Spring 2017

# Scalable algorithms for distributed beamforming and nullforming

Amy Kumar
*University of Iowa*

# SCALABLE ALGORITHMS FOR DISTRIBUTED BEAMFORMING AND NULLFORMING

by

Amy Kumar

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

May 2017

Thesis Supervisors: Associate Professor Raghuraman Mudumbai
Professor Soura Dasgupta

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

---

PH.D. THESIS

---

This is to certify that the Ph.D. thesis of

Amy Kumar

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Electrical and Computer Engineering at the May 2017 graduation.

Thesis Committee: _____
Raghu Mudumbai, Thesis Supervisor

_____
Soura Dasgupta, Thesis Supervisor

_____
Er-Wei Bai

_____
Anton Kruger

_____
Ralph P. Russo

To my family.

*That one must do some work seriously and must be independent and not merely amuse oneself in life* , this our mother (Marie Curie) has told us always, but never that science was the only career worth following. - Irene Joliot Curie

## ACKNOWLEDGEMENTS

I am grateful to God for having bestowed upon me his blessings and giving me strength and wisdom.

I would like to take this opportunity to thank my advisors, Professor Raghuraman Mudumbai and Professor Soura Dasgupta, for their persistent and immense support and guidance right from the beginning of this research work. I am highly indebted to them for their insights and dedication to this project and will always hold them in high regards.

I am also grateful to my Committee members for their support and for sharing their perspective, to University of Iowa Professors for knowledge sharing through courses, and department staff for help at several occasions. I am also thankful to our collaborators in other universities, Professor Upamanyu Madhow and Professor D. Richard Brown, for their help and feedback.

I extend my profound gratitude to my family, especially, my parents for their innumerable sacrifices, my younger sister for being a constant source of motivation, and my husband for his patience, wisdom and whole-hearted support during my study.

I would also like to thank my lab mates, department peers and lab seniors for the valuable time they spent with me.

## ABSTRACT

Constant evolution requirements of Wi-Fi and cellular standards to meet the demands of better power efficiency, longer range and higher throughput of wireless networks has drawn attention to multiple antenna transmitters and receivers, i.e., multi-input multi-output(MIMO) systems. This research falls in the larger context of distributed MIMO, or DMIMO systems, wherein groups of cooperating transceivers organize themselves into virtual antenna arrays which can, in principle, emulate any MIMO technique that a centralized array can support.

Beamforming and nullforming are techniques that can be employed by centralized or distributed antenna array, and can be building blocks for MIMO communication systems; these impart directionality to the array and can help cater to the demands of today's wireless networks. In beamforming, a set of distributed transmitters in a wireless network cooperatively transmit a common message signal in such a way that their individual transmissions add up to a desired SNR level at the set of designated receivers while in nullforming, cooperative transmission ensures that the individual transmissions cancel each other at the set of designated receivers. The key bottleneck in the practical realization of DMIMO is synchronization. Distributed nullforming specifically poses challenges that call for special attention. Here, we develop a set of scalable algorithms for beamforming and nullforming using distributed transmitters by forming a virtual antenna array and overcome the involved challenges in a purely distributed fashion.

Under a per-antenna power constraint and assuming equal-gain channels, an ideal N-antenna beamformer provides an N squared-fold coherent power gain on target. Ideal nullforming on the other hand results in zero power on the target. These properties motivate applications in cooperative jamming or communications, where the goal is to maximize the net transmitted power using multiple transmitters while simultaneously protecting a designated receiver. For example, in a cognitive radio system where the transmit array is a secondary user of licensed spectrum which seeks to communicate with a set of secondary receivers (beam targets) without causing any interference at primary receivers (null targets). Another possible application is a cellular network where adjacent Base Stations form a transmit array and coordinate their transmissions to avoid cochannel interference. Recent algorithms on wireless security critically rely on nodes blanketing a landscape with full power jamming signals while protecting a cooperating receiver through nullforming. So a third application can be electronic warfare where a transmit array broadcasts strong jamming signals that disable an enemy's communication infrastructure while protecting friendly stations (null targets) from interference due to the jamming signal. The joint beam and nullforming specifically can be more generally thought of as a fundamental building block for increased spatial spectrum reuse and toward achieving the full spatial multiplexing gains available from MIMO techniques with distributed antenna arrays.

## PUBLIC ABSTRACT

The human race has come a long way since the birth of radio: from the first television transmissions to cellular mobile telephony. The number of internet and mobile phone users and the demand for higher data rates is going up by the day. Multiple antenna transmitter-receiver (MIMO) systems can provide a possible solution to meet the demands of better power efficiency, longer range and higher throughput of wireless networks. This research falls under distributed MIMO, or DMIMO systems, wherein groups of cooperating transceivers organize themselves into virtual antenna arrays which can, in principle, emulate any MIMO technique that a centralized array can support.

Beamforming, in which distributed transmitters in a wireless network cooperatively transmit a common message signal in such a way that their individual transmissions add up to a desired level at the designated receivers, and nullforming, in which cooperative transmission ensures that the individual transmissions cancel each other at the designated receivers, are techniques that can be employed to impart directionality to the array and can be building blocks for MIMO communication systems. Here, we develop a set of scalable algorithms for beamforming and nullforming using distributed transmitters by forming a virtual antenna array and overcome the involved challenges, like synchronization, in a purely distributed fashion. Cooperative jamming in cognitive radio systems, cochannel interference avoidance in a cellular network, and wireless security are a few applications.

## TABLE OF CONTENTS

## LIST OF FIGURES

Figure

# CHAPTER 1
# INTRODUCTION

It's been more than hundred years since the era of Hertz, Marconi, Jagdish Chandra Bose and the first successful wireless communication across the Atlantic ocean. The human race has come a long way since the birth of radio: from the first television transmissions to the demonstration of frequency modulation to cellular mobile telephony and wireless internet. With an ever expanding number of internet and mobile phone users (close to half and total world population respectively as of 2016), the demand for higher data rates is going up drastically by the day and so is the cost of frequency spectrum. This has led to constant evolution of Wi-Fi and cellular standards. Multiple antenna transmitters and receivers (MIMO), compared to conventional transmission links consisting of a singe antenna transmitter and receiver, can provide a possible solution to meet the demands of better power efficiency, longer range and higher throughput of wireless networks. Distributed MIMO systems (DMIMO) which can be used for larger number of antennas as these overcome constraints posed by antenna size, number and form-factor to centralized MIMO, might help to achieve higher capacity. Techniques such as beamforming, in which transmitters transmit a common message signal in such a way that their individual transmissions add up to a desired level at the designated receivers, and nullforming, in which individual transmissions cancel each other at the designated receivers, impart directionality to the array and can be employed by centralized or distributed antenna array to help cater to the demands of today's wireless networks.

## 1.1 Motivation

We are interested in developing a set of scalable algorithms for beamforming and nullforming using distributed transmitters by forming a virtual antenna array. Beamforming and nullforming are building blocks for spatial multiplexing (the art of transmitting multiple messages or signals simultaneously using a common frequency band) and multi-input multi-output communication systems. DMIMO systems pose significant challenges in implementation; the ones posed by distributed nullforming specifically call for special attention. Our research interest is in developing algorithms to achieve the mentioned goals and overcome the involved challenges in a purely distributed fashion.

## 1.2 Background

Performance gains due to Multiple Input Multiple Output (MIMO) techniques such as beamforming, spatial multiplexing, space division multiple access (SDMA), and space-time coding are well established [1, 2], and MIMO forms an integral part of current wireless standards [3, 4]. This work falls in the larger context of distributed MIMO, or DMIMO, systems, wherein groups of cooperating transceivers organize themselves into *virtual antenna arrays* which can, in principle, emulate any MIMO technique that a centralized array can support. Beamforming and nullforming are of particular interest, since they provide building blocks for techniques such as spatial division multiple access (SDMA) and interference alignment. In distributed beamforming, a set of distributed transmitters in a wireless network cooperatively transmit

a common message signal in such a way that their individual transmissions add up to a desired SNR level at the set of designated receivers while in distributed nullforming, cooperative transmission ensures that the individual transmissions cancel each other at the set of designated receivers.

While the number of antennas in a centralized MIMO transceiver is limited by size and cost, DMIMO allows us to scale up to large virtual antenna arrays by exploiting the natural geographical distribution of cooperating nodes. Since the required array size scales with wavelength, DMIMO techniques are of particular significance at lower carrier frequencies. The key bottleneck in the practical realization of DMIMO is synchronization. Unlike conventional centralized MIMO where transmit antennas are driven by a single oscillator, transceivers in DMIMO have independent oscillators with unpredictable phase offsets relative to each other. This complicates distributed transmission strategies such as beam and nullforming, which require precise control over the phase of the transmitted signals to ensure that these signals arrive at the receiver with the appropriate phase relationships.

Approaches to DMIMO include a significant body of work that uses high-bandwidth wired backhaul links for the synchronization needed to synthesize virtual arrays from base stations in cellular [5] and access points in WiFi networks [6]. This work focuses, by contrast, on DMIMO techniques that are amenable to *all-wireless* deployments (e.g., *ad hoc* networks for communication and sensing) with little coordination overhead. We are particularly interested in techniques that *scale* with the number of cooperating nodes forming the virtual array, in terms of overhead and

protocol complexity.

While the idea of cooperative communication has been studied for decades [7], the early work in this area neglected the RF synchronization issues that are crucial for the practical implementation of these ideas. Recently, there has been a significant amount of research activity on distributed transmit beamforming [8], which has led to the development of a menu of sophisticated synchronization techniques [9] featuring different sets of tradeoffs between simplicity, performance and overheads. Furthermore, the practical feasibility of the virtual array concept has been convincingly demonstrated in multiple experimental demonstrations, for example [10], of distributed beamforming. In essence it has been shown that substantial SNR gains can be achieved with modest overheads using simple signal processing on commodity hardware [11, 12]. One approach that stands out amongst the work on synchronization of all-wireless DMIMO for its scalability is for each transmitter to adapt separately based on a *common aggregate feedback from the receiver.* The earliest example of this approach is the one-bit feedback algorithm for distributed beamforming in [13], where each transmitter perturbs its phase randomly, and the receiver broadcasts a *single bit* of feedback indicating whether the received power is better or worse. If better, the transmitters keep their phase perturbations; if worse, they undo them. This completely decentralized randomized ascent was proven to converge to phase coherence at the receiver [14]. The aggregate feedback for the one-bit algorithm allows the receiver to be oblivious of the number and identity of transmitters, thus providing protocol-level scalability, and simplifying prototyping efforts like [11, 15]. This work,

sets the groundwork for a similar approach to distributed nullforming or joint beam and nullforming.

Though there is scope for adapting the synchronization techniques originally developed for distributed beamforming for the case of nullforming, there are important differences that make nullforming significantly more challenging. These are described in Section 1.3. While there is a substantial literature on nullforming using centralized antenna arrays, [16, 17], work on distributed nullforming is limited. The three papers of note are [18, 19], and [20]. What makes the algorithms of [18, 19], not scalable however, is the fact that the receiver feeds back to *each node, the complex channel gain seen by every other transmitter*. The only truly scalable algorithms are in [20, 21].

In contrast to [18, 19], the algorithms developed in this thesis require that the receiver(s) send to each transmitter only its complex channel gain and an aggregate feedback common to all transmitters. This means our algorithms do not require the transmitters to acquire knowledge of the global channel state information but instead need each transmitter to only gather information regarding its own channel gain to the receiver(s). Since the aggregate feedback required by our algorithms is common to all transmitters, it can be broadcast by the receiver to all transmitters at once and doesn't need to be sent individually to the transmitters. These differences in requirements allow our algorithms to scale very well with number of transmitters in terms of feedback overhead.

Also, receivers in our formulation cooperate with the distributed transmit array, for example by explicitly sending required feedback, yielding a far simpler

problem formulation in contrast to certain interference avoidance techniques proposed in [22, 23, 24] which require the transmitters to implicitly learn the nullspace by probing the MIMO channel with different precoding vectors and observing indirect measures of the SINR, e.g., power control, at the receivers to which the nulls are steered, and thereby determine a suitable nullforming precoding vector.

In all, we can say that the body of algorithms developed in this work can be implemented in a distributed fashion at the transmit antenna array through an iterative process to form beams and nulls at designated receiver(s). These algorithms possess some very interesting properties like *scalability* with large number of transmitters, not only in terms of feedback overhead but also in terms of convergence speed, that is, the convergence speed of these algorithms does not increase with an increase in the number of transmitters, and performance *robustness* to noise and feedback errors. We begin with developing algorithms for performing nullforming at single receiver in Chapters 2 and 4 and later generalize to the case of performing beamforming and nullforming at multiple receivers in Chapter 5. All algorithms require the transmitters to apply suitable weights to their transmitted signals and adaptively adjust those in subsequent iterations of the algorithm till their transmissions effect the desired power levels at the receiver(s).

The nullforming algorithm developed in Chapter 2, requires that the transmitters apply and adjust only phases [21] unlike the zero-forcing techniques in [16]-[19] which control both transmit gains and phases. In Chapter 3, we discusses how this algorithm is robust to channel estimation errors in both gain and phase. The key

deficiency of the algorithm developed in Chapter 2 is that it requires prior frequency synchronization among the transmitters. In Chapter 4, we propose a way to incorporate frequency synchronization process into the nullforming algorithm itself [25] by letting each transmitter adjust both its frequency and phase. Thereafter in Chapter 5 where we consider joint beamforming and nullforming(JBNF) which includes beamforming and nullforming as special cases [26, 27], we let the transmitters apply a complex weight and adjust both phase and amplitude of applied weights. Using complex weights helps us mathematically represent the problem as a distributed gradient search of a simple, suitably chosen cost function which is convex in the weights, and is guaranteed to converge to a manifold on which all the beam and nullforming constraints of the JBNF problem are satisfied under very general conditions. For the nullforming only algorithms developed in Chapters 2 and 4, we avoid employing both phase and amplitude adaptation as this could lead the algorithm to drive the transmitters to apply zero amplitudes to their transmitted signals in order to achieve the desired zero power levels at null-targets and hence would not serve our purpose.

### 1.3 Challenges in distributed nullforming

In a DMIMO system, since each antenna is driven by a separate independent oscillator, phase errors result in some loss of performance with respect to ideal and added stringency is required for nullforming as compared to beamforming.

- While beamforming gains are highly robust and insensitive to small phase errors (upto about 30 degrees [8]), nullforming is substantially more sensitive [19] to

even modest errors. One implication of this sensitivity to small phase errors is that the simple 1-bit feedback algorithm [13] that has proved to be effective for beamforming does not work for nullforming. For a system with $N = 10$ transmit nodes and time-invariant channels with unit magnitude, if the transmitting nodes have perfect channel state knowledge and are perfectly synchronized at the start, beam and nullforming performance are near ideal when the elapsed time from synchronization is small, but degrade as oscillators begin to drift. For example, if the oscillators drift independently with a standard deviation of 62 ps per second [28] and the channel phase estimates become stale, to maintain a null 10 dB better than incoherent transmission, the nodes must resynchronize within approximately 120 ms. Nullforming performance tends to degrade more quickly than beamforming and requires a commensurately tighter synchronization [21].

- For beamforming, each transmitter only needs the knowledge of the phase of its own transmitted signal at the receiver. In contrast for nullforming, the phase of the transmitted signal at each node cannot be chosen independently of the amplitudes and phases of other nodes [19]. Nullforming essentially depends on a node's transmitted signal cancelling the signals from all other transmitters. Therefore state-of-the-art distributed nullforming algorithms, [18] and [19] assume that each transmitter knows *every transmitter's* complex channel gain to the receiver. This requirement poses a severe challenge for scalability.

## 1.4 Scalability to large transmit arrays

As discussed in Section 1.3, nullforming requires each transmitter to know the global channel state information. This CSI requirement for nullforming poses a severe challenge for scalability: in [18, 19], the receiver feeds back to *each node, the complex channel gain seen by every other transmitter*. In other words, *global channel knowledge* is assumed to be available to all nodes in the network. In a network of $N$ transmitters and $M$ receivers, this requires $N \times M$ complex channel gains to be sent to each of the $N$ transmitters, representing total overhead scaling as $O(MN^2)$.

In this work, we take an iterative approach. We assume only knowledge at each transmitter of *its own complex channel gains* to the receivers, and a set of common feedback messages from each receiver broadcast to all the transmitters. The feedback messages contain simply the complex amplitude (i.e. the gain and phase) of the aggregate received RF signal at each receiver. Thus, in this body of algorithms, only a total of $M$ channel gains are required at each transmitter in addition to the common feedback signals, representing a total overhead of $O(MN+M)$ per iteration. Furthermore, it turns out that the number of iterations required for convergence of our algorithm *does not increase* with the number of transmitters.

In other words, the *total synchronization overhead* over all the iterations until convergence in our proposed algorithm could be a factor of $O(N)$ smaller than what is required in [18, 19]. We should note that in Chapters 2, 3 and 4, $M = 1$ as we discuss the single receiver setting unlike in Chapter 5 where we consider multiple receivers.

## 1.5 Practical considerations

The set of algorithms developed in this thesis have a few common properties, for example, all of them are all iterative in nature and also involve a common step wherein the receiver(or receivers) broadcast a feedback message to the transmitters. Practical considerations allow us in making the following simplifying assumptions in general:

**Prior frequency synchronization** We know that effective baseband complex channel gain is the cumulative effect of the propagation channel, the RF transmit and receive hardware, and carrier frequency offsets between transmitter and receiver. Out of these, the largest and most dominant effects arise in practice from carrier frequency offsets. Algorithms developed in Chapters 2 and 5 assume that the transmitters are frequency synchronized at the start. The following are a few ways to do the same:

- **GPS** The effect of carrier frequency offsets can be mitigated without any centralized coordination by having the transmitters lock themselves to a common reference, e.g., a global positioning system (GPS) frequency reference.

- **Common feedback** If GPS is not available or undesirable, another possibility is to simply use the common feedback messages broadcast by the receiver for carrier frequency synchronization.

A variety of procedures have been developed in the literature for frequency synchronization, e.g., those described in [29], [18], [19], [30], and any of these are appropriate for use with the algorithms developed in this body of work. We also provide a way of

addressing the prior frequency synchronization requirement for nullforming in Chapter 4 by incorporating a method to do frequency synchronization into the algorithm itself.

**Time-slotting.** We assume a synchronous time-slotted network with a time-division multiplexed schedule of transmissions. This requires some coarse-level timing synchronization shared across all nodes in the transmit array as well as all the receivers, achieved using network synchronization protocols such as [31].

**Slowly varying phase offsets.** We assume that channel gains and oscillator offsets vary relatively slowly in time so that they can be reasonably taken to be static over several iterations of our algorithm. With time-slots of duration $\approx 50$ ms, this requires that the channels and offsets are roughly constant over several seconds. This condition can be assured by using standard filtering techniques [32] that dynamically track motions-related Doppler effects and clock dynamics.

**Local channel state information.** We assume the availability of local channel state information, specifically knowledge at each transmitter of its own complex channel gains to each receiver. These can be obtained either using an explicit training, estimation and channel feedback procedure, or implicitly by transmitters inferring their downlink channels using uplink transmissions from the receivers using reciprocity. Some details are as under:

- **Explicit channel feedback.** This is the most direct way of obtaining the necessary CSI. Under this method, there is a dedicated time-slot for channel state estimation, which starts with transmitter 1 individually transmitting a

known sequence of training symbols; these symbols are received by the beam and null-targets which use their knowledge of the training symbols to estimate the channel gains $h_{1p},\ p \in 1 \ldots M$ where $M$ is the number of receivers. These channel gains are then sent as feedback to transmitter 1. Note that the estimates of transmitter 1's channel gains only need to be sent to transmitter 1. The process is then repeated for transmitter 2 and so on.

- **Aggregate rich feedback method.** Under this class of techniques [29], the transmitters concurrently send uncorrelated training sequences, and the resulting fluctuations in the aggregate received signal can be quantized, encoded and fed back to the transmitters which are then able to extract estimates of their own channel gains to the receiver using only the knowledge of their own training sequences. Since the channel feedback signal is common to all transmitters, it can be combined with the feedback required by the algorithms in a natural and scalable way.

- **Reciprocity-based method.** Under this method, the transmitters observe the uplink feedback signals sent by the beam and null-targets, and are automatically able to infer their downlink channel gains using reciprocity. Note that this method only allows each transmitter to learn *its own channel gains* to the receiver and is thus perfectly suited to our algorithms. However, reciprocity does require some prior calibration [33] of the transmitting nodes.

## 1.6 Applications

Under a per-antenna power constraint and assuming equal-gain channels, an ideal $N$-antenna beamformer provides an $N^2$-fold coherent power gain on target. Incoherent transmission, e.g., transmitting with random phases, provides an $N$-fold power pooling gain, on average. Ideal nullforming results in zero power on the target. Motivating applications of nullforming and beamforming include cooperative jamming or communications, where the goal is to maximize the net transmitted power using multiple transmitters while simultaneously protecting a designated receiver. For example in a cognitive radio system [34] where the transmit array is a secondary user of licensed spectrum which seeks to communicate with a set of secondary receivers (beam targets) without causing any interference at primary receivers (null targets). Another possible application is a cellular network where adjacent Base Stations form a transmit array and coordinate their transmissions to avoid cochannel intereference. Recent algorithms on wireless security critically rely on nodes blanketing a landscape with full power jamming signals while protecting a cooperating receiver through nullforming [35]. So a third application can be electronic warfare where a transmit array broadcasts strong jamming signals [36] that disable an enemy's communication infrastructure while protecting friendly stations (null targets) from interference due to the jamming signal. The joint beam and nullforming specifically can be more generally thought of as a fundamental building block for increased spatial spectrum reuse [37] and toward achieving the full spatial multiplexing gains available from MIMO techniques [38] with distributed antenna arrays.

## 1.7 Organization

The rest of this thesis is organized as follows. Chapter 2 describes the phase-only update of transmit array setup for single receiver nullforming. Chapter 3 provides robustness analysis of the single receiver nullforming algorithm. This leads to Chapter 4 which provides another algorithm to perform nullforming at single receiver but without prior frequency synchronization requirement. Next Chapter 5 addresses the joint beam and nullforming setup where we want to form beams at certain beam-target receivers while simultaneously forming nulls at other receivers by iterative updates to the entire complex weight instead of just the phase. Chapter 6 concludes the work presented in the Thesis and mentions scope for future research.

# CHAPTER 2
# DISTRIBUTED NULLFORMING WITH PHASE-ONLY ADAPTATION

This chapter considers the problem of distributed nullforming at a single receiver, in which multiple wireless transmitters steer a null toward the designated receiver by only adjusting their carrier phases. Since each transmitter transmits at full power, the system maximizes "power pooling" gains for cooperative communication or jamming, while simultaneously protecting a designated receiver. Analysis in a noiseless setting shows that, while the received power at the designated receiver, as a function of the transmitted phases, is non-convex with multiple critical points, all of its local minima are also global minima. This implies that a null can be formed using a distributed, scalable protocol based on gradient descent: each transmitter adapts its phase based only on *aggregate* feedback broadcast by the receiver (so that feedback overhead does not increase with the number of transmitters), along with an estimate of its own channel gain (which can be obtained, for example, via reciprocity). Simulations show that the convergence rate actually improves with the number of transmitters, and that the algorithm is robust to noise, substantial channel estimation errors, and oscillator drift.

The proposed algorithm is scalable on several fronts: (i) it only requires aggregate feedback from the receiver, so that the feedback overhead remains constant as the number of transmitters increases; (ii) each transmitter only requires local information (i.e., an estimate of its own channel to the receiver), which can be acquired

efficiently using reciprocity in TDD systems using broadcast from the receiver (e.g., using the feedback packets themselves); and (iii) nulls are formed more effectively for larger transmit clusters.

This algorithm, formulated as a gradient-descent minimization of the received power, and is presented in Section 2.1[21]. Section 2.2 characterizes the critical points of the cost function (which is the received power as a function of the transmitter phases) and shows that all local minima are global minima. Section 2.3 argues that this guarantees the *practical* convergence of gradient descent to a global minimum of the cost function. While our formal analysis is for an idealized model, simulations in Section 2.4 verify the robustness to *large channel estimation errors* and also shows that the convergence speed actually *increases* with the number of transmitters. The latter observation leads to the remarkable fact that the total amount of feedback overhead required to achieve nullforming — not merely the overhead per iteration — actually decreases as the number of transmitters increases.

## 2.1 Algorithm description

We should recall that prior frequency synchronization and a time-slotted structure have been assumed as discussed in Section 1.5. Any of the frequency synchronization techniques enumerated there are appropriate for use with the gradient descent nullforming algorithm developed here.

Consider an $N$-transmitter array. Suppose the channel phase from the $i$-th transmitter to the receiver is $r_i e^{\nu_i[k]}$ at time slot $k$ and denote the estimate of

the channel phase as $\hat{\nu}_i[k]$. Each transmitter precompensates its channel phase to the receiver by the estimate $\hat{\nu}_i[k]$, and inserts an additional phase $\theta_i[k]$, which it adapts in response to receiver feedback, to effect nullforming. Thus, at time slot $k$, assuming each node transmits an unmodulated carrier for notational simplicity, the $i$-th node transmits the phase compensated baseband signal $e^{j(\theta_i[k]-\hat{\nu}_i[k])}$. With $w[k] \sim \mathcal{CN}(0, \sigma_w^2)$, the aggregate baseband signal at the receiver is thus

$$s[k] = \sum_{i=1}^{N} r_i e^{j(\theta_i[k]+\phi_i[k])} + w[k] = R[k] + jI[k], \tag{2.1}$$

where $\phi_i[k] = \nu_i[k] - \hat{\nu}_i[k]$ is the channel estimation error.

Our algorithm for nullforming is the gradient descent minimization of the objective function $J(\boldsymbol{\theta})$ defined as

$$J(\boldsymbol{\theta}) \doteq \left| \sum_{i=1}^{N} r_i e^{j(\theta_i+\phi_i)} \right|^2 \tag{2.2}$$

where $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_N]^\top$. Note that $J(\boldsymbol{\theta}[k])$ is the *received power* in the $k$-th time slot. For a suitably small $\mu > 0$, the gradient descent is specified as

$$\boldsymbol{\theta}[k+1] = \boldsymbol{\theta}[k] - \mu \left. \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}[k]} \tag{2.3}$$

where, $\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \left[ \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1}, \cdots, \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_N} \right]^\top$.

According to (2.3), the $i^{\text{th}}$ transmitter then updates its phase as

$$\theta_i[k+1] = \theta_i[k] - \mu r_i \text{Im}\left[ e^{-j(\theta_i[k]+\phi_i[k])} s[k] \right]. \tag{2.4}$$

In practice this requires the knowledge of the channel phase error $\phi_i$ and the actual

channel gain $r_i$. As these are not available, we instead use:

$$\theta_i[k+1] = \theta_i[k] - \mu\hat{r}_i \mathrm{Im}\left[e^{-j\theta_i[k]} s[k]\right] \tag{2.5}$$

$$= \theta_i[k] + \mu\hat{r}_i \left(\sin\left(\theta_i[k]\right) R[k] - \cos\left(\theta_i[k]\right) I[k]\right).$$

We note from (2.5), that transmitter $i$ only requires knowledge of its own estimated channel gains $\hat{r}_i$ and one additional complex number $s[k]$ which is common to all transmitters. Hence, given the common feedback $s[k]$ and local channel knowledge, the gradient descent (2.5) can be implemented by each transmitter independently in a *purely decentralized manner*. Furthermore, the common feedback $s[k]$ ensures that the nullforming *feedback overhead is fixed* and independent of the size of the transmitter array. The feedback signals $\{s[k]\}$ can be broadcast to the transmitters over, for example, a packetized digital wireless link.

Another important consideration is how best to obtain the local channel state information required to implement (2.5). Once again we can chose from the methods mentioned in Section 1.5.

## 2.2 Preliminaries of stability analysis: critical points and nulls

In this section, we investigate the structure of the cost function $J(\boldsymbol{\theta})$ in terms of its critical points. The next section considers the convergence of the decentralized gradient descent algorithm. To develop analytical insight, we will assume an idealized setting of no noise, zero channel estimation errors and static channel gains and make the following simplifying *standing assumptions* for all results in this and the next section: That the channel phases are time-invariant ($\nu_i[k]$ is constant over $k$ for all

transmitters $i \in \{1, ..., N\}$), the channel estimation errors are zero ($\phi_i[k] \equiv 0$ for all $k$ and all $i \in \{1, ..., N\}$)), the estimated channel gains $\hat{r}_i$ equal the true channel gains $r_i$ ($\hat{r}_i = r_i$), and that there is no noise in the receiver feedback ($w[k] = 0$). To avoid triviality, we assume that at least two transmitters have nonzero channel gains (i.e., $r_i > 0$ for at least two values of $i \in \{1, ..., N\}$).

**Assumption 2.2.1.** *In (2.1-2.5), $r_i > 0$, $\phi_i = 0$, $w[k] = 0$ and $\hat{r}_i = r_i$.*

Under Assumption 2.2.1, we have

$$J(\boldsymbol{\theta}) \doteq \left| \sum_{i=1}^{N} r_i e^{j\theta_i} \right|^2 \tag{2.6}$$

$$R[k] = \sum_{i=1}^{N} r_i \cos(\theta_i[k]), \;\; I[k] = \sum_{i=1}^{N} r_i \sin(\theta_i[k]). \tag{2.7}$$

and

$$\begin{aligned} \theta_i[k+1] &= \theta_i[k] - \mu r_i \mathrm{Im}\left[e^{-j\theta_i[k]} s[k]\right] \\ &= \theta_i[k] + \mu r_i \left(\sin\left(\theta_i[k]\right) R[k] - \cos\left(\theta_i[k]\right) I[k]\right). \end{aligned} \tag{2.8}$$

The critical points of the algorithm by definition satisfy

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0. \tag{2.9}$$

The $ij$-th element of the Hessian $\boldsymbol{H}(\boldsymbol{\theta})$ is

$$\left[\boldsymbol{H}(\boldsymbol{\theta})\right]_{ij} = \frac{\partial^2 J(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j}. \tag{2.10}$$

If a critical point is a local minimum, then the Hessian at that point is positive semidefinite. In general, we can have critical points which are not local minima, and this is indeed the case for the cost function in (2.6).

The technical properties derived in this section, and their consequences, are summarized as follows:

- All nulls and critical points of $J(\boldsymbol{\theta})$ lie along manifolds. All points in the null manifold $J(\boldsymbol{\theta}) = 0$ are critical points but not all critical points are in the null manifold.

- The Hessian of $J(\boldsymbol{\theta})$ is singular everywhere (unlike standard settings for gradient descent, where the Hessian is strictly positive definite at local minima). This precludes the use of standard gradient descent convergence and stability results and requires a more careful analysis of the properties of $J(\boldsymbol{\theta})$.

- All local minima are global minima. We characterize the depth of the null corresponding to this global minimum in terms of the channel gains $\{r_i\}$.

- We characterize all critical points which are not local minima (i.e., such that the Hessian has at least one negative eigenvalue).

- The key technical take away from this section used in Section 2.3 is that the only critical points at which the Hessian is positive semidefinite are global minima.

The following section provides a formal analysis the properties of the critical points of $J(\boldsymbol{\theta})$.

### 2.2.1 Properties of critical points

For every $\alpha > 0$, the set of $\boldsymbol{\theta}$ for which $J(\boldsymbol{\theta}) = \alpha$ is either empty or is a nontrivial manifold as

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N} r_i^2 + 2\sum_{i=1}^{N}\sum_{l=1,l\neq i}^{N} r_i r_l \cos(\theta_i - \theta_l). \tag{2.11}$$

That is, the cost function depends only on phase differences, and does not change when we add a constant offset to all phases. Thus, for all scalar $\beta$ and the $N$-vector $\boldsymbol{u} = [1, \cdots, 1]^\top$, $J(\boldsymbol{\theta}) = \alpha$ implies $J(\boldsymbol{\theta}+\beta\boldsymbol{u}) = \alpha$. For example, $J(\boldsymbol{\theta}) = 0$ corresponds to a *null manifold.*

Similarly,

$$\begin{aligned}
\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_i} &= -2r_i \sin\theta_i \left(\sum_{l=1}^{N} r_l \cos\theta_l\right) \\
&+ 2r_i \cos\theta_i \left(\sum_{l=1}^{N} r_l \sin\theta_l\right) \qquad (2.12) \\
&= -2\sum_{l=1,l\neq i}^{N} r_i r_l \sin(\theta_i - \theta_l), \qquad (2.13)
\end{aligned}$$

Again, the gradient depends only on phase differences, hence we can add constant offsets without changing it. Thus, any critical point lies on a *critical manifold.* As explained in Section 2.3, this complicates stability analysis.

All members of the null manifold $J(\boldsymbol{\theta}) = 0$ satisfy $\sum_{l=1}^{N} r_l \cos\theta_l = 0$ and $\sum_{l=1}^{N} r_l \sin\theta_l = 0$. Substituting these equalities into (2.12), we see that any point on the null manifold is also a critical point. However, there are critical points that do not lie on the null manifold. From (2.12), these other critical points *must satisfy*

$$\tan\theta_i = \tan\theta_l \quad \forall\{i,l\} \subset \{1, \cdots, N\}. \tag{2.14}$$

This corresponds to the phases being offset by integer multiples of $\pi$, i.e.,

$$(\theta_i - \theta_l) \mod \pi = 0 \quad \forall \{i, l\} \subset \{1, \cdots, N\}. \tag{2.15}$$

From (2.13), the condition in (2.15) is also sufficient for the gradient to be zero. Thus, (2.15) and the null manifold together constitute *all* the critical points of $J(\boldsymbol{\theta})$.

We define the minimum value of $J(\boldsymbol{\theta})$

$$J^* = \min_{\boldsymbol{\theta} \in \mathbb{R}^N} J(\boldsymbol{\theta}). \tag{2.16}$$

Should the null manifold be non-empty, then $J^* = 0$. However, for some choices of the gains $r_i$ there may not be phases $\theta_i$ for which $J(\boldsymbol{\theta}) = 0$. In essence, if one channel gain is larger than the sum of all the rest, then it is clear that the best we can do is to make sure we coherently subtract all of the smaller gains from the largest one to minimize $J(\boldsymbol{\theta})$. The more interesting result is that, whenever this condition is not satisfied (i.e., whenever no one gain is larger than the sum of the rest), then an ideal null is possible. The theorem below, proved in our work in [21], formalizes this characterization.

**Theorem 2.1.** *Consider Assumption 2.2.1. Assume $r_i \geq r_{i+1} > 0$ and $N > 1$. Then $J^* > 0$ iff*

$$r_1 > \sum_{l=2}^{N} r_l. \tag{2.17}$$

*Under (2.17) the $\theta_i$ that minimize $J(\boldsymbol{\theta})$ obey: For integer $m_l$, and $l \in \{2, \cdots, N\}$*

$$\theta_1 - \theta_l = (2m_l + 1)\pi \tag{2.18}$$

*resulting in the coherent subtraction of the smaller gains yielding the following minimum:*

$$J^* = \left( r_1 - \sum_{l=2}^{N} r_l \right)^2 . \tag{2.19}$$

We now examine the structure of the Hessian. From (2.13) we have

$$[\boldsymbol{H}(\boldsymbol{\theta})]_{il} = \begin{cases} -2 \sum_{\substack{l=1 \\ l \neq i}}^{N} r_i r_l \cos(\theta_i - \theta_l) & i = l \\ 2 r_i r_l \cos(\theta_i - \theta_l) & i \neq l. \end{cases} \tag{2.20}$$

The Hessian is *always singular* because all row sums are zero, i.e.,

$$\sum_{l=1}^{N} [\boldsymbol{H}(\boldsymbol{\theta})]_{il} = 0. \tag{2.21}$$

for all $i$. If we define

$$\boldsymbol{c}(\boldsymbol{\theta}) = [\cos\theta_1, \cdots, \cos\theta_N]^\top \text{ and} \tag{2.22}$$

$$\boldsymbol{s}(\boldsymbol{\theta}) = [\sin\theta_1, \cdots, \sin\theta_N]^\top \tag{2.23}$$

then it is readily seen that

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{\theta}) &= 2\mathrm{diag}\{\boldsymbol{r}\} \left( \boldsymbol{c}(\boldsymbol{\theta}) \boldsymbol{c}^\top(\boldsymbol{\theta}) + \boldsymbol{s}(\boldsymbol{\theta}) \boldsymbol{s}^\top(\boldsymbol{\theta}) \right) \mathrm{diag}\{\boldsymbol{r}\} \\ &- 2\mathrm{diag}\{\delta_i\}_{i=1}^{N}, \end{aligned} \tag{2.24}$$

where

$$\delta_i = r_i \cos\theta_i \sum_{l=1}^{N} r_l \cos\theta_l + r_i \sin\theta_i \sum_{l=1}^{N} r_l \sin\theta_l. \tag{2.25}$$

*Hessian at global minima:* As characterized by Theorem 2.1, the global minimum is an ideal null with $J^* = 0$ if (2.17) does not hold, and is given by (2.19) if it does hold. In either case (i.e., whether or not (2.17) holds), the Hessian will be positive semidefinite

at these global minima, since global minima are also local minima. However, the Hessian is *never positive definite* because it is singular everywhere. This is consistent with the fact that the global minima are not isolated but rather lie on nontrivial manifolds.

The next Lemma provides a compact expression for the Hessian at the critical points that do not lie on the null manifold. This is then used to show in Theorem 2.3 that critical points that do not correspond to a global minimum are unstable.

**Lemma 2.2.** *Under Assumption 2.2.1, suppose $\boldsymbol{\theta}$ is a critical point that is not a null. Then the following hold:*

*(i) There exist $\mathcal{I}$ and $\mathcal{I}^c$ that partition $\{1, \cdots, N\}$, and obey*

$$(\theta_i - \theta_l) \mod 2\pi = \begin{cases} 0 & \forall \{i,l\} \subset \mathcal{I} \\ \pi & \forall\, i \in \mathcal{I} \text{ and } l \in \mathcal{I}^c \end{cases}. \tag{2.26}$$

*(ii) The Hessian defined in (2.20) can be expressed as:*

$$\boldsymbol{H}(\boldsymbol{\theta}) = 2\left[\boldsymbol{x}\boldsymbol{x}^\top - \operatorname{diag}\{\boldsymbol{x}\} \sum_{i=1}^{N} x_i\right], \tag{2.27}$$

*where $\boldsymbol{x} = [x_1, \cdots, x_N]^\top$ obeys:*

$$x_i = \begin{cases} r_i & \forall\, i \in \mathcal{I} \\ -r_i & \forall\, i \in \mathcal{I}^c \end{cases}. \tag{2.28}$$

The proof of this result follows directly from (2.15) and (2.20).

The following section analyzes the properties of the Hessian of $J(\boldsymbol{\theta})$ to further characterize the critical points.

### 2.2.2 Eigenvalues of the Hessian

The following theorem shows that $\boldsymbol{H}(\boldsymbol{\theta})$ has a negative eigenvalue at any critical point that is not a global minimum.

**Theorem 2.3.** *Consider Assumption 2.2.1. Assume $r_i \geq r_{i+1} > 0$ and $N > 1$. If $\boldsymbol{\theta}$ is a critical point that is not a global minimum, then $\boldsymbol{H}(\boldsymbol{\theta})$ has a negative eigenvalue.*

*Proof.* First consider $N = 2$. Suppose $\boldsymbol{\theta}$ is a critical point that is not a global minimum. If a null is not possible then from Theorem 2.1 and (2.18), $(\theta_1 - \theta_2) \mod 2\pi = 0$. The same holds when a null is possible as $r_1 = r_2$. Thus (2.20) implies that both diagonal elements of the symmetric matrix $\boldsymbol{H}(\boldsymbol{\theta})$ are negative, i.e. $\boldsymbol{H}(\boldsymbol{\theta})$ must have a negative eigenvalue.

For $N > 2$ from Lemma 2.2, (2.27) holds under (2.28). Note $|x_i| \geq |x_{i+1}| > 0$. Define

$$s_l = x_l \sum_{i=1}^{N} x_i. \tag{2.29}$$

We assert that at a critical point that is not a global minimum, $s_l > 0$ for at least two distinct $l \in \{1, \cdots, N\}$. To prove this consider two cases.

**Case I: A null is impossible.** Then from Theorem 2.1, (2.17), (2.18) and (2.27) there exists $i \in \{2, \cdots, N\}$ such that

$$x_1 > \sum_{l=2}^{N} |x_l| \text{ and } x_1 x_i > 0. \tag{2.30}$$

Thus, as $\sum_{i=1}^{N} x_i > 0$, $s_l > 0$ for $l \in \{1, i\}$.

**Case II: A null is possible.** From Theorem 2.1 for all $i$,

$$|x_i| \leq \sum_{l=1, l\neq i}^{n} |x_l|. \tag{2.31}$$

Clearly there must exist at least two indices $k$ and $l$ such $x_k$ and $x_l$ have the same sign as $\sum_{i=1}^{N} x_i$. Thus again there are two distinct indices for which $s_i > 0$.

Now observe from Lemma 2.2 that with

$$\boldsymbol{D} = \text{ diag } \{\boldsymbol{x}\} \sum_{i=1}^{N} x_i = \text{ diag } \{d_1, \cdots, d_N\}, \tag{2.32}$$

$$\boldsymbol{H}(\boldsymbol{\theta}) = 2\left(\boldsymbol{x}\boldsymbol{x}^\top - \boldsymbol{D}\right)$$

with at least two elements of $\boldsymbol{D}$ positive.

As eigenvalues do not change under symmetric permutations, without loss of generality assume $d_1 > 0$ and $d_2 > 0$. As $N > 1$, and all $x_i \neq 0$, there exist nonzero scalars $p_1$ and $p_2$, such that $\left[p1, p2, \mathbf{0}^\top\right] \boldsymbol{x} = 0$, where the zero vector is in $\mathbb{R}^{N-2}$. As $\boldsymbol{H}(\theta) = \boldsymbol{H}^\top(\boldsymbol{\theta})$ it has a negative eigenvalue as

$$\left[p_1, p_2, \mathbf{0}^\top\right] \left(\boldsymbol{x}\boldsymbol{x}^\top - \boldsymbol{D}\right) \begin{bmatrix} p_1 \\ p_2 \\ \mathbf{0} \end{bmatrix} = -d_1 p_1^2 - d_2 p_2^2 < 0$$

□

As at a local minimum $\boldsymbol{H}(\theta)$ cannot have a negative eigenvalue, Theorem 2.3 shows that a critical point that is not a global minimum cannot be a local minimum. This has implications to the stability analysis in the next section.

It is worth noting that at a null $\delta_i = 0$ in (2.25), i.e. from (2.24)

$$\boldsymbol{H}(\boldsymbol{\theta}) = 2\text{diag}\{\boldsymbol{r}\} \left[\boldsymbol{c}(\boldsymbol{\theta})\boldsymbol{c}^\top(\boldsymbol{\theta}) + \boldsymbol{s}(\boldsymbol{\theta})\boldsymbol{s}^\top(\boldsymbol{\theta})\right] \text{diag}\{\boldsymbol{r}\} \tag{2.33}$$

Thus, at a null, $\boldsymbol{H}(\boldsymbol{\theta})$ is positive semidefinite and in fact has rank at most 2. Of

course as $\boldsymbol{H}(\boldsymbol{\theta})$ is positive semidefinite at a global minimum, its eigenvalues must be nonnegative with at least one zero.

## 2.3 Stability analysis

Having characterized the nature of critical manifolds in the last section, we now establish the *practical uniform stability* of the gradient descent algorithm under our idealized setting (no noise, ideal channel phase estimates, time-invariant channel), by showing three things. (A) That the phase estimates *uniformly converge to a single point* on a critical manifold, where uniformity is with respect to the initial time. (B) That all critical points that are not global minima are locally unstable. (C) That the global minima are locally stable.

The practical implication of (A-C) are as follows. Uniform convergence to a point under idealized assumptions assures that such convergence is robust to departures from idealizations, [39]. The role of (B) and (C) is to show that in practical terms the point to which such robust convergence can occur must be a global minimum. This is so as (B) shows that convergence to a critical point that is not a global minimum if at all possible, will never be sustained as the slightest noise will drive the phase trajectories away from them.

While (B) is based on standard arguments, the proofs of (A) and (C) are nontrivial because in our setting *the Hessian is never positive definite.* Since the gradient and the Hessian are bounded and the gradient is Lipschitz continuous, arguments similar to that in [40] show the following: For sufficiently small $\mu$, there exists

$0 < \epsilon(\mu) < 1$ such that along the trajectories of (2.8) or equivalently (2.3), there holds,

$$J(\boldsymbol{\theta}[k+1]) \leq J(\boldsymbol{\theta}[k]) - (1-\epsilon(\mu)) \left\| \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}[k]} \right\|^2 . \tag{2.34}$$

As $J(\boldsymbol{\theta}[k]) \geq 0$, this does imply that the gradient is in $\ell_2$. However, as is well known (see [41] for examples), this does not by itself imply that the *gradient actually goes to zero*. Unless the gradient goes to zero the updates in (2.3) or equivalently in (2.8), will not cease and convergence will not occur. Indeed example convergence analyses of descent based algorithms (e.g., Newton-Raphson) in [40], that go beyond just showing that the gradient is in $\ell_2$, assume a positive definite Hessian. Even in the classical LMS algorithm, one cannot conclude that convergence to a point occurs without a condition known as persistent excitation, which is a variation of the positive definiteness condition on the Hessian [42], .

The local stability of the global minimum is also complicated by the fact that, without a positive definite Hessian, linearization around a minimum yields a transition matrix that has eigenvalues at unity. Indeed, to address the lack of positive definiteness, the recent paper [43] invokes the highly technical *center manifold theory*. Given these difficulties, we prove (A) from (2.34) by appealing to the further device of Lasalle's invariance principle, [44], summarized in Theorem 2.4. This convergence result is stated in Theorem 2.5. Theorem 2.6 proves (B), the local instability of critical points that are not global minima. Theorem 4.7 proves (C) without having to appeal to center manifold theory.

*(A) Convergence:* Theorem 2.4 summarizing Lasalle's invariance principle refers to

the *lack of explicit dependence on time* in an update kernel. An example is the update kernel in (2.8). The update depends on $k$ only through the value of $\theta_i[k]$ at that $k$.

**Theorem 2.4.** *Consider the state equation*

$$\boldsymbol{\xi}[k+1] = \boldsymbol{f}(\boldsymbol{\xi}[k]), \quad \forall k \geq k_0 \tag{2.35}$$

*where $k$ and $k_0$ are integers, and $f(\boldsymbol{\xi}[k])$ has no explicit dependence on $k$. Suppose the following conditions hold: (a) $\boldsymbol{\xi}[k]$ is uniformly bounded for every finite $\boldsymbol{\xi}[k_0]$. (b) There exists a nonegative function $V(\boldsymbol{\xi}[k])$ such that the following holds for all $k \geq k_0$ along the trajectories of (2.35):*

$$V(\boldsymbol{\xi}[k+1]) \leq V(\boldsymbol{\xi}[k]). \tag{2.36}$$

*(c) For all finite $\boldsymbol{\xi}[k_0]$, $V(\boldsymbol{\xi}[k])$ is uniformly bounded. Then $\boldsymbol{\xi}[k]$ uniformly converges to a trajectory of (2.35) on which $V(\boldsymbol{\xi}[k])$ is a constant.*

The next theorem establishes (A) and the fact that along the trajectories of (2.8) $J(\boldsymbol{\theta}[k])$ is nonincreasing.

**Theorem 2.5.** *Under (2.7), (2.8),(2.6) and Assumption 2.2.1, there exists a $\mu^* > 0$, such that for all $0 < \mu < \mu^*$, initial time $k_0$ and $\boldsymbol{\theta}[k_0] \in \mathbb{R}^N$ the following hold:*

$$J(\boldsymbol{\theta}[k+1]) \leq J(\boldsymbol{\theta}[k]) \quad \forall k \geq k_0 \tag{2.37}$$

*and*

$$\lim_{k\to\infty} \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}[k]} = 0 \tag{2.38}$$

*Further, the convergence in (2.38) is uniform in $k_0$ and there exists a critical point $\boldsymbol{\theta}^*$ such that*

$$\lim_{k\to\infty} \boldsymbol{\theta}[k] = \boldsymbol{\theta}^*. \tag{2.39}$$

*Proof.* Because of (2.34), (2.37) holds. Now we invoke Theorem 2.4 whose application requires the boundedness of $\boldsymbol{\theta}[k]$ *ab initio*. For this difficulty, we reformulate the state space to make it *a priori* bounded, by choosing $\xi_i = (\theta_i \mod 2\pi)$ as the elements of the new state vector. Define $\boldsymbol{f(\cdot)} = [f_1(\cdot), \cdots, f_N(\cdot)]^\top$, and $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_N]^\top$. The definition of the $\xi_i$, ensures that this state space is bounded. Observe, $J(\boldsymbol{\theta}) = J(\boldsymbol{\xi})$ and

$$\left.\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}[k]} = \left.\frac{\partial J(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}[k]}. \tag{2.40}$$

Define,

$$g_i(\boldsymbol{\xi}) = \left(\xi_i - \mu \frac{\partial J(\boldsymbol{\xi})}{\partial \xi_i}\right) \mod 2\pi.$$

Under these definitions (2.8) leads to (2.35). By defintion this state space is bounded. Identify $V(\cdot)$ with $J(\cdot)$. Then because of (2.34), $V(\cdot)$ satisfies all the conditions in Theorem 2.4. Consequently, $\boldsymbol{\xi}[k]$ converges uniformly to a trajectory where $J(\boldsymbol{\xi}[k]) = J(\boldsymbol{\theta}[k])$ is a constant. From (2.34) and (2.40) this must correspond to the trajectory:

$$\left.\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}[k]} = \left.\frac{\partial J(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}[k]} = 0.$$

Further along this trajectory $\boldsymbol{\xi}[k+1] = \boldsymbol{\xi}[k]$. It remains to show that this also implies that $\boldsymbol{\theta}[k+1] = \boldsymbol{\theta}[k]$. Observe,

$$|\theta_i[k+1] - \theta_i[k]| \leq 2\mu r_i \sum_{l=1}^{N} r_l.$$

Suppose $0 < \mu \leq \bar{\mu}$ assures (2.34). Choose

$$\mu^* < \min\left\{\bar{\mu}, \min_{i\in\{1,\cdots,N\}}\left\{\frac{\pi}{r_i \sum_{l=1}^{N} r_l}\right\}\right\}. \tag{2.41}$$

Then as $\xi_i = (\theta_i \mod 2\pi)$, for all $0 < \mu < \mu^*$, $\xi_i[k+1] = \xi_i[k]$ implies $\theta_i[k+1] = \theta_i[k]$. This completes the proof.

□

Thus, $\boldsymbol{\theta}[k]$ is guaranteed to uniformly converge to a *point* in the critical manifold.

*Intuition behind proof of convergence:* While the details are in the appendix, the intuition behind the preceding development is as follows. First, (2.37) follows from (2.34). Second to show (2.39) by invoking Lasalle's invariance principle, we observe that gradient descent operates on the unwrapped phases $\boldsymbol{\theta}$, which can therefore be unbounded. Lasalle's principle is applied to the bounded *wrapped* phases $\boldsymbol{\xi}$ to conclude that these converge. We then show that, for small enough adaptation gain $\mu$, we can guarantee that the unwrapped phases do not jump around too much under our gradient descent algorithm, and hence inherit the convergence of the wrapped phases.

*(B) Instability of critical points which are not minima:* Linearization of (2.8) around any $\boldsymbol{\theta}^*$ is given by, $\boldsymbol{\eta}[k+1] = [\boldsymbol{I} - \mu \boldsymbol{H}(\boldsymbol{\theta^*})]\boldsymbol{\eta}[k]$, with $\boldsymbol{\eta} = \boldsymbol{\theta} - \boldsymbol{\theta}^*$. From Theorem 2.3, at a critical point that is not a global minimum, $[\boldsymbol{I} - \mu \boldsymbol{H}(\boldsymbol{\theta^*})]$ has a positive eigenvalue. Thus we have:

**Theorem 2.6.** *Consider (2.8), under (2.7) and Assumption 2.2.1. Then (2.8) is locally unstable around any critical point that is not a global minimum.*

*(C) Local stability of global minima:* This proof is complicated by the fact that the Hessian is singular. Thus, at every $\boldsymbol{\theta}$, $\boldsymbol{I} - \mu \boldsymbol{H}(\boldsymbol{\theta})$ has an eigenvalue at 1 and standard theory does not prove local stability. Suppose $J_1$ is the smallest value $J(\boldsymbol{\theta})$ takes at a critical point that is not a global minumum. Suppose, at the initial condition, $\boldsymbol{\theta}[k_0]$, $J(\boldsymbol{\theta}[\boldsymbol{k_0}]) < J_1$. Then because of (2.37) $J(\boldsymbol{\theta}[\boldsymbol{k}]) < J_1$, for all $k \geq k_0$. Thus the only critical points that can be attained are global minima. As (2.38) assures convergence to a critical point, the limit point has to be a global minimum. We have thus shown the following.

**Theorem 2.7.** *Consider (2.8) under (2.7) and Assumption 2.2.1. With $k_0$ the initial time suppose $J(\boldsymbol{\theta}[k_0]) < J_1$ above. Then uniformly in $k_0$ there holds*

$$\lim_{k \to \infty} J(\boldsymbol{\theta}[k]) = J^*.$$

## 2.4 Simulation results

We now present simulations. Throughout, the receiver noise is AWGN and oscillator drift is Brownian motion. Observe that a feedback rate of even 100 packets per second, with 16 bytes/packet to represent a double-precision floating point complex number represents a rate of only 1600 bits/sec, far below the capacity of typical feedback channels. SNR is defined as the ratio between the signal power of a single transmitter at the receiver at the null target and the noise power. Received power is computed by averaging over several runs. *Incoherent* power is the expectation of the total received power when the received phases are random, equalling $N$ for $N$ transmitters when $r_i = 1$.

### 2.4.1 Noise, channel errors and oscillator drift

For $N = 10$, Fig. 2.1 depicts the received power achieved by our algorithm as a function of SNR without oscillator drift, but with large time-invariant channel phase estimation errors (modeling severe quantization or imprecise channel estimation), with $\phi_i$ uniformly distributed over $[0, \pi/2]$. Figure 2.2 models channel estimation errors in both gain and phase. The transmitter assumes that each channel gain $\hat{r}_i = 1$, even though each $r_i$ is actually uniformly distributed over $[1, 2]$ and each $\phi_i$ is uniformly distributed over $[0, \pi/2]$. These results show the robustness of our nullforming algorithm: despite these very substantial channel estimation errors, the received signal power nears the SNR floor. This suggests that for slowly varying channels, infrequent, even highly inaccurate or heavily quantized channel estimation suffices.

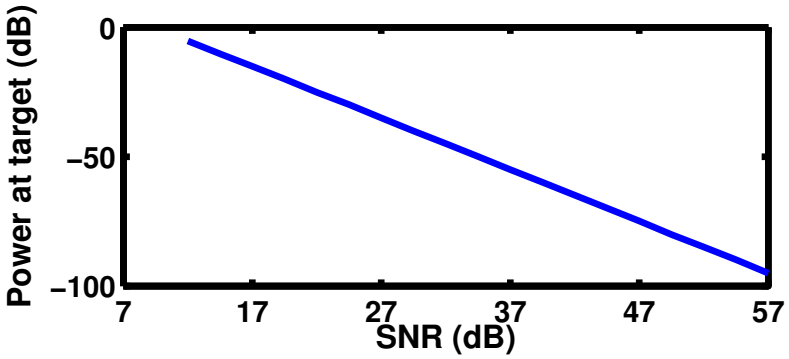



Figure 2.1: Power at null target vs. SNR with constant $\phi_i \sim \mathcal{U}[0, \pi/2]$.

Figure 2.3 shows performance under channel time variations. The transmitters

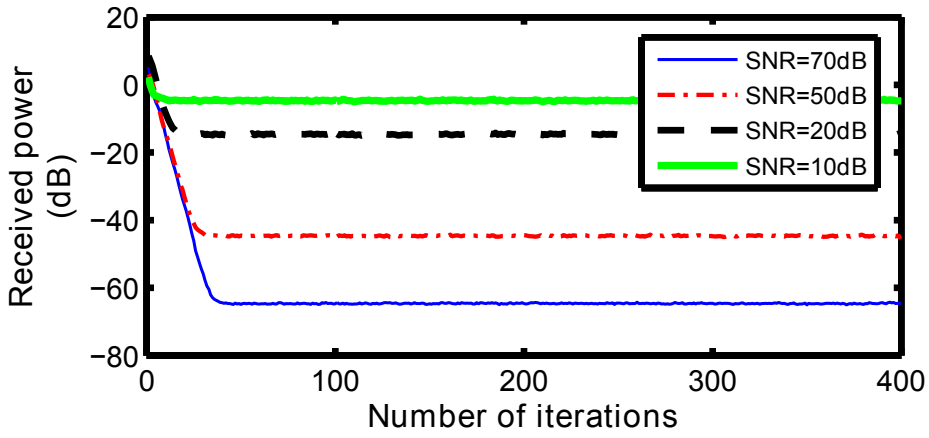



Figure 2.2: Power at null target with constant channel error: $r_i \sim \mathcal{U}[1,2]$ and $\phi_i \sim \mathcal{U}[0,\pi/2]$.

*always assume unit channel gains.* Each transmitter has an *initial error in channel phase estimation, uniformly distributed in* $[0°, 45°]$ *that is never corrected.* Thereafter, each channel changes every $\mathcal{C}$-th iteration of the algorithm. Each change in $r_i$ is by a factor of $\epsilon_i \sim \mathcal{U}[.99, 1.01]$, representing a one percent change. The change in $\phi_i$ is additive by $\delta_i \sim \mathcal{U}[-1.5°, 1.5°]$. Thus, with $\mathcal{C} = 1$ at a feedback rate of 100 packets/sec the gain can change by as much as 170% over a second and the phase by 150°, even discounting the *initial error*. We call $\mathcal{C}$ the *coherence*.

Suppose $\theta_i$ are such that one has a *perfect null* with unit channel gains and no phase errors. The actual received power due to a change in channel phase by $\delta_i$ and gain by a factor $\epsilon_i$ is:

$$J_{\text{change}}(\boldsymbol{\theta}) = \mathrm{E}\left[\left|\sum_{i=1}^{N} \epsilon_i e^{j(\theta_i+\delta_i)}\right|^2\right],$$

where the expectation is over $\epsilon_i \sim \mathcal{U}[.99, 1.01]$ and $\delta_i \sim \mathcal{U}[-1.5°, 1.5°]$. Consider

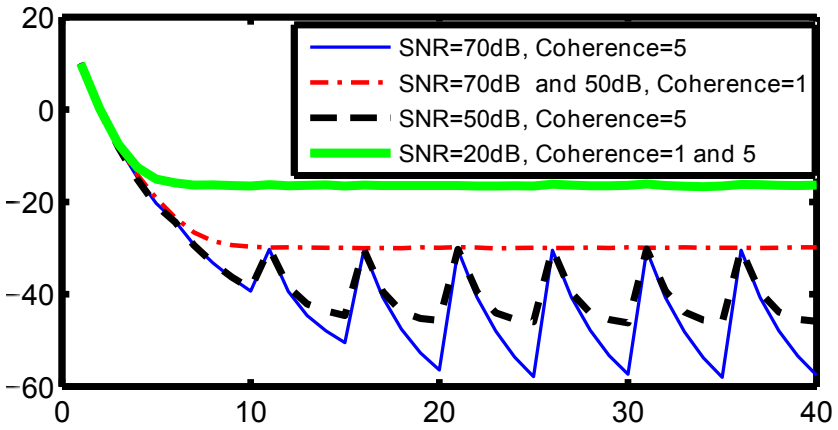



Figure 2.3: Power at null target with time varying channel gain and phase errors. Coherence, $\mathcal{C}$ means that the channel changes at every $\mathcal{C}$-th iteration.

$J_{\text{change}}(\boldsymbol{\theta})$ averaged over all $\theta_i$ such that $\sum_{i=1}^{N} e^{j\theta_i} = 0$. This would represent a theoretical floor for the algorithm performance for $\mathcal{C} = 1$ *without any initial channel error*. Figure (2.4) provides an estimate of this average, by averaging the power over 1000 runs with $\theta_i$ obtained independently in each run by running our algorithm in the noise free case. Evidently, while for SNRs of 50 and 70 dB our algorithm matches this performance, it is less than 3dB away for SNR of 20dB, *even though Figure 2.4 does not account for the initial phase error of as much as* $45°$.

Also interesting are the plots with $\mathcal{C} = 5$. At SNR of 50 and 70 dB, the received power rapidly declines between the channel transitions, and then expectedly returns to the $\mathcal{C} = 1$ level at transitions. The fact that the $\mathcal{C} = 1$ curve coincides at these two SNRs accords with the fact that in Figure 2.4 the received power at these two SNRs are identical, and that at the channel transitions the phase and gain change by the

requisite amounts. As interesting is the fact that performance for both $\mathcal{C} \in \{1, 5\}$ is identical for SNR of 20 dB; evidently the noise at this SNR swamps the effect of the channel transitions. Again note that Figure 2.4 ignores the initial phase error.

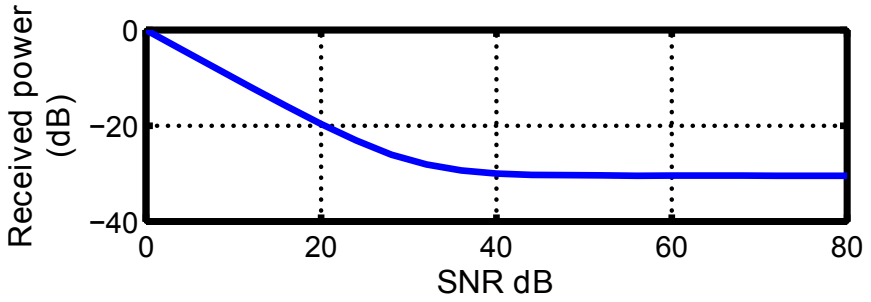



Figure 2.4: Ideal performance with $\mathcal{C} = 1$ and channel change as in Figure 2.3.

Received power vs. phase noise is also plotted in our paper [21]. Received power vs. the rms oscillator drift between two iterations of the algorithm, for different SNRs and with unit channel gains is plotted in Fig. 2.5. The null power is determined by SNR for small drifts, but the effect of drift dominates for when the rms drift between iterations exceeds 0.1°. Fig. 2.6 has comparable results with unequal Rayleigh distributed channel gains.

### 2.4.2 Convergence speed and scalability

As a common packetized feedback is broadcast to all transmitters, scalability is determined by how the convergence speed depends on $N$. Figure 2.7 depicts the relation between convergence speed and $N$: For $N = 50$ a $-40dB$ null is attained in

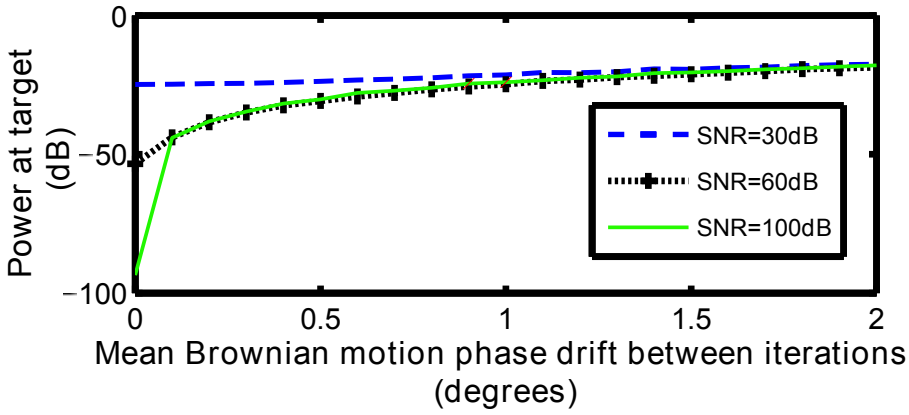



Figure 2.5: Power at null target vs. oscillator drift for equal channel gains.

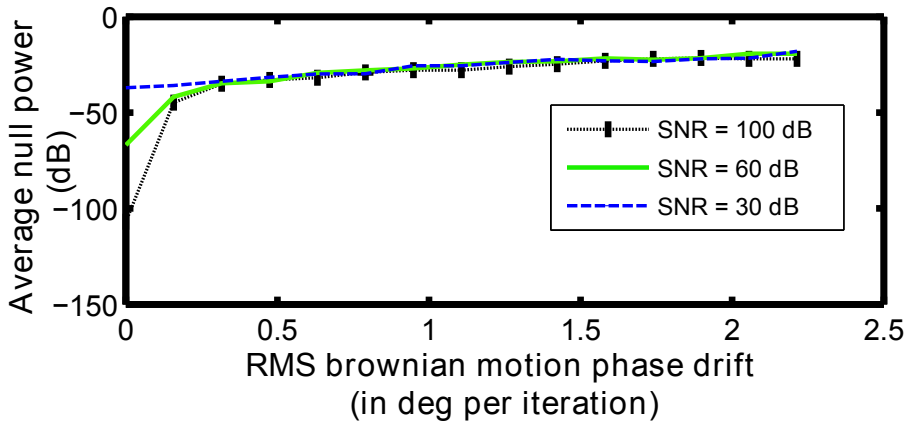



Figure 2.6: Power at null target vs. oscillator drift for unequal channel gains.

just five iterations. Compellingly, a null of $-40$ dB is acquired in *just 5 iterations,* which would, for example, only take 50 ms at a feedback rate of 100 packet/s. These simulation results attest to the scalability of our algorithm, showing that *the convergence speed improves with $N$,* even though we decrease the adaptation gain $\mu$ with $N$. Intuitively, this is because the dimension of the null manifold grows with $N$, shrinking its distance from generic points. Thus, the overhead associated with the packetized common feedback does not grow with the number of transmitters.

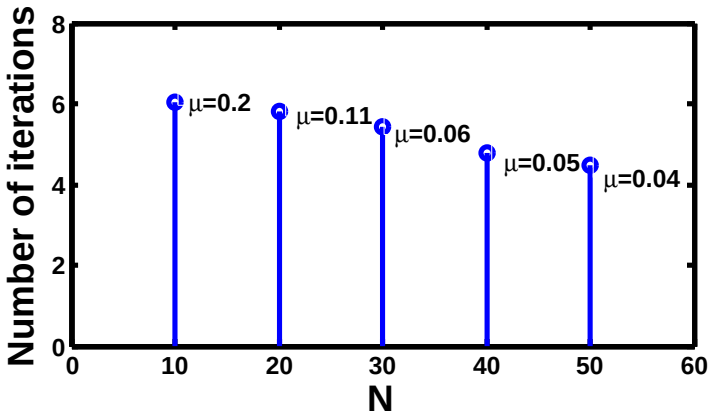



Figure 2.7: Number of iterations to achieve a 40-dB null vs. array size.

### 2.4.3 Effect on coherent beams

Fig. 2.8 shows performance with phases initialized to form a coherent beam at a location and then deploying our algorithm to nullform at a random target. A thousand random null targets were selected. The figure has the beam and null power averaged over these 1000 runs, as a result of the phases generated by our algorithm.

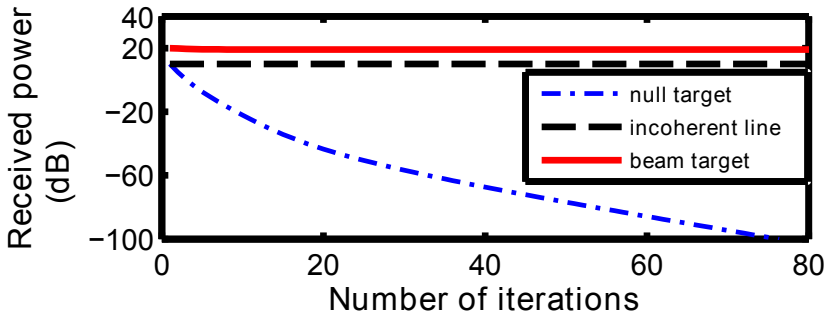



Figure 2.8: Effect of nullforming algorithm on power at a desired receiver.

The adjustments made by our algorithm barely dent the beam at the original location, despite achieving a deep null at the null targets, at least on average. Intuitively, this could be because a coherent beam is insensitive to small phase adjustments while nulls are very sensitive. A beam at one location could therefore be largely preserved with minor degradation, while applying small phase perturbations to synthesize a deep null at another designated location. This raises the intriguing possibility of forming both beams and nulls with phase-only adjustments, providing a building block for SDMA.

The scalable, distributed nullforming algorithm we developed in this chapter allows each transmitter to transmit at full power while steering a null toward a designated receiver through adaptation of the transmission phases using decentralized gradient descent. Unlike standard amplitude-phase adaptation with quadratic cost functions, the proof of convergence for phase-only adaptation required detailed examination of a highly non-convex cost function to prove that all local minima are global minima. Simulations show scalability in terms of nullforming performance: conver-

gence time does not increase with the number of transmitters. In the next chapter we provide an analysis for the robustness of this algorithm to errors in channel estimation.

# CHAPTER 3
# ROBUSTNESS ANALYSIS OF NULLFORMING ALGORITHM

The previous chapter presents a nullforming algorithm (2.5) with phase only adaptation, and demonstrates its uniform convergence under the idealized assumptions of zero channel estimation errors and noise (see Assumption 2.2.1). Simulations show that the (2.5) withstands significant violations of this assumption. This chapter provides a theoretical justification for this robustness.

Our approach is as follows. The difference between the algorithm (2.5) we actually implement, and the true gradient descent algorithm (2.4) is that $\phi_i \neq 0$ and $\hat{r}_i \neq r_i$. As explained further in this chapter, and as is well understood in the literature, (2.5) will still converge if the inner product between vector of phase updates in actual, and the true gradient exemplified in (2.4) were positive. In this chapter we give sufficient conditions under which this is true on the average.

## 3.1 Problem formulation

We assume that the channel estimation is performed at the start of the nullforming algorithm for example by using explicit channel information sent by receiver to transmitters. Since, the channel is assumed to be slowly varying for the duration of the few time-slots required for a few iterations of the algorithm, channel phases, $\nu_i[k]$, are assumed to be constant over $k$ for all transmitters $i \in \{1, ..., N\}$. Channel gains, are also assumed to be constant over the iterations for all transmitters $i \in \{1, ..., N\}$ and hence denoted simply as $r_i$ for the $i$-th transmitter. The algorithm (2.5) uses the

channel gain estimates $\hat{r}_i$ which are also assumed constant across the duration of the updates. Since we estimate the channel at the start in this setting, estimation errors remain for the duration of the algorithm. Therefore the channel phase estimation errors $\phi_i[k]$ are constant over $k$ for all $i$. Hereafter we will refer to these as $\phi_i$. We will denote the implemented gradient in (2.5) by

$$\frac{\partial J(\boldsymbol{\theta}, \mathbf{0}, \hat{\boldsymbol{r}})}{\partial \theta_m} \tag{3.1}$$

where $\hat{\boldsymbol{r}} = [\hat{r}_1, \cdots, \hat{r}_N]^T$, is the vector of estimated channel gains. The update equation in (2.5) is thus

$$\theta_m[k+1] = \theta_m[k] - \mu \left. \frac{\partial J(\boldsymbol{\theta}, \mathbf{0}, \hat{\boldsymbol{r}})}{\partial \theta_m} \right|_{\theta_m = \theta_m[k]} \tag{3.2}$$

where $\theta_m$ is the $m$-th element of $\theta$ corresponding to the $m$-th transmitter. Simulations showed that even if $\boldsymbol{\phi} \neq 0$ and $\hat{r}_i \neq r_i$, the (2.5) would converge. When idealized setting is not considered($\boldsymbol{\phi} \neq 0$, $\hat{r}_i \neq r_i$), the cost function is

$$J(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{r}) \tag{3.3}$$

The true gradient is as follows:

$$\frac{\partial J(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{r})}{\partial \theta_m} \tag{3.4}$$

where, $\boldsymbol{r} = [r_1, \cdots, r_N]^T$, is the vector of actual channel gains. The update equation using this true gradient is of the form:

$$\theta_m[k+1] = \theta_m[k] - \mu \left. \frac{\partial J(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{r})}{\partial \theta_m} \right|_{\theta_m = \theta_m[k]} \tag{3.5}$$

We will consider the average behavior of (3.2) under a variety of channel estimation errors. Section 3.3 considers just phase estimation errors. Section 3.4 considers

multiplicative errors involving both channel phase and gain. The basic approach is explained in Section 3.2.

### 3.2 The approach

Consider the minimization of a scalar function $f(x)$ of $x \in \mathbb{R}^N$. Gradient descent minimization proceeds as:

$$x_{k+1} = x_k - \mu \nabla f(x_k).$$

The descent direction $-\mu \nabla f(x_k)$ is known as the steepest descent direction. Consider now the alternative algorithm

$$z_{k+1} = z_k - \mu d_k.$$

Then it is well known that the update direction $d_k$ represents a descent direction if the inner product of $\nabla f(x_k)$ and $d_k$ is positive [45]. Translated to our setting the algorithm in (3.2) will converge if

$$\left[\frac{\partial J(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{r})}{\partial \theta_m}\right]^T \left[\frac{\partial J(\boldsymbol{\theta}, \mathbf{0}, \hat{\boldsymbol{r}})}{\partial \theta_m}\right] > 0 \tag{3.6}$$

i.e., the inner product between the true gradient and the implemented gradient is greater than 0. Instead of (3.6) in the case where $\boldsymbol{\phi}$ and $\boldsymbol{r}$ are random, it suffices to establish conditions under which

$$E\left[\frac{\partial J(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{r})}{\partial}\right]^T \left[\frac{\partial J(\boldsymbol{\theta}, \mathbf{0}, \hat{\boldsymbol{r}})}{\partial}\right] \geq 0. \tag{3.7}$$

We proceed to show that (3.7) holds under a variety of assumptions on $\boldsymbol{r}$ and $\boldsymbol{\phi}$.

### 3.3 Robustness to channel phase estimation errors

Recall in the simulation depicted in Fig. 2.1 the underlying assumption is

$$\hat{\boldsymbol{r}} = \boldsymbol{r} \text{ and } \phi_i \text{ i.i.d. } \sim \mathcal{U}[0, \alpha],\ \alpha < \pi/2. \tag{3.8}$$

We will now prove (3.7) under (3.8). First, lets denote the implemented gradient by $\boldsymbol{G}(\mathbf{0})$. From (3.1) under (3.8), we have

$$\begin{aligned} G_m(\mathbf{0}) &= \frac{\partial J(\boldsymbol{\theta}, \mathbf{0}, \boldsymbol{r})}{\partial \theta_m} \\ &= \frac{\partial \left| \sum_{i=1}^{N} r_i e^{j\theta_i} \right|^2}{\partial \theta_m} \\ &= r_m \sum_{i=1}^{N} r_i \sin(\theta_i - \theta_m) \end{aligned} \tag{3.9}$$

We can re-write (3.3) as follows:

$$J(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{r}) = |\sum_{i=1}^{N} r_i e^{j(\theta_i + \phi_i)}|^2 \tag{3.10}$$

$$= (\sum_{i=1}^{N} r_i \cos(\theta_i + \phi_i))^2 + (\sum_{i=1}^{N} r_i \sin(\theta_i + \phi_i))^2 \tag{3.11}$$

Let us denote the true gradient by $G(\boldsymbol{\phi})$. $G_m(\boldsymbol{\phi})$ can be written as

$$\frac{\partial J(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{r})}{\partial \theta_m} = -2r_m \sin(\theta_m + \phi_m)(\sum_{i=1}^{N} r_i \cos(\theta_i + \phi_i)) + 2r_m \cos(\theta_m + \phi_m)(\sum_{i=1}^{N} r_i \sin(\theta_i + \phi_i))$$

and can be further simplified as

$$G_m(\boldsymbol{\phi}) = 2r_m \sum_{i=1}^{N} r_i \sin(\theta_i + \phi_i - \theta_m - \phi_m) \tag{3.12}$$

$$\begin{aligned} E\{G_m(\boldsymbol{\phi})\} &= 2\sum_{i=1}^{N} E\{\sin((\theta_i - \theta_m) + (\phi_i - \phi_m))\} \\ &= 2\sum_{i=1}^{N} [\sin(\theta_i - \theta_m)E\{\cos(\phi_i - \phi_m)\} + \cos(\theta_i - \theta_m)E\{\sin(\phi_i - \phi_m)\}] \end{aligned} \tag{3.13}$$

Given that the $\phi_i$'s are independent, we have

$$E\{\cos(\phi_i - \phi_m)\} = E\{\cos(\phi_i)\}E\{\cos(\phi_m)\} + E\{\sin(\phi_i)\}E\{\sin(\phi_m)\} \tag{3.14}$$

$$E\{\sin(\phi_i - \phi_m)\} = E\{\sin(\phi_i)\}E\{\cos(\phi_m)\} + E\{\cos(\phi_i)\}E\{\sin(\phi_m)\} \tag{3.15}$$

Observe

$$E\{\sin(\phi_i)\} = \int_0^{\alpha} \sin(\phi_i)\frac{1}{\alpha}d\phi_i = \frac{1 - \cos(\alpha)}{\alpha} \tag{3.16}$$

and

$$E\{\cos(\phi_i)\} = \int_0^{\alpha} \cos(\phi_i)\frac{1}{\alpha}d\phi_i = \frac{\sin(\alpha)}{\alpha} \tag{3.17}$$

Putting these values in (3.14) and (3.15) we have

$$\begin{aligned} E\{\cos(\phi_i - \phi_m)\} &= \frac{1}{\alpha^2}[\sin(\alpha)^2 + (1 - \cos(\alpha))^2] \\ &= 2\frac{1 - \cos(\alpha)}{\alpha^2} \end{aligned} \tag{3.18}$$

and

$$\begin{aligned} E\{\sin(\phi_i - \phi_m)\} &= \frac{1}{\alpha^2}[\sin(\alpha)(1 - \cos(\alpha)) - \sin(\alpha)(1 - \cos(\alpha))] \\ &= 0 \end{aligned} \tag{3.19}$$

We can now write $E\{G_m(\boldsymbol{\phi})\}$ as

$$\begin{aligned} E\{G_m(\boldsymbol{\phi})\} &= 2\sum_{i=1}^{N}[\sin(\theta_i-\theta_m)E\{\cos(\phi_i-\phi_m)\}+\cos(\theta_i-\theta_m)E\{\sin(\phi_i-\phi_m)\}] \\ &= 2\sum_{i=1}^{N}[\frac{2(1-\cos(\alpha))}{\alpha^2}\sin(\theta_i-\theta_m)+0] \\ &= \frac{4}{\alpha^2}(1-\cos(\alpha))\sum_{i=1}^{N}\sin(\theta_i-\theta_m) \end{aligned} \tag{3.20}$$

We know $G_m(\mathbf{0})$ from (3.9). For $r_i = 1$, this will become

$$G_m(\mathbf{0}) = \sum_{i=1}^{N}\sin(\theta_i-\theta_m) \tag{3.21}$$

Now we can go ahead and evaluate $E\{G_m(\boldsymbol{\phi})G_m(\mathbf{0})\}$ as under

$$\begin{aligned} E\{G_m(\boldsymbol{\phi})G_m(\mathbf{0})\} &= G_m(0)E\{G_m(\phi)\} \\ &= \frac{4}{\alpha^2}(1-\cos(\alpha))(\sum_{i=1}^{N}\sin(\theta_i-\theta_m))^2 \end{aligned} \tag{3.22}$$

Thus indeed (3.7) holds if $\alpha < \pi/2$.

## 3.4 Robustness to multiplicative channel estimation errors in both phase and gain

We will now consider the case where apart from errors in channel phase, there can be errors in gain too. True channel gain is denoted by $r_i$ where,

$$r_i = \beta_i \hat{r}_i \tag{3.23}$$

$$\beta_i \text{ is i.i.d.} \sim \mathcal{U}[1-\epsilon, 1+\epsilon],\ \phi_i \text{ is i.i.d.} \sim \mathcal{U}[0,\alpha], \alpha < \pi/2 \tag{3.24}$$

The channel gain used by $i$-th transmitter for its update equation is $\hat{r}_i$. Hence there is an error since the actual channel gain is $\beta_i$ times that estimated and used by the

transmitter. The new cost function under these considerations will be based on the power received at the null-target and can be written as

$$J(\theta, \boldsymbol{\phi}, \boldsymbol{r}) = |\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{r_i e^{j(\theta_i + \phi_i)}}|^{\mathbf{2}} \tag{3.25}$$

Denote the $m$-th element of true gradient $G(\boldsymbol{\phi}, \beta)$ as under:

$$G_m(\boldsymbol{\phi}, \beta) = \frac{\partial J(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{r})}{\partial \theta_m} = 2\beta_m \hat{r}_m \sum_{i=1}^{N} \beta_i \hat{r}_i \sin(\theta_i + \phi_i - \theta_m - \phi_m) \tag{3.26}$$

Once again, in order to prove (3.7) we need to evaluate $E\{G_m(\boldsymbol{\phi}, \beta) G_m(\mathbf{0})\}$. Since, the implemented gradient $G_m(\mathbf{0})$, is evaluated for ideal conditions, it does not depend on $\boldsymbol{\phi}$ and $\beta_i$. That is, for $G_m(\mathbf{0})$, $\boldsymbol{\phi} = 0$ and $\beta_i = 1$. So, we can evaluate $E\{G_m(\boldsymbol{\phi}, \beta)$ separately. Since $\beta_i$ and $\phi_i$ are independent, we have

$$\begin{aligned} E\{G_m(\boldsymbol{\phi}, \beta)\} &= 2\hat{r}_m \sum_{i=1}^{N} E\{\beta_m \hat{r}_i \beta_i \sin((\theta_i - \theta_m) + (\phi_i - \phi_m))\} \\ &= 2\hat{r}_m \sum_{i=1}^{N} \hat{r}_i E\{\beta_m\} E\{\beta_i\} [\sin(\theta_i - \theta_m) E\{\cos(\phi_i - \phi_m)\} \\ &+ \cos(\theta_i - \theta_m) E\{\sin(\phi_i - \phi_m)\}] \end{aligned} \tag{3.27}$$

Now,

$$\begin{aligned} E\{\beta_i\} &= \int_{1-\epsilon}^{1+\epsilon} \beta_i \frac{1}{2\epsilon} d\beta_i \\ &= \frac{(1+\epsilon)^2 - (1-\epsilon)^2}{4\epsilon} \\ &= 1 \end{aligned} \tag{3.28}$$

Also,

$$E\{\beta_m\} = 1 \tag{3.29}$$

Putting values from (3.18), (3.19), (3.28) and (3.29) into (3.27), we have

$$\begin{aligned} E\{G_m(\boldsymbol{\phi},\beta)\} &= 2\hat{r}_m \sum_{i=1}^{N} \hat{r}_i.1.[2\frac{1-\cos(\alpha)}{\alpha^2}\sin(\theta_i-\theta_m)+0] \\ &= 4\hat{r}_m \frac{1-\cos(\alpha)}{\alpha^2} \sum_{i=1}^{N} \hat{r}_i \sin(\theta_i-\theta_m) \end{aligned} \tag{3.30}$$

From (3.20) and (3.30) we can observe that $E\{G_m(\boldsymbol{\phi},\beta)\} = E\{G_m(\boldsymbol{\phi})\}$.

Also, from (3.9) we have

$$G_m(\mathbf{0}) = r_m \sum_{i=1}^{N} r_i \sin(\theta_i-\theta_m) = \hat{r}_m \sum_{i=1}^{N} \hat{r}_i \sin(\theta_i-\theta_m) \tag{3.31}$$

since $\beta_i = 1$ for $G_m(\mathbf{0})$. So,

$$E\{G_m(\boldsymbol{\phi},\beta)G_m(\mathbf{0})\} = \frac{4\hat{r}_m^2}{\alpha^2}(1-\cos(\alpha))(\sum_{i=1}^{N} \hat{r}_i \sin(\theta_i-\theta_m))^2 \tag{3.32}$$

Since $\alpha < \pi/2$, so, $(1-\cos(\alpha)) > 0$. Thus (3.7) holds as long as $\alpha < \pi/2$.

We showed that the single receiver gradient descent nullforming algorithm (2.3) is robust to channel estimation errors in both phase and gain. In all we proved why (2.3) performed as expected in simulations under non-idealized conditions even though its properties had been analyzed theoretically under only ideal conditions in Chapter 2. In the next chapter we propose an algorithm for nullforming at single receiver that also tries to address the prior frequency synchronization requirement of nullforming instead of simply assuming it. The presence of time dependent frequency term renders the system non-autonomous making it a challenging problem.

# CHAPTER 4
# DISTRIBUTED NULLFORMING WITHOUT PRIOR FREQUENCY SYNCHRONIZATION

In this chapter we propose an algorithm for distributed nullforming at a single receiver without assuming prior frequency synchronization among the transmitters. In Chapter 2, our underlying assumptions of prior frequency synchronization and slowly varying oscillator offsets, simplified the overall problem wherein the transmitters only adjusted the phase of their transmitted signals as part of algorithm implementation. In this chapter we try to incorporate a possible way to address the frequency synchronization issue into the algorithm itself. Specifically, we do so by allowing each transmitter to iteratively make an adjustment to both the phase and frequency of its transmitted RF signal, thereby effectively implementing an algorithm to reduce the amplitude of the overall received signal to zero. As discussed in Section 1.5, distributed nullforming is an extremely challenging problem as each transmitter usually obtains its RF signal from a separate local oscillator (LO), and signals obtained from different LOs invariably have Brownian motion driven phase and frequency drifts, [46] due to manufacturing tolerances and temperature variations which the nullforming algorithm must estimate, track and compensate for the effect of these drifts.

Section 4.1 describes the general problem formulation. The algorithm presented in Section 4.2 [25], can be implemented in a purely distributed fashion at each transmitter as each transmitter needs only an estimate of its own channel gain to the receiver, and a feedback signal from the receiver, that is common across all the trans-

mitters. We prove analytically that the algorithm practically, globally converges to a null at the designated receiver. Following some preliminaries in Section 4.3, Section 4.4 presents this analysis of the stability and convergence properties of the algorithm. By practical convergence we mean that (i) the algorithm always converges to a stationary trajectory, (ii) that though some of these trajectories may not correspond to a null, those that do not are locally unstable, while those that do are locally stable.

## 4.1 Problem formulation

We assume that at the beginning of a nullforming epoch, each transmitter has access to its own complex channel gain to the receiver, using which it equalizes its channel to the receiver. Time slotted structure is assumed and the local channel information can be gathered using a method like explicit channel feedback as discussed in Section 1.5. Unlike Chapter 2 we do not assume that these transmitters have been prior synchronized in frequency.

Denote $\theta_i(t)$ to be the equalized phase of the $i$-th transmitter. Further, assume that $\omega_i(t)$, is a frequency offset of the $i$-th transmitter, from a nominal frequency to which each transmitter should ideally have been synchronized, but oscillator drift prevents the maintenance of such synchronization. Like Chapter 2, $N$ denotes the total number of transmitters.

Then the complex baseband signal received at the cooperatng receiver is:

$$s(t) = R(t) + jI(t) \tag{4.1}$$

where

$$R(t) = \sum_{i=1}^{N} \cos(\omega_i(t)t + \theta_i(t)) \tag{4.2}$$

and

$$I(t) = \sum_{i=1}^{N} \sin(\omega_i(t)t + \theta_i(t)). \tag{4.3}$$

The equivalent total baseband signal the receiver sees is

$$\sum_{i=1}^{N} e^{j((\omega_i(t)t+\theta_i(t))}$$

which is equivalent to knowing $s(t)$ in (4.1). Define $\theta(t) = [\theta_1(t), \cdots, \theta_N(t)]^\top$ and $\omega(t) = [\omega_1(t), \cdots, \omega_N(t)]^\top$. The equivalent total received power is:

$$J(\theta, \omega, t) = I^2(t) + R^2(t). \tag{4.4}$$

## 4.2 Algorithm description

The receiver braodcasts the received complex baseband signal $s(t)$ to all transmitters. This aggregate feedback is used by the $i$-th node to adjust its phase and frequency to drive $J(\theta, \omega, t)$ to zero. In practice the adjustment and feedback will be in discrete time. However, we propose and analyze here a continuous time algorithm. Should this algorithm be uniformly asymptotically stable (u.a.s), then standard averaging theory, [47] tells us that for sufficiently high feedback rates, and sufficiently small adaptation gains, an obvious discretized version will also be well behaved.

Specifically, with,

$$z(t) = [\theta^\top(t), \omega^\top(t)]^\top, \tag{4.5}$$

we propose a control law:

$$\dot{z} = f(z, t). \tag{4.6}$$

The obvious discretized version of this law is:

$$z(t+\Delta) = z(t) + \Delta f(z,t). \tag{4.7}$$

We will analyze (4.6), in the belief, to be verified by simulations, that should (4.6) be (u.a.s) then for sufficiently small $\Delta$, so would be (4.7).

We now consider a point of conceptual departure from Chapter 2, where $\omega = 0$, the cost function $J$ is autonomous, and a gradient descent minimization suffices. By contrast in the present setting, frequency offsets render the cost function non-autonomous, in that it may change its value even if the phases and frequencies are not adjusted. Accordingly, We consider a Lyapunov based design of the null forming algorithm. This exploits the following self-evident relationships.

$$\frac{\partial J}{\partial \theta_i} = -2R(t)\sin(\omega_i(t)t + \theta_i(t)) + 2I(t)\cos(\omega_i(t)t + \theta_i(t)), \tag{4.8}$$

$$\begin{aligned} \frac{\partial J}{\partial t} &= -2R(t)\sum_{i=1}\omega_i(t)\sin(\omega_i(t)t + \theta_i(t)) \\ &+ 2I(t)\sum_{i=1}\omega_i(t)\cos(\omega_i(t)t + \theta_i(t)), \\ &= \omega^{\top}(t)\frac{\partial J}{\partial \theta} \end{aligned} \tag{4.9}$$

and

$$\frac{\partial J}{\partial \omega} = t\frac{\partial J}{\partial \theta}. \tag{4.10}$$

Observe also, that the $i$-th node can implement (4.8) as long as it has access to its frequency, phase and the common feedback signals $I(t)$ and $R(t)$. Thus, (4.6) can be

implemented in a totally distributed fashion should one choose:

$$\dot{\theta} = -\frac{\partial J}{\partial \theta} - \frac{\omega}{2} \tag{4.11}$$

$$\dot{\omega} = -\frac{1}{2}\frac{\partial J}{\partial \theta}. \tag{4.12}$$

### 4.3 Preliminaries of stability analysis

In this section we present certain preliminary results that among other things show the uniform convergence of the gradient of $J$ with respect to $\theta$, and explore the properties of the stationary points of (4.11,4.12).

But first, a result used in [48].

**Lemma 4.1.** *Suppose on a closed interval $\mathcal{I} \subset \mathbb{R}$ of length $T$, a signal $w : \mathcal{I} \to \mathbb{R}$ is twice differentiable and for some $\epsilon$ and $M'$*

$$|w(t)| \leq \epsilon_1 \text{ and } |\ddot{w}(t)| \leq M' \quad \forall \quad t \in \mathcal{I}.$$

*Then for some $M$ independent of $\epsilon_1$, $\mathcal{I}$ and $M'$, and $M'' = \max(M', 2\epsilon_1 T^{-2})$ one has:*

$$|\dot{w}(t)| \leq M(M''\epsilon_1)^{1/2} \quad \forall \quad t \in \mathcal{I}.$$

We begin by showing that under (4.11,4.12) $J$ is nonincreasing and its gradient with respect to $\theta$ converges uniformly to zero.

**Lemma 4.2.** *Consider (4.11,4.12) with (4.2-4.4) initial time $t_0 \geq 0$. Then the following hold:*

*(a) For all $t \geq t_0$,*

$$V(t) = J(t) + \frac{\|\omega(t)\|^2}{2} \tag{4.13}$$

*is nonincreasing.*

*(b) The following occurs uniformly in $t_0$.*

$$\lim_{t\to\infty} \frac{\partial J}{\partial \theta}(t) = 0. \tag{4.14}$$

*Proof.* Because of (4.4), (4.8)-4.10) and (4.11,4.12), there holds:

$$\begin{aligned} \dot{J} + \frac{d}{dt}\left\{\frac{\omega^\top \omega}{2}\right\} &= \frac{\partial J}{\partial t} + \dot{\theta}^\top \frac{\partial J}{\partial \theta} + \dot{\omega}^\top \frac{\partial J}{\partial \omega} + \omega^\top \dot{\omega} \\ &= \omega^\top \frac{\partial J}{\partial \theta} - \left\|\frac{\partial J}{\partial \theta}\right\|^2 - \frac{\omega^\top}{2}\frac{\partial J}{\partial \theta} \\ &- \frac{\omega^\top}{2}\frac{\partial J}{\partial \theta} - \frac{t}{2}\left\|\frac{\partial J}{\partial \theta}\right\|^2 \\ &= -\left(1+\frac{t}{2}\right)\left\|\frac{\partial J}{\partial \theta}\right\|^2 \end{aligned} \tag{4.15}$$

Consequently (a) holds. Further, $\omega$ is uniformly bounded. Consequently from (4.8) there is an $M_1$, independent of $t_0$, such that for all $t \geq t_0$

$$\left\|\frac{d}{dt}\left\{\frac{\partial J}{\partial \theta}(t)\right\}\right\| \leq M_1.$$

Equally, there exists an $M_2$, also independent of $t_0$, such that for all $t \geq t_0$,

$$\left\|\frac{\partial J}{\partial \theta}(t)\right\| \leq M_2.$$

Further, since the initial time $t_0 \geq 0$, from (4.15) and $V(t)$ is nonnegative, one obtains that for all $t \geq t_0$:

$$\begin{aligned} \int_{t_0}^{t} \left\|\frac{\partial J}{\partial \theta}(s)\right\|^2 \, \mathrm{d}s &\leq \int_{t_0}^{t} \left(1+\frac{s}{2}\right)\left\|\frac{\partial J}{\partial \theta}(s)\right\|^2 \, \mathrm{d}s \\ &\leq V(t_0). \end{aligned}$$

Thus, for every $\epsilon > 0$, there exists a $T$ independent of $t_0$ such that for all $t \geq T + t_0$,

$$\int_{T+t_0}^{t} \left\|\frac{\partial J}{\partial \theta}(s)\right\|^2 \, \mathrm{d}s \leq \epsilon.$$

Then from Lemma 4.1, there is a $K$ independent of $t_0$ such that for all $\epsilon > 0$, there exists a $T$ independent of $t_0$ such that for all $t \geq T + t_0$,

$$\left\| \frac{\partial J}{\partial \theta}(s) \right\|^2 \leq \epsilon, \quad \forall s \geq T + t_0.$$

Thus indeed (b) holds uniformly in $t_0$. □

Evidently, the algorithm converges uniformly to a trajectory where:

$$\frac{\partial J}{\partial \theta} = 0 \tag{4.16}$$

and for some constant $\omega^*$ the frequency offsets

$$\omega = \omega^*. \tag{4.17}$$

Some of these trajectories correspond to the desired null. Others do not, and will be dubbed spurious. Our goal is to demonstrate that the latter are locally unstable. Thus they are rarely attained, and even if attained not practically maintained as the slightest noise would drive the trajectories away from them. Thus, by showing the local stability of the null manifold, we would have demonstrated the practical uniform convergence of the algorithm to a null.

Observe, (4.16) and (4.17) hold under the following circumstances.

- [A] $R(t) = I(t) = 0$.
- [B] If $R(t) \neq 0$, then for all $i$,

$$\tan(\omega_i^* t + \theta_i) = \frac{I(t)}{R(t)}.$$

- [C] If $I(t) \neq 0$, then for all $i$,

$$\cot(\omega_i^* t + \theta_i) = \frac{R(t)}{I(t)}.$$

Clearly [A] corresponds to stationary points reflecting nulls. The set of points it represents will henceforth be called the *null manifold*. Both [B] and [C] reflect the condition that for all $i, l$, there holds:

$$\tan(\omega_i^* t + \theta_i) = \tan(\omega_l^* t + \theta_l), \quad \forall t. \tag{4.18}$$

Some of these may still correspond to nulls. The rest are spurious.

It is also evident that under all circumstances and a scalar $\bar{\omega}$,

$$\omega = \bar{\omega}[1, \cdots, 1]^\top. \tag{4.19}$$

The local analysis of these stationary trajectories, will require the examination of the Hessian with respect to the $\theta$. In particular from (4.8) one has:

$$\begin{aligned} \frac{\partial J(\theta)}{\partial \theta_i} &= -2R(t)\sin(\omega_i(t)t + \theta_i(t)) \\ &+ 2I(t)\cos(\omega_i(t)t + \theta_i(t)) \\ &= 2\sum_{l=1} \sin((\omega_l(t) - \omega_i(t))t + \theta_l(t) - \theta_i(t)) \end{aligned}$$

Thus, in view of (4.19), along (4.16) and (4.17) the $il$-th element of the Hessian along the stationary trajectory is given by:

$$[H(\theta)]_{il} = \frac{\partial^2 J(\theta)}{\partial \theta_i \partial \theta_l} = \begin{cases} -2\sum_{m \neq i}^{N} \cos(\theta_i - \theta_m) & i = l \\ 2\cos(\theta_i - \theta_l) & i \neq l \end{cases} \tag{4.20}$$

## 4.4 Stability analysis

Armed with the preliminary results in Section 4.3 we now complete our stability analysis. Lemma 4.2 shows that uniform convergence to a stationary trajectory is guaranteed. Some of these trajectories correspond to a null. Other do not. In this section we show that *only those that correspond to a null are locally stable. The others are not.* Consequently, one is assured of practical uniform convergence in the sense that stationary trajectories that do not correspond to the desired nulls if at all attained, cannot be practically maintained. Thus for all practical purposes the algorithm defined in (4.11,4.12) achieves a desired null.

First we demonstrate the local instability of spurious stationary trajectories.

To this end we present two Lemmas.

**Lemma 4.3.** *The linear system below with scalar $a > 0$ is unstable:*

$$\dot{\eta} = \begin{bmatrix} a & ta + \frac{1}{2} \\ \frac{a}{2} & \frac{at}{2} \end{bmatrix} \eta \tag{4.21}$$

*Proof.* Consider the initial condition $\eta(0) = [1, 0]^\top$. Then it is evident that both elements of the state are nonegative for all $t > 0$. Then the first element of the state vector is

$$\eta_1(t) \geq e^{at}.$$

Thus the system is unstable. □

We next show that $H(\theta)$ at a false spurious stationary point has a negative eigenvalue.

**Lemma 4.4.** *Consider a stationary point $H(\theta)$ in (4.20) when (4.16), (4.17) such that along this trajectory $J \neq 0$. Then $H(\theta)$ has at least one negative eigenvalue.*

*Proof.* Because of (4.18) and (4.19) there holds, for all $i \neq l$, $\cos(\theta_i - \theta_l) = \pm 1$. Thus, as $H(\theta)$ is symmetric, it suffices to show that at least one diagonal element is not equal is either zero or negative.

Without loss of generality, assume $\theta \in [0, 2\pi)^N$. As only the differences of $\theta_i$ appear in the expression for $H(\theta)$, in view of (4.18), again without loss of generailty, one can partition $\{1, \cdots, N\}$ into two sets $\mathcal{I}_0$ and $\mathcal{I}_\pi$ such that

$$\theta_i = \begin{cases} 0 & i \in \mathcal{I}_0 \\ \pi & i \in \mathcal{I}_\pi \end{cases}$$

Also observe that

$$\cos(\theta_i - \theta_l) = \begin{cases} 1 & \{i, l\} \subset \mathcal{I}_0 \text{ or } \{i, l\} \subset \mathcal{I}_\pi \\ -1 & \text{else} \end{cases}$$

We need to show that for at least one $i \in \{1, \cdots, N\}$ there holds:

$$\sum_{\substack{l=1 \\ l \neq i}}^{N} \cos(\theta_i - \theta_l) \geq 0. \tag{4.22}$$

We consider the following three cases:

**Case I:** $|\mathcal{I}_0| - 1 = |\mathcal{I}_\pi|$**.** Consider $i \in \mathcal{I}_0$. Then the number of summands in (4.22) that are 1, equals the number that are -1, and the sum equals zero.

**Case II:** $|\mathcal{I}_0| + 1 = |\mathcal{I}_\pi|$**.** Consider $i \in \mathcal{I}_\pi$. Then the number of summands in (4.22) that are 1, equals the number that are -1, and the sum equals zero.

**Case III: Neither Case I nor Case II holds.** Then there is at least one $i \in \{1, \cdots, n\}$ for which the number of summands in (4.22) that are 1, is greater than the number that are -1, and the sum is positive.

□

We now prove that a spurious stationary point is unstable.

**Theorem 4.5.** *Consider (4.11,4.12), and a stationary trajectory defined by (4.16), (4.17) such that along these trajectory $J \neq 0$. Then this trajectory is unstable.*

*Proof.* Following the discussion in the previous section, and noting that for all $i, l$

$$\frac{\partial^2 J(\theta)}{\partial \theta_i \partial \omega_l} = t \frac{\partial^2 J(\theta)}{\partial \theta_i \partial \theta_l}$$

(4.16), (4.17) linearized around such a trajectory is given by:

$$\dot{x} = \begin{bmatrix} -H(\theta) & -tH(\theta) + \frac{I}{2} \\ -\frac{H(\theta)}{2} & -\frac{t}{2} H(\theta) \end{bmatrix} x. \tag{4.23}$$

In view of Lemma 4.4 and the symmetry of $H(\theta)$, there an orthogonal matrix $\Omega$ and real $\lambda_i$, with $\lambda_1 > 0$, such that with

$$\Lambda = \text{ diag } \{-\lambda_1, \cdots, \lambda_N\},$$

$$H(\theta) = \Omega \Lambda \Omega^T$$

Define $\beta = \text{ diag } \{\Omega, \Omega\} x$. Then the linearized systems is equivalent to:

$$\dot{\beta} = \begin{bmatrix} -\Lambda & -t\Lambda + \frac{I}{2} \\ -\frac{\Lambda}{2} & -\frac{t}{2}\Lambda \end{bmatrix} \beta.$$

Then instability follows from Lemma 4.3. □

Thus indeed spurious stationary points are unstable. To complete the result, it suffices to show that the null manifold is locally stable. It is tempting to examine the linearized algorithm around a null trajectory. To this end we must examine the Hessian $H(\theta)$ around this trajectory. However, the test on the Hessian is inconclusive. To be precise, at a stationary point corresponding to a null, i.e. when $R = I = 0$, there holds:

$$\begin{aligned}[H(\theta)]_{il} &= \begin{cases} 2 & i = l \\ 2\cos(\theta_i - \theta_l) & i \neq l \end{cases} \\ &= 2c(\theta)c^\top(\theta) + 2s(\theta)s^\top(\theta)\end{aligned}$$

where

$$c(\theta) = [\cos(\theta_1), \cdots, \cos(\theta_N)]^\top$$

and

$$s(\theta) = [\sin(\theta_1), \cdots, \sin(\theta_N)]^\top.$$

This has rank at most two. Thus, for $N > 2$ at least one eigenvalue of the Hessian is zero and the linearized analysis is inconclusive. One could adopt a center manifold based approach to showing stability, as was done in [49]. Instead, we adopt a more direct approach.

For this we require a final Lemma.

**Lemma 4.6.** *Suppose under (4.16), (4.17), $J(\theta) < 1$. Then*

$$\frac{\partial J(\theta)}{\partial \theta} = 0 \Leftrightarrow J(\theta) = 0.$$

*Proof.* At a spurious stationary point, for all $i, l \in \{1, \cdots, N\}$, $\cos(\theta_i - \theta_l) = \pm 1$. Further, from (4.4), (4.2) and (4.3),

$$\begin{aligned} J(\theta, \omega, t) &= \left( \sum_{i=1}^{N} \cos(\omega_i(t)t + \theta_i(t)) \right)^2 \\ &+ \left( \sum_{i=1}^{N} \sin(\omega_i(t)t + \theta_i(t)) \right)^2 \\ &= N + 2 \sum_{i=1}^{N} \sum_{\substack{l=1 \\ l \neq i}}^{N} \cos((\omega_l(t) - \omega_i(t))t \\ &+ \theta_l(t) - \theta_i(t)) \end{aligned}$$

Thus at a spurious stationary point $J(\theta)$ is a nonnegative integer. The result follows.

□

We can now prove local stability of the null manifold.

**Theorem 4.7.** *Under the conditions of Lemma 4.2 at $t_0 \geq 0$,*

$$V(t_0) < 1.$$

*Then*

$$\lim_{t \to \infty} J(\theta(t)) = 0, \tag{4.24}$$

*and the convergence is uniform in $t_0$.*

*Proof.* Item (b) of Lemma 4.2 holds uniformly in $t_0$. Further for all $t \geq t_0$

$$J(\theta(t)) \leq V(t) \leq V(t_0) < 1.$$

Thus the result holds from Lemma 4.6. □

The distributed nullforming algorithm we developed in this chapter overcame the prior frequency synchronization requirement of the algorithm developed in Chapter 2. We proved practical uniform convergence of the algorithm to a null. The robustness to noise and frequency drift is verified by simulations involving 10 transmitters and nontrivial Brownian motion driven oscillator drift in our paper [25]. In the next chapter, we let the transmitters apply a complex weight, and adapt both its phase and amplitude instead of just the phase as in Chapter 2. This makes the objective function convex in applied weights and helps simplify the problem. Also, in previous chapters and current chapter, we dealt with nullforming at single receiver, but in the next chapter we generalize to multiple receivers.

# CHAPTER 5
# JOINT BEAM AND NULLFORMING

In this chapter we consider the *distributed joint beam and nullforming* (JBNF) problem shown in Fig. 5.1 where $N$ single antenna transmitters seek to cooperatively broadcast a common message signal to a subset of $M_1$ single antenna receivers while simultaneously canceling at another set of $M-M_1$ receivers. In effect, the transmitters form a *virtual antenna array* and transmit with particular phases and amplitudes to shape the array's pattern such that beams and nulls are created at desired locations. By simultaneously transmitting beams and nulls, coherent combining gains can be achieved toward intended receivers while avoiding interference toward unintended receivers.

Previously we developed algorithms for nullforming to one receiver that do not require global channel knowledge. These algorithms were limited to a single null target, and considered phase-only adaptation at the transmitters. Here we consider a natural generalization of this approach to the multiple receiver case. By allowing both amplitude and phase adaptation and by targeting received complex amplitudes rather than power, we are able to consider a much simpler quadratic optimization framework while generalizing to multiple beam and null targets. We reported preliminary results in a conference paper [26] and further details in a journal paper [27]. The algorithm we develop in this chapter is also adaptive in nature in which each transmitter in the distributed array iteratively adjusts its transmitted signal in a purely distributed fashion just like other algorithms discussed so far. We assume cooperation from both

beam and null targets in terms of periodic aggregate feedback messages. Again only local channel information and the feedback messages are required by transmitters for implementing the algorithm.It is important to note that we assume prior frequency synchronization just like we did in Chapter 2 in order to avoid the complications of dealing with non-autonomous systems.

Given a set of desired power levels at the intended receivers, it is shown that in the absence of noise, the adaptive algorithm converges to a solution that achieves the target complex amplitudes and hence the corresponding power levels at the desired receivers while also achieving nulls with zero power toward the null targets. Moreover, with appropriate initialization, the algorithm converges to the minimum transmit power solution. The property of requiring only local channel knowledge with aggregate feedback results in reduced overhead and increased scalability compared to previous techniques which require global channel knowledge. It is shown that the convergence speed is nondecreasing in the number of transmitters $N$ if a step-size parameter is kept constant, and arbitrarily fast for large $N$ with suitable choice of the step-size, with probability one for Rayleigh fading channel gains. Our algorithm, when properly initialized, achieves the beams and nulls with *vanishing total transmit power* when $N$ becomes large, even in the presence of noise, again with probability one for Rayleigh fading channel gains. Taken together, these results add up to some remarkable scalability properties: the feedback overhead does not grow with the number of transmitters, and with high probability, the algorithm can be configured to converge arbitrarily fast and use vanishingly small total transmit power.

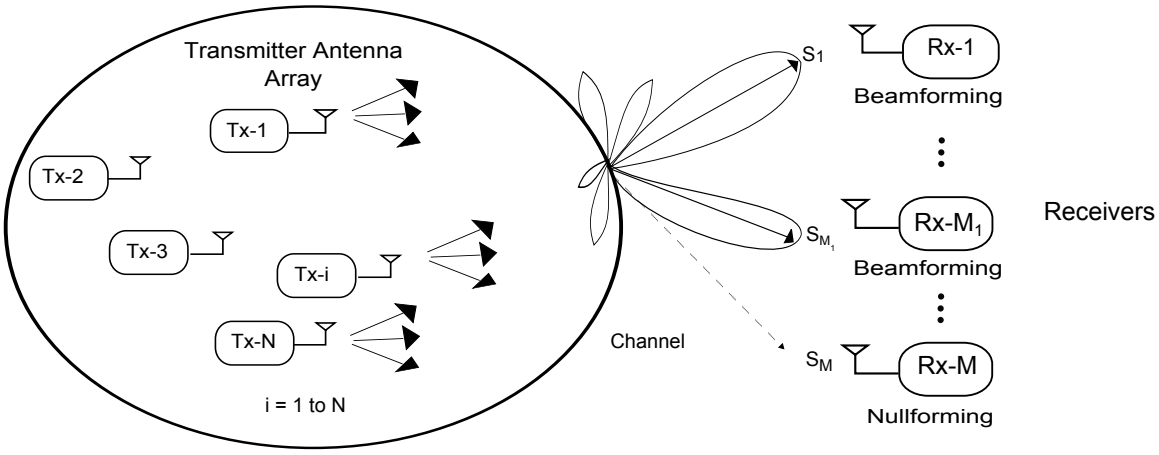



Figure 5.1: The problem of joint beam and nullforming using a distributed array.

**Contributions** Apart from developing a scalable, distributed algorithm for JBNF problem that has fast convergence and power efficiency properties, we also analyse these properties as detailed below:

**(a) Characterizing limit points:** Our JBNF problem is under constrained: there are no isolated critical points. Instead there is an entire affine subspace of critical points (identified in (5.9)), all of which are global minima. Such a situation is very common in adaptive control and is often linked to the lack of persistent excitation (pe) [50]-[51]. We show (Theorem 5.1) that, in the noise-free setting, our algorithm converges to the projection of the initial estimate on the affine subspace. With noise, convergence is to this same point in the mean with bounded variance (Theorem 5.2).

**(b) Power efficiency of the limit point:** We characterize conditions under which the limit point corresponds to the solution with minimum transmit power: specifically, the initial iterate of the gradient descent algorithm must lie in the range space

of a matrix comprising the complex channel gains.

**(c) The absence of drift due to noise:** When an adaptive algorithm converges to an affine subspace rather than an isolated point in the absence of noise, the possibility arises that noise may cause the adaptations to drift along this affine subspace [52, 53, 54]. We show that our algorithm avoids this problem, as noise has no effect along the minimizing affine subspace.

**(d) Scalability:** For a given set of beam and null targets, if we keep a step size parameter of the algorithm fixed, the convergence rate of our algorithm is nondecreasing in $N$.

**(e) Asymptotic scalability:** For independent Rayleigh fading channels: as $N$ tends to infinity, with probability one, the convergence speed can be made arbitrarily fast. Further, if the algorithm is initialized in the range space of the channel matrix described above, then again with probability one, even in the presence of noise, the iterates converge in the mean to zero with zero covariance as $N$ tends to infinity.

The rest of the chapter is organized as follows. Section 5.1 formulates the JBNF problem as a convex minimization and 5.2 describes our algorithm. The convergence of this algorithm is examined in Section 5.3, and its scaling to large number of transmitters in Section 5.4. Section 5.5 presents simulation results.

## 5.1 Problem formulation

This chapter considers a more general setting, compared to Chapters 2 and 4, with $M$ receivers, joint beam and nullforming, and adaptation of *both the phase*

*and amplitude* at each transmit node. We formulate a gradient descent algorithm to minimize a quadratic cost function that is convex in the complex amplitudes, with global minima that correspond to gain values for which the beam and nullforming objectives of the JBNF problem are satisfied. This algorithm can be implemented in a *distributed* fashion at each transmitter. Each transmitter adjusts its transmission knowing only its own $M$ complex channel gains to the receivers and common feedback messages broadcast by the receivers to all of the transmitters. The feedback messages contain estimates of the complex amplitude (i.e., the gain and phase) of the aggregate received radio frequency signal at each receiver.

Note that while the JBNF problem is much more general than the nullforming to a single receiver considered in Chapter 2, the latter considers phase-only adaptation to avoid the trivial solution where the null is achieved by driving all transmitted signals to zero. Phase only adaptation renders the minimization considered in Chapter 2 nonconvex, whereas the adaptation of both the phase and gain permits the minimization here to be convex. Thus in some respects, our proposed algorithm for the JBNF problem is actually *simpler* than nullforming to a single receiver.

We consider a distributed array of $N$ transmitters as in Chapters 2 and 4 but unlike one receiver in those chapters, here we consider $M$ receivers. Receivers 1 through $M_1$, where $M_1 \leq M$, are desired receivers or *beam targets* where we want to direct the transmission, and receivers $M_1 + 1$ to $M$ are *null targets* where the transmitted signals must cancel each other.

Time-slotting, slowly varying channel phase offsets and frequency

pre-synchronization are assumed as mentioned in Section 1.5. The time-division multiplexed schedule of transmissions pertaining to the time-slotted structure is shown in Fig.5.2. Any of the channel estimation methods mentioned earlier or any combinations of those can be used for our JBNF algorithm. Note that if channels are slowly varying, the channel estimation process may only need to be performed infrequently.

**Notation and problem statement**

In this chapter we denote the $N \times M$ channel matrix by $\boldsymbol{H}$ whose $ij$-th entry $h_{ij}$ is the complex channel from the $i$-th transmitter to the $j$-th receiver. With $\boldsymbol{h}_j$ its $j$-th column we can write $\boldsymbol{H} = [\boldsymbol{h}_1\ \boldsymbol{h}_2\ \ldots\ \boldsymbol{h}_M]$.

We denote by $\boldsymbol{x} = [x_1, x_2, ..., x_N]^T$ the $N \times 1$ transmit weight vector, where our convention is that $x_i^*$ is the complex weight applied by the $i$th transmitter, $1 \leq i \leq N$. Thus, the complex baseband message signal transmitted by the $i$th transmitter is $c_i(t) = x_i^* m(t)$ and the noiseless complex baseband signal received at the $j$th receiver is $r_j(t) = \left(\boldsymbol{x}^H \boldsymbol{h}_j\right) m(t)$.

Since the message signal waveform $m(t)$ is immaterial to the JBNF problem, we set $m(t) \equiv 1$ without loss of generality. This leads to the total complex baseband signal seen by receiver $j$ is $\boldsymbol{r}_j = \boldsymbol{x}^H \boldsymbol{h}_j$. For notational simplicity, we work with the complex conjugate of this received signal amplitude $s_j \doteq (\boldsymbol{r}_j)^* \equiv \boldsymbol{h}_j^H \boldsymbol{x}$. The complex number $s_j$ is the feedback broadcast by receiver $j$ to all the transmitters to drive each iteration of the JBNF algorithm.

Thus, we can collect the feedback broadcast by the receivers in time slot $k$

into the following vector:

$$\boldsymbol{s}[k] = \boldsymbol{H}^H \boldsymbol{x}[k] + \boldsymbol{w}[k], \quad \boldsymbol{w}[k] \sim CN\left(0, \sigma_w^2 I\right), \forall k \tag{5.1}$$

where $\boldsymbol{w}[k] = (w_1[k], ..., w_M[k])^T$ represents complex Gaussian noise assumed to be i.i.d. across receivers and time slots.

We would like to adapt $\boldsymbol{x}[k]$ in a *distributed fashion* so that $\boldsymbol{x}^H[k]\boldsymbol{h}_j$ are driven towards specified nonzero values for the beam targets $1 \leq j \leq M_1$, and towards zero for the null targets $M_1 + 1 \leq j \leq M$. Even though we are interested only in nonzero *power* at a beam target, it is convenient to over-constrain the problem, setting a target complex amplitude instead, in order to obtain a quadratic framework for optimizing $\boldsymbol{x}$. Specifically, suppose that the *desired complex amplitude* at receiver $j$ is $b_j$, where $b_j \neq 0, 1 \leq j \leq M_1$ and $b_j = 0$ for $M_1 + 1 \leq j \leq M$. We would like to set $\boldsymbol{x}$ such that, $s_j = \boldsymbol{h}_j^H \boldsymbol{x} = b_j$, $1 \leq j \leq M$, or in vector form $\boldsymbol{H}^H \boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{b} = (b_1, ..., b_M)^T$.

Thus, under (5.1), the JBNF problem can be recast as the asymptotic minimization of the following quadratic objective function over $\boldsymbol{x}$:

$$J_w(\boldsymbol{x}) = E_{\boldsymbol{w}}\left[\|\boldsymbol{s} - \boldsymbol{b}\|^2\right] = E_{\boldsymbol{w}}\left[\left\|\boldsymbol{H}^H \boldsymbol{x} + \boldsymbol{w} - \boldsymbol{b}\right\|^2\right] = \left\|\boldsymbol{H}^H \boldsymbol{x} - \boldsymbol{b}\right\|^2 + M\sigma_w^2. \tag{5.2}$$

The minimization of $J_w$ is therefore equivalent to that of

$$J(\boldsymbol{x}) = \|\boldsymbol{H}^H \boldsymbol{x} - \boldsymbol{b}\|^2. \tag{5.3}$$

We show how this minimization can be achieved in a scalable, distributed fashion in the next section.

## 5.2 Algorithm description

In the noise free case, the conjugate gradient of $J(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ can be written as:

$$\boldsymbol{\nabla J}(\boldsymbol{x}) = \left(\boldsymbol{H}(\boldsymbol{H}^H\boldsymbol{x} - \boldsymbol{b})\right)^H \tag{5.4}$$

Thus, in view of (5.1), in the noise-free case, the exact gradient descent minimization of $J(\boldsymbol{x})$ can be accomplished by

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] - \mu\boldsymbol{H}(\boldsymbol{s}[k] - \boldsymbol{b}), \tag{5.5}$$

where the step size $\mu$ is suitably small and positive. If we explicitly write down the adaptation (5.5) for transmitter $i$, we get:

$$x_i[k+1] = x_i[k] + \mu\sum_{j=1}^{M} h_{ij}\left(s_j[k] - b_j\right) \tag{5.6}$$

We can clearly see from (5.6), that the adaptation equation (5.5) for transmitter $i$ only requires knowledge of its own channel gains to the receivers $h_{ij},\ \forall j$, and the aggregate feedback $s_j[k]$. Specifically, transmitter $i$ does *not* require knowledge of the channel gains of other transmitters $h_{mj},\ m \neq i$ to the receivers.

Fig. 5.2 illustrates a decentralized implementation of (5.5) in a time-slotted system. At the beginning of each time-slot, the distributed transmit array uses the weight vector $\boldsymbol{x}[k]$ to transmit a message signal to the receivers. This is the part of the time-slot where the useful communication takes place. After this follows the feedback phase of the time-slot where, each receiver successively broadcasts a common information signal to the transmit array. Specifically, the $j$-th receiver sends a

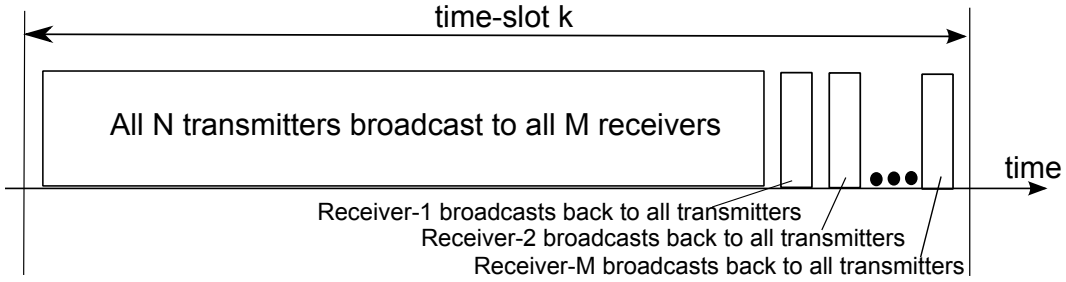



Figure 5.2: Description of transmission in time-slot $k$.

short packet that contains a single complex number representing the total complex baseband received signal, $s_j[k]$. At the end of each time-slot the transmitters update their weights using (5.5). At the beginning of the next time-slot $k+1$, weight vector $\boldsymbol{x}[k+1]$ is used for the transmission from the distributed array. The communication between transmitters and receivers proceeds in this fashion until the desired beam/nullforming is achieved at designated receivers.

### 5.3 Analytical characterization

In this section, we first discuss the geometric structure of the problem and then investigate the convergence properties of the iterative algorithm (5.5). Given that $J(\boldsymbol{x})$ is convex, and in the absence of noise (5.5) is an exact gradient descent, the convergence of $\boldsymbol{x}[k]$ follows. The focus of our analysis is on characterizing the effects of initial condition, noise, and the attainment of the so called *power efficient solution*. Subsequently, we also study issues of scalability.

Since we are interested in scaling to a large number of transmitters $N$, we focus on the regime $N > M$ (distributed array size larger than the number of receivers),

and indeed, on $N\gamma M$. With high probability, therefore, the $N \times 1$ channels $\{\boldsymbol{h}_j, 1 \leq j \leq M\}$ are linearly independent. For most of our analysis, we make the latter assumption, stated formally below.

**Assumption 5.3.1.** *The $N \times M$ channel matrix $\boldsymbol{H}$ has full column rank.*

This assumption on the tall matrix $\boldsymbol{H}$ implies that the $M \times M$ correlation matrix $\boldsymbol{H}^H\boldsymbol{H}$ is full rank and positive definite. We denote its ordered eigenvalues by $\lambda_1 \geq \lambda_2 ... \geq \lambda_M > 0$.

### 5.3.1 Geometric interpretation of optimum solution

Under Assumption 5.3.1, with $\mathcal{R}(\boldsymbol{H}^H)$ denoting the range space of $\boldsymbol{H}^H$, one has $\boldsymbol{b} \in \mathcal{R}(\boldsymbol{H}^H)$, which guarantees the existence of $\boldsymbol{x}$ such that (5.3) is zeroed out:

$$\boldsymbol{H}^H\boldsymbol{x} = \boldsymbol{b}. \tag{5.7}$$

Since this is an underdetermined system in the typical regimes of interest ($N > M$), there exists an entire *affine subspace* of vectors satisfying (5.7), from which we would like to choose the power-efficient solution, $\boldsymbol{x}_e = \arg\min_{\boldsymbol{x}\in\mathcal{H}} \|\boldsymbol{x}\|$ with $\mathcal{H} = \left\{ \boldsymbol{x} \in \mathbb{C}^N \middle| \boldsymbol{H}^H\boldsymbol{x} = \boldsymbol{b} \right\}$.

Under Assumption 5.3.1, the $M \times M$ matrix $\boldsymbol{H}^H\boldsymbol{H}$ is invertible, and the unique power-efficient solution is given by

$$\boldsymbol{x}_e = \boldsymbol{H}\left(\boldsymbol{H}^H\boldsymbol{H}\right)^{-1}\boldsymbol{b} = \boldsymbol{H}\boldsymbol{a}. \tag{5.8}$$

Observe that the power efficient solution $\boldsymbol{x}_e$ must lie in the *signal space* $\mathcal{S} = \mathcal{R}(\boldsymbol{H})$ spanned by the columns $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_M$ of $\boldsymbol{H}$. Fig.5.3 illustrates this geometric interpretation of $\boldsymbol{x}_e$

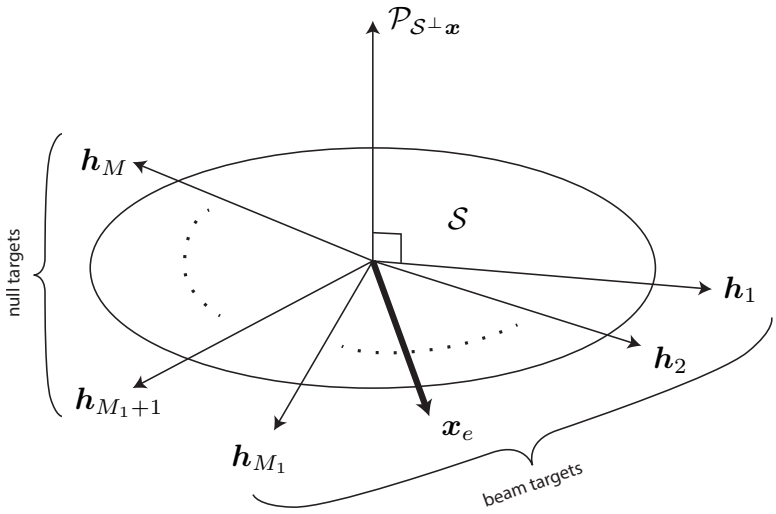



Figure 5.3: Geometric interpretations of the power efficient solution.

We can now completely characterize the affine subspace $\mathcal{H}$ of solutions as

$$\mathcal{H} = \{\boldsymbol{x} : \boldsymbol{x} - \boldsymbol{x}_e \in \mathcal{S}^\perp\} \tag{5.9}$$

where $\mathcal{S}^\perp = \mathcal{N}(\boldsymbol{H}^H)$ is the null space of $\boldsymbol{H}^H$ [55]. To see this, note that for any $\boldsymbol{x} \in \mathcal{H}$, $\boldsymbol{H}^H(\boldsymbol{x} - \boldsymbol{x}_e) = \boldsymbol{b} - \boldsymbol{b} = \boldsymbol{0}$. Thus, the affine subspace of solutions to (5.7) is the translation of the "undesired" subspace $\mathcal{S}^\perp$ by the power-efficient solution $\boldsymbol{x}_e$ (or indeed, by any solution $\boldsymbol{x}$ of (5.7)). We will characterize how the particular solution in this affine subspace that the iteration (5.5) converges to depends on the initial condition. Before that, we provide an alternative geometric characterization of the power-efficient solution $\boldsymbol{x}_e$, working within the signal space $\mathcal{S}$.

**Remark.** As long as Assumption 5.3.1 holds, a solution exists for *any* choice of target complex amplitudes $\boldsymbol{b}$. The minimum transmit power, corresponding to the

minimum norm solution $\boldsymbol{x}_e$, is

$$P_{TX} = \boldsymbol{x}_e^H \boldsymbol{x}_e = \boldsymbol{b}^H \left(\boldsymbol{H}^H \boldsymbol{H}\right)^{-1} \boldsymbol{b}$$

This depends on the target complex amplitudes $\boldsymbol{b}$, which can be written as $[\boldsymbol{b}_{beam}^T \mathbf{0}^T]^T$ where, $\boldsymbol{b}_{beam} = [b_1, ..., b_{M_1}]^T$ corresponds to the complex conjugates of the desired complex amplitudes at the beam targets and $\mathbf{0}$ is a $M - M_1$ column vector of zeroes corresponding to the desired zero power at null targets. Hence in principle, we could choose these complex amplitudes to further optimize the value of the minimum transmit power. Such a choice might not be admissible, because we may wish to constrain the magnitudes of the entries of $\boldsymbol{b}_{beam}$ to some values based on the desired SNR at each beam location. We generally focus on a fixed, and arbitrary, choice of $\boldsymbol{b}_{beam} = \mathbf{1}$ for the rest of the chapter.

### 5.3.2 Convergence in noiseless regime

Behavior of the updates (5.5) in the noiseless case is provided in Theorem 5.1 below. A few features are instructive. First observe that, while it is well known that convergence must occur to the affine subspace $\mathcal{H}$, existing analyses of such algorithms fail to characterize the the precise limit point $\boldsymbol{x}_\infty$ on $\mathcal{H}$. In contrast this theorem proves that this limit point is in fact *the projection of the intial iterate* $\boldsymbol{x}[0]$ *on* $\mathcal{H}$. Design implications of this fact is described after the theorem.

The second fact that impacts subsequent noise analysis is as follows. Consider the *error vector* defined as $\boldsymbol{\Delta}[k] \doteq \boldsymbol{x}[k] - \boldsymbol{x}_\infty$. It is shown in the proof that this vector

evolves according to:

$$\boldsymbol{\Delta}[k+1] = \boldsymbol{\Delta}[k] - \mu \boldsymbol{H}\boldsymbol{H}^H \boldsymbol{\Delta}[k] = \left(\boldsymbol{I} - \mu \boldsymbol{H}\boldsymbol{H}^H\right) \boldsymbol{\Delta}[k] \tag{5.10}$$

Observe as $\boldsymbol{H}$ is tall, the transition matrix $\left(\boldsymbol{I} - \mu \boldsymbol{H}\boldsymbol{H}^H\right)$ has eigenvalues at 1, representing modes that are orthonormal to $\mathcal{H}$, and do not decay. To facilitate the convergence analysis, the theorem in fact proves that that the $N \times 1$ vector $\boldsymbol{\Delta}[k]$ lies in the signal subspace, and can therefore be expressed in terms of a lower dimensional $M \times 1$ vector $\boldsymbol{\delta}[k]$ as

$$\boldsymbol{\Delta}[k] = \boldsymbol{H}\boldsymbol{\delta}[k] \tag{5.11}$$

This vector on the other hand evolves as

$$\boldsymbol{H}\boldsymbol{\delta}[k+1] = \left(\boldsymbol{I}_N - \mu \boldsymbol{H}\boldsymbol{H}^H\right) \boldsymbol{H}\boldsymbol{\delta}[k] = \boldsymbol{H}\left(\boldsymbol{I}_M - \mu \boldsymbol{H}^H\boldsymbol{H}\right) \boldsymbol{\delta}[k] \tag{5.12}$$

where we put subscripts on the identity matrices to specify their dimension. As $\boldsymbol{H}$ has full column rank, this becomes

$$\boldsymbol{\delta}[k+1] = \left(\boldsymbol{I}_M - \mu \boldsymbol{H}^H\boldsymbol{H}\right) \boldsymbol{\delta}[k] \tag{5.13}$$

and the *de facto* transition matrix $\boldsymbol{I}_M - \mu \boldsymbol{H}^H\boldsymbol{H}$ does not have eigenvalues at 1. Thus, (5.13) reflects movement along the signal subspace. We explain later why this reduced state space has important implications to the convergence analysis in the presence of noise.

**Theorem 5.1.** *Consider (5.5) under (5.1) and Assumption 5.3.1. In the absence of noise, and assuming that the adaptation gain $\mu$ satisfies*

$$|1 - \mu\lambda_i| < 1, \;\; i \in \{1, \cdots, M\}. \tag{5.14}$$

*the weight sequence converges to the following limit:*

$$\begin{aligned} \boldsymbol{x}_\infty &= \lim_{k\to\infty} \boldsymbol{x}[k] = \boldsymbol{x}_e + \mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}[0] = \boldsymbol{x}_e + \boldsymbol{x}[0] - \mathbf{P}_{\mathcal{S}}\boldsymbol{x}[0] \\ &= \boldsymbol{x}_e + \boldsymbol{x}[0] - \boldsymbol{H}(\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H\boldsymbol{x}[0] \end{aligned} \tag{5.15}$$

*Equivalently, we can express this limit as the projection of* $\boldsymbol{x}[0]$ *onto the affine subspace* $\mathcal{H}$*:*

$$\boldsymbol{x}_\infty = Proj_{\mathcal{H}}\left(\boldsymbol{x}[0]\right) \tag{5.16}$$

*Proof.* Recall that any vector $\boldsymbol{x} \in \mathbb{C}^N$ can be expressed as the sum of its projection onto $\mathcal{S} = \mathcal{R}(\boldsymbol{H})$ and its orthogonal complement, i.e., $\boldsymbol{x} = \mathbf{P}_{\mathcal{S}}\boldsymbol{x} + \mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}$. Observe that the update term in (5.5) lies in the signal space $\mathcal{S}$, for any values of the feedback vector $\boldsymbol{s}[k]$ and desired complex amplitudes $\boldsymbol{b}$. Thus, the component of $\boldsymbol{x}[0]$ orthogonal to $\mathcal{S}$ is unaffected by the iterations. Decomposing the weight sequence into its projection in the signal space and orthogonal to it, as $\boldsymbol{H}(\boldsymbol{s}[k] - \boldsymbol{b}) \in \mathcal{S}$, we may rewrite the iteration as follows:

$$\begin{aligned} \mathbf{P}_{\mathcal{S}}\boldsymbol{x}[k+1] &= \mathbf{P}_{\mathcal{S}}\boldsymbol{x}[k] - \mu\boldsymbol{H}(\boldsymbol{s}[k] - \boldsymbol{b}) \\ \mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}[k+1] &= \mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}[k] \equiv \mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}[0] \end{aligned} \tag{5.17}$$

Thus, the component orthogonal to the signal space remains stuck at $\mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}[0]$. On the other hand, the component restricted to the signal space is follows gradient descent on a quadratic cost function with a unique global minimum $\boldsymbol{x}_e$, and therefore converges to $\boldsymbol{x}_e$. To see this, without presuming the existence of a limit point, and treating $\boldsymbol{x}_\infty$ as the well defined vector on the right hand side of (5.15), consider $\boldsymbol{\Delta}[k] = \boldsymbol{x}[k] - \boldsymbol{x}_\infty$.

In view of (5.17), $\boldsymbol{x}[k] - \boldsymbol{x}[0] = \mathbf{P}_{\mathcal{S}}(\boldsymbol{x}[k] - \boldsymbol{x}[0])$. Thus,

$$\begin{aligned}
\boldsymbol{\Delta}[k] &= \boldsymbol{x}[k] - \boldsymbol{x}_\infty \\
&= \boldsymbol{x}[k] - (\boldsymbol{x}_e + \mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}[0]) \\
&= \boldsymbol{x}[k] - \boldsymbol{x}_e - \boldsymbol{x}[0] + \mathbf{P}_{\mathcal{S}}\boldsymbol{x}[0] \\
&= \mathbf{P}_{\mathcal{S}}(\boldsymbol{x}[k] - \boldsymbol{x}[0]) - \boldsymbol{x}_e + \mathbf{P}_{\mathcal{S}}\boldsymbol{x}[0] \\
&= \mathbf{P}_{\mathcal{S}}\boldsymbol{x}[k] - \boldsymbol{x}_e. \qquad (5.18)
\end{aligned}$$

Note that this $N \times 1$ vector lies in the signal space $\mathcal{S}$, and can therefore be written as in (5.11) for an $M \times 1$ error vector $\boldsymbol{\delta}[k]$. In the absence of noise,

$$\boldsymbol{s}[k] - \boldsymbol{b} = \boldsymbol{H}^H\boldsymbol{x}[k] - \boldsymbol{H}^H\boldsymbol{x}_\infty = \boldsymbol{H}^H\boldsymbol{\Delta}[k] = \boldsymbol{H}^H\boldsymbol{H}\boldsymbol{\delta}[k]$$

Thus (5.5) becomes (5.10), or in terms of the $M \times 1$ error vector $\boldsymbol{\delta}[k]$ as in (5.13). Under (5.14), all eigenvalues of $\boldsymbol{I}_M - \mu\boldsymbol{H}^H\boldsymbol{H}$ are strictly smaller than one in magnitude, hence the right-hand side converges to zero. Denoting the limiting value of the error vectors $\boldsymbol{\Delta}[k]$ and $\boldsymbol{\delta}[k]$ by $\boldsymbol{\Delta}_\infty$ and $\boldsymbol{\delta}_\infty$, respectively, we obtain that $\boldsymbol{\Delta}_\infty = \boldsymbol{H}\boldsymbol{\delta}_\infty = \mathbf{0}$, so that, under Assumption 5.3.1, $\boldsymbol{\delta}_\infty = \mathbf{0}$. This proves that $\mathbf{P}_{\mathcal{S}}\boldsymbol{x}[k]$ converges to $\boldsymbol{x}_e$ and hence (5.15).

Finally, we derive (5.16) for the limiting weight. The projection of any $N$-vector $\mathbf{0}$ onto the affine subspace $\mathcal{H} = \boldsymbol{x}_e + \mathcal{S}^{\perp}$ is $\boldsymbol{x}_e + \boldsymbol{y}$, where $\boldsymbol{y} \in \mathcal{S}^{\perp}$ minimizes the distance of $\mathbf{0}$ from $\mathcal{H}$:

$$\min_{\boldsymbol{y}\in\mathcal{S}^{\perp}} ||\mathbf{0} - (\boldsymbol{x}_e + \boldsymbol{y})||^2 = \min_{\boldsymbol{y}\in\mathcal{S}^{\perp}} ||\mathbf{P}_{\mathcal{S}}\mathbf{0} - \boldsymbol{x}_e||^2 + ||\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{0} - \boldsymbol{y}||^2$$

where we have decomposed the squared distance across $\mathcal{S}$ and $\mathcal{S}^{\perp}$. We cannot change the first term on the right hand side, but can set the second term to zero by setting

$\boldsymbol{y} = \mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{0}$, so that

$$\mathrm{Proj}_{\mathcal{H}}(\mathbf{0}) = \boldsymbol{x}_e + \mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{0} \tag{5.19}$$

Plugging in $\mathbf{0} = \boldsymbol{x}[0]$ completes the proof. □

**Design prescription for minimizing transmit power:** A key implication of the theorem is that, as long as the initial condition $\boldsymbol{x}[0]$ is in the signal space (i.e., it can be written as $\boldsymbol{x}[0] = \boldsymbol{H}\boldsymbol{\eta}$ for some $M \times 1$ vector $\boldsymbol{\eta}$), the iterations converge to the power-efficient (minimum norm) solution $\boldsymbol{x}_e$. To see this, substitute $\boldsymbol{x}[0] = \boldsymbol{H}\boldsymbol{\eta}$ into (5.15) and verify that $\boldsymbol{x}_\infty \equiv \boldsymbol{x}_e$. For example, the initialization $\boldsymbol{x}[0] = \mathbf{0}$, or to a spatial matched filter to one of the beam targets, say $\boldsymbol{x}[0] = \boldsymbol{h}_1$, guarantees convergence to the power efficient solution. When initialization in the signal space is not feasible, then leakage-type mechanisms can be introduced to dissipate the $\mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}[0]$ term that our present algorithm is unable to perturb. This is explored further in Section 5.4.3.

**Effect of linear dependence:** Should Assumption 5.3.1 be violated (i.e., if the channel vectors $\{\boldsymbol{h}_i\}$ are not linearly independent), then $\boldsymbol{b}$ may not be in the range space of $\boldsymbol{H}^H$. In the latter instance, the minimum value of the cost function $J(\boldsymbol{x})$ is not zero. Nonetheless under (5.14), in the noise free case the gradient asymptotically converges to zero:

$$\lim_{k\to\infty} \boldsymbol{H}(\boldsymbol{H}^H\boldsymbol{x}[k] - \boldsymbol{b}) = 0 \tag{5.20}$$

even though $\boldsymbol{H}^H\boldsymbol{x}[k]$ does not converge to $\boldsymbol{b}$. As $J(\boldsymbol{x})$ is convex, such a limit point is still its global minimum.

### 5.3.3 The effect of noise

We now extend the preceding arguments and analyze the impact of noise in the feedback:

$$\boldsymbol{s}[k] = \boldsymbol{H}^H \boldsymbol{x}[k] + \boldsymbol{w}[k], \quad \boldsymbol{w}[k] \sim \mathcal{CN}(\boldsymbol{0}, \sigma_w^2 \boldsymbol{I}). \tag{5.21}$$

We characterize the means and covariances of the weight vectors $\{\boldsymbol{x}[k]\}$ in Theorem 5.2 below. The noisy version of (5.13) is used to show in the theorem that noise does not cause the $\{\boldsymbol{x}[k]\}$ to drift along the affine subspace $\mathcal{H}$, and the error covariance is bounded with a limit point.

We first note that even with noise, the update term in (5.5) lies in $\mathcal{S}$, hence we still have

$$\mathbf{P}_{\mathcal{S}}^{\perp} \boldsymbol{x}[k] \equiv \mathbf{P}_{\mathcal{S}}^{\perp} \boldsymbol{x}[0]$$

Define the error vectors $\boldsymbol{\Delta}[k]$ and $\boldsymbol{\delta}[k]$ as before, using (5.18) and (5.11). The error term driving the iterations is now given by

$$\boldsymbol{s}[k] - \boldsymbol{b} = \boldsymbol{H}^H \boldsymbol{x}[k] + \boldsymbol{w}[k] - \boldsymbol{H}^H \boldsymbol{x}_\infty = \boldsymbol{H}^H \boldsymbol{\Delta}[k] + \boldsymbol{w}[k] = \boldsymbol{H}^H \boldsymbol{H} \boldsymbol{\delta}[k] + \boldsymbol{w}[k]$$

Thus we obtain,

$$\boldsymbol{\Delta}[k+1] = \boldsymbol{\Delta}[k] - \mu \boldsymbol{H} \left( \boldsymbol{H}^H \boldsymbol{\Delta}[k] + \boldsymbol{w}[k] \right) = \left( \boldsymbol{I} - \mu \boldsymbol{H} \boldsymbol{H}^H \right) \boldsymbol{\Delta}[k] - \mu \boldsymbol{H} \boldsymbol{w}[k] \tag{5.22}$$

$$\boldsymbol{H}\boldsymbol{\delta}[k+1] = \left( \boldsymbol{I}_N - \mu \boldsymbol{H} \boldsymbol{H}^H \right) \boldsymbol{H} \boldsymbol{\delta}[k] = \boldsymbol{H} \left( \boldsymbol{I}_M - \mu \boldsymbol{H}^H \boldsymbol{H} \right) \boldsymbol{\delta}[k] - \mu \boldsymbol{H} \boldsymbol{w}[k] \tag{5.23}$$

There are two key points to make about these equations. First with $\lambda_i$ as in Theorem 5.1, the transition matrix in (5.23) is asymptotically stable, though that in (5.22) is not. More importantly, in both equations, the effect of noise is masked by the

channel matrix $\boldsymbol{H}$, precluding the possibility of Brownian motion orthogonal to the signal space.

**Theorem 5.2.** *Consider (5.5) under (5.1) and Assumption 5.3.1, with noisy feedback modeled as in (5.21). Assume that the adaptation gain $\mu$ satisfies (5.14). Then the mean of the weight sequence converges to the same limit as in the noiseless setting:*

$$\begin{aligned} \lim_{k\to\infty} E\left[\boldsymbol{x}[k]\right] &= \boldsymbol{x}_\infty = \boldsymbol{x}_e + \mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}[0] = \boldsymbol{x}_e + \boldsymbol{x}[0] - \mathbf{P}_{\mathcal{S}}\boldsymbol{x}[0] \\ &= \boldsymbol{x}_e + \boldsymbol{x}[0] - \boldsymbol{H}(\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H\boldsymbol{x}[0] = Proj_{\mathcal{H}}\left(\boldsymbol{x}[0]\right) \end{aligned} \tag{5.24}$$

*The covariance $\boldsymbol{\Sigma}_{\boldsymbol{x}}[k] = E\left[\left(\boldsymbol{x}[k] - E[\boldsymbol{x}[k]]\right)\left(\boldsymbol{x}[k] - E[\boldsymbol{x}[k]]\right)^H\right]$ converges to the following limit:*

$$\boldsymbol{\Sigma} = \lim_{k\to\infty} \boldsymbol{\Sigma}_{\boldsymbol{x}}[k] \;\; = \;\; \mu\sigma_w^2\boldsymbol{H}(2\boldsymbol{H}^H\boldsymbol{H} - \mu(\boldsymbol{H}^H\boldsymbol{H})^2)^{-1}\boldsymbol{H}^H. \tag{5.25}$$

*Proof.* Observe (5.22) and (5.23) hold. Define the mean vectors

$$\boldsymbol{m}_{\boldsymbol{\Delta}}[k] = E\left[\boldsymbol{\Delta}[k]\right], \quad \boldsymbol{m}_{\boldsymbol{\delta}}[k] = E\left[\boldsymbol{\delta}[k]\right] \tag{5.26}$$

and the corresponding covariance matrices

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{\Delta}}[k] &= E\left[\left[\boldsymbol{\Delta}[k] - E[\boldsymbol{\Delta}[k]]\right]\left[\boldsymbol{\Delta}[k] - E[\boldsymbol{\Delta}[k]]\right]^H\right], \\ \boldsymbol{\Sigma}_{\boldsymbol{\delta}}[k] &= E\left[\left[\boldsymbol{\delta}[k] - E[\boldsymbol{\delta}[k]]\right]\left[\boldsymbol{\delta}[k] - E[\boldsymbol{\delta}[k]]\right]^H\right] \end{aligned} \tag{5.27}$$

Taking expectations on both sides of (5.22) and (5.23), it is easy to see that the means follow the same trajectories as in the noiseless setting, and therefore converge to zero under the assumptions of Theorem 5.1. Furthermore, subtracting out the means from (5.22) and (5.23) and then taking outer products, it is easy to see that we obtain the

following:

$$\boldsymbol{\Sigma}_{\boldsymbol{\Delta}}[k+1] = \left(\boldsymbol{I} - \mu \boldsymbol{H}\boldsymbol{H}^H\right) \boldsymbol{\Sigma}_{\boldsymbol{\Delta}}[k] \left(\boldsymbol{I} - \mu \boldsymbol{H}\boldsymbol{H}^H\right) + \mu^2 \sigma_w^2 \boldsymbol{H}\boldsymbol{H}^H \tag{5.28}$$

$$\boldsymbol{H}\boldsymbol{\Sigma}_{\boldsymbol{\delta}}[k+1]\boldsymbol{H}^H = \boldsymbol{H}\left(\boldsymbol{I} - \mu \boldsymbol{H}^H\boldsymbol{H}\right) \boldsymbol{\Sigma}_{\boldsymbol{\delta}}[k] \left(\boldsymbol{I} - \mu \boldsymbol{H}^H\boldsymbol{H}\right) \boldsymbol{H}^H + \mu^2 \sigma_w^2 \boldsymbol{H}\boldsymbol{H}^H \tag{5.29}$$

While $\boldsymbol{\Sigma}_{\boldsymbol{\Delta}}[k] = \boldsymbol{\Sigma}_{\boldsymbol{x}}[k]$ is the covariance of the original error vector, we find it more convenient to work with the second recursion, which is the covariance of the representation of the error vector in the signal space. From (5.29), we see that the limiting covariance $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$ must satisfy

$$\boldsymbol{H}\boldsymbol{\Sigma}_{\boldsymbol{\delta}}\boldsymbol{H}^H = \boldsymbol{H}\left(\boldsymbol{I} - \mu \boldsymbol{H}^H\boldsymbol{H}\right) \boldsymbol{\Sigma}_{\boldsymbol{\delta}} \left(\boldsymbol{I} - \mu \boldsymbol{H}^H\boldsymbol{H}\right) \boldsymbol{H}^H + \mu^2 \sigma_w^2 \boldsymbol{H}\boldsymbol{H}^H$$

Pre-multiplying by $\boldsymbol{H}^H$ and post-multiplying by $\boldsymbol{H}$, and then pre- and post-multiplying by $(\boldsymbol{H}^H\boldsymbol{H})^{-1}$, we obtain that

$$\boldsymbol{\Sigma}_{\boldsymbol{\delta}} = \left(\boldsymbol{I} - \mu \boldsymbol{H}^H\boldsymbol{H}\right) \boldsymbol{\Sigma}_{\boldsymbol{\delta}} \left(\boldsymbol{I} - \mu \boldsymbol{H}^H\boldsymbol{H}\right) + \mu^2 \sigma_w^2 \boldsymbol{I} \tag{5.30}$$

Under the convergence condition (5.14), this Lyapunov equation has a unique positive definite solution. We can now verify that this solution is given by

$$\boldsymbol{\Sigma}_{\boldsymbol{\delta}} = \mu\sigma_w^2 (2\boldsymbol{H}^H\boldsymbol{H} - \mu(\boldsymbol{H}^H\boldsymbol{H})^2)^{-1} \tag{5.31}$$

To see this, set $\boldsymbol{A} = \boldsymbol{H}^H\boldsymbol{H}$, and note that $(2\boldsymbol{A} - \mu\boldsymbol{A}^2)^{-1}$ commutes with $\boldsymbol{I} - \mu\boldsymbol{A}$, as both can be expressesd as power series in $\boldsymbol{A}$. Let us now substitute $\boldsymbol{\Sigma}_{\boldsymbol{\delta}} = \mu\sigma_w^2(2\boldsymbol{A} - \mu\boldsymbol{A}^2)^{-1}$ in the right-hand side of (5.31) and check that it simplifies to yield the left-

hand side:

$$
\begin{aligned}
&(\boldsymbol{I}-\mu\boldsymbol{A})\,\mu\sigma_w^2(2\boldsymbol{A}-\mu\boldsymbol{A}^2)^{-1}\,(\boldsymbol{I}-\mu\boldsymbol{A})+\mu^2\sigma_w^2\boldsymbol{I}\\
&\quad=\mu\sigma_w^2(2\boldsymbol{A}-\mu\boldsymbol{A}^2)^{-1}\,(\boldsymbol{I}-\mu\boldsymbol{A})\,(\boldsymbol{I}-\mu\boldsymbol{A})+\mu^2\sigma_w^2\boldsymbol{I}\\
&\quad=\mu\sigma_w^2(2\boldsymbol{A}-\mu\boldsymbol{A}^2)^{-1}\left(\boldsymbol{I}-2\mu\boldsymbol{A}+\mu^2\boldsymbol{A}^2\right)+\mu^2\sigma_w^2\boldsymbol{I}\\
&\quad=\mu\sigma_w^2(2\boldsymbol{A}-\mu\boldsymbol{A}^2)^{-1}\left(\boldsymbol{I}-\mu(2\boldsymbol{A}-\mu\boldsymbol{A}^2)\right)+\mu^2\sigma_w^2\boldsymbol{I}\\
&\quad=\mu\sigma_w^2(2\boldsymbol{A}-\mu\boldsymbol{A}^2)^{-1}-\mu^2\sigma_w^2\boldsymbol{I}+\mu^2\sigma_w^2\boldsymbol{I}\\
&\quad=\mu\sigma_w^2(2\boldsymbol{A}-\mu\boldsymbol{A}^2)^{-1}=\boldsymbol{\Sigma}_{\boldsymbol{\delta}}
\end{aligned}
$$

This proves (5.31). Since $\boldsymbol{\Delta}[k]=\boldsymbol{H}\boldsymbol{\delta}[k]$, we have that $\boldsymbol{\Sigma}_{\boldsymbol{\Delta}}[k]=\boldsymbol{H}\boldsymbol{\Sigma}_{\boldsymbol{\delta}}[k]\boldsymbol{H}^H$. Plugging in (5.31) yields the final covariance limit (5.25). □

Theorem 5.2 shows that, even with noise in the feedback, the mean of the weight vector $\boldsymbol{x}[k]$ converges to the same limit as in the noiseless setting, and the limiting covariance is finite. It is worth highlighting the structure of the error revealed through the proof. As before, the $N$-dimensional error vector $\Delta[k]=\boldsymbol{x}[k]-\boldsymbol{x}_\infty$ is constrained to the $M$-dimensional signal space $\mathcal{S}$, and can therefore be described in terms of an $M$-dimensional error vector $\boldsymbol{\delta}[k]$. The limiting $M$-dimensional covariance $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$ is positive definite under our assumptions, whereas the limiting $N$-dimensional covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Delta}}=\boldsymbol{H}\boldsymbol{\Sigma}_{\boldsymbol{\delta}}\boldsymbol{H}^H$ is positive semi-definite, with $M$ positive eigenvalues, and $N-M$ zero eigenvalues.

We also emphasize that the component of $\boldsymbol{x}[k]$ orthogonal to the signal space remains fixed at $\mathbf{P}_{\mathcal{S}}^{\perp}\boldsymbol{x}[0]$ throughout the iterations, whether or not there is noise in the feedback. The implication of this is that noise in the feedback cannot cause drift in $\boldsymbol{x}[k]$. This is in stark contrast to standard adaptive filtering, where noise components

orthogonal to the signal space can cause coefficient drift, which is typically remedied by mechanisms such as tap leakage. The key difference in our setting is that the effect of noise at the *receivers,* when used for *transmitter adaptation,* is seen through the channel matrix $\boldsymbol{H}$, and hence is restricted to the $M$-dimensional signal space. Thus, no additional mechanisms are required to handle feedback noise, even in the regime $N\gamma M$.

We note that, even if Assumption 5.3.1 is violated (i.e., the channel vectors $\{\boldsymbol{h}_i\}$ are not linearly independent), noise does not induce drift. To see this, note that drift must occur along the null space of $\boldsymbol{H}^H$. This is so as should $\boldsymbol{x} = \boldsymbol{x}^*$ minimize $J(\boldsymbol{x})$ and $\boldsymbol{\eta}$ be in the null space of $\boldsymbol{H}^H$ then as $\boldsymbol{H}^H\boldsymbol{\eta} = 0$, $\boldsymbol{x} = \boldsymbol{x}^* + \boldsymbol{\eta}$ must also minimize $J(\boldsymbol{x})$. Now for any such $\boldsymbol{\eta}$ there holds:

$$\begin{aligned}
\boldsymbol{\eta}^H\boldsymbol{x}[k+1] &= \boldsymbol{\eta}^H\left(\boldsymbol{x}[k] - \mu\boldsymbol{H}(\boldsymbol{s}[k] - \boldsymbol{b})\right) \\
&= \boldsymbol{\eta}^H\left(\boldsymbol{x}[k] - \mu\boldsymbol{H}(\boldsymbol{H}^H\boldsymbol{x}[k] + \boldsymbol{w}[k] - \boldsymbol{b})\right) \\
&= \boldsymbol{\eta}^H\boldsymbol{x}[k].
\end{aligned}$$

Thus, the noise has no impact along the null space of $\boldsymbol{H}^H$. This argument can be formalized further to show that even when $\boldsymbol{H}$ does not have full column rank, noise does not induce drift.

### 5.3.4 Convergence speed vs. residual variance

One of our goals is to study convergence speed and residual variance as the number of transmitters $N$ increases, with the number of receivers $M$ fixed. To this end, we first quantify the effect of $\lambda_i$ and the selection of $\mu$.

Define $\boldsymbol{\Lambda} = \text{ diag } \{\lambda_1, \cdots, \lambda_m\}$ as the eigenvalue matrix of $\boldsymbol{H}^H\boldsymbol{H}$ under the ordering $\lambda_i \geq \lambda_{i+1} > 0$. Then, with $\boldsymbol{U} \in \mathbb{C}^{N\times N}$, $\boldsymbol{V} \in \mathbb{C}^{M\times M}$ unitary matrices, one has

$$\boldsymbol{H} = \boldsymbol{U} \begin{bmatrix} \boldsymbol{\Lambda}^{\frac{1}{2}} \\ 0 \end{bmatrix} \boldsymbol{V}. \tag{5.32}$$

Then from (5.11) and (5.13) one readily obtains:

$$\boldsymbol{\Delta}[k] = \boldsymbol{U} \begin{bmatrix} (I - \mu\boldsymbol{\Lambda})^k & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{U}^H \boldsymbol{\Delta}[0]. \tag{5.33}$$

Thus, the convergence rate is constrained by the largest among $|1-\mu\lambda_i|$. Specifically $\|\boldsymbol{\Delta}[k]\| \leq |1-\mu\lambda_i|^k \|\boldsymbol{\Delta}[0]\|$. Subject to (5.14), the $\mu$ that minimizes the largest among $|1-\mu\lambda_i|$ is, [56],

$$\mu^* = \frac{2}{\lambda_1 + \lambda_M}. \tag{5.34}$$

In this case, the largest value of $|1-\mu\lambda_i|$ is given by $-(1-\mu\lambda_1) = 1-\mu\lambda_M > 0$, which simplifies to

$$\frac{C-1}{C+1} \equiv \left(1 - \frac{2}{C+1}\right) \tag{5.35}$$

where $C = \frac{\lambda_1}{\lambda_M}$ is the condition number of the channel matrix $\boldsymbol{H}$. Unsurprisingly, the convergence rate improves as $C$ decreases (i.e., as the eigenvalue spread gets smaller), and $C = 1$ (no spread) yields deadbeat one step convergence.

A more conservative choice,

$$\mu = \frac{1}{\lambda_1}, \tag{5.36}$$

ensures that $0 < 1 - \mu\lambda_i \leq 1$ for all $i$. In this case, the convergence rate is given by $1 - \mu\lambda_M = 1 - \frac{\lambda_M}{\lambda_1} = 1 - \frac{1}{C}$. As with the optimal choice, the convergence rate improves with declining $C$ and achieves deadbeat status when $C = 1$.

The choice of $\mu = \mu^*$, however, may lead to a larger residual variance. To see this observe that (5.25) and (5.32) yield:

$$\begin{aligned}
\boldsymbol{\Sigma} &= \mu\sigma_w^2 \boldsymbol{H}(2\boldsymbol{H}^H\boldsymbol{H} - \mu(\boldsymbol{H}^H\boldsymbol{H})^2)^{-1}\boldsymbol{H}^H \\
&= \mu\sigma_w^2 \boldsymbol{H}\left(\boldsymbol{V}^H\left[2\boldsymbol{\Lambda} - \mu\boldsymbol{\Lambda}^2\right]\boldsymbol{V}\right)^{-1}\boldsymbol{H}^H \\
&= \mu\sigma_w^2 \boldsymbol{U}\begin{bmatrix}\Lambda^{\frac{1}{2}} \\ \\ 0\end{bmatrix}\boldsymbol{V}\left(\boldsymbol{V}^H\left[2\boldsymbol{\Lambda} - \mu\boldsymbol{\Lambda}^2\right]\boldsymbol{V}\right)^{-1}\boldsymbol{V}^H\begin{bmatrix}\Lambda^{\frac{1}{2}} & 0\end{bmatrix}\boldsymbol{U}^H \\
&= \mu\sigma_w^2 \boldsymbol{U}\begin{bmatrix}\Lambda^{\frac{1}{2}} \\ \\ 0\end{bmatrix}\boldsymbol{V}\boldsymbol{V}^H\left[2\boldsymbol{\Lambda} - \mu\boldsymbol{\Lambda}^2\right]^{-1}\boldsymbol{V}\boldsymbol{V}^H\begin{bmatrix}\Lambda^{\frac{1}{2}} & 0\end{bmatrix}\boldsymbol{U}^H \\
&= \mu\sigma_w^2 \boldsymbol{U}\begin{bmatrix}(2I - \mu\boldsymbol{\Lambda})^{-1} & 0 \\ \\ 0 & 0\end{bmatrix}\boldsymbol{U}^H. \qquad (5.37)
\end{aligned}$$

Of course a smaller $\mu$ results in a smaller steady state covariance. At the same time under (5.34) there obtains

$$\begin{aligned}
\frac{\mu^*}{2 - \mu^*\lambda_i} &\leq \frac{\mu^*}{2 - \mu^*\lambda_1} \\
&= \frac{2}{2(\lambda_1 + \lambda_M) - 2\lambda_1} \\
&= \frac{1}{\lambda_M}.
\end{aligned}$$

Thus,

$$0 \leq \boldsymbol{\Sigma} \leq \frac{\sigma_w^2}{\lambda_M} I. \qquad (5.38)$$

On the other hand if $\mu\lambda_1 \leq 1$, a condition that guarantees convergence, but may not be satisfied by $\mu = \mu^*$, one obtains the smaller bound of

$$\begin{aligned} \frac{\mu}{2-\mu\lambda_i} &\leq \frac{\mu}{2-\mu\lambda_1} \\ &\leq \mu \\ &= \frac{1}{\lambda_1}, \end{aligned}$$

leading to

$$0 \leq \mathbf{\Sigma} \leq \frac{\sigma_w^2}{\lambda_1} I. \tag{5.39}$$

## 5.4 Behavior with large $N$

In this section we study the performance of the JBNF algorithm as $N$, the number of transmitters, becomes large, with the total number of beam and null targets fixed at $M$. Our goal is to study the convergence rate, the power efficient solution $\boldsymbol{x}_e$ and noise performance. We now introduce subscripts to explicitly denote dependence on $N$. For example, $\boldsymbol{H}_N$ is the corresponding channel matrix, the eigenvalues of $\boldsymbol{H}^H\boldsymbol{H}$, are $\lambda_{i,N}$. Section 5.4.1 shows that the convergence rate is bounded from below as $N$ grows, under mild deterministic assumptions on $\boldsymbol{H}_N$. Section 5.4.2 examines convergence speed, steady state residual variance and the nature of the power efficient solution as $N$ grows to infinity, and the channels are Rayleigh fading.

### 5.4.1 Convergence speed with deterministic channels

First suppose that $\mu_N$ is fixed at some value $\mu_0$ such that $\mu_0\lambda_{1,N} < 1$. In this case, (5.14) is satisfied and convergence rate does not decline if $\mu_0\lambda_{M,N}$ does not

decline with $N$.

The channel matrix grows from $N$ transmitters to $N+1$ as:

$$\boldsymbol{H}_{N+1} = \begin{pmatrix} \boldsymbol{H}_N \\ \\ \boldsymbol{g}_{N+1}^H \end{pmatrix} \tag{5.40}$$

with

$$\boldsymbol{g}_{N+1} = [h_{N+1,1}^*,\ h_{N+1,2}^*,\ \ldots,\ h_{N+1,M}^*]^\top, \tag{5.41}$$

where $\boldsymbol{g}_{N+1}^H$ is the $(N+1)$-th row of $\boldsymbol{H}_N$, see (5.1). We then have the following result.

**Theorem 5.3.** *Consider the family of JBNF algorithms (5.5) with an increasing number of transmitters $N > M$ while keeping the step size $\mu_N$ fixed at $\mu_0$. Then the convergence rate of the algorithm is nondecreasing in the number of transmitters $N$ provided $\mu_0 \lambda_{1,N} < 1$.*

*Proof.* As $\mu$ is fixed at $\mu_0$ and $\mu_0 \lambda_{1,N} < 1$, it suffices to show that for all $l < N$ (a) $\mu_0 \lambda_{M,N} < 1$, and (b) $\mu_0 \lambda_{M,l}$ is nondecreasing in $l$. Observe from (5.40) that

$$H_{l+1}^H H_{l+1} = H_l^H H_l + g_{L+1} g_{L+1}^H.$$

Then (a) and (b) follow from the fact that for any pairs of compatibly dimensioned Hermitian matrices $\boldsymbol{A}$ and $\boldsymbol{B}$,

$$\lambda_{\min}(\boldsymbol{A}+\boldsymbol{B}) \geq \lambda_{\min}(\boldsymbol{A}) + \lambda_{\min}(\boldsymbol{B}) \text{ and } \lambda_{\max}(\boldsymbol{A}+\boldsymbol{B}) \leq \lambda_{\max}(\boldsymbol{A}) + \lambda_{\max}(\boldsymbol{B}). \tag{5.42}$$

□

Note that this result does not depend on any specific channel model and holds for all fading and LoS channels. However, using a fixed step-size $\mu_0$ while increasing

the number of transmitters is too restrictive: for large $N$, the value of $\mu_0$ required is unnecessarily small. Thus, we now consider the setting when $\mu_N$ is allowed to change with $N$. Recall from Section 5.3.4 that the convergence rate is bounded from below if and only if the condition number

$$C_N = \frac{\lambda_{1,N}}{\lambda_{M,N}}. \tag{5.43}$$

is upper bounded. To this end we provide a sufficient condition on the $\boldsymbol{h}_i$ that assures the uniform boundedness of $C_N$. The condition is known in the adaptive control literature as the *persistently spanning or excitation condition*, [50]-[51]. It requires that channel submatrices seen by each new batch of transmitters should be sufficiently well conditioned. A feature of this condition is that it can be checked for any deterministic set of channels. However, we show later that i.i.d. complex Gaussian channels asymptotically meet the condition.

**Theorem 5.4.** *Suppose $\boldsymbol{H}_N \in \mathbb{C}^{N\times M}$ is as in (5.40) and $\boldsymbol{g}_i \in \mathbb{C}^M$ as in (5.41). Define $\lambda_{i,N}$ as the eigenvalues of $\boldsymbol{H}_N^H\boldsymbol{H}_N$ and $C_N$ as in (5.43). Suppose there exist $0 < \alpha_l$ and an $L$ such that for all $i$ and $g_i$ defined in (5.40),*

$$0 < \alpha_1 I \leq \sum_{m=i}^{i+L} \boldsymbol{g}_m \boldsymbol{g}_m^H \leq \alpha_2 I.$$

*Then for all $N \geq L$, $C_N$ is uniformly bounded in $N$.*

*Proof.* The result follows from the facts that

$$\boldsymbol{H}_N^H\boldsymbol{H}_N = \sum_{m=1}^{N} \boldsymbol{g}_m \boldsymbol{g}_m^H;$$

and the inequalities (5.42). □

### 5.4.2 Asymptotics with rayleigh fading channels

We now derive a variety of results for large $N$, assuming i.i.d. (across all transmitter-receiver pairs) complex Gaussian channels. All draw upon the following result from [57].

**Theorem 5.5.** *Suppose the channel coefficients* $h_{ij} \sim \mathcal{CN}(0,1),\ i \in \{1 \dots N\},\ j \in \{1 \dots M\}$ *and are i.i.d. Then for any given* $M$*, the condition number* $C_N$ *of the matrix* $\boldsymbol{H}_N^H \boldsymbol{H}_N$ *satisfies* $\lim_{N\to\infty} C_N = 1$ *with probability one. Further, with probability one, there holds*

$$\lim_{N\to\infty} \frac{\lambda_{1,N}}{N} = \lim_{N\to\infty} \frac{\lambda_{M,N}}{N} = 1 \tag{5.44}$$

Referring to the discussion in Section 5.3.4, this implies that the optimal choice (5.34) and the conservative choice (5.36) of $\mu_N$ are asymptotically equivalent, and that we asymptotically obtain arbitrarily fast convergence, both with probability one as $N \to \infty$. Further, the covariance bounds (5.38) and (5.39) imply that the covariance tends to zero. We summarize these results in the following theorem.

**Theorem 5.6.** *Consider the algorithm*

$$\boldsymbol{x}_N[k+1] = \boldsymbol{x}_N[k] - \mu_N \boldsymbol{H}_N(\boldsymbol{s}[k] - \boldsymbol{b}) \tag{5.45}$$

*and suppose the conditions of Theorem 5.5 hold. Then there exists a sequence of* $\mu_N$ *such that in the noise free case convergence to* $\boldsymbol{x}_{N,\infty}$ *occurs arbitrarily fast, and in the presence of noise*

$$\lim_{N\to\infty} \boldsymbol{\Sigma}_N = 0. \tag{5.46}$$

*with probability one.*

Finally, recall that any initialization in the range space of $\boldsymbol{H}_N$ causes convergence to the power efficient solution $\boldsymbol{x}_{e,N}$ in the noise free case. In the presence of noise, convergence to the same point occurs in the mean. In view of Proposition 5.5 and (5.8) there holds:

$$\lim_{N\to\infty} \boldsymbol{x}_{e,N} \;=\; \boldsymbol{H}_N(\boldsymbol{H}_N^H\boldsymbol{H}_N)^{-1}\boldsymbol{b}_N = 0.$$

Further $\boldsymbol{H}_N(\boldsymbol{H}_N^H\boldsymbol{H})_N^{-1}\boldsymbol{H}_N^H = \boldsymbol{U}_N \begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{U}_N^H$. Thus as, $\boldsymbol{U}_N$ is unitary, for every $N$, we have $\left\| \left(I - \boldsymbol{H}_N(\boldsymbol{H}_N^H\boldsymbol{H})_N^{-1}\boldsymbol{H}_N^H\right)\boldsymbol{x}[0] \right\| \leq \|\boldsymbol{x}[0]\|$. Thus, one obtains with probability one that $\lim_{N\to\infty} \left\|\mathrm{Proj}_{\mathcal{H}_N}(\boldsymbol{x}[0])\right\| \leq \|\boldsymbol{x}[0]\|$, and the following.

**Theorem 5.7.** *Suppose the conditions of Theorem 5.6 hold with* $|1-\mu_N\lambda_{i,N}| < 1$ *for all* $i$. *Then in the noise free case for every* $\boldsymbol{x}_N[0]$ *there holds with probability one,* $\lim_{N\to\infty} \|\boldsymbol{x}_{N,\infty}\| \leq \|\boldsymbol{x}_N[0]\|$. *Further when* $\boldsymbol{x}_N[0]$ *is in the range space of* $\boldsymbol{H}_N$ *then with probability one,* $\lim_{N\to\infty} \boldsymbol{x}_{N,\infty} = 0$.

To summarize, for channels that are i.i.d. $\mathcal{CN}(0,1)$, we have established the following results with probability one as $N$ goes to infinity.

(i) Convergence is arbitrarily fast.

(ii) Residual variance goes to zero.

(iii) Initialization in the signal space drives the steady state trasmit power to zero.

(iv) Regardless of initialization the steady state transmit power is no greater than the initial transmit power.

(v) In the presence of noise the last two occur in the mean.

Items (ii), (iii) and (v) together demonstrate the following. As $N$ tends to infinity, should one initialize in the signal space using e.g. the design prescription on Section 5.3.2, then even with noise the limit point approaches almost surely, a *zero transmit power solution in the mean with zero covariance.*

### 5.4.3 Leakage to minimize the total transmit power

Initialization in the signal space ensures the attainment of the power efficient solution in the noise free case, and that in the presence of noise this solution is attained in the mean. However, such initialization may not always be feasible; for instance, if the channel matrix changes, a weight vector that was previously in the subspace $\mathcal{H}$ may no longer be in that space. In such cases, we would like our algorithm to automatically adapt to the new power efficient solution. To accomplish this, we can introduce *leakage*, a popular device both in adaptive filtering and control [58]-[54].

Leakage involves the addition of a *penalty term* proportional to the total transmit power to the objective function in (5.3) to get a new objective function:

$$J_2(\boldsymbol{x}) = \left(\boldsymbol{x}^H\boldsymbol{H} - \boldsymbol{b}^H\right)\left(\boldsymbol{H}^H\boldsymbol{x} - \boldsymbol{b}\right) + \alpha\left(\boldsymbol{x}^H\boldsymbol{x}\right) \tag{5.47}$$

where $\alpha > 0$ is a constant that can be chosen to penalize power inefficiency to a greater or lesser degree as desired. Note that the new objective function $J_2(\boldsymbol{x})$ is also quadratic and convex, and also allows a distributed gradient search implementation:

$$\boldsymbol{x}[k+1] = (1 - \mu\alpha)\boldsymbol{x}[k] - \mu\boldsymbol{H}(\boldsymbol{s}[k] - \boldsymbol{b}). \tag{5.48}$$

In this leaky algorithm (5.48), each transmitter only needs local channel knowledge and aggregate feedback, just as in the original JBNF algorithm (5.5). Furthermore, it can be easily shown that (5.48) is guaranteed to achieve the solution to the JBNF problem with the *minimum total transmit power* for any arbitrary initialization $\boldsymbol{x}[0]$. However, the new algorithm (5.48) suffers from one important limitation compared to (5.5): Proposition 5.5 that guarantees that the convergence can be made arbitrarily fast as the network size increases to infinity, may not hold. In practice, we can use our freedom in choosing the parameter $\alpha$ that determines the size of the penalty term in (5.48) to gain the benefits of power minimization without compromising convergence speed. We illustrate this with a numerical example in Section 5.5.

## 5.5 Simulation results

We now present results of some numerical simulations to illustrate the properties of our proposed JBNF algorithm, specifically its scalability, power efficiency and flexibility to adapt to channel variations. We consider a JBNF system with $N = 20$ transmitters and $M = 5$ receivers of which $M_1 = 2$ are beam targets and the remaining 3 receivers are null targets. All channel gains are modeled as i.i.d. $\sim \mathcal{CN}(0, 1)$, and the noise level is taken to be $-40$ dB at each receiver.

First, we consider initialization of transmit weights to zero. Fig. 5.4 shows the resulting variation of the cost function as well as the individual received signal levels at each receiver under the JBNF algorithm. We see that, within about 40 iterations,

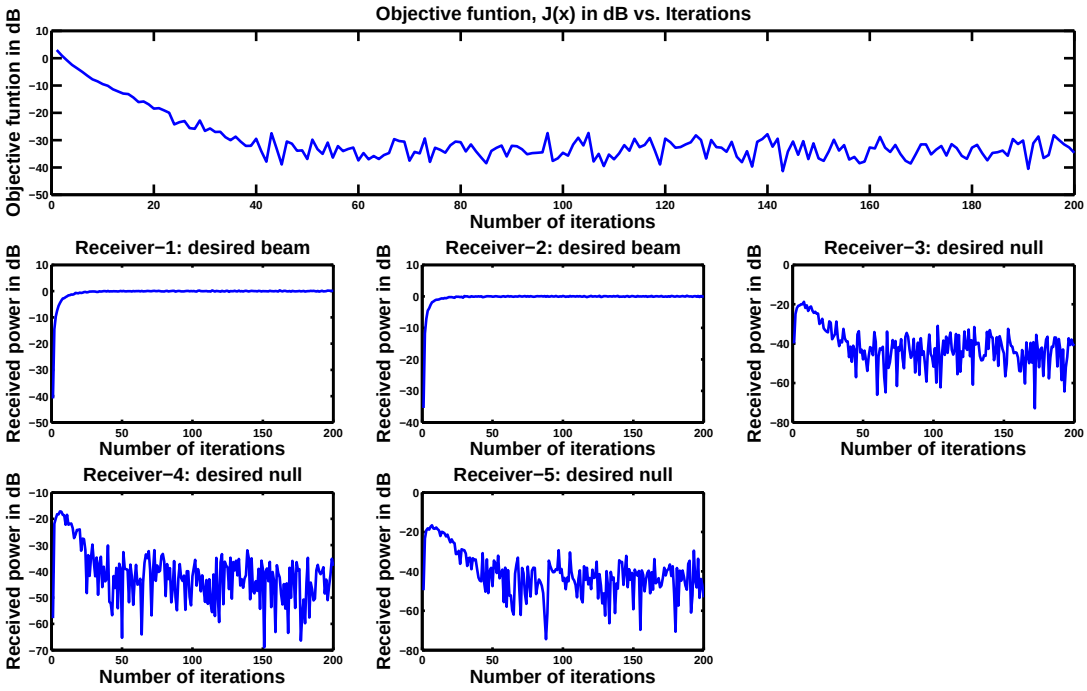
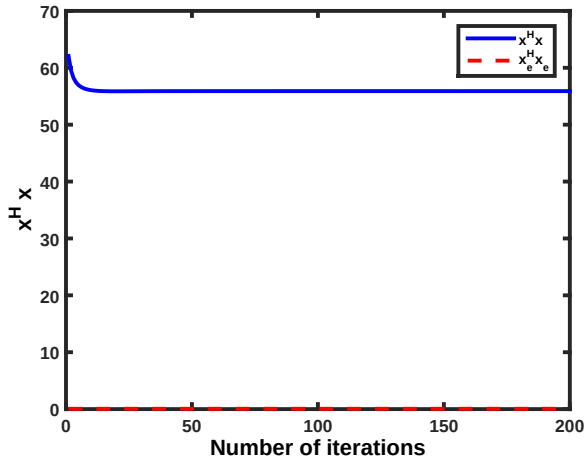


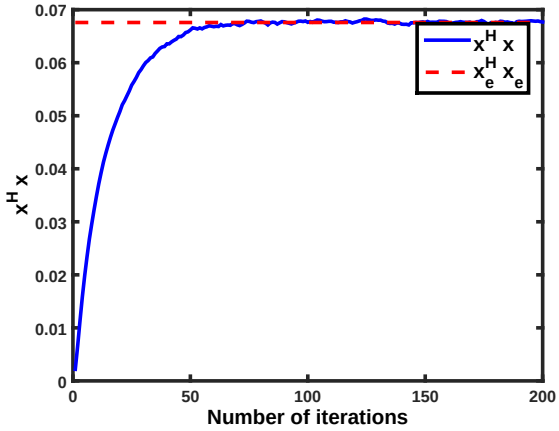

Figure 5.4: Convergence of JBNF algorithm with initialization of transmit weights as zeros.

the cost function as well as the power levels at the null targets have converged to a level close to the noise floor of $-40$ dB. The convergence at the beam targets is even faster.

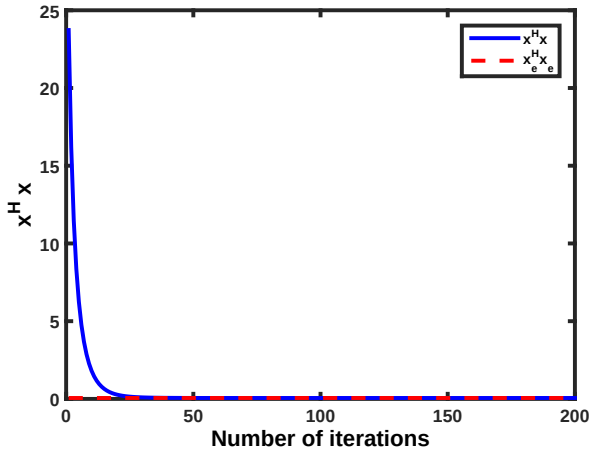
If the weights are initialized randomly from a complex Gaussian distribution on the other hand, we expect a non-zero component orthogonal to the signal space which remains constant, and leads to wasted transmit power. This is confirmed by Fig. 5.5-(a)which shows that the transmit power does not converge to that corresponding to the optimum weights. In contrast, for zero initialization of weights, we see from




(a) random initialization.

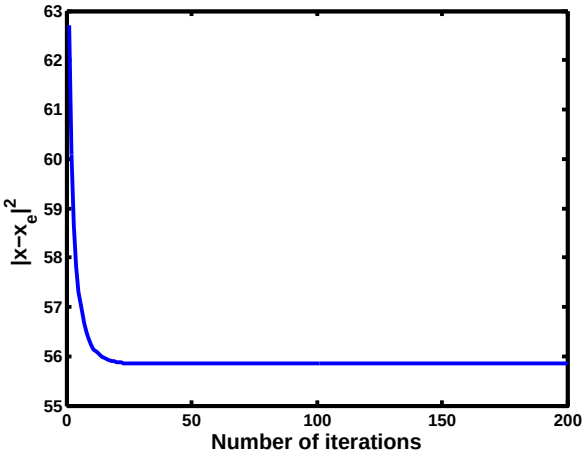



(b) zero initialization.

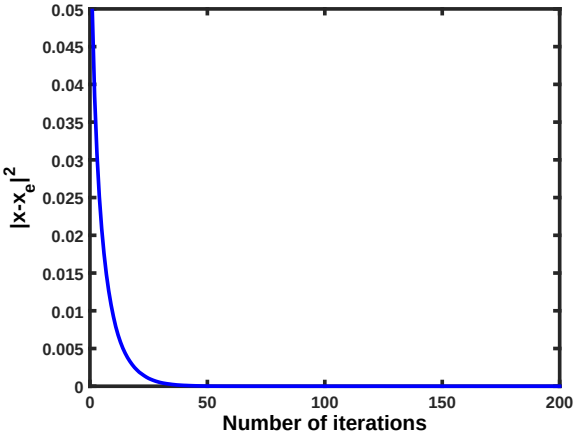



(c) linear combination initialization.

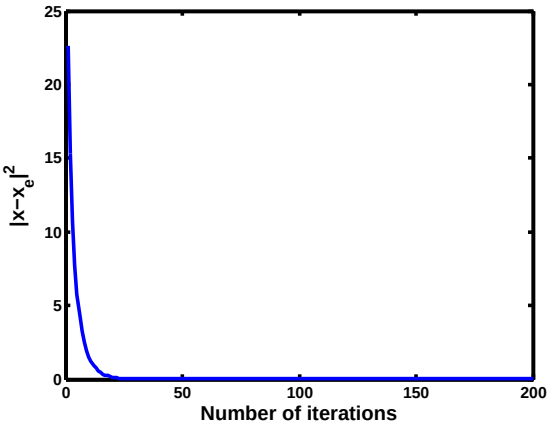
Figure 5.5: In (a,b,c), blue line represents the total transmit power, and red dashed line represents the power corresponding to power efficient solution.




(a) random initialization.




(b) zero initialization.




(c) linear combination initialization.

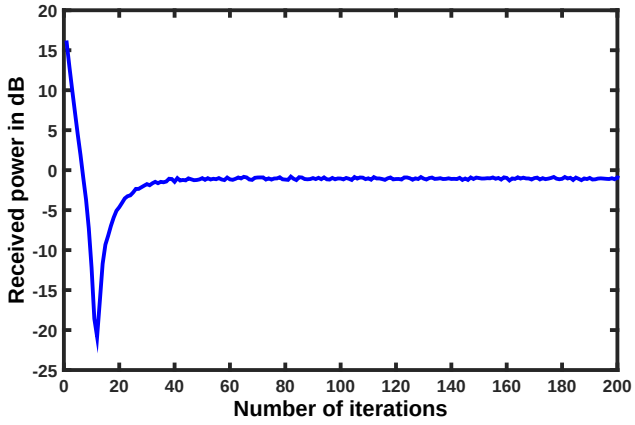
Figure 5.6: Deviation of transmit weights from optimal weights under the JBNF algorithm.

Fig. 5.5-(b), that the efficient power solution is reached within 50 iterations. Another interesting initialization within the signal space is to set the transmit weights to a linear combination of the channel vectors to the beam targets only. Fig. 5.5-(c) shows that this converges to the efficient power solution even faster, within about 20 iterations, possibly because we start with a solution which is already focusing towards the beam targets. Fig. 5.6 show the deviation of transmit weights from optimal weights under the JBNF algorithm corresponding to the three cases in Fig. 5.5 in the same order and confirm the observations stated. Note that, since each transmitter knows its own channel, the only coordination required among transmitters for such an initialization is agreement on the coefficients of the linear combination to be used.

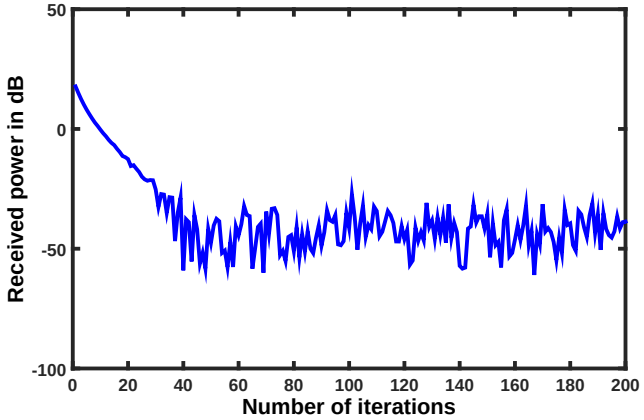
Fig. 5.7 shows the convergence of the JBNF algorithm when the weights are randomly initialized, but using the leakage term described in Section 5.4.3. It shows that we attain the desired beams and nulls, as well as the minimum transmit power.

Fig. 5.8 shows the converged average value of the objective function, $J(\mathrm{x})$ for different noise levels. Clearly, the power at the null-targets is reduced to approximately the noise floor for the range of SNRs plotted in Fig. 5.8.

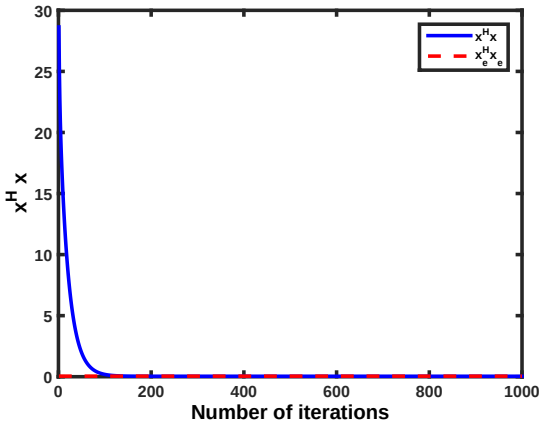
Next, we consider the effects of phase drifts on the performance of the JBNF algorithm. We know from the previous literature [19] that the nullforming process is highly sensitive to phase errors. So this is an important measure of the robustness of our algorithm. Fig. 5.8 shows the converged average value of $J(\mathrm{x})$ against the rate of phase drifts. For this plot, we assume that the time between iterations of the algorithm is $T_s = 50$ ms, and the phase drift over this interval varies from 0 to about




(a) Beam Power.




(b) Null Power.




(c) Total Transmit Power.

Figure 5.7: Convergence of JBNF algorithm with leakage($\alpha = 5$). In (c), blue line represents the total transmit power, red dashed line represents the power corresponding to power efficient solution.

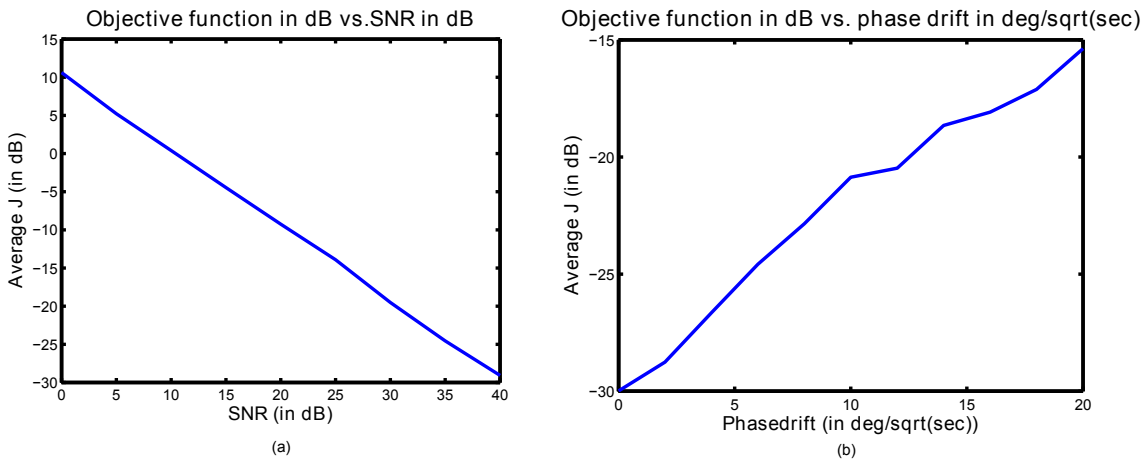


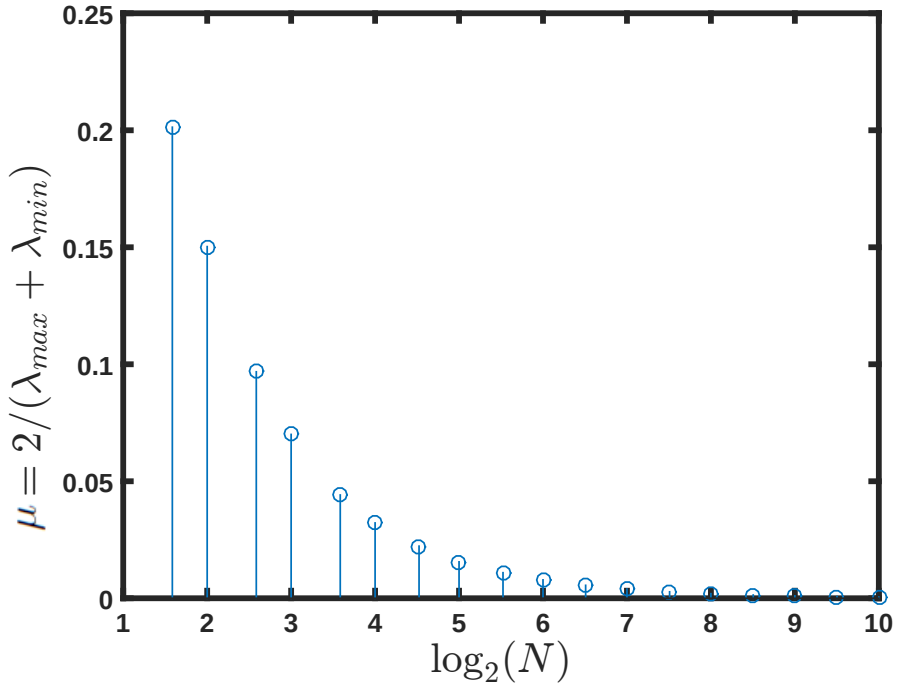

Figure 5.8: JBNF objective function as a function of SNR and phasedrift($N = 20$, $M = 5$, $\mu = 0.005$).

5° in the plot.Noise power is $-40$dB. We can see that the algorithm is able to achieve deep nulls even with such large phase errors.

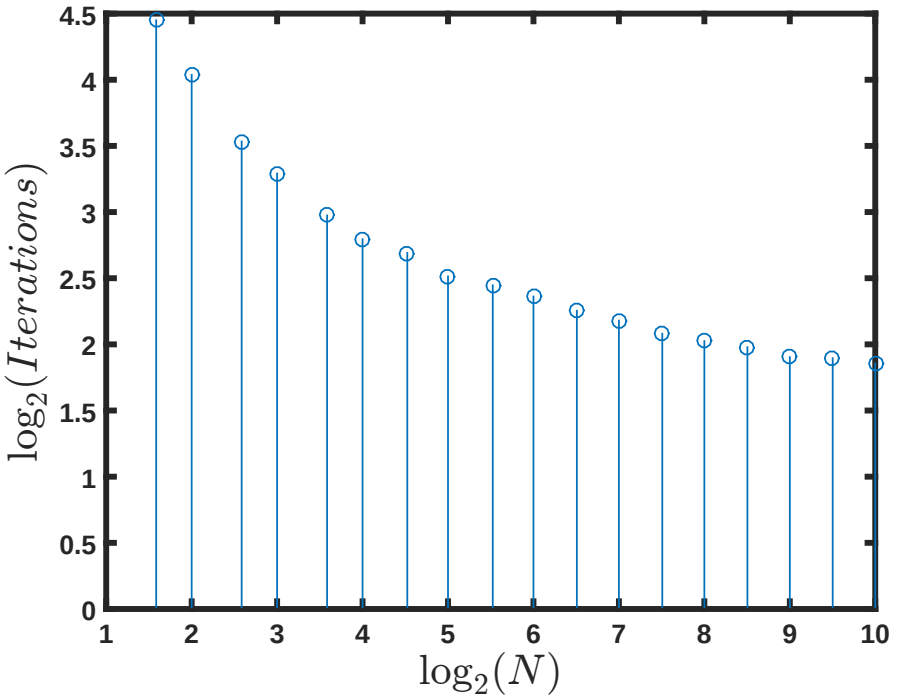
Fig. 5.9 shows scalability, in terms of improved convergence rate, as the number of transmitters $N$ increases. The step-size parameter $\mu$ is selected as per (5.34), the channels are chosen i.i.d. complex Gaussian, and there is no noise. We set $M = 2$ with one beam and one null target. We see that the number of iterations required for the algorithm to drive the cost function to $-60$ dB decreases with $N$ as described in 5.4.2.

We presented scalable distributed algorithms for nullforming in previous chapters and for joint beam and nullforming in this chapter. We showed that these algorithms are scalable to large arrays and robust to noise and phase drifts. In the next chapter, we provide conclusion of the body of work presented in this thesis and also

provide some areas of opportunity for future work in this field.




(a) $\mu$ for different number of transmitters ($N$).




(b) Iterations required for convergence to $-60$ dB.

Figure 5.9: Rate of Convergence of JBNF algorithm with different number of transmitters and $M = 2$.

# CHAPTER 6
# CONCLUSION AND FUTURE RESEARCH

In this thesis, we have presented a set of scalable algorithms for nullforming at a single receiver and joint beam and nullforming at multiple receivers using distributed transmitters by forming a virtual antenna array. We also presented scalable algorithm for nullforming at single receiver which does not assume prior frequency synchronization. Robustness analysis of the single receiver nullforming algorithm was shown.

These algorithms are iterative in nature and have a distributed and scalable quality since each transmitter only needs knowledge of its own channel gain to the receiver( or receivers) in place of the entire channel state information which is generally required by existing nullforming algorithms. This knowledge alongwith feedback from the receiver(or receivers) is enough for the transmitters to evaluate their update equation for the next iteration thereby driving the desired beam or null at the receiver(or receivers). Distributed nullforming specifically poses challenges that call for special attention. Research work presented in this thesis achieves the mentioned goals and overcomes these challenges in a purely distributed fashion.

## 6.1 Open problems

Some open problems that are logical extensions to this thesis are as under

1. **Distributed nullforming at multiple receivers with phase-only adaptation**

Chapter 2 discussed a provably convergent, scalable, distributed nullforming algorithm that allows each transmitter to transmit at full power while steering a null toward a designated receiver through adaptation of the transmission phases using decentralized gradient descent. A natural extension to this scenario would be to try to form nulls at multiple receivers using phase-only adaptation. This would be non-trivial but interesting. The JBNF algorithm that uses amplitude and phase adaptations for forming beams at some receivers and nulls at others simultaneously could be used to achieve this goal but since it would be a multiple-null case, JBNF might lead to the trivial solution of driving all amplitudes of weight coefficients to zero. Excellent applications of this would be in cooperative jamming or communications. For example for wireless security at physical level relying on nodes blanketing a landscape with full power jamming signals while protecting a set of cooperating receivers through forming nulls at their location. For a distributed array of $N$ transmitters and $M$ desired null target receivers, this would mean minimizing the following received signal power:

$$J(\theta) = \sum_{l=1}^{M} |s_k|^2 \tag{6.1}$$

where, $s_k$ is the total complex baseband signal received at the $k$-th receiver provided the same message be transmitted by all transmitters. This task of finding a set of algorithms which would be distributed in nature and also scalable in terms of number of feedback messages and required channel state information at the transmitters, is complicated by the fact that unlike Chapter 2, there are

several receivers to take care of and hence the cost function which was already highly non-convex for single null case has become even more complicated.

2. **Scalable algorithms for spatial multiplexing**

   Chapter 5 discussed a gradient descent algorithm scalable with number of transmitters which could be used as building block for spatial multiplexing. Parallel joint beam and nullforming algorithms can be run at the transmitters for different set of messages to transmit multiple messages simultaneously. But this would call for an initial training phase the time complexity of which would be dependent on the number of messages that need to be sent and hence would not be scalable with increasing number of receivers and hence increasing number of messages that might be expected to be sent simultaneously. A next step could be to design an improved set of algorithms to implement spatial multiplexing which would be scalable in nature.

Apart from these, some other interesting avenues for future exploration are listed as follows

- **Scalability with receivers** Scalability properties of present body of algorithms with increasing number of receivers can be studied. Also, work can be done to find an approximate relationship between the number of transmitters and receivers that would be required for nullforming using our algorithms. Such bounds would also help from a practical implementation point of view.

- **Receive-side beamforming** The present body of algorithms can be extended

to employ the receivers instead of transmitters for distributed beamforming. Essentially the receivers can be made to use smart methods to combine the received signals. Compression and/or quantization can be used to improve performance over existing systems.

- **MAC for distributed antenna arrays** From practical implementation point of view, our proposed methods for the physical layer need corresponding structure at the higher layers. Medium access control for distributed antenna arrays in general is an open area for research. Specifically, simple questions like when should the transmitters in the virtual array decide to transmit, does it makes sense for all transmitters to do carrier sensing, etc., need to be addressed.

- **Experimental demonstration of nullforming** Some work done by us to experimentally demonstrate the ideas presented here can be brought to completion. Specifically, the work done to build the experimental setup to demonstrate beamforming using reciprocity on software-defined radios was used and some parts of the setup were modified to try to effect the nullforming algorithm that requires phase-only updates. This work can be completed in order to present one of the first demonstrations of distributed nullforming on fully wireless platform.

- **Experimental demonstration of JBNF** Once nullforming has been experimentally demonstrated the next step can be to try to implement the joint beam and nullfomring algorithm practically.

# REFERENCES


[1] Ezio Biglieri, Robert Calderbank, Anthony Constantinides, Andrea Goldsmith, Arogyaswami Paulraj, and H. Vincent Poor. *MIMO Wireless Communications*. Cambridge University Press, 2007.

[2] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[3] Ieee draft standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE P802.11-REVmc/D5.0, January 2016 (Revision of IEEE Std 802.11-2012 as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, IEEE Std 802.11ad-2012, IEEE Std 802.11ac-2013, and IEEE Std 802.11af-2013)*, 2016.

[4] Stefania Sesia, Issam Toufik, and Matthew Baker. *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009.

[5] P. Marsch and eds. Fettweis, G.P. *Coordinated Multi-Point in Mobile Communications: From Theory to Practice*. Cambridge University Press, 2011.

[6] H. Rahul, S. Kumar, and D. Katabi. Megamimo: Scaling wireless capacity with user demands, 2012.

[7] T. Cover and A.E. Gamal. Capacity theorems for the relay channel. *Information Theory, IEEE Transactions on*, 25(5):572–584, Sep 1979.

[8] R. Mudumbai, G. Barriac, and U. Madhow. On the feasibility of distributed beamforming in wireless networks. *Wireless Communications, IEEE Transactions on*, 6(5):1754–1763, May 2007.

[9] R. Mudumbai, D.R. Brown, U. Madhow, and H.V. Poor. Distributed transmit beamforming: challenges and recent progress. *Communications Magazine, IEEE*, 47(2):102–110, February 2009.

[10] Munkyo Seo, M. Rodwell, and U. Madhow. A feedback-based distributed phased array technique and its application to 60-ghz wireless sensor network. In *Microwave Symposium Digest, 2008 IEEE MTT-S International*, pages 683–686, June 2008.

[11] M.M. Rahman, H.E. Baidoo-Williams, R. Mudumbai, and S. Dasgupta. Fully wireless implementation of distributed beamforming on a software-defined radio platform. In *Information Processing in Sensor Networks (IPSN), 2012 ACM/IEEE 11th International Conference on*, pages 305–315, April 2012.

[12] F. Quitin, M. Mahboob Ur Rahman, R. Mudumbai, and U. Madhow. Distributed beamforming with software-defined radios: Frequency synchronization and digital feedback. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 4787–4792, Dec 2012.

[13] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac. Scalable feedback control for distributed beamforming in sensor networks. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 137–141, Sept 2005.

[14] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac. Distributed transmit beamforming using feedback control. *Information Theory, IEEE Transactions on*, 56(1):411–426, Jan 2010.

[15] Franeois Quitin, Muhammad Mahboob Ur Rahman, Raghuraman Mudumbai, and Upamanyu Madhow. A scalable architecture for distributed transmit beamforming with commodity radios: Design and proof of concept. *Wireless Communications, IEEE Transactions on*, 12(3):1418–1428, March 2013.

[16] M.D. Zoltowski. Synthesis of sum and difference patterns possessing common nulls for monopulse bearing estimation with line arrays. *Antennas and Propagation, IEEE Transactions on*, 40(1):25–37, Jan 1992.

[17] Hoon Huh, A.M. Tulino, and G. Caire. Network mimo with linear zero-forcing beamforming: Large system analysis, impact of channel estimation, and reduced-complexity scheduling. *Information Theory, IEEE Transactions on*, 58(5):2911–2934, May 2012.

[18] D.R. Brown, U. Madhow, P. Bidigare, and S. Dasgupta. Receiver-coordinated distributed transmit nullforming with channel state uncertainty. In *Information Sciences and Systems (CISS), 2012 46th Annual Conference on*, pages 1–6, March 2012.

[19] D.R. Brown, P. Bidigare, S. Dasgupta, and U. Madhow. Receiver-coordinated zero-forcing distributed transmit nullforming. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 269–272, Aug 2012.

[20] Muhammad M. Rahman, Soura Dasgupta, and Raghuraman Mudumbai. A scalable feedback-based approach to distributed nullforming. April 2013.

[21] A. Kumar, R. Mudumbai, S. Dasgupta, M. M. U. Rahman, D. R. Brown III, U. Madhow, and T. P. Bidigare. A scalable feedback mechanism for distributed nullforming with phase-only adaptation. *IEEE Transactions on Signal and Information Processing over Networks*, 1(1):58–70, 2015.

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

[23] 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*, 12(4):1722–1734, 2013.

[24] A. Manolakos, Y. Noam, and A. J. Goldsmith. Null space learning in cooperative mimo cellular networks using interference feedback. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 3983–3989, 2013.

[25] A. Kumar, S. Dasgupta, and R. Mudumbai. Distributed nullforming without prior frequency synchronization. In *Control Conference (AUCC), 2013 3rd Australian*, pages 207–211, 2013.

[26] A. Kumar, R. Mudumbai, and S. Dasgupta. Scalable algorithms for joint beam and null-forming using distributed antenna arrays. In *Global Communications Conference (GLOBECOM), 2014 IEEE*, pages 4042–4047, 2014.

[27] A. Kumar, R. Mudumbai, S. Dasgupta, U. Madhow, and R. Brown. Distributed mimo multicast with protected receivers: A scalable algorithm for joint beamforming and nullforming. *IEEE Transactions on Wireless Communications*, PP(99):1–1, 2016.

[28] W. Klepczynski and P. Ward. Frequency stability requirements for narrow band receivers. *32nd Annual Precise Time and Time Interval Meeting*, Nov 2000.

[29] R. Mudumbai, P. Bidigare, S. Pruessing, S. Dasgupta, M. Oyarzun, and D. Raeman. Scalable feedback algorithms for distributed transmit beamforming in wireless networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5213–5216, March 2012.

[30] M.M. Rahman, S. Dasgupta, and R. Mudumbai. A distributed consensus approach to synchronization of rf signals. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 281–284, Aug 2012.

[31] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.

[32] D.R. Brown, P. Bidigare, and U. Madhow. Receiver-coordinated distributed transmit beamforming with kinematic tracking. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5209–5212, March 2012.

[33] Slock D.T.M. Guillaud, M. and R. Knopp. A practical method for wireless channel reciprocity exploitation through relative calibration. *ISSPA*, pages 403–406, 2005.

[34] T. Yucek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys Tutorials, IEEE*, 11(1):116–130, First 2009.

[35] M. Ghaderi, D. Goeckel, A. Orda, and M. Dehghan. Efficient wireless security through jamming, coding and routing. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*, pages 505–513, June 2013.

[36] Lun Dong, Zhu Han, A.P. Petropulu, and H.V. Poor. Cooperative jamming for wireless physical layer security. In *Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on*, pages 417–420, Aug 2009.

[37] A. Ozgur, O. Leveque, and D.N.C. Tse. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *Information Theory, IEEE Transactions on*, 53(10):3549–3572, Oct 2007.

[38] E. Telatar. Capacity of multi-antenna gaussian channels. *European Transactions on Telecommunications*, 10(6):585–595, 1999.

[39] W. Hahn. *Stability of motion*. Springer, 1967.

[40] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag New York, 1999.

[41] C. A. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties*. Academic Press, New York, 1975.

[42] C. R. Johnson Jr. Prentice hall, 1988.

[43] Broucke M. Krick, L. and B. Francis. Stabilization of infinitesimally rigid formations of multi-robot networks, 2009.

[44] J.P. LaSalle. *The Stability and Control of Discrete Processes*. Springer Verlag New York, 1986.

[45] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific Press, 1999.

[46] D.R. Brown, R. Mudumbai, and S. Dasgupta. Fundamental limits on phase and frequency tracking and estimation in drifting oscillators. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5225–5228, March 2012.

[47] R. R. Bitmead Jr. Johnson P. V. Kokotovic R. L. Kosut I.M.Y. Mareels L. Praly B. D Riedle Anderson, B.D.O. *Stability of Adaptive Systems: Passivity and Averaging Analysis*. MIT Press, 1986.

[48] S. Dasgupta, B.D.O. Anderson, and Ah Chung Tsoi. Input conditions for continuous-time adaptive systems problems. *Automatic Control, IEEE Transactions on*, 35(1):78–82, Jan 1990.

[49] T.H. Summers, Changbin Yu, S. Dasgupta, and B.D.O. Anderson. Control of minimally persistent leader-remote-follower and coleader formations in the plane. *Automatic Control, IEEE Transactions on*, 56(12):2778–2792, Dec 2011.

[50] S. Dasgupta, B. D. O. Anderson, and A. C. Tsoi. Input conditions for continuous-time adaptive systems problems. *IEEE Transactions on Automatic Control*, 35(1):78–82, 1990.

[51] S. Dasgupta, Y. Shrivastava, and G. Krenzer. Persistent excitation in bilinear systems. *Automatic Control, IEEE Transactions on*, pages 305–313, 1991.

[52] E. W. Bai S. Dasgupta and R. Tempo. Anchored consensus in multiagent systems. In *Proceedings of the 30th Chinese Control Conference*, 2011.

[53] G.J. Rey, R.R. Bitmead, and C.R. Johnson. The dynamics of bursting in simple adaptive feedback systems with leakage. *Circuits and Systems, IEEE Transactions on*, pages 476–488, May 1991.

[54] W.A. Sethares, D.A. Lawrence, C.R. Johnson, and R.R. Bitmead. Parameter drift in lms adaptive filters. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, pages 868–879, Aug 1986.

[55] Anderson.B. D. O. and J.B. Moore. *Optimal Filtering*. Prentice Hall, 1979.

[56] M. Tarrab and A. Feuer. Convergence and performance analysis of the normalized lms algorithm with uncorrelated gaussian data. *Information Theory, IEEE Transactions on*, pages 680–691, Jul 1988.

[57] Edelman A. Eigenvalues and condition numbers of random matrices. *SIAM Journal on Matrix Analysis and Applications*, pages 543–560, 1988.

[58] P. A. Ioannou and P. V. Kokotovic. Instability analysis and improvement of robustness of adaptive control,. *Automatica*, 20(5):583–594, 1985.