1}$ using Cyclic Dykstra's Projection algorithm.

Algorithm description

Initialization: Define $\mathbf{x}_{old} = \mathbf{y}_0 = \tilde{\mathbf{w}}_{n+1}$, $\mathbf{x}_{new} = \mathbf{0}$, $\delta = 10^{-3}$ and auxiliary variable vectors $\mathbf{p}_{j,old} = \mathbf{0}$, $\forall j = 0, 1, 2, ..., J$, Denote the set $\mathcal{S}_0 = \{\mathbf{w} : \|\mathbf{w}\|^2 \leq 1\}$ and $\mathcal{S}_j = \{\mathbf{w} : \mathbf{w}^H \mathbf{G}_j \mathbf{w} \leq 1\}$, $\forall j = 1, 2, ..., J$.

while (1)

  for $j = 1$ to $J + 1$

    $\mathbf{y}_j = \mathcal{P}_{\mathcal{S}_{j-1}} [\mathbf{y}_{j-1} - \mathbf{p}_{j-1,old}]$

    $\mathbf{p}_{j,new} = \mathbf{y}_j - (\mathbf{y}_{j-1} - \mathbf{p}_{j-1,old})$

  end

  $\mathbf{x}_{new} = \mathbf{y}_{J+1}$

if $(\|\mathbf{x}_{new} - \mathbf{x}_{old}\| \leq \delta)$

break

else

$\mathbf{x}_{old} = \mathbf{x}_{new}$

$\mathbf{p}_{j,old} = \mathbf{p}_{j,new},\ \forall j = 1, 2, ..., J$

end if

end while

where $\mathcal{P}_{\mathcal{S}_0}[.]$ is the projection of the argument onto the set $\mathcal{S}_0$ and $\mathcal{P}_{\mathcal{S}_j}[.]$ is the projection onto the set $\mathcal{S}_j,\ \forall j = 1, 2, ..., J$.

$$\mathcal{P}_{\mathcal{S}_0}[\mathbf{x}] = \begin{cases} \mathbf{x}, & \|\mathbf{x}\|^2 \leq 1; \\ \frac{\mathbf{x}}{\|\mathbf{x}\|}, & \|\mathbf{x}\|^2 > 1. \end{cases} \tag{5.6}$$

5.3.2 Bisection search for $\mathcal{P}_{\mathcal{S}_j}[\mathbf{x}] = \min_{\mathbf{y}^H \mathbf{G}_j \mathbf{y} \leq 1} \|\mathbf{x} - \mathbf{y}\|^2$

Assume that the eigen-decomposition of $\mathbf{G}_j$ is known beforehand i.e., $\mathbf{G}_j = \mathbf{U}_j \mathbf{L}_j \mathbf{U}_j^H$. Define

$$\mathbf{z} = \mathbf{L}_j^{\frac{1}{2}} \mathbf{U}_j^H \mathbf{y},\ \tilde{\mathbf{x}} = \mathbf{U_j}^H \mathbf{x},\ \mathbf{D}_j = \mathbf{L}_j^{-\frac{1}{2}}. \tag{5.7}$$

The projection over the set $\mathcal{S}_j$ can be written as $\min_{\|\mathbf{z}\|^2 \leq 1} \|\tilde{\mathbf{x}} - \mathbf{D}_j \mathbf{z}\|^2$. The augmented Lagrangian function for the problem is given by

$$\mathcal{L}(\mathbf{z}, \lambda) = \sum_{i=1}^{N} \left| \tilde{\mathbf{x}}(i) - \mathbf{D}_j(i,i)\mathbf{z}(i) \right|^2 + \lambda \left(\sum_{i=1}^{N} \left| \mathbf{z}_i \right|^2 - 1 \right) \tag{5.8}$$

where $\lambda \geq 0$, $\tilde{\mathbf{x}}(i), \mathbf{z}(i)$ and $\mathbf{D}_j(i,i)$ are the i^{th} entries and i^{th} diagonal entry of $\tilde{\mathbf{x}}, \mathbf{z}$ and $\mathbf{D}_j$ respectively. From the KKT conditions and complementary slackness condition, we get

$$\mathbf{z}^*(i) = \frac{\tilde{\mathbf{x}}(i)\overline{\mathbf{D}}_j(i,i)}{\lambda + |\mathbf{D}_j(i,i)|^2} \tag{5.9}$$

$$F(\mathbf{z}^*, \lambda) = \|\mathbf{z}^*\|^2 = \sum_{i=1}^{N} \frac{|\tilde{\mathbf{x}}(i)\overline{\mathbf{D}}_j(i,i)|^2}{\left(\lambda^* + |\mathbf{D}_j(i,i)|^2\right)^2} = 1 \tag{5.10}$$

where $\overline{\mathbf{D}}_j(i,i)$ is the conjugate of $\mathbf{D}_j(i,i)$, $\mathbf{z}^*(i), \forall i = 1, 2, .., N$ and λ^* are the optimal value of the optimization variables and the Lagrangian multiplier. If the vector $\mathbf{x}$ is such that it satisfies $\mathbf{x}^H \mathbf{G}_j \mathbf{x} < 1$, then the constraint is inactive and $\lambda^* = 0$ which means $\mathcal{P}_{\mathcal{S}_j}[\mathbf{x}] = \mathbf{x}$. Since

$F(\mathbf{z}^*, \lambda)$ is a monotonically decreasing function of $\lambda \geq 0$, the optimal λ^* can be obtained by a bisection search until the condition in (5.10) is satisfied. Subsequently the optimal projection can be obtained as $\mathcal{P}_{\mathcal{S}_j}[\mathbf{x}] = \mathbf{U}_j \mathbf{D}_j \mathbf{z}^*$.

The above algorithm is applicable when the matrices $\{\mathbf{G}_j\}_{j=1}^{J}$ are full-rank. When some of the matrices are rank-deficient (say the rank of $\mathbf{G}_j$ is $r_j < N$ for a subset of j's $\in \{1, 2, .., J\}$), then the corresponding bisection search should be slightly modified as follows. We can now split the optimal projection $\mathbf{y}^*$ into two parts; i.e.,

$$\mathbf{y}^* = \mathbf{y}_r^* + \mathbf{y}_n^* \tag{5.11}$$

Assuming that the eigenvalues in the matrix $\mathbf{L}_j$ are arranged in the decreasing order of magnitude along the principal diagonal, $\mathbf{y}_n^* = [\mathbf{u}_j^{r_j+1} \mathbf{u}_j^{r_j+2} ... \mathbf{u}_j^N]^H \mathbf{x}$ and $\mathbf{u}_j^i$ is the i^{th} column of $\mathbf{U}_j$, since these columns represent the eigenvectors corresponding to the zero eigenvalues of $\mathbf{G}_j$ and hence they do not affect the constraint. The vector $\mathbf{y}_r$ can be obtained using the bisection search mentioned above using the reduced version of the matrices $\mathbf{L}_j$ and $\mathbf{U}_j$ (i.e., after removing the rows and columns corresponding to the zero eigenvalues from $\mathbf{L}_j$ and after removing the corresponding columns from $\mathbf{U}_j$).

It can be seen that the complexity of $\mathcal{P}_{\mathcal{S}_j}[.]$ for every bisection step is $\approx \mathcal{O}(N)$ and so the complexity per iteration of the of the additive update and cyclic projection combined is $\approx \mathcal{O}(N^2 + JN)$ (which is significantly less compared to $\mathcal{O}(N + K + J)^{3.5}$ for the SLA iteration and $\mathcal{O}(N + 3(K + J))^{3.5}$ for the FPP-SCA iteration). It has been shown in [48] that the Cyclic Dykstra's algorithm converges to the optimal projection as opposed to any other point in the feasible set. Therefore the proposed is indeed a projected gradient algorithm and so the iterates converge to a KKT point of $\mathbf{\Pi_{18}}$.

5.3.3 Refining step using one SLA iteration

Denote the vector obtained after the convergence of iterations in (5.2) - (5.3) as $\mathbf{w}_{PA}$. The vector $\mathbf{w}_{PA}$ is used as an initialization to one SLA iteration [27] shown in $\mathbf{\Pi_{17}}$. The resultant vector $\mathbf{w}_{PA-SLA}$ is the solution obtained from the Projected AU-SLA algorithm. Subsequently, the scalar $t_{PA-SLA} = \max_k \mathbf{w}_{PA-SLA}^H \mathbf{R}_k \mathbf{w}_{PA-SLA}$ is computed and if $t_{PA-SLA} \geq 1$, then the Projected AU-SLA algorithm has successfully found a feasible solution for a problem instance of $\mathbf{\Pi_{15a}}$ in polynomial time.

5.3.4 Application to Cognitive Multicast Beamforming

Consider an underlay cognitive single group multicast network with a secondary Tx with N antennas serving K single antenna secondary users; while the primary network comprises of J Tx-Rx pairs with a single antenna at every Rx. The objective is to design beamforming vectors at the secondary Tx that maximize the minimum SNR among the secondary users while restricting the interference caused to every primary Rx below a known threshold. The max-min SNR is chosen as the performance metric because it determines the maximum multicast downlink rate in the secondary system. Assume that the secondary Tx has perfect knowledge of the correlation matrices / outer product of the channels to secondary users $\{\mathbf{R}_k\}_{k=1}^K$ and to every primary Rx $\{\mathbf{G}_j\}_{j=1}^J$ and the primary threshold. This problem formulation is exactly the same as $\mathbf{\Pi_2}$ and so the proposed Projected AU-SLA, SLA SDR-G algorithms can be applied. The FPP-SCA algorithm is the same as the SLA algorithm here because it is easy to obtain a feasible solution as initialization for this setting.

5.4 Simulation Results

For simulation purposes, $\mathbf{R}_k, \mathbf{G}_j$ $k \in \{1, 2, .., K\}, j \in \{1, 2, .., J\}$ were obtained by generating random orthonormal matrices $\mathbf{U}_k$ and $\mathbf{V}_j$, random diagonal matrices $\mathbf{\Lambda}_{\mathbf{R}k}$ and $\mathbf{\Lambda}_{\mathbf{G}j}$ with positive diagonals, and setting $\mathbf{R}_k = \mathbf{U}_k \mathbf{\Lambda}_{\mathbf{R}k} \mathbf{U}_k^H$, $\mathbf{G}_j = \mathbf{V}_j \mathbf{\Lambda}_{\mathbf{G}j} \mathbf{V}_j^H$.

The success probability and average computational time for obtaining feasible solutions of multiple problem instances of $\mathbf{\Pi_{1a}}$ using the Projected AU-SLA algorithm is compared with the FPP-SCA algorithm [45], SLA algorithm [27] and the SDR-G algorithm [46] (with 10^4 Gaussian randomizations) in Tables 1 and 2. The codes are executed using CVX [39] and YALMIP [49] as modeling languages and the generic conic programming solver SeDuMi [50] is chosen as the solver for the FPP-SCA, SLA and SDR-G algorithms. The plots are obtained after averaging over 100 Monte-Carlo (MC) runs. For each run, the proposed algorithm is executed until $\|\mathbf{w}_{n+1} - \mathbf{w}_n\| \leq 10^{-4}$ or a maximum of 1000 iterations, whichever comes first.

The matrices $\mathbf{R}_k$ and $\mathbf{G}_j$ were generated as $\mathbf{R}_k = \mathbf{A}_k \mathbf{A}_k^H$, $\mathbf{G}_j = \mathbf{B}_j \mathbf{B}_j^H$, where $\mathbf{A}_k$ and $\mathbf{B}_j \in \mathcal{C}^{N \times N}$ and their entries are drawn from an i.i.d. $\mathcal{CN}(0, 1)$ distribution. In these tables, we have also included Projected AU-SLA with best of three random initializations, where we use three different initializations instead of one and then choose the initialization that has the best objective at the end of the Projected AU step and use that for the final SLA refining step.

It can be seen that the Projected AU-SLA with one and three random initializations attain feasibility with a very high success probability (i.e., $98.7\%, 100\%$ respectively for $(N_t, K, J) = (5, 5, 5)$ and $93.4\%, 96.6\%$ respectively for $(N_t, K, J) = (10, 10, 10)$)and require a lower average computational time (i.e., 0.1728 s, 0.1915 s respectively for $(N_t, K, J) = (5, 5, 5)$ and 0.8574 s, 1.3812 s respectively for $(N_t, K, J) = (10, 10, 10)$) than compared to the state-of-the-art FPP-SCA ($100\%, 100\%$ feasibility and 1.4150 s and 5.4127 s for $(5, 5, 5)$ and $(10, 10, 10)$). Furthermore, it can also be seen that the $\%$ feasibility improves as the number of randomizations for the Projected AU-SLA increases from one to three.

We compare the minimum SNR in cognitive multicast beamforming obtained from the Projected AU-SLA, SLA, SDR-G algorithms in Fig. 5.1 and Fig. 5.2. The performance of the plain Projected AU algorithm (without the final refining SLA iteration) is also included. The plots are obtained after averaging over 100 Monte-Carlo (MC) runs. Fig. 5.1 compares the average minimum SNR and the average computational time versus N for $K = 50$ secondary users and

Table 5.1: % Feasibility comparison

(N_t, K, J)	(5,5,5)	(5,10,10)	(10,10,10)	(10,25,25)
Proj. AU-SLA	98.7	97.1	93.4	91.5
Proj. AU-SLA (Best of 3 initializations)	100	98.7	96.6	94.2
FPP - SCA	100	100	100	99.1
SLA	100	100	97.8	95.1
SDR-G	81.2	63.5	23.4	0.5

Table 5.2: Comparison of average computational time (s)

(N_t, K, J)	(5,5,5)	(5,10,10)	(10,10,10)	(10,25,25)
Proj. AU-SLA	0.1728	0.3528	0.8574	5.322
Proj. AU-SLA (Best of 3 initializations)	0.1915	0.4573	1.3812	7.424
FPP - SCA	1.4150	2.9684	5.4127	32.143
SLA	1.3542	2.6478	4.8521	29.1248
SDR-G	0.3846	1.5169	4.1278	21.048

$J = 50$ primary receivers. First of all, it can be seen that the Projected AU-SLA algorithm performs very close ($\approx$ 0.3 dB less than SLA at $N = 25$) to the SLA algorithm and outperforms the SDR-G algorithm in terms of the average minimum SNR. Furthermore, the complexity of the Projected AU-SLA algorithm is approximately 4-5 times lesser than the SLA or the SDR-G algorithms.

In Fig. 5.2, we simulate a difficult beamforming scenario for the secondary Tx. Here, we align every primary Rx along the same direction (w.r.t. secondary Tx) as another secondary Rx. Therefore, when the received signal power at a secondary Rx increases, the interference power to the primary user aligned with that secondary Rx also increases simultaneously, thereby making the beamformer design at the secondary Tx a challenging job. However, from Fig. 5.2, it can be clearly seen that the Projected AU-SLA algorithm performs almost similar ($\approx$ 0.5 dB less than SLA at $N = 25$) to the SLA (both of which perform close (i.e., $\approx 1dB$ below) to the SDR upper bound as N increases) and outperforms SDR-G algorithm, but at a much lower complexity than both the algorithms.

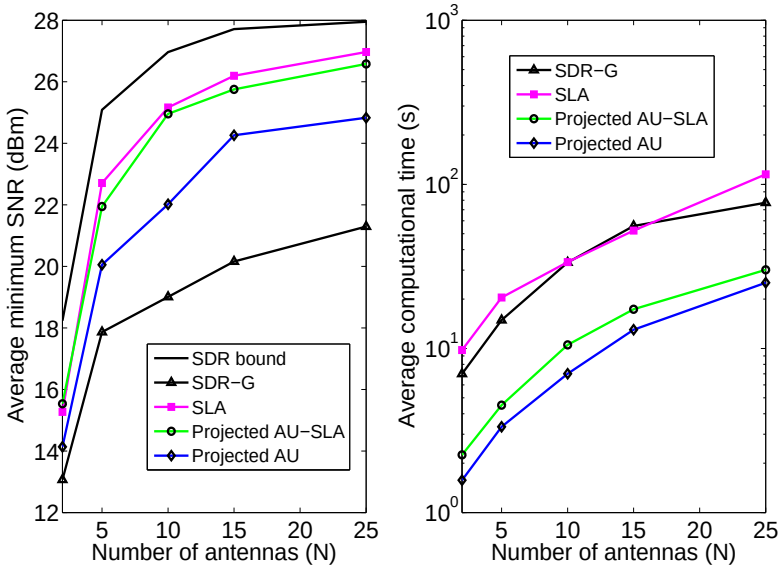

Figure 5.1: Comparison of average minimum SNR and computational time versus N for K = 50 and J = 50.

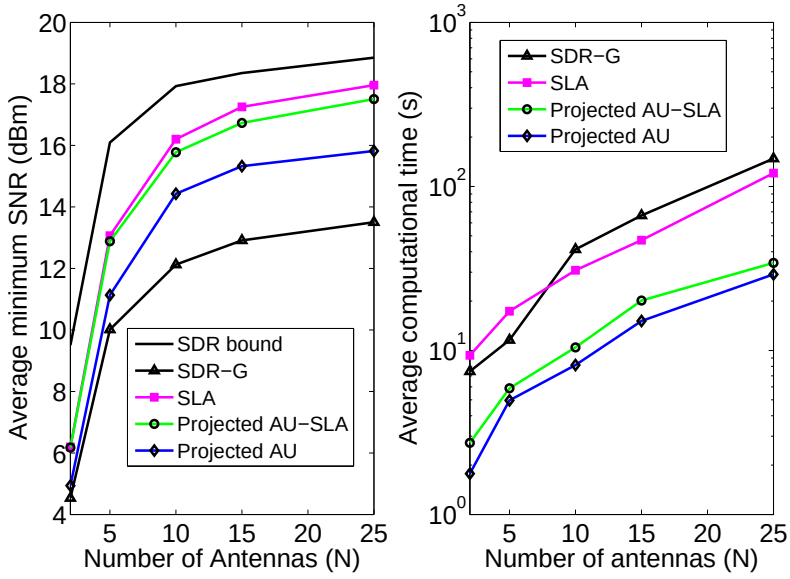

Figure 5.2: Comparison of average minimum SNR and computational time versus N for K = 50 and J = 50 when the channel from secondary Tx to each of the primary Rx is aligned with a channel to a secondary Rx.

5.5 Summary

We considered the class of non-convex QCQP with two-sided constraints when the associated matrices are positive semi-definite and proposed a low complexity Projected AU-SLA algorithm for obtaining feasible solutions (feasibility problem is NP-hard) with a high success probability. The projected AU-SLA comprises of a additive gradient update step followed by a low complexity projection step over a set of constraints computed using the cyclic Dykstra's projection algorithm. The convergence of the iterates of the projected AU-SLA algorithm is derived using the convergence proof of the Gradient Projection algorithm and the Descent Lemma. The success probability of obtaining feasibility is compared with the state-of-the-art FPP-SCA and SLA algorithms and simulations show that the Projected AU-SLA algorithm achieves a favorable performance-complexity tradeoff by achieving a $\%$ feasibility very close to the FPP-SCA and SLA algorithms; but requiring a much lesser computational time which is remarkable considering the fact. The Projected AU-SLA algorithm was also applied to an underlay cognitive single-group multicast beamforming setup (which is a special case of the non-convex QCQP considered before; where the problem is NP-hard, but obtaining feasible solutions is easy) and simulations shown that the minimum SNR among the secondary users / downlink multicast rate obtained from the Projected AU-SLA algorithm performs almost as well as the state-of-the-art SLA algorithm, but with a much lower computational complexity.

Chapter 6

Conclusions

The focus of this thesis was to propose adaptive transmit beamforming algorithms for various types of wireless scenarios ranging from isolated point to point MISO links to single group multicast networks. In each of these scenarios, we assumed that the Tx of interest did not have any initial CSI about the channel to its intended Rx. Furthermore, it was also assumed that the Rx had limited computational resources and restricted communication capability. The common underlying theme between the algorithms proposed for various scenarios is the ability of the Tx to learn the channel matrix / matrices of interest and design good transmit beamforming vectors for transmitting data to the intended Rx simultaneously, when there is no channel reciprocity, using a sequence of feedback bits from the intended Rx(s). Extensive analysis and relevant simulation results were used throughout to highlight the superior performance of the proposed algorithms as compared to the state of the art.

This thesis started by considering an isolated point to point MISO link. Here, we proposed an efficient way to accurately estimate the channel correlation matrix at the Tx of a MISO link based on binary feedbacks from the Rx, obtained by comparing the average received SNR with a threshold that is varied adaptively by the Tx and communicated to the Rx. The Tx starts without any initial CSI and uses these sequence of feedback bits to periodically update the estimate of the channel matrix. The concepts of Analytic Center Cutting Plane Method were used by the Tx to design the new beamforming vector at every time instant in such a way that the uncertainty about the channel matrix in the previous time slot is significantly reduced using the feedback bit in the subsequent time slot (irrespective of the orientation of the new beamforming vector used). It was shown that in the absence of SNR measurement errors at

the Rx and communication errors (w.r.t. the 1-bit feedback), the proposed algorithm enables the Tx to converge to an optimal beamforming vector and attain the maximum SNR at the Rx (attainable with *a priori* perfect CSIT). To accommodate the scenario with measurement errors and bit flips during the communication of the feedback bits, we proposed a maximum likelihood formulation which converges (in simulations) to the maximum achievable SNR with perfect CSIT. A discounted maximum likelihood formulation was also proposed to handle the scenario when the channel matrix / channel correlation matrix itself changes slowly over time, in which the proposed algorithm successfully tracks the channel variations.

The algorithms proposed for the isolated MISO links were used as a foundation and were extended to the case of an underlay cognitive radio network for designing long-term beamforming vectors at the secondary Tx to maximize the average received SNR at the secondary Rx without causing excessive interference to the primary Rx. The situation here is much more challenging than the isolated MISO link case considering the fact that the secondary Tx has to learn not only the correlation matrix of the channel to the secondary Rx, but also the correlation matrix of the interference channel to the primary Rx without any cooperation from the primary network, and no channel reciprocity can be assumed. Here we proposed a novel algorithm that enables the Tx to learn these matrices using a series of 1-bit channel quality indicator feedback bits from the secondary Rx (obtained after comparing the average SNR with a pre-determined threshold) and by overhearing the ACKs-NACKs that the primary Rx sends back to the primary Tx. Furthermore, it was proven and verified using simulations that the estimates of channel correlation matrices converge to the actual channel correlation matrices and the secondary Tx starts from no initial CSI and is asymptotically successful in designing the transmit beamforming that attains the maximum achievable average SNR at the secondary Rx while restricting the interference caused to the primary Rx, provided the primary interference threshold is known at the secondary Tx. When the primary interference threshold is unknown, an alternating optimization based algorithm followed by a power back-off procedure was proposed. Simulations were used to show that the average SNR attained at the secondary Rx is very close to the maximum achievable SNR using perfect CSIT. Furthermore, the interference caused at the primary Rx converges to the interference threshold even though it is unknown at the secondary Tx.

Subsequently, the focus of the thesis shifted to transmit beamforming for single group multicast networks. In this context, novel adaptive algorithms of low complexity, namely the AU, MU and MU-SLA were proposed that obtain transmit beamforming vectors which attain a high

minimum SNR when the Tx has perfect CSI. The proof of convergence of the AU algorithm to a KKT point of a proportionally fair beamforming problem, was derived and extensive simulations were used to show that the MU / AU algorithm achieves a higher minimum SNR than the state-of-the-art with the exception of SLA at orders of magnitude lower complexity. Furthermore, the MU-SLA algorithm (where the solution of the MU algorithm was refined by using it as an initialization for a single SLA iteration) outperforms the SLA by attaining a higher minimum SNR among the users at a much lower complexity (very close to that of the MU / AU), thereby achieving a favorable performance-complexity trade-off. When the Tx does not have any initial CSI and the receivers have limited computational resources, an online CMU algorithm based on ACCPM and appropriate threshold selection techniques were proposed to enable the Tx to learn the channel correlation matrices and design long-term transmit beamforming vectors simultaneously, using binary feedback from every Rx. Asymptotic convergence of the CMU algorithm to perfect-CSIT performance was established by invoking convergence results for the ACCPM from convex optimization, and verified in pertinent simulations. It was shown using simulations that the CMU algorithm enables the Tx to converge to the perfect CSIT performance using a very limited feedback requirement.

The approach used for single-group multicast transmit beamforming was finally extended to a very interesting application for finding feasible solutions to non-convex QCQPs (NP-hard in general) with two-sided constraints where the associated matrices are positive semi-definite. In this context, a low complexity Projected AU-SLA algorithm was proposed , which comprises of a additive gradient update step followed by a low complexity projection step over a set of convex constraints computed using the cyclic Dykstra's projection algorithm. The convergence of the iterates of the projected AU-SLA algorithm was derived using the convergence proof of the Gradient Projection algorithm and the Descent Lemma. It was shown using simulations that the Projected AU-SLA algorithm was successful in obtaining a feasible solution with high probability at a much lower complexity compared to the state of the art.

References

[1] A. B. Gershman, N. D. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, "Convex optimization-based beamforming," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 62–75, 2010.

[2] E. Telatar, "Capacity of multi-antenna gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

[3] D. J. Love, R. W. Heath, and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2735–2747, 2003.

[4] P. Xia and G. B. Giannakis, "Design and analysis of transmit-beamforming based on limited-rate feedback," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1853–1863, 2006.

[5] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Distributed transmit beamforming using feedback control," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 411–426, 2010.

[6] B. C. Banister and J. R. Zeidler, "A simple gradient sign algorithm for transmit antenna weight adaptation with feedback," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1156–1171, 2003.

[7] J. Xu and R. Zhang, "Energy beamforming with one-bit feedback," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3513–3517, 2014.

[8] B. Gopalakrishnan and N. D. Sidiropoulos, "Cognitive transmit beamforming from binary CSIT," *IEEE Transactions on Wireless Communication*, vol. 14, no. 2, pp. 895–906, 2014.

[9] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994, vol. 15.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[11] ——, "Localization and cutting-plane methods," *Lecture notes, Stanford University, www. stanford. edu/class/ee392o/localization-methods. pdf*, 2003.

[12] W. K. Newey and D. McFadden, "Large sample estimation and hypothesis testing," *Handbook of econometrics*, vol. 4, pp. 2111–2245, 1994.

[13] E. Tsakonas, J. Jalden, N. D. Sidiropoulos, and B. Ottersten, "Sparse conjoint analysis through maximum likelihood estimation," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5704–5715, 2013.

[14] Y. Noam and J. Tabrikian, "Marginal likelihood for estimation and detection theory," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 3963–3974, 2007.

[15] H. White, *Estimation, inference and specification analysis*. Cambridge university press, 1996, no. 22.

[16] Y. Noam and A. J. Goldsmith, "The one-bit null space learning algorithm and its convergence," *IEEE Transactions on Signal Processing*, vol. 61, no. 24, pp. 6135–6149, 2013.

[17] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 894–914, 2009.

[18] Y. Noam and A. J. Goldsmith, "Blind null-space learning for mimo underlay cognitive radio with primary user interference adaptation," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1722–1734, 2013.

[19] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, 2007.

[20] R. A. Tannious and A. Nosratinia, "Cognitive radio protocols based on exploiting hybrid arq retransmissions," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2833–2841, 2010.

[21] Y. Huang and D. P. Palomar, "Rank-constrained separable semidefinite programming with applications to optimal beamforming," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 664–678, 2010.

[22] Y. Ye and S. Zhang, "New results on quadratic minimization," *SIAM Journal on Optimization*, vol. 14, no. 1, pp. 245–267, 2003.

[23] R. Zhang, F. Gao, and Y.-C. Liang, "Cognitive beamforming made practical: Effective interference channel and learning-throughput tradeoff," *IEEE Transactions on Communications*, vol. 58, no. 2, pp. 706–718, 2010.

[24] F. Gao, R. Zhang, Y.-C. Liang, and X. Wang, "Multi-antenna cognitive radio systems: Environmental learning and channel training," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 2329–2332, 2009.

[25] R. Zhang, "On active learning and supervised transmission of spectrum sharing based cognitive radios by exploiting hidden primary radio feedback," *IEEE Transactions on Communications*, vol. 58, no. 10, pp. 2960–2970, 2010.

[26] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2239–2251, 2006.

[27] L.-N. Tran, M. F. Hanif, and M. Juntti, "A conic quadratic programming approach to physical layer multicasting for large-scale antenna arrays," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 114–117, 2014.

[28] O. T. Demir and T. Tuncer, "Alternating maximization algorithm for the broadcast beamforming," *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, pp. 1915–1919, 2014.

[29] A. Lozano, "Long-term transmit beamforming for wireless multicasting," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, vol. 3, pp. III–417, 2007.

[30] E. Matskani, N. D. Sidiropoulos, Z.-Q. Luo, and L. Tassiulas, "Efficient batch and adaptive approximation algorithms for joint multicast beamforming and admission control," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4882–4894, 2009.

[31] M. J. Lopez, "Multiplexing, scheduling, and multicasting strategies for antenna arrays in wireless networks," DTIC Document, Tech. Rep., 2002.

[32] I. H. Kim, D. J. Love, and S. Park, "Optimal and successive approaches to signal design for multiple antenna physical layer multicasting," *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2316–2327, 2011.

[33] Y. Huang, Q. Li, W.-K. Ma, and S. Zhang, "Robust multicast beamforming for spectrum sharing-based cognitive radios," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 527–533, 2012.

[34] Y. Noam and A. J. Goldsmith, "One-bit null space learning for mimo underlay cognitive radio," *Information Theory and Applications Workshop (ITA)*, pp. 1–7, 2013.

[35] J. Xu and R. Zhang, "Energy beamforming with one-bit feedback," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5370–5381, 2014.

[36] B. Gopalakrishnan and N. D. Sidiropoulos, "Cognitive transmit beamforming from binary CSIT," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 895–906, 2014.

[37] ——, "Adaptive multicast beamforming: Guaranteed convergence and state-of-art performance at low complexity," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2015.

[38] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, 2008.

[39] M. Grant, S. Boyd, and Y. Ye, *CVX: Matlab software for disciplined convex programming*, 2008.

[40] N. Jindal and Z.-Q. Luo, "Capacity limits of multiple antenna multicast," *IEEE International Symposium on Information Theory*, pp. 1841–1845, 2006.

[41] J. Xu, L. Liu, and R. Zhang, "Multiuser miso beamforming for simultaneous wireless information and power transfer," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4754–4758, 2013.

[42] Z.-Q. Luo, W.-K. Ma, A.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[43] A. d'Aspremont and S. Boyd, "Relaxations and randomized methods for nonconvex qcqps," *EE392o Class Notes, Stanford University*, 2003.

[44] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[45] O. Mehanna, K. Huang, B. Gopalakrishnan, A. Konar, and N. Sidiropoulos, "Feasible point pursuit and successive approximation of non-convex qcqps," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 804–808, 2014.

[46] K. T. Phan, S. A. Vorobyov, N. D. Sidiropoulos, and C. Tellambura, "Spectrum sharing in wireless networks via qos-aware secondary multicast beamforming," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2323–2335, 2009.

[47] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[48] N. Gaffke and R. Mathar, "A cyclic projection algorithm via duality," *Metrika*, vol. 36, no. 1, pp. 29–54, 1989.

[49] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," *IEEE International Symposium on Computer Aided Control Systems Design*, pp. 284–289, 2004.

[50] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

Appendix A

Proofs

This appendix contains a collection of proofs for theorems used in this thesis

Appendix A1

Taking the scalar dot product with $\mathbf{w}_{fp}$ on both sides of (4.5) and using the fact that $\|\mathbf{w}_{fp}\|^2 = 1$, we get $c = 2$. It can be seen from (4.5) that $\mathbf{w}_{fp}$ is a linear combination of $\mathbf{h}_1$ and $\mathbf{h}_2$. Furthermore $\mathbf{w}_{fp}\, e^{j\phi}$ is also a fixed point of (4.5), $\forall \phi \in [-\pi, \pi]$ (i.e., the set of fixed points of (4.5) is closed under rotation). Consider one such fixed point.

$$\mathbf{w}_{fp} = a\mathbf{h}_1 + be^{j\theta}\mathbf{h}_2, \quad a, b \in \mathbb{R}^1,\ \theta \in [-\pi, \pi] \tag{A.1}$$

Using the closure property of the set of fixed points of (4.5), we can also assume that $a, b \in \mathbb{R}^1_+$ without loss of generality. Equating the right hand side of (4.5) and (A.1) after substituting for $\mathbf{w}_{fp}$ from (A.1) and using the fact that $c = 2$, we get

$$2a\mathbf{h}_1 + 2be^{j\theta}\mathbf{h}_2 =$$

$$\frac{\mathbf{h}_1}{a\|\mathbf{h}_1\|^2 + be^{-j\theta}\mathbf{h}_2^H\mathbf{h}_1} + \frac{\mathbf{h}_2}{a\mathbf{h}_1^H\mathbf{h}_2 + be^{-j\theta}\|\mathbf{h}_2\|^2} \tag{A.2}$$

Assuming that $\mathbf{h}_1$ and $\mathbf{h}_2$ are linearly independent, we can equate their corresponding coefficients on both sides.

$$2a^2\|\mathbf{h}_1\|^2 + 2abe^{-j\theta}\mathbf{h}_2^H\mathbf{h}_1 = 1 \tag{A.3}$$

$$2abe^{j\theta}\mathbf{h}_1^H\mathbf{h}_2 + 2b^2\|\mathbf{h}_2\|^2 = 1 \tag{A.4}$$

In (A.3), since $a, b \in \mathbb{R}_+^1$, it is clear that $\theta = \angle(\mathbf{h}_2^H \mathbf{h}_1)$. Substituting this value of θ in (A.3) and (A.4) followed by equating the corresponding terms in the left hand side, we get

$$2a^2\|\mathbf{h}_1\|^2 + 2ab|\mathbf{h}_2^H\mathbf{h}_1| = 2b^2\|\mathbf{h}_2\|^2 + 2ab|\mathbf{h}_1^H\mathbf{h}_2| \tag{A.5}$$

This implies $a\|\mathbf{h}_1\| = b\|\mathbf{h}_2\|$. Using the fact that $\|\mathbf{w}_{fp}\|^2 = 1$ and $\theta = \angle(\mathbf{h}_2^H\mathbf{h}_1)$, from (A.3) and (A.5), we get

$$a = \frac{1}{\sqrt{2\left[\|\mathbf{h}_1\|^2 + \left(\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}\right)|\mathbf{h}_2^H\mathbf{h}_1|\right]}} \tag{A.6}$$

Now the characterization of the fixed point is complete. We next compute the SNR at each of the receivers using this transmit beamforming vector $\mathbf{w}_{fp}$. The SNR at the k^{th} Rx is $|\mathbf{w}_{fp}^H\mathbf{h}_k|^2$, $k \in \{1, 2\}$.

$$\begin{aligned} |\mathbf{w}_{fp}^H\,\mathbf{h}_1|^2 &= \left(a\|\mathbf{h}_1\|^2 + b|\mathbf{h}_2^H\mathbf{h}_1|\right)^2 \\ &= a^2\left[\|\mathbf{h}_1\|^2 + \left(\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}\right)|\mathbf{h}_2^H\mathbf{h}_1|\right]^2 \\ &= \frac{1}{2}\left[\|\mathbf{h}_1\|^2 + \left(\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}\right)|\mathbf{h}_2^H\mathbf{h}_1|\right] \end{aligned} \tag{A.7}$$

Similarly

$$\begin{aligned} |\mathbf{w}_{fp}^H\,\mathbf{h}_2|^2 &= \left|a\mathbf{h}_1^H\mathbf{h}_2 + be^{-j\theta}\|\mathbf{h}_2\|^2\right|^2 \\ &= a^2\left[|\mathbf{h}_1^H\mathbf{h}_2| + \|\mathbf{h}_1\|\|\mathbf{h}_2\|\right]^2 \\ &= \frac{1}{2}\frac{\left[|\mathbf{h}_1^H\mathbf{h}_2| + \|\mathbf{h}_1\|\|\mathbf{h}_2\|\right]^2}{\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}\left[|\mathbf{h}_1^H\mathbf{h}_2| + \|\mathbf{h}_1\|\|\mathbf{h}_2\|\right]} \\ &= \frac{1}{2}\left[\|\mathbf{h}_2\|^2 + \left(\frac{\|\mathbf{h}_2\|}{\|\mathbf{h}_1\|}\right)|\mathbf{h}_2^H\mathbf{h}_1|\right] \end{aligned} \tag{A.8}$$

From (A.7) and (A.8), the minimum SNR among the receivers SNR_{min} and the associated multicast rate r_{min} are given by

$$\begin{aligned} SNR_{min} &= \min\left(|\mathbf{w}_{fp}^H\,\mathbf{h}_1|^2, |\mathbf{w}_{fp}^H\,\mathbf{h}_2|^2\right) \\ &= \frac{1}{2}\Bigg[\min\left(\|\mathbf{h}_1\|^2, \|\mathbf{h}_2\|^2\right) + \\ &\qquad \min\left(\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|}, \frac{\|\mathbf{h}_2\|}{\|\mathbf{h}_1\|}\right)|\mathbf{h}_1^H\mathbf{h}_2|\Bigg] \\ r_{min} &= \log_2(1 + SNR_{min}) \end{aligned} \tag{A.9}$$

Appendix A2

The gradient of $f(\mathbf{w})$ (the objective function of $\mathbf{\Pi_9}$) at $\mathbf{w} = \mathbf{w}_n$ is given by

$$\nabla_{\mathbf{w}} f(\mathbf{w}_n) = \sum_{k=1}^{K} \frac{\mathbf{R}_k \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + \varepsilon} \tag{A.10}$$

Now suppose that a projected gradient update algorithm is used for finding the local maxima of the constrained non-concave maximization problem $\mathbf{\Pi_9}$, where the update step at iteration $n+1$ is given by $\tilde{\mathbf{w}}_{n+1} = \mathbf{w}_n + \alpha \nabla_{\mathbf{w}} f(\mathbf{w}_n)$, $\mathbf{w}_{n+1} = \mathcal{P}_{S_w}(\tilde{\mathbf{w}}_{n+1})$, $\mathcal{P}_{S_w}(.)$ is the projection of the argument onto the set $S_w = \{\mathbf{w} : \|\mathbf{w}\|^2 = 1\}$ and α is a positive step size (same as in (4.7)). It can be seen that $\mathbf{w}_{n+1}$ in (4.10) is the optimal projection of the gradient update $\tilde{\mathbf{w}}_{n+1}$ onto the unit ball S_w. Furthermore, it can be shown that

- $\nabla f(\mathbf{w})$ is Lipschitz continuous in $\mathbf{w}$ with a Lipschitz constant $L_{\nabla f}$; See Appendix C.
- $\|\nabla^2 f(\mathbf{w})\|_F \leq \sum_{k=1}^{K} \left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda_{\max}^2(\mathbf{R}_k)}{\varepsilon^2} \right) =: L_{\nabla f}, \forall \, \|\mathbf{w}\| \leq 1$; See Appendix A3.

In simplifying the upper bound for $\|\nabla^2 f(\mathbf{w})\|_F$, we have used that $\mathbf{R}_k = \mathbf{h}_k \mathbf{h}_k^H$, $\forall k = 1, 2, ..., K$. Using the convergence results for the projected gradient method in [47, Chapter 2, p. 240], it can be shown that iterates of the AU algorithm in (4.7) converge to a Karush-Kuhn-Tucker(KKT) point of $\mathbf{\Pi_9}$ if $0 < \alpha \leq \frac{2}{L_{\nabla_f}}$.

Appendix A3

$$\begin{aligned}
\nabla^2 f(\mathbf{w}) &= \sum_{k=1}^{K} \frac{(\mathbf{w}^H \mathbf{R}_k \mathbf{w} + \varepsilon)\mathbf{R}_k - 2\mathbf{R}_k \mathbf{w}\mathbf{w}^H \mathbf{R}_k}{\left(\mathbf{w}^H \mathbf{R}_k \mathbf{w} + \varepsilon\right)^2} \\
\|\nabla^2 f(\mathbf{w})\| &\leq \sum_{k=1}^{K} \left\| \frac{\mathbf{R}_k}{\mathbf{w}^H \mathbf{R}_k \mathbf{w} + \varepsilon} \right\| + 2 \left\| \frac{\mathbf{R}_k \mathbf{W} \mathbf{R}_k}{\left(\mathbf{w}^H \mathbf{R}_k \mathbf{w} + \varepsilon\right)^2} \right\| \\
\|\nabla^2 f(\mathbf{w})\| &\leq \sum_{k=1}^{K} \left(\frac{\|\mathbf{R}_k\|_F}{\varepsilon} + \frac{2\lambda_{\max}^2(\mathbf{R}_k)}{\varepsilon^2} \right)
\end{aligned} \tag{A.11}$$

where $\mathbf{W} = \mathbf{w}\mathbf{w}^H$ and the last two inequalities use the fact that $\|\mathbf{w}\| = 1$, $\mathbf{w}^H \mathbf{R}_k \mathbf{w} \geq 0, \forall \mathbf{w}$, and $\mathbf{w}^H \mathbf{R}_k \mathbf{w} \leq \lambda_{\max}(\mathbf{R}_k), \forall \mathbf{w} : \|\mathbf{w}\| = 1$. From (A.11), it can be seen that $\nabla^2 f(\mathbf{w})$ can be universally bounded over the feasible region. Furthermore it can also be seen that $\nabla f(\mathbf{w})$ is

continuously differentiable. Hence, it can be seen that $\nabla f(\mathbf{w})$ is Lipschitz continuous in $\mathbf{w}$. It is straightforward to see that $\nabla u(\mathbf{w}, \mathbf{w}(n))$ is a linear combination of two Lipschitz continuous functions i.e., $\nabla f(\mathbf{w}_n)$ and $\frac{(\mathbf{w}-\mathbf{w}_n)}{\alpha}$. Therefore, $\nabla u(\mathbf{w}, \mathbf{w}(n))$ is also Lipschitz continuous.

Appendix A4

When the matrices $\{\mathbf{R}_k\}_{k=1}^{K}$ have rank > 1, the optimization problem in [27] is given by $\mathbf{\Pi_{13}}$ and the n^{th} SLA iteration is the solution of $\mathbf{\Pi_{14}}$.

$$\mathbf{\Pi_{13}} \qquad \min_{\mathbf{w} \in \mathbb{C}^N} \|\mathbf{w}\|^2$$
$$\text{s.t.} \quad \mathbf{w}^H \mathbf{R}_k \mathbf{w} \geq 1, \quad \forall k \in \{1, 2, ..K\}$$

$$\mathbf{\Pi_{14}} \qquad \mathbf{w}_{n+1} = \arg \min_{\mathbf{w} \in \mathbb{C}^N} \|\mathbf{w}\|^2$$
$$\text{s.t. } \mathbf{w}_n^H \mathbf{R}_k \mathbf{w}_n + 2\Re\left[(\mathbf{R}_k \mathbf{w}_n)^H (\mathbf{w} - \mathbf{w}_n)\right] \geq 1$$

$\forall k \in \{1, 2, ...K\}$

Appendix B

Glossary and Acronyms

This appendix contains a table of acronyms used in this thesis and their meaning.

B.1 Acronyms

Table B.1: Acronyms

Acronym	Meaning
Tx	Transmitter
Rx	Receiver
CSIT	Channel State Information at Transmitter
QoS	Quality of Service
MISO	Multiple-Input Single-Output
SNR	Signal to Noise Ratio
ACCPM	Analytical Center Cutting Plane Method
CRN	Cognitive Radio Networks
sTx	Secondary Transmitter
sRx	Secondary Receiver
pTx	Primary Transmitter
pRx	Primary Receiver

Continued on next page

Table B.1 – continued from previous page

Acronym	Meaning
SDR	Semi-Definite Relaxation
QCQP	Quadratically Constrained Quadratic Program
SLA	Successive Linear Approximation
p.s.d.	Positive Semi-Definite
KKT	Karush Kuhn Tucker