# Performance Analysis and Algorithm Design for Distributed Transmit Beamforming

*Shuo Song*


A thesis submitted for the degree of Doctor of Philosophy.
**The University of Edinburgh**.
June 2011

# Abstract

Wireless sensor networks has been one of the major research topics in recent years because of its great potential for a wide range of applications. In some application scenarios, sensor nodes intend to report the sensing data to a far-field destination, which cannot be realized by traditional transmission techniques. Due to the energy limitations and the hardware constraints of sensor nodes, distributed transmit beamforming is considered as an attractive candidate for long-range communications in such scenarios as it can reduce energy requirement of each sensor node and extend the communication range. However, unlike conventional beamforming, which is performed by a centralized antenna array, distributed beamforming is performed by a virtual antenna array composed of randomly located sensor nodes, each of which has an independent oscillator. Sensor nodes have to coordinate with each other and adjust their transmitting signals to collaboratively act as a distributed beamformer. The most crucial problem of realizing distributed beamforming is to achieve carrier phase alignment at the destination. This thesis will investigate distributed beamforming from both theoretical and practical aspects.

First, the bit error ratio performance of distributed beamforming with phase errors is analyzed, which is a key metric to measure the system performance in practice. We derive two distinct expressions to approximate the error probability over Rayleigh fading channels corresponding to small numbers of nodes and large numbers of nodes respectively. The accuracy of both expressions is demonstrated by simulation results. The impact of phase errors on the system performance is examined for various numbers of nodes and different levels of transmit power.

Second, a novel iterative algorithm is proposed to achieve carrier phase alignment at the destination in static channels, which only requires one-bit feedback from the destination. This algorithm is obtained by combining two novel schemes, both of which can greatly improve the convergence speed of phase alignment. The advantages in the convergence speed are obtained by exploiting the feedback information more efficiently compared to existing solutions.

Third, the proposed phase alignment algorithm is modified to track time-varying channels. The modified algorithm has the ability to detect channel amplitude and phase changes that arise over time due to motion of the sensors or the destination. The algorithm can adjust key parameters adaptively according to the changes, which makes it more robust in practical implementation.

# Declaration of originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the School of Engineering at The University of Edinburgh.

Shuo Song

# Acknowledgements

This is probably the most difficult part of thesis writing since several sentences cannot fully express my heartfelt gratitude to their help, support and understand.

In the first place I would like to express my sincere appreciation to my principle supervisor, Dr. John S Thompson, without whom this thesis would not have been possible. Whenever I need help, he is always there with his invaluable guidance, wise advice, bright ideas and broad knowledge. His enthusiasm, patience, and encouragement kept me motivated at challenging times during my PhD study, at moments when I got stuck in deriving math equations, when I was struggling with simulation results, when I felt disappointed with reviewers' comments . . . It was really a wonderful experience working with him as a PhD student.

I would like to thank my second supervisor Dr. Pei-Jung Chung and Prof. Peter M Grant for their expert knowledge, constant encouragement, and insightful discussions which led to a part of the contributions in this thesis. I would also like to thank Prof. Vahid Tarokh at Harvard University for offering me an opportunity to visit his lab. It is always a pleasure and honor to work with them and learn from them.

My deepest gratitude goes to my parents and my grandma, whose love, understand, support and encouragement doubled my efforts to do a better job, helped me grow through pains and kept me feel warm, even in pouring rains. It was my father's first visit to Leeds University in 1995 which initially inspired me to pursue a PhD degree in UK. This thesis is a late reply to their earlier expectations and I hope it will make them proud of me.

My special appreciation goes to Zeyang Zhou, who accompanied me, took care of me, and supported me all the time throughout the PhD journey. It was her love that boosted my confidence in every footstep, eased my stress in every struggle and made my life in Edinburgh full of happiness. Meeting her is my biggest harvest in Edinburgh besides this thesis.

I would like to thank all my colleagues at IDCOM, University of Edinburgh, especially Yan Yan and Chao Wang, who provided me a lot of help at the beginning of my PhD research.

Last but not least, I gratefully acknowledge the financial support from China Scholarship Council/University of Edinburgh Joint Scholarship Program.

# Contents

# List of figures

# List of tables

# Acronyms and abbreviations

| | |
|---|---|
| AWGN | Additive White Gaussian Noise |
| BER | Bit Error Ratio |
| BPSK | Binary Phase Shift Keying |
| CDF | Cumulative Distribution Function |
| CLT | Central Limit Theorem |
| CSI | Channel State Information |
| EGC | Equal Gain Combining |
| EGT | Equal Gain Transmission |
| GPS | Global Positioning System |
| i.i.d. | Independent and Identically Distributed |
| LMS | Least Mean Square |
| MAC | Medium Access Control |
| MISO | Multiple-Input Single-Output |
| MRC | Maximal Ratio Combining |
| MRT | Maximal Ratio Transmission |
| MUSIC | Multiple Signal Classification |
| ODE | Ordinary Differential Equation |
| PAM | Pulse Amplitude Modulation |
| pdf | Probability Density Function |
| QAM | Quadrature Amplitude Modulation |
| RSS | Received Signal Strength |
| SDMA | Space Division Multiple Access |
| SIMO | Single-Input Multiple-Output |
| SNR | Signal-to-Noise Ratio |
| ULA | Uniform Linear Array |

# Nomenclature

[1]

| | |
|---|---|
| $d$ | Fixed antenna spacing in uniform linear array |
| $R$ | Disk radius |
| $N$ | Number of sensor nodes/transmitters |
| $\lambda$ | Wavelength of carrier signal |
| $\mathbf{a}(\theta)$ | Steering vector of the array in the direction $\theta$ |
| $\mathbf{w}$ | Beamforming weighting vector |
| $(r_k, \alpha_k)$ | Position of sensor node $k$ in polar coordinates |
| $(A, \phi_d, \theta_d)$ | Position of the destination in spherical coordinates |
| $\theta^s$ | Elevation angle in spherical coordinates |
| $\phi^s$ | Azimuth angle in spherical coordinates |
| $D_k(\phi_d, \theta_d)$ | Distance between the $k$th node and the destination |
| $F$ | Array factor |
| $BP$ | Beam pattern |
| $DG$ | Directivity gain |
| $P_R$ | Received power at the destination defined in (2.13) |
| $h_i(t)$ | Channel gain from sensor node $i$ to the receiver |
| $\phi_i(t)$ | Cumulative phase error of the carrier signal at the receiver for sensor node $i$ |
| $\phi_0$ | Phase error range |
| $r(t)$ | Received signal |
| $m(t)$ | Common message signal |
| $\gamma$ | Signal-to-noise ratio per bit |
| $P$ | Total transmit power |
| $P_e$ | Bit error ratio |
| $n(t)$ | Additive white Gaussian noise |
| $\sigma_n^2$ | Power of additive white Gaussian noise at the receiver |
| $\sigma_c^2$ | Parameter of Rayleigh fading channel |
| $\sigma_d^2$ | Residual variance |

[1] Only the notations that are frequently used throughout the thesis are listed. Others will be given and explained specifically in certain parts of the thesis.

| | |
|---|---|
| $\widetilde{\sigma}_n^2$ | Total noise power defined in (3.15) |
| $\eta$ | A factor defined in (3.13) for expectation adjustment |
| $H$ | Equivalent channel for the case of EGT |
| $H_M$ | Equivalent channel for the case of MRT |
| $\lambda^2$ | Noncentrality parameter of a non-central chi-square distribution |
| $\sigma^2$ | Variance of a non-central chi-square distribution |
| $m$ | Parameter of Nakagami $m$-distribution |
| $\mathscr{A}$ | Second moment of $\|H\|$ |
| $\mathscr{B}$ | Fourth moment of $\|H\|$ |
| $\mathscr{C}$ | Second moment of $\|H_M\|$ |
| $\mathscr{D}$ | Fourth moment of $\|H_M\|$ |
| $\varphi_i[n]$ | Adaptive component adjusted by transmitter $i$ in time slot $n$ |
| $\theta_i[n]$ | Best known phase recorded by transmitter $i$ in time slot $n$ |
| $\delta_i[n]$ | A random perturbation on the phase of transmitter $i$ in time slot $n$ |
| $\delta_0$ | Phase perturbation size |
| $\Delta_0$ | Optimal perturbation size |
| $\epsilon_i[n]$ | A modifying factor to the adaptive component of transmitter $i$ in time slot $n$ |
| $\Psi_i[n]$ | Channel phase response from sensor node $i$ to the receiver in time slot $n$ |
| $\psi_i$ | Static component in the channel phase response from sensor node $i$ to the receiver |
| $\lambda_i[n]$ | Time-varying component in the channel phase response from sensor node $i$ to the receiver in time slot $n$ |
| $\Lambda_0$ | Phase drift speed |
| $\gamma_i$ | Unknown phase offset at transmitter $i$ |
| $\Phi_i[n]$ | Phase of the received signal at the receiver from transmitter $i$ in time slot $n$ |
| $R[n]$ | Received signal strength at the receiver in time slot $n$ |
| $R_{\text{best}}[n]$ | Best received signal strength recorded at the receiver in time slot $n$ |
| $R_{\min}[n]$ | Minimum received signal strength recorded at the receiver in time slot $n$ |
| $R_{\text{opt}}$ | Optimal received signal strength with perfect phase alignment at the receiver |
| $C_N$ | Number of successive negative feedback steps |
| $C_T$ | Threshold for successive negative feedback steps |
| $R_D$ | Decreasing ratio of perturbation size |
| $S_C$ | Estimation of the channel drift speed within one size period |
| $S_P$ | Estimation of the perturbation catch-up speed within one size period |
| $\rho$ | A discounting factor defined in (5.3) |

| | |
|---|---|
| $[\cdot]^T$ | Matrix transpose operation |
| $[\cdot]^H$ | Matrix Hermitian transpose operation |
| $\lvert \cdot \rvert$ | Absolute value |
| $p(\cdot)$ | Probability density function |
| $\mathrm{E}[\cdot]$ | Expectation |
| $\mathrm{Var}[\cdot]$ | Variance |
| $(\cdot)!$ | Factorial |
| $\max$(a,b) | Maximum value between $a$ and $b$ |
| $\min$(a,b) | Minimum value between $a$ and $b$ |

# Chapter 1
# Introduction

This thesis addresses issues of distributed transmit beamforming in the context of wireless sensor networks. We consider the application scenario that the destination is located far away from the sensor network. Due to energy limitations and hardware constraints of sensor nodes, traditional transmission techniques for sensor networks, such as direct transmission and multi-hop transmission, cannot establish reliable communication links between the sensor network and the destination in such a scenario. Instead, an innovative transmission technique, distributed beamforming, has been put forward to realize long-range communications for wireless sensor networks. The crucial problem of realizing distributed beamforming is to achieve carrier phase alignment at the destination and the consequent feature of distributed beamforming, which is different from conventional beamforming, is the unavoidable phase errors. The thesis will analyze the effect of phase errors on the distributed beamforming performance in theory, and present novel schemes to achieve phase alignment and minimize phase errors for distributed beamforming in practice.

## 1.1 Introduction and motivations

Wireless sensor networks has been one of the key research challenges in recent years because of its great potential for a wide range of applications. A typical wireless sensor network is shown in Figure 1.1. Due to size and cost constraints, sensor nodes are usually supplied by power-limited batteries, equipped with a single antenna, and randomly scattered in the sensing area. These characteristics of the wireless sensor network make distributed transmit beamforming a good candidate for long-range communications. The traditional transmission techniques used for within-network communications, both direct transmission and multi-hop transmission, have limited communication ranges and are inapplicable to long-range communications because of the constraints of power supply and the effect of path loss in wireless transmission. Instead, distributed beamforming can reduce the energy requirement of each sensor node by having the transmission power focused in the desired direction and sharing the energy cost among sensor

nodes. It is well known that transmit beamforming can provide a high signal-to-noise ratio (SNR) gain in proportion to the number of antenna elements. Therefore, the communication range of a sensor network can be significantly increased by simply adding more sensor nodes to constitute a distributed beamformer. Distributed beamforming is a form of cooperative coherent transmission. It is "cooperative" because sensor nodes, each of which equipped with a single antenna, act cooperatively as a virtual antenna array to transmit a common message signal to the far field destination. It is "coherent" because sensor nodes adjust their carrier frequency and phase settings to ensure that signals transmitted from each node will add coherently at the destination, which is similar to a centralized phased array.

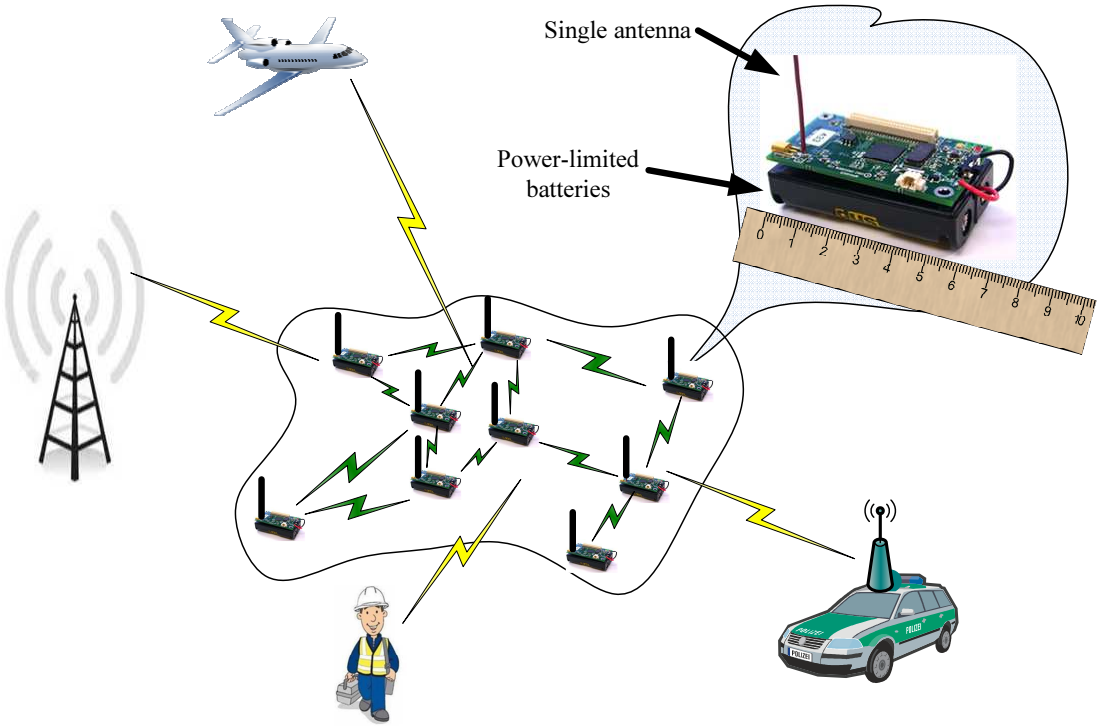



**Figure 1.1:** *A typical wireless sensor network*

While transmit beamforming has been studied for decades, to the best of the author's knowledge, the concept of distributed transmit beamforming was first published in [2] in 2004. In [2], the authors discussed the practical challenges of realizing distributed beamforming, compared it with centralized beamforming, and briefly analyzed the mean and variance of beamforming gains. The work in [2] was later expanded and further studied in [3] and [4], which set a fundamental understanding of distributed beamforming in the research community. Also in 2004, the far-field beam pattern of a random antenna array using collaborative beamforming was studied

in [5]. Although using different words, the phrase "collaborative beamforming" and the phrase "distributed beamforming" in the literature refer to the same transmission technique. In [5], the authors show that randomly located sensor nodes, acting as a distributed transmit beamformer, can form a beam pattern with a narrow main lobe in the desired direction. More interesting work based on [2], [3], [4], [5] came out in the literature and the research progress on this topic was comprehensively addressed in [6]. In [6], the authors also reviewed the key issues of distributed beamforming and challenges we face in future work.

The performance of centralized beamforming largely depends on the knowledge of channel state information (CSI) at transmitter side. Unlike a centralized beamformer, each sensor node contributing to the distributed beamformer has an independent oscillator to generate carrier waves. Even with correct phase settings calculated at each sensor node, phase errors among signals arriving at the destination cannot be eliminated due to oscillator internal noise. Moreover, while centralized beamforming is usually operated with a uniform antenna array, distributed beamforming is performed by randomly located sensor nodes with unknown phase offsets among them. The geometry of sensor nodes estimated by employing existing position estimation techniques is not accurate enough for implementing distributed beamforming. Even with accurate position information, computing the correct phase settings for each sensor node has a high complexity. Therefore, in addition to obtaining CSI at transmitters, the most crucial problem of distributed transmit beamforming is to synchronize sensor nodes in a distributed manner to achieve carrier phase alignment at the destination. Based on this point, publications in the literature about distributed beamforming can be generally classified into two categories. One is to analyze the effects of phase errors on the beamforming performance, the other is to design practical schemes to achieve phase alignment.

## 1.2 Objectives and contributions of the thesis

### 1.2.1 Objectives

Generally, the aim of the thesis is to study the performance of distributed beamforming and to design practical schemes which can improve its performance. Specifically, our study has the following objectives:

- Analyze the achievable bit error ratio (BER) performance of distributed beamforming in

terms of the number of nodes, transmit power and phase errors.

- Design novel schemes to achieve carrier phase alignment at the destination and improve the performance of the schemes.

### 1.2.2 Contributions

The contributions of the thesis are summarized as follows.

- The performance of distributed transmit beamforming in terms of the beam pattern and received power has been analyzed in the literature. From a more practical point of view, we analyze the BER performance of distributed beamforming with phase errors and derive two distinct formulae to approximate the error probability corresponding to a small number of nodes and a large number of nodes respectively. The effects of the number of sensor nodes, transmit power, and phase errors on the BER performance are carefully examined. Simulation results show a good match with the analytical results. With a given number of nodes and a specified transmit power constraint, one can use the BER expressions to bound the permissable phase errors, which gives a quantitative understanding of the impact of phase errors on the beamforming performance.

- Besides theoretical analysis on the beamforming performance, we also contribute to the practical realizations of distributed beamforming in achieving phase alignment and minimizing phase errors at the destination. A simple iterative algorithm using one-bit feedback from the destination in each iteration was proposed to achieve carrier phase alignment in static channels. The one-bit feedback algorithm has many advantages compared to other approaches which make it an attractive candidate in the literature. For example, it is simple in implementation, scalable to a large number of nodes and it does not need knowledge of the CSI. We propose two novel schemes to improve the convergence performance of the feedback algorithm using two different mechanisms. Both schemes keep all the advantages of the original algorithm, and require no extra hardware or information exchange. Then we show that the two schemes can be combined to yield a hybrid algorithm, which can largely enhance the convergence speed of phase alignment by over $40\%$ compared to the original algorithm in static channels.

- There is not much work in the literature focusing on the realization of distributed beamforming in time-varying channels. We further modify the hybrid algorithm to track time-

varying channels. The modified hybrid algorithm has the ability to detect variations in the speed of channel phase changes and adjust phase settings accordingly. It can achieve a reasonable beamforming gain in time-varying channels without the knowledge of CSI. This ability makes the one-bit feedback algorithm much more robust to channel variations in practical implementations.

## 1.3 Structure of the thesis

The rest of the thesis is organized as follows. Chapter 2 gives a background review and motivation of the thesis. This includes an introduction to the special features of wireless sensor networks, a comparison between conventional beamforming and distributed beamforming, key challenges brought to the research community, and major progresses made in the literature. Chapter 3 studies the BER performance of distributed beamforming with phase errors and presents two methods of deriving approximate expressions which can accurately predict the error probability. Chapter 4 reviews the iterative one-bit feedback algorithm in the literature, which can achieve carrier phase alignment at the destination, and presents a novel scheme to improve its convergence speed of phase alignment. Chapter 5 further improves the algorithm by employing a variable step size scheme and extends the algorithm to track time-varying channels. Chapter 6 draws conclusions for the thesis and discusses possible future work on distributed beamforming.

# Chapter 2
# Background

In this chapter, we will give a background review for distributed beamforming techniques, including application scenarios, key challenges it brings to the researchers and some major progress made in the literature. The concept of distributed beamforming was initially brought out in the context of wireless sensor networks for long range communications. We first review some key features of wireless sensor networks and discuss their impacts on the research in distributed beamforming. Then we describe the principles and fundamental problems of realizing distributed beamforming in practice and compare it with the well known conventional beamforming. Last, we present some key results done in the research community on the performance evaluation of distributed beamforming, including its beam pattern performance and the analysis of the received power.

## 2.1 Basic background

Our work on distributed transmit beamforming was based on the idea of applying beamforming techniques into the environment of wireless sensor networks for long distance communications. In this section, we will review the basic background and present a brief survey of the two areas: wireless sensor networks and conventional beamforming.

### 2.1.1 Wireless sensor networks

Advances in microelectronics, sensing, wireless communications, and networking has enabled the deployment of a large number of low-cost, low power, multifunctional sensor nodes in a sensing area, which can collect information, coordinate with each other and form a network via wireless communications. Each sensor node is equipped with a sensing unit, a small processor, a short-range wireless transceiver, and power-limited batteries. Such a network composed of sensor nodes is called a wireless sensor network. Wireless sensor networks were listed as one of the ten emerging technologies that will change the world by MIT's Technology Review in

2003 [7]. It motivated many interesting research problems and has been a key research topic in recent years.

Wireless sensor networks are expected to have a great potential in a wide range of applications [8]. Wireless sensor networks are usually deployed in a sensing area to collect information on demand, either for on-line data collection, e.g. periodic sampling of a parameter of interest, or for alarm triggering, e.g. abnormal parameter variation in the monitored environment. With diversified sensing functionalities, such as light, motion, temperature, humidity, pressure and oxygen, wireless sensor networks can be applied into environmental monitoring, medical treatment, industrial automation, weather sensing, battlefield surveillance, infrastructure maintenance, etc. For example, a smart infrastructure project led by the civil engineering department in Cambridge University used inclinometer sensors to monitor the health of London Underground tunnels [9], [10]. These sensors can detect deteriorations in the structure and avoid the need for routine maintenance conducted in the past, which was time-consuming and costly. Recently, the project researchers are substituting a large number of camera sensors for inclinometer sensors to obtain more precise measurements, which brings challenges for wireless communications as image transmission requires a higher data transmission rate. Other challenges in wireless communications for sensor networks also arise in resource allocation and management, cross-layer design, Medium Access Control (MAC) protocols, location estimation, cooperative transmission, synchronization, etc.

Below we address some key features of wireless sensor networks which are highly related to our project. More details can be found in some textbooks, such as [11], [12].

1. **Power constraints:** In wireless sensor networks, sensor nodes are usually powered by batteries. In most application scenarios, sensor nodes are deployed in a harsh environment where human access is not available, and replacing batteries is considered impossible. Therefore, in order for the lifetime of the sensor nodes to be as long as possible, one of the most important design criteria is energy efficiency. All operations including sensing, computing, storage, communication are considered expensive, among which communication is typically most energy consuming. The wireless communication range of a sensor node is usually very limited due to the power constraints. The network may experience sensor node failures when the node batteries run down. Limited energy also makes node mobility impossible unless the sensor nodes are installed on vehicles.

2. **Randomly scattered:** In most application scenarios, sensor nodes are randomly scattered in the sensing area. The precise location of each node is unknown and the distance between any two nodes is unknown. This sets a tough problem for applying beamforming techniques as signal processing in conventional beamforming is based on precise knowledge of the geometry of antenna array. However, the density of sensor nodes within an area may be approximately controlled during deployment.

3. **Ad hoc operation:** Sensor nodes have to form a network in an ad hoc manner which can provide stable performance even when facing a dynamic network. This is because sensor nodes are often randomly located with no global identification set before implementation, and due to unexpected node failures, the topology of the network changes frequently. Transmission techniques have to operate in an adaptive manner to cope with unexpected changes.

4. **Single antenna:** The size of the sensor node may vary from the order of millimeters to the order of meters. But in most applications, their sizes are expected to be a few square centimeters. Due to size limitations and hardware constraints, each sensor node is usually equipped with a single antenna.

5. **High quantity:** Along with their cheap cost and uncertainty in lifetime, sensor nodes are usually densely deployed in the sensing area. Therefore, scalability to large number of nodes is a key metric considered in the design of communication techniques. In order to reduce interference and traffic load, sensing data are usually processed and compressed locally before transmitting through the network.

6. **Low-cost configurations:** The hardware usually has low energy, limited memory and computational capacities. All protocols and algorithms for communications have to operate under these constraints. In addition, due to internal noise in individual oscillators, the carrier signal of each node undergoes uncompensated phase drift, which will have a negative impact on beamforming performance.

Although wireless sensor networks have many features in common, from the perspective of wireless communications, the area of wireless sensor networks is very application specific. This is because the quality of wireless links and the selection of transmission techniques largely depends on the wave propagation environment. There are different technical issues needing to be resolved for different application scenarios. For example, the signal transmission techniques

used in body sensing to monitor patients' health could be quite different from those adopted in a wireless sensor network designed for forest fire detection. In this thesis, we consider the scenario that the sensor nodes are intended to send information collaboratively to a far field destination, which cannot be reached by a single node due to energy constraints. One of the examples in real world applications is that astronauts intend to collect some information about a planet surface but the spaceship cannot land on the planet due to some technical limitations. Instead, the task can be completed by dropping a large number of sensor nodes to the planet surface. The sensor nodes collect information on demand and report it to the spaceship in the air which may be far away from the sensor network. In such a scenario, transmit beamforming is a very promising form of transmission as it can provide high SNR gains.

In a wireless sensor network, when sensors collect some data which should be reported to a destination, the most common technique used to transmit the data is multihop transmission [13]. Since the path loss is in proportional to the square of transmission distance [14], multihop transmission may consume less energy compared to direct transmission between the source node and the destination node. Also, multihop transmission shares the energy cost among the sensor nodes involved in the multihop chain, which can prevent sensor node failure due to energy shortage. A typical wireless sensor network using multihop transmission is illustrated in Figure 2.1.

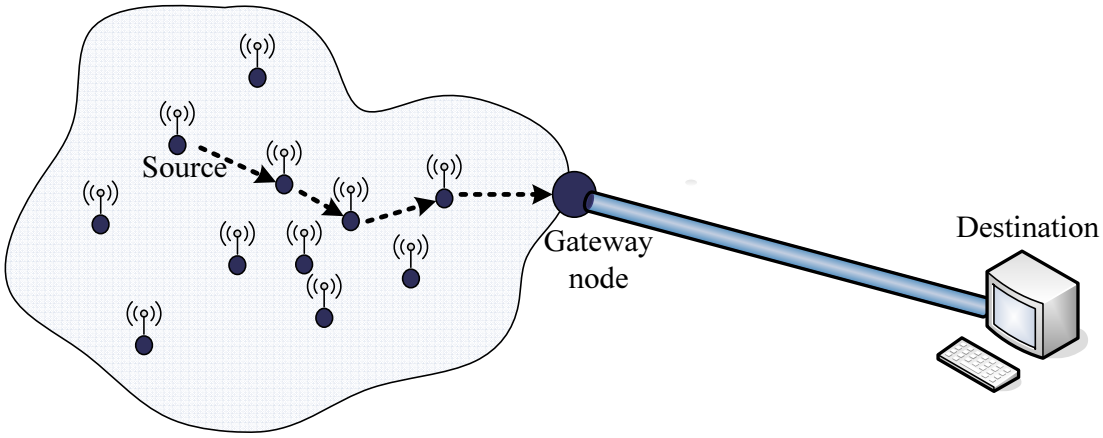



**Figure 2.1:** *Illustration of a wireless sensor network using multihop transmission.*

The gateway node is a specialized node which usually has more energy, memory, computational capacities and other resources compared to other sensor nodes. It is typically located closest to the destination where users analyze the sensing data or it has a wired connection with the

destination. However, in some application scenarios, it is impossible to deploy these gateway nodes in the sensing area and the destination is located far away from the sensor network. Then multihop transmission cannot successfully report the data to the destination and other transmission techniques, such as transmit beamforming, are required.

From the perspective of the operating structure, wireless sensor networks can be classified into two types: centralized and decentralized [15] as shown in Figure 2.2. In the centralized structure, the whole network is divided into several clusters. In each cluster, there is an advanced sensor node, called the head node, which coordinates the operations in the cluster and usually has more functions compared to other sensor nodes in the cluster, which are called slave nodes. For example, the head node may be equipped with more batteries and computational capabilities. Within a cluster, slave nodes transmit data to the head node. The head node then performs data aggregation and exchanges data with other cluster head nodes. When applying beamforming techniques to sensor networks, this type of structure enables the head node to coordinate the synchronization of other nodes in phase and frequency. The corresponding beamforming schemes operate in an open-loop fashion, using minimal coordination with the destination [3]. In the decentralized structure, all sensor nodes are equal, and communication may be established between any two nodes as long as their radio range can reach each other, which results in a more complex beamforming network formation. Within network control and synchronization are difficult to perform in such circumstances. Sensor nodes have to adjust their carrier phase settings with the aid of periodic feedback from the destination. The corresponding beamforming schemes usually operate in a closed-loop fashion [4].

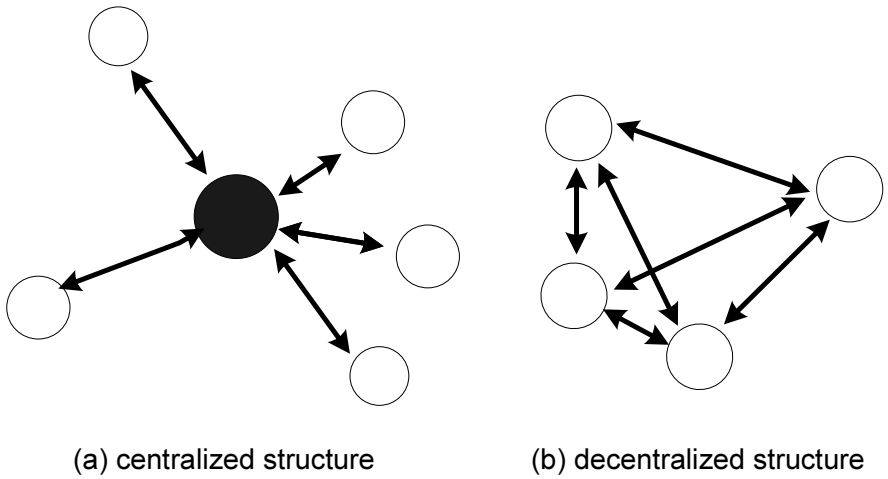



**Figure 2.2:** *Structures of wireless sensor networks*

### 2.1.2 Conventional beamforming

Beamforming techniques use antenna or sensor arrays for directional signal transmission or reception. In the case of a receiving beamformer, the sensor array collects spatial samples of propagating wave fields and processes them with specific weighting vectors to form a linear combination of the outputs. The receiving beamformer can enhance signals from a desired direction and attenuate signals from other directions. In the case of a transmitting beamformer, the antenna array controls the phase and amplitude of the signal transmitted on each antenna in order to create a beam pattern of constructive and destructive interference on the wavefront in space [16], [17]. The advantages of beamforming techniques are well-known. First, it can provide high SNR gain by adding signals coherently. Transmit beamforming techniques can dramatically reduce the energy consumption to achieve a certain SNR at the destination. For example, we consider a beamformer with isotropic antennas transmitting under ideal channel conditions. If a single antenna transmitting with power $P_T$ achieves an SNR of $\rho_1$, then a beamformer with $N$ array elements transmitting with the same total power $P_T$, i.e. each element transmits with power $\frac{P_T}{N}$, can achieve $N$ times of SNR $\rho_N = N\rho_1$. Second, beamforming can provide high directivity gain. In the case of a transmitting beamformer, directivity gain represents the radiation intensity in the desired direction divided by the average radiation intensity over the sphere. In the case of a receiving beamformer, directivity gain represents power arriving from desired direction divided by the noise power at the array over a sphere [18]. Beamforming can work as a spatial filter, which can separate desired signals from interference within the same frequency band but from different spatial locations. It enables space-division multiple access (SDMA) by creating parallel spatial transmission pipes, which may significantly increase communication rates and reduce power consumptions. Beamforming can also be used to suppress interferences from particular directions by performing null-steering operations.

Research in the area of beamforming techniques in the literature is based on the condition that the antenna array is regularly placed, normally with equal distance among the antennas. A typical conventional beamformer is the delay-and-sum beamformer with a uniform linear array (ULA), as illustrated in Figure 2.3. There are $N$ antennas located in a line with uniform spacing equal to $d$. If the channels are ideal, the signals coming from the far-field source in the direction $\theta$ will reach every antenna at different time instants. For the beamformer shown in Figure 2.3, the source signal will reach antenna number 1 first, and then antenna number 2 with a relative delay, and reach the following antennas with an increasing relative delay. The

quantity of relative delay between two antennas located next to each other can be calculated from the equation $\tau = \frac{d\cos\theta}{c}$, where $c$ represents the speed of light. For narrowband signals, these propagation delays turn into phase differences among signals received on each antenna, which can be compensated by phase shifters. If the beamformer introduces a delay of $\tau$ to the received signal from antenna number $N-1$, a delay of $2\tau$ to the signal from antenna number $N-2$, a delay of $(N-i)\tau$ to the signal from antenna number $i$, and so on, then signals from all antennas can be added coherently in phase and the output of the beamformer provides an SNR gain of $N$ as stated above.

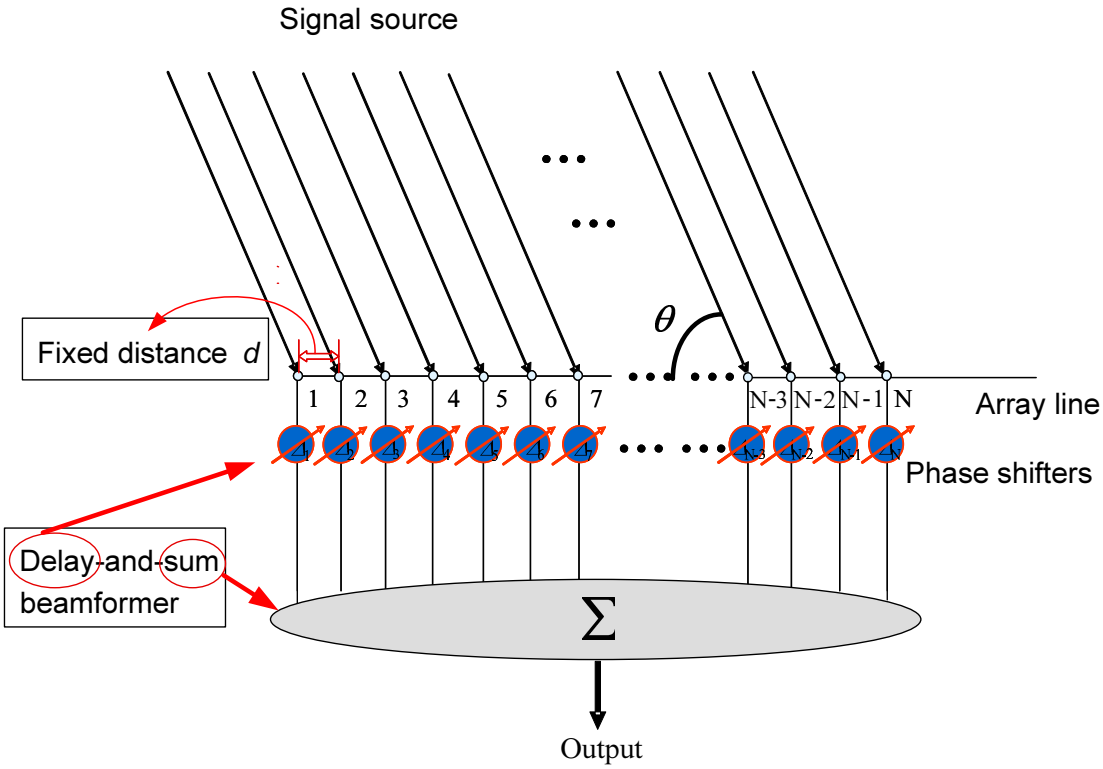



**Figure 2.3:** *Delay-and-sum beamformer with uniform linear array*

At time $t$, the signal received at the antenna array can be expressed as an $N \times 1$ vector:

$$\mathbf{r}(t) = \mathbf{a}(\theta) \cdot s(t) + \mathbf{n}(t) \tag{2.1}$$

where $s(t)$ is the source signal, the $N \times 1$ vector $\mathbf{n(t)}$ is the additive noise at all antenna elements, the $N \times 1$ vector $\mathbf{a}(\theta) = \left[1 \quad \exp\left(-j\frac{2\pi d}{\lambda}\cos\theta\right) \quad \cdots \quad \exp\left(-j\frac{2\pi(N-1)d}{\lambda}\cos\theta\right)\right]^T$ is

called the steering vector of the array in the direction $\theta$, and $\lambda$ is the wavelength of the source signal. The superscript $T$ denotes the matrix transpose. The signals received at different antenna elements are the same except the phase differences due to different propagating distances among antenna elements. The beamformer processes the signals received at each antenna element with a weighting coefficient to combine them coherently and the source signal can be estimated based on the output of the beamformer:

$$\hat{s}(t) = \mathbf{w}^H \cdot \mathbf{r}(t) = s(t) \cdot \mathbf{w}^H \mathbf{a}(\theta) + \mathbf{w}^H \mathbf{n}(t) \tag{2.2}$$

where the $N \times 1$ vector $\mathbf{w}$ is the weighting vector and the superscript $H$ denotes the Hermitian transpose. For a uniformly weighted ULA, the signals received at each antenna are phase shifted and scaled with equal weights $\frac{1}{N}$ before summing. The weighting vector is chosen as $\mathbf{w} = \frac{1}{N}\mathbf{a}(\theta_0)$, where $\theta_0$ is the desired direction.

The beam pattern is a key element in determining the array performance and shows the directivity gain of a beamformer. The beam pattern is defined as:

$$BP(\theta) = \left|\mathbf{w}^H \mathbf{a}(\theta)\right|^2. \tag{2.3}$$

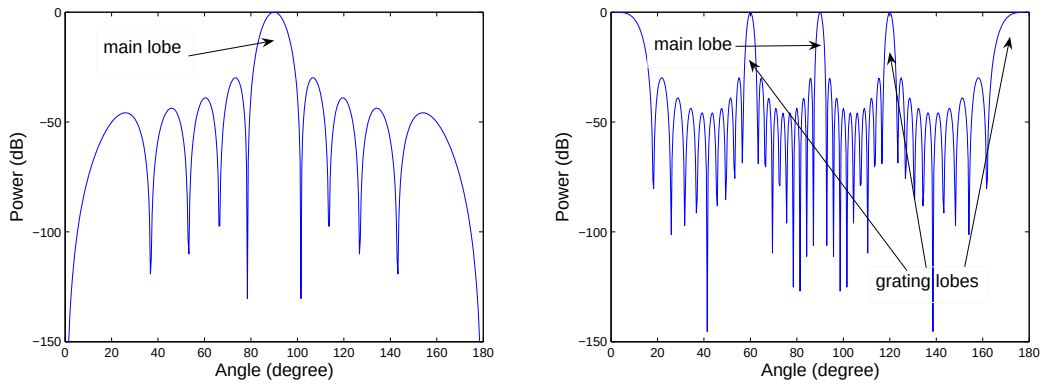
For a uniformly weighted ULA, if the desired direction is $90^\circ$, then $\mathbf{w} = \frac{1}{N}\mathbf{1}$, where $\mathbf{1}$ is the $N \times 1$ unity vector. The corresponding beam pattern becomes:

$$\begin{aligned} BP(\theta) &= \left|\frac{1}{N}\sum_{n=0}^{N-1} e^{-j\frac{2\pi n d}{\lambda}\cos\theta}\right|^2 \\ &= \left|\frac{1}{N}\cdot\frac{1-e^{-j\frac{2\pi N d}{\lambda}\cos\theta}}{1-e^{-j\frac{2\pi d}{\lambda}\cos\theta}}\right|^2 \\ &= \left|\frac{1}{N}\frac{\sin(\frac{N}{2}\cdot\frac{2\pi}{\lambda}\cos\theta\cdot d)}{\sin(\frac{1}{2}\cdot\frac{2\pi}{\lambda}\cos\theta\cdot d)}\right|^2 \end{aligned} \tag{2.4}$$

Figure 2.4 shows the beam patterns of the ULA with the same number of antennas $N = 10$ but different antenna spacings $d = \frac{\lambda}{2}$, $2\lambda$ and the desired direction is set as $\theta_0 = 90^\circ$. As we can see, the fixed distance $d$ among antennas has a big effect on the beam pattern. In Figure

2.4 (a), when the fixed distance $d = \frac{\lambda}{2}$, there is only one main beam in the desired direction $90°$. In such a case, the sensor array can distinguish signals from half of the space, that is, $0° \sim 180°$, which is called the visible region in textbooks [18]. In Figure 2.4 (b), with the same number of antennas, $d = 2\lambda$ results in a much narrower main beam in the desired direction $90°$ and several grating lobes in other directions, which brings a problem of estimating the angle of arrival of incoming signals. Since the delay-and-sum beamformer only adjusts the phases of the received signals, the shape of the beam pattern remains unchanged when the desired direction changes. If each antenna scales its signals with different weights, the corresponding shape of beam pattern also changes.

Various superresolution techniques have been proposed for the angle of arrival estimation in the literature, and the most well studied one is the multiple signal classification (MUSIC) algorithm [19]. The MUSIC algorithm can be used to locate several closely spaced signals and produce sharp peaks in the vicinity of the angle of arrivals. Its resolution capability depends on the received SNR and the number of snapshots. The performance of the MUSIC algorithm degrades rapidly when the SNR or the number of snapshots fall below a certain threshold. The weakness of the MUSIC algorithm is that it cannot provide a good performance when the source signals are highly correlated.




(a) Number of sensors $N = 10$, fixed distance $d = \frac{\lambda}{2}$ (b) Number of sensors $N = 10$, fixed distance $d = 2\lambda$

**Figure 2.4:** *Beam pattern of uniform linear array with different antenna spacings* $d = \frac{\lambda}{2}, 2\lambda$

In conventional beamforming, antenna elements are regularly placed with known distances between them. If the direction of the source signals is available, it is easy to calculate the propagation delays of signals arriving at different antennas. The relative delays between antennas can be compensated by appropriate phase shifts to ensure signals are added up coherently. It is also

easy to adjust these phase shifters in an adaptive way when the desired direction is changing.

## 2.2 Distributed transmit beamforming

In this section, we introduce the concept of distributed transmit beamforming in the context of wireless sensor networks and the challenges it brings to the researchers.

### 2.2.1 Concept of distributed beamforming

As we discussed in Section 2.1.1, in some application scenarios, the user or destination may be sufficiently far away from the sensing area that signal transmission between the sensor network and the destination cannot be accomplished by a single node due to the high power cost of wireless transmission over a long distance and the low battery power constraints of sensor nodes. The traditional way of using a multihop chain to transmit the sensing data is no longer applicable in such scenarios. However, sharing knowledge of sensing data within the local network among sensors may be relatively low in cost and can be easily realized by broadcasting. Then sensor nodes may transmit the data in a collaborative way, i.e. several sensor nodes transmit a common message signal simultaneously and adjust their phase settings to ensure that the signals transmitted from different nodes will combine constructively at the destination. In principle, this method of transmission reduces power consumption by having the sensor nodes form a virtual antenna array to perform transmit beamforming. Since sensor nodes are operating in a distributed manner to complete the task, this technique is called distributed beamforming or collaborative beamforming in the literature.

Figure 2.5 shows an illustration of the application scenarios of distributed beamforming. Distributed beamforming can be used to establish communications between a sensor network and a distant user, either a base station or a vehicle. It can also be used to establish communications between two clusters of sensors which are located far away from each other. As discussed in Section 2.1.2, under ideal channel conditions, the SNR gain of beamforming grows linearly with the number of transmitters. Therefore, the low-power limitation on each sensor node can be compensated by having more nodes involved in the distributed beamformer. Distributed beamforming not only reduces the overall energy cost which can prolong the lifetime of the whole network, it also shares the power consumption among sensor nodes, which can prevent single node failure. It also has the potential to reduce interference to other users by having the

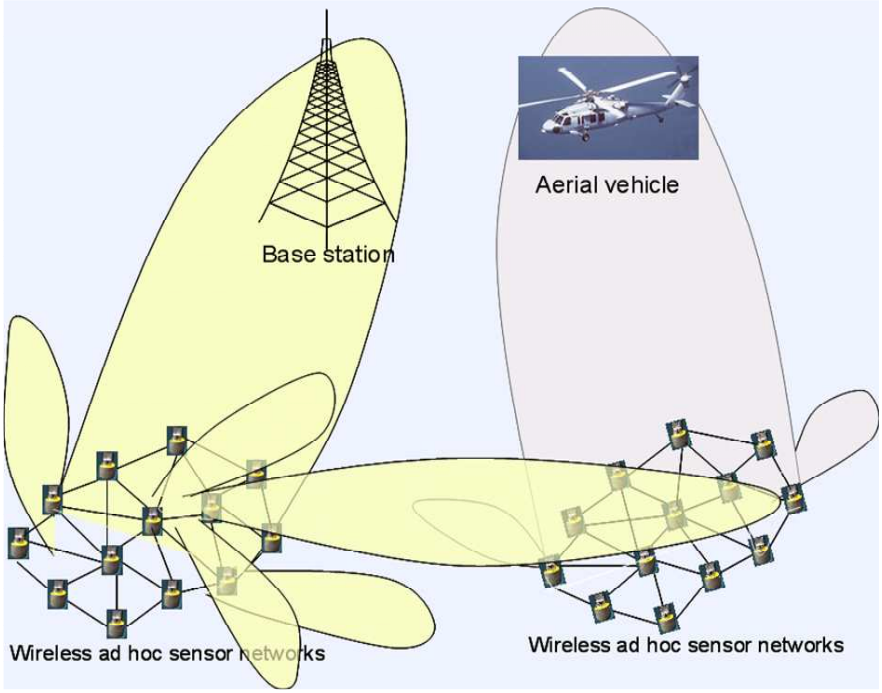



**Figure 2.5:** *Illustration of the application scenarios of distributed beamforming*

transmission power focused in one direction. Depending on the objectives, the SNR gain can be transferred into increments in the communication rate or range [6].

### 2.2.2 Challenges in practical realizations

Although distributed beamforming techniques could bring many attractive advantages in wireless sensor networks, especially improved energy efficiency, a number of challenges arise in its practical applications at the same time. The fundamental problem of realizing distributed beamforming is that there is no central control connected to all elements forming the beamformer and all operations of the beamforming process have to be organized and implemented in a distributed manner.

The principle behind the transmit beamforming technique is that the signals transmitted from each antenna should be frequency synchronized and phase adjusted so that the signals will add coherently at the destination. While conventional beamforming is implemented by a central-controlled device equipped with a regularly placed antenna array, distributed beamforming is performed by a virtual antenna array composed of randomly located sensor nodes, each of which is equipped with a single antenna. As discussed in Section 2.1.2, the steering vector of

a conventional beamformer can be easily computed with the knowledge of the fixed spacing between antennas and the desired direction in space. In contrast, the steering vector or the correct phase settings at transmitters are hard to compute for distributed beamforming because sensor nodes have unknown distances between them. Several steps require to be followed to tackle the practical problems of realizing distributed beamforming:

First, the sensing data must be shared or disseminated within the sensor network in an energy-and-time efficient way [20]. If the sensing data gathered from sensor nodes are strongly correlated, a data fusion process is required to cut the load in data communications [21]. Before beamforming to the far-field destination, all sensor nodes should share amongst each other the same message signal. The overhead of this information sharing process grows with the number of sensor nodes and partly depends on the topology of the network. There exists a tradeoff between the cost of within-network dissemination and the beamforming array gain. Therefore, the number of sensor nodes forming a beamformer should be carefully chosen to optimize the energy efficiency. If we consider one node broadcasting the sensing data to other nodes, the information sharing process can also be viewed as the first phase of a relaying process where other nodes are considered as relays. Such a relay network experiences both a total transmit power constraint and an individual relay power constraint. The performance of distributed beamforming in relay networks with perfect or partial channel state information and different relaying strategies has been studied in [22], [23], [24]. In this thesis, our work focuses on the issue of phase alignment at the destination and we assume perfect information sharing among sensor nodes in the following chapters to conduct BER analysis and algorithm design. To obtain a more comprehensive understanding of the beamforming performance, one may include the impact of the errors in the information sharing process.

Second, the carrier frequency and phase offset generated from each sensor node must be synchronized and adjusted to secure phase alignment at the receiver. Frequency synchronization can be achieved by employing a master-slave scheme presented in [3], where the slave nodes lock their frequencies to a reference signal periodically broadcasted by the master node. But phase adjustment on each sensor node needs much more effort to be resolved. As discussed in Section 2.1.2, in conventional beamforming, antennas are usually regularly placed with known distances among them and all antennas are connected by wires to a central control unit. The propagation delays among antennas can be easily calculated based on the known geometry of the antenna array and the desired direction for beamforming. If CSI is available, phase align-

ment can be easily achieved by compensating the delays using phase shifters. However, in distributed beamforming, sensor nodes are randomly scattered and the distances among them are unknown. Certain location estimation schemes have been designed for wireless sensor networks in the literature [25], [26]. However, the accuracy of these schemes remains in the order of meters which is not accurate enough to satisfy the requirement of beamforming. One may also consider using the Global Positioning System (GPS). However, the estimation obtained by using GPS is not accurate enough either and it is necessary to have a line of sight from the sensor node to the satellite which makes the GPS inapplicable in some application scenarios. According to the IEEE 802.15.4 standard for low-rate wireless personal area networks, the operating frequency of sensor networks is around 900MHz or 2.4GHz, which correspond to a wavelength of 0.34 meters or 0.125 meters. For example, a carrier signal of 2 GHz has a wavelength of 15 centimetres, which means an error of 7.5 centimetres in location estimation can turn the constructive interference into destructive interference. Moreover, each sensor node has an individual oscillator to generate carrier waves which undergoes uncompensated phase drifts due to oscillator internal noise. Therefore, phase errors among signals arriving at the destination cannot be avoided in distributed beamforming and phase alignment is the key obstacle to realizing distributed beamforming in practice. While research on conventional beamforming usually focuses on optimum weight design, especially for power allocation among antennas, research on distributed beamforming requires more emphasis on phase adjustment. This is because the beamforming performance is sensitive to phase errors, which has a much stronger impact compared to power allocations.

Third, all sensor nodes must transmit the message signal at "the same time" which raises an issue of timing synchronization. Errors in timing synchronization contribute to the unknown phase offsets at transmitters and also cause inter symbol interference. Timing synchronization techniques for wireless sensor networks are summarized in [27], [28].

## 2.3 Performance evaluations in the literature

Given the many advantages of using distributed beamforming techniques, we require to investigate the factors which control the beamforming performance. The first factor is the number of nodes. As discussed in Section 2.1.2, the SNR gain of beamforming grows linearly with the number of nodes. Increasing the number of nodes can dramatically reduce the energy cost of each sensor node for long-range communications, and reduce interference to other co-channel

users. The second factor is the node density. The exact geometry of the network is hard to measure, but the approximate density of nodes in the sensing area can be roughly controlled during deployment in practice. The node density can be measured as the number of nodes per unit area, or the average distance between adjacent nodes. Third, we must consider the impact of phase errors on performance. Phase errors among signals arriving at the receiver may be caused by errors in node position estimation, channel estimation, timing synchronization or carrier synchronization.

### 2.3.1 Analysis of beampattern

The beam pattern intuitively shows the performance of a beamformer, including its directivity gain and SNR gain. In Section 2.1.2, we reviewed the beam pattern of a ULA conventional beamformer which can form a narrow main lobe in the desired direction. The question is for the case of ideal channels, correct phase settings and perfect timing synchronization among sensor nodes, whether a distributed beamformer, viewed as a random antenna array, can form a useful beam pattern with a narrow main beam in the desired direction, and what are the impacts of the number of nodes and the node density on the beam pattern. The beam pattern of distributed beamforming has been well studied in the literature. Below we discuss some fundamental features of the beam pattern of distributed beamforming, more details can be found in [29], [30], [31], [32], [33].

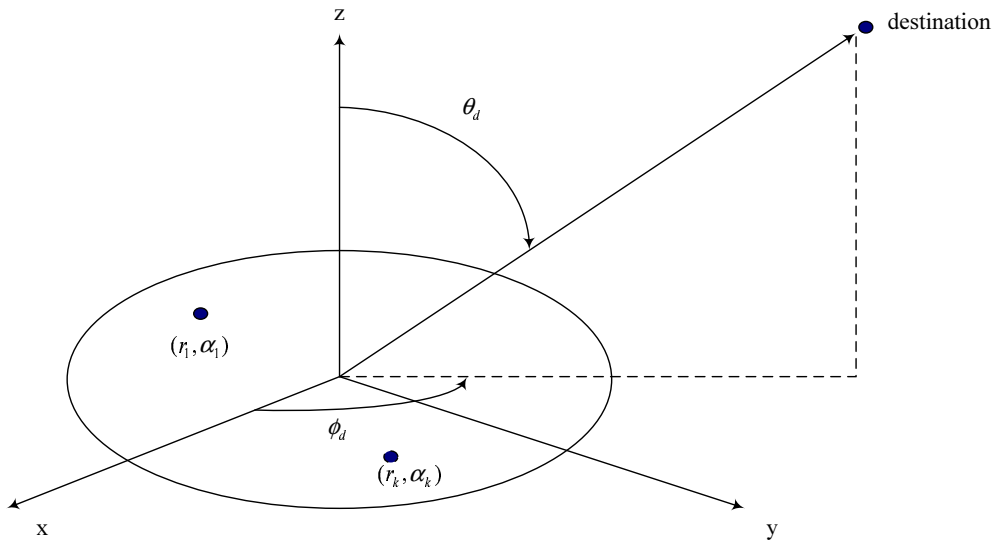



**Figure 2.6:** *Coordinate positions of the system*

We consider a system model that all sensor nodes are located randomly, following a uniform distribution within a disk of radius $R$ in the plane. We assume all sensor nodes transmit with unit power under ideal channel conditions and the path losses of all sensor nodes to the destination are identical. The coordinate positions of sensor nodes are defined in Figure 2.6. We denote ($r_k$, $\alpha_k$) as the position of sensor node $k$ where $r_k$ is the distance from the $k$th node to the disk center, and $\alpha_k$ is the angle to the common reference direction in polar coordinates. We denote angle $\theta^s \in [0, \pi]$ as the elevation direction, $\phi^s \in [-\pi, \pi]$ as the azimuth direction in spherical coordinates, and $(A, \phi_d, \theta_d)$ is the location of the destination. The array factor can be mathematically expressed as:

$$F(\phi^s, \theta^s \mid \mathbf{r}, \boldsymbol{\alpha}) = \frac{1}{N} \sum_{k=1}^{N} e^{j\frac{2\pi}{\lambda}[D_k(\phi^s,\theta^s) - D_k(\phi_d,\theta_d)]} \tag{2.5}$$

where $\mathbf{r} = [r_1, r_2, \cdots, r_N] \in [0, R]^N$ and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_N] \in [-\pi, \pi]^N$ represents the given realization of all the sensor nodes locations. The scalar $N$ is the number of sensor nodes and $\lambda$ is the wavelength of carrier signals. The scalar $D_k(\phi^s, \theta^s)$ denotes the distance between the $k$th node and the reference location, and $D_k(\phi_d, \theta_d)$ denotes the distance between the $k$th node and the destination. We assume that the destination is located far away from the sensor nodes, i.e. $A \gg r_k$, and therefore:

$$D_k(\phi_d, \theta_d) \approx A - r_k \sin\theta_d \cos(\phi_d - \alpha_k). \tag{2.6}$$

The array factor can be approximated as:

$$\begin{aligned} F(\phi^s, \theta^s \mid \mathbf{r}, \boldsymbol{\alpha}) &\approx \frac{1}{N} \sum_{k=1}^{N} e^{j\frac{2\pi}{\lambda} r_k [\sin\theta_d \cos(\phi_d - \alpha_k) - \sin\theta^s \cos(\phi^s - \alpha_k)]} \\ &\triangleq \widetilde{F}(\phi^s, \theta^s \mid \mathbf{r}, \boldsymbol{\alpha}) \end{aligned} \tag{2.7}$$

For simplicity, we assume the destination node locates on the same plane of the disk where sensor nodes are located and we only study the beam pattern in this plane, i.e. the elevation angle $\theta^s = \theta_d = \frac{\pi}{2}$. Equation (2.7) can be simplified as:

$$\widetilde{F}(\phi^s \mid \mathbf{r}, \boldsymbol{\alpha}) = \frac{1}{N} \sum_{k=1}^{N} e^{j\frac{4\pi}{\lambda} r_k \sin\left(\frac{\phi_d - \phi^s}{2}\right) \sin\left(\alpha_k - \frac{\phi_d + \phi^s}{2}\right)} \tag{2.8}$$

and the far-field beam pattern is defined as:

$$BP(\phi^s \mid \mathbf{r}, \boldsymbol{\alpha}) \triangleq \left| \widetilde{F}(\phi^s \mid \mathbf{r}, \boldsymbol{\alpha}) \right|^2 \tag{2.9}$$

The average beam pattern taken over all realizations of $(\mathbf{r}, \boldsymbol{\alpha})$ is given by:

$$BP_{av}(\phi^s) \triangleq \mathrm{E}_{\mathbf{r},\boldsymbol{\alpha}} \{BP(\phi^s \mid \mathbf{r}, \boldsymbol{\alpha})\}. \tag{2.10}$$

The directivity gain is defined as:

$$DG(\mathbf{r}, \boldsymbol{\alpha}) = \frac{\int_{-\pi}^{\pi} BP(\phi_d \mid \mathbf{r}, \boldsymbol{\alpha}) d\phi^s}{\int_{-\pi}^{\pi} BP(\phi^s \mid \mathbf{r}, \boldsymbol{\alpha}) d\phi^s}. \tag{2.11}$$

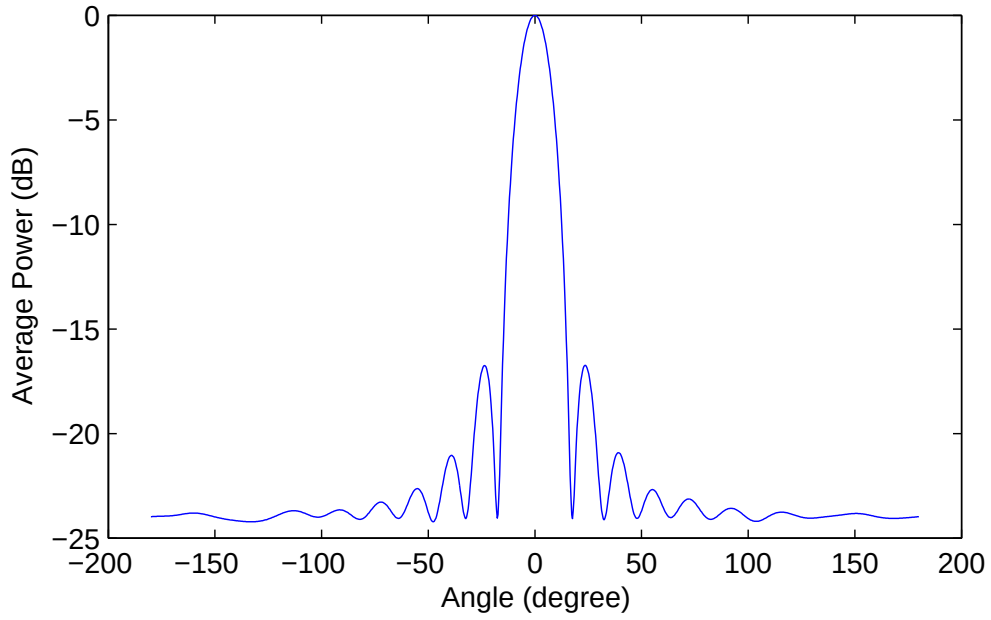



**Figure 2.7:** *Average beam pattern of distributed beamforming with $N = 256$ sensor nodes*

Figure 2.7 shows the average beam pattern of distributed beamforming taken over 3000 realizations of random arrays with the same number of nodes $N = 256$ and the same disk radius

$R = 2\lambda$. It shows that the average beam pattern has a narrow main lobe in the desired direction without grating lobes. In [29], it is proved that the side lobes on average approach $\frac{1}{N}$ as the beam angle moves away from the desired direction.

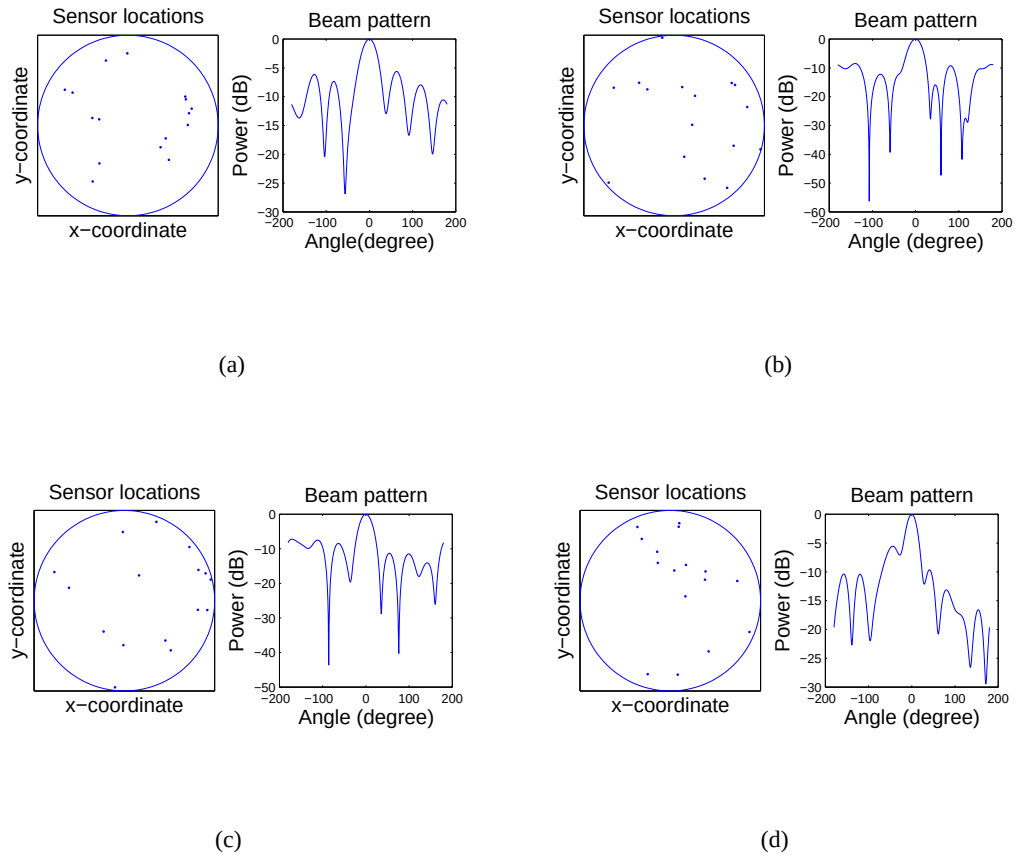
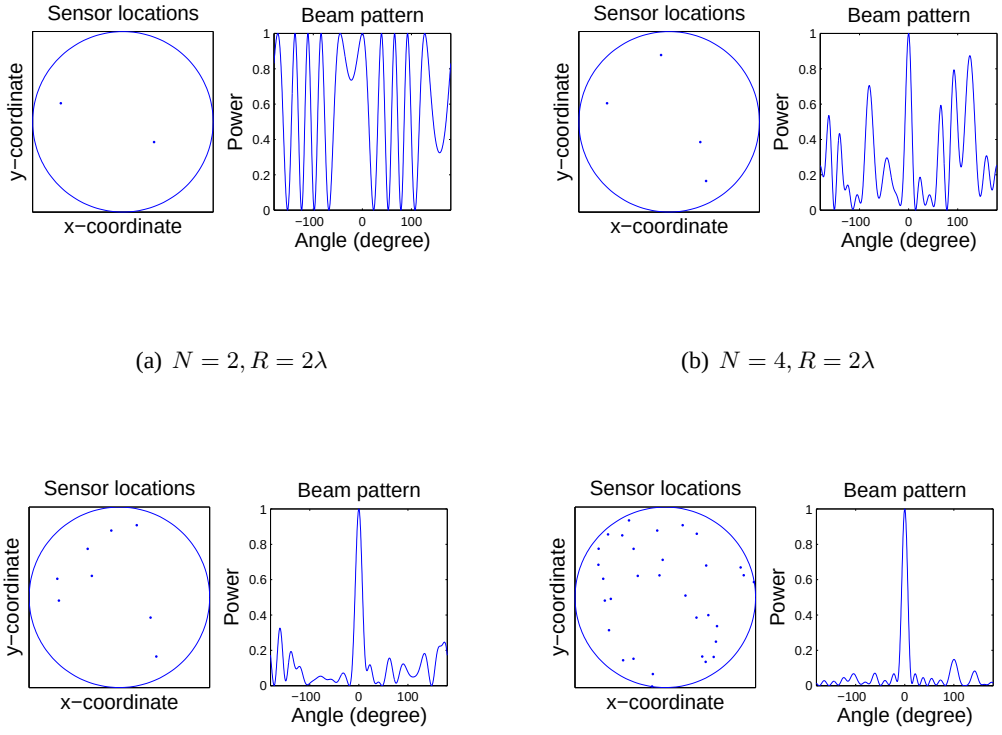



**Figure 2.8:** *Four instances of the beam pattern of distributed beamforming with different sensor node locations. The simulation parameters are otherwise the same with: number of nodes* $N = 16$*, disk radius* $R = \lambda$*.*

Although the average beam pattern of distributed beamforming provides encouraging results with a narrow main lobe in the desired direction and small side lobes in other directions, the beam pattern with one given realization of sensor array largely depends on the positions of sensor nodes. Four instances of the beam pattern with the same number of nodes $N = 16$ and the same disk radius $R = \lambda$ but different sensor node positions are shown in Figure 2.8. It shows that all the four instances can form a main lobe pointing at the desired direction but some of them simultaneously generate large side lobes in other directions, which may be considered

unacceptable in some application scenarios.

Sensor node positions have a big effect on the beam pattern, but it is hard to control them in practice as sensor nodes are usually randomly scattered in the sensing area. Instead, below we investigate the effects of the number of nodes and the node density on the beam pattern performance as these parameters are much easier to be controlled in practice. In order to view the effect of the number of nodes on the beam pattern, we fix the disk of radius as $R = 2\lambda$, add sensor nodes one by one into the disk, and view the trend of the beam pattern when increasing the number of nodes. The positions of sensor nodes are randomly chosen, following a uniform distribution. Figure 2.9 shows the positions of sensor nodes and the corresponding beam patterns in normalized power when $N = 2$, $4$, $8$, and $32$. It shows that the fluctuation range of side lobes decreases when the number of nodes increases. Therefore, large side lobes can be reduced and avoided by adding more nodes into the sensing area.




(a) $N = 2, R = 2\lambda$

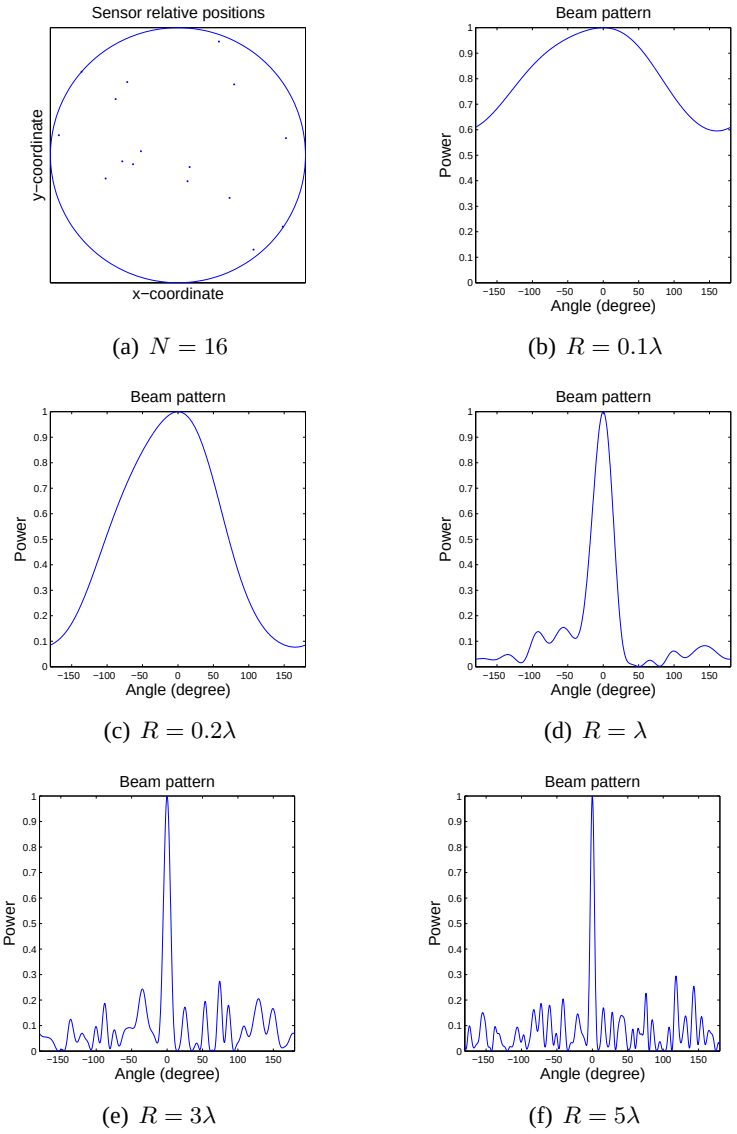
(b) $N = 4, R = 2\lambda$

(c) $N = 8, R = 2\lambda$

(d) $N = 32, R = 2\lambda$

**Figure 2.9:** *Change of the beam pattern when adding sensor nodes into a fixed disk.*

In order to view the effect of node density on the beam pattern, we keep the geometry of sensor array or the node relative positions unchanged while increasing the disk radius and view the change of the corresponding beam patterns in normalized power. This is the same as decreasing the node density. Figure 2.10 shows the geometry of the sensor array composed of $16$ sensor nodes and the beam patterns with disk radius $R = 0.1\lambda$, $R = 0.2\lambda$, $R = \lambda$, $R = 3\lambda$, $R = 5\lambda$.




(a) $N = 16$ (b) $R = 0.1\lambda$

(c) $R = 0.2\lambda$ (d) $R = \lambda$

(e) $R = 3\lambda$ (f) $R = 5\lambda$

**Figure 2.10:** *Change of the beam pattern when increasing the disk radius while keeping node relative positions unchanged.*

It shows that when the disk radius increases, the number of side lobes increases and the width of the main lobe decreases. Therefore, the directivity gain of distributed beamforming can

be improved by spreading the sensor nodes in a larger area. It is proved in [29], [30] that the directivity gain can asymptotically approach $N$ if the sensor nodes are located sparsely enough.

### 2.3.2 Analysis of received power

Besides the beam pattern performance, the effect of the number of nodes and phase errors on the received power at the destination using distributed beamforming was initially studied in [3]. Below we review some key results in [3]. We consider the model of $N$ sensor nodes performing distributed beamforming to a far-field destination. The individual carrier signals transmitted from each node arrive coherently at the destination with phase errors $\phi_i$, which are independently and uniformly distributed in the range $(-\phi_0, \phi_0)$. The channel coefficients, denoted as $h_i(t) \sim CN(0,1)$, are independent circularly symmetric complex normal random variables with zero mean and unit variance. We assume that the overall power transmitted by all the sensor nodes is fixed to 1, i.e. each node transmits with power $\frac{1}{N}$. Sensor nodes apply maximum ratio transmission (MRT) to achieve the maximum received power at the destination, i.e. each sensor node pre-amplifies the signal with power equal to the channel gain. The received signal at the destination can be expressed as:

$$r(t) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |h_i(t)|^2 e^{j\phi_i(t)} \cdot m(t) + n(t), \tag{2.12}$$

where $m(t)$ represents the common message signal and $n(t)$ is the additive noise. We define the received power as:

$$P_R = \frac{1}{N} \left| \sum_{i=1}^{N} |h_i(t)|^2 e^{j\phi_i(t)} \right|^2 \tag{2.13}$$

Figure 2.11 shows the average received power normalized to its maximum value, $\frac{E(P_R)}{N}$, with different number of nodes and phase error ranges. It shows that even with a large phase error range $\phi_0 = 72°$, a large beamforming gain is still available.

Figure 2.12 shows the histograms of $P_R$ to view the variance of the received power. There are no measurement units for the received power because we assume unit total transmit power by all nodes and unit channel gain from each node to the destination. The value on the x-axis in

each subfigure can be viewed as normalized to the total transmit power and the channel gains. It shows that when the number of nodes increases, the received power becomes more concentrated around its mean value.

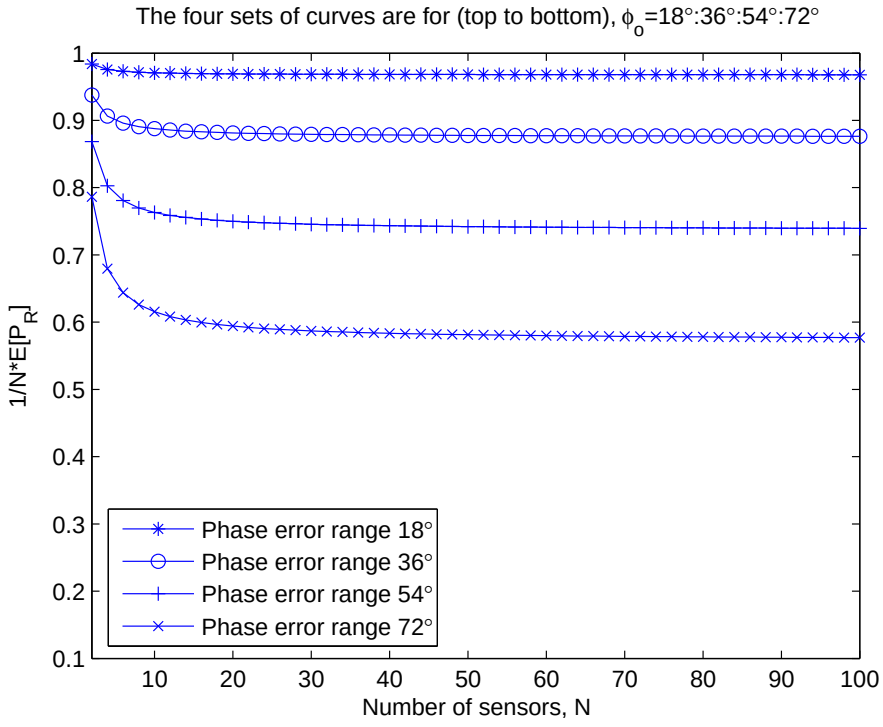


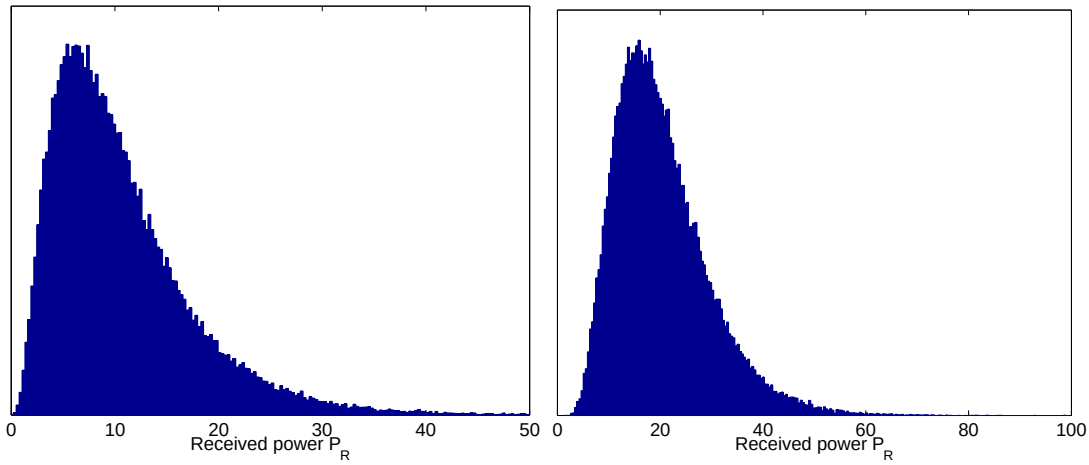

**Figure 2.11:** *$E[P_R]/N$ with different number of nodes and phase error ranges*

It is proved in [3] that the mean of $P_R$ grows linearly with $N$ while its standard deviation is proportional to $\sqrt{N}$. According to the central limit theorem (CLT), when $N$ is large enough,

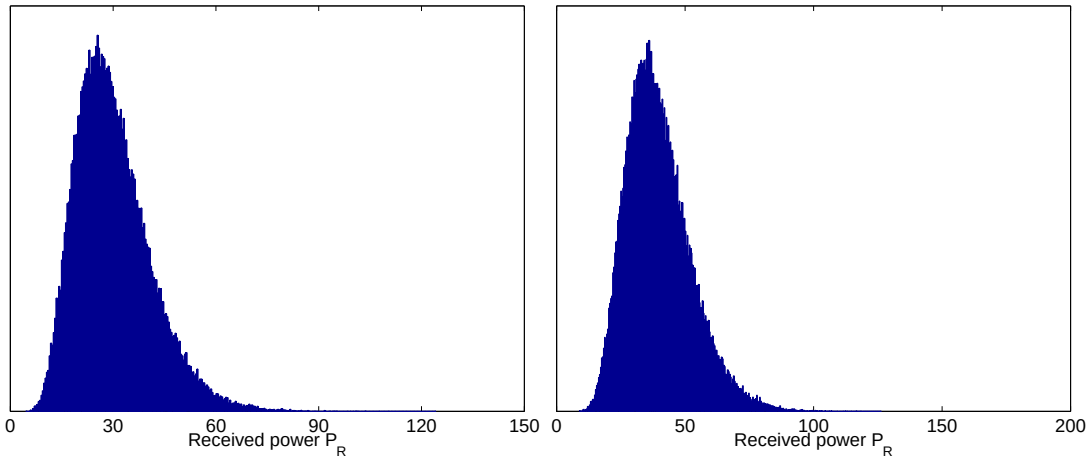
$$\begin{aligned} P_R &= \frac{1}{N}\left|\sum_{i=1}^{N}\left|h_i(t)\right|^2\cos\phi_i(t) + j\sum_{i=1}^{N}\left|h_i(t)\right|^2\sin\phi_i(t)\right|^2 \\ &\approx X_r^2 + X_i^2 \end{aligned} \tag{2.14}$$

where $X_r \sim N(m_r, \sigma_r^2)$, $X_i \sim N(0, \sigma_i^2)$, and $m_r$, $\sigma_r^2$, $\sigma_i^2$ are given by:

$$m_r = \sqrt{N}\mathrm{E}\big[\cos\phi_i(t)\big] \tag{2.15}$$




(a) $N = 10, \phi_0 = 18^\circ$ (b) $N = 20, \phi_0 = 18^\circ$




(c) $N = 30, \phi_0 = 18^\circ$ (d) $N = 40, \phi_0 = 18^\circ$

**Figure 2.12:** *Histograms of received power $P_R$ with the same phase error range $\phi_i(t) \sim (-18^\circ, 18^\circ)$ but different number of nodes $N = 10, 20, 30, 40$.*

$$\begin{aligned} \sigma_r^2 &= \mathrm{Var}\Big[\big|h_i(t)\big|^2 \cos\phi_i(t)\Big] \\ &= 2\mathrm{E}\Big[\cos^2\phi_i(t)\Big] - \Big(\mathrm{E}\big[\cos\phi_i(t)\big]\Big)^2 \end{aligned} \tag{2.16}$$

$$\begin{aligned} \sigma_i^2 &= \mathrm{Var}\Big[\big|h_i(t)\big|^2 \sin\phi_i(t)\Big] \\ &= 2\mathrm{E}\Big[\sin^2\phi_i(t)\Big] \end{aligned} \tag{2.17}$$

The mean and variance of $P_R$ can be computed as:

$$\mathrm{E}[P_R] = m_r^2 \tag{2.18}$$

$$\mathrm{Var}[P_R] = 4m_r^2\sigma_r^2 + 2\sigma_r^2 + 2\sigma_i^2 \tag{2.19}$$

Following equations (2.15)-(2.19), one can conclude that both the mean and variance of $P_R$ grow linearly with $N$.

## 2.4 Summary

In this Chapter, we provided some background of distributed transmit beamforming in the context of wireless sensor networks. The features of sensor networks make distributed beamforming a promising form of transmission for long-range communications as it can provide high SNR gain and reduce the energy requirement for each sensor node. Distributed beamforming is performed by a virtual antenna array composed of randomly located sensor nodes, each of which is equipped with a single antenna and an independent oscillator. By comparing it with conventional beamforming and reviewing the principles of beamforming techniques, we discussed the challenges of realizing distributed beamforming in practice, among which the most critical one is to achieve carrier phase alignment at the destination. We reviewed some key results and progresses on the performance analysis existed in the literature. The study of the

beam pattern shows that sensor nodes, acting as a distributed beamformer, may form a beam pattern with a narrow main lobe in the desired direction by carefully controlling some determinate factors. The study of the received power shows that acceptable beamforming gains can be obtained in distributed beamforming, even with moderately large phase errors. These results give us a fundamental understanding of the technique. In the next three chapters, we will contribute to both the theoretical and practical aspects: performance analysis and practical realization of distributed beamforming.

# Chapter 3
# BER Performance of Distributed Beamforming with Phase Errors

A key distinguishing feature of distributed beamforming, different from conventional beamforming, is the unavoidable phase errors. This is mainly because distributed beamforming is performed by a virtual antenna array composed of randomly located sensor nodes, each of which has an independent carrier oscillator, while conventional beamforming is implemented on a device with a centralized antenna array. In distributed beamforming, synchronization and coordination among transmitters are achieved wirelessly while in conventional beamforming, regularly placed antennas are connected and controlled by wires. Previous researchers have studied the effects of phase errors on the distributed beamforming performance from various aspects, showing that moderately large phase errors may be acceptable in achieving beamforming gains. To accurately predict the beamforming performance, the bit error ratio expression of distributed beamforming with phase errors is both theoretically and practically important but not available. In this Chapter, we investigate the bit error ratio performance for distributed beamforming and derive two distinct formulae to approximate the error probability performance over Rayleigh fading channels corresponding to small numbers of nodes and large numbers of nodes respectively. The effects of phase errors on the bit error ratio performance are examined for various numbers of nodes and different levels of total transmit power.

## 3.1 Introduction

As we mentioned in Chapter 2, unlike conventional beamforming, phase errors among the signals arriving at the receiver cannot be avoided in distributed beamforming. This may arise from the noise in individual carrier oscillators, sensor node position errors, channel estimation errors or timing synchronization errors. In [2], [29], [34], phase errors in distributed beamforming have been modeled to follow a uniform distribution, while in [3], the dominant component of the phase error has been modeled as a Gaussian variable. In [35], the authors show that

the phase errors follow a "exp-cosine" distribution in their proposed feedback algorithm for distributed beamforming. To measure the beamforming performance, the BER expression of distributed beamforming with phase errors is both theoretically and practically important.

The effects of phase errors on the beamforming performance have been investigated in several ways in the literature. It is shown in [36] that the received SNR only depends on the phase errors among the signals arriving at the receiver rather than the absolute phase values. When the phase errors change from $0^\circ$ to $180^\circ$, constructive interference at the receiver changes into destructive interference. In [3] and [37], the authors discussed a simple model of two transmitters, as illustrated in Figure 3.1, to show that moderately large phase errors at the receiver can still be used to achieve acceptable beamforming gains. As shown in Figure 3.2, signals from two equal power transmitters arriving at the receiver with a phase error $\delta$ result in a superposition signal with amplitude $|1 + e^{j\delta}| = 2\cos(\delta/2)$. Specifically, with a large phase error $\delta = 60^\circ$, the superposition signal at the receiver still has a gain of 1.732 in the amplitude. In [3], the authors also studied the phase error effect on the average beamforming gain with more transmitters and the variance of the received SNR with phase errors. In [38], the authors quantitatively studied the phase error effect on the average far-field beam pattern for random arrays. In [39], the authors studied the phase error effect on a cross-layer scheme for distributed beamforming, which can reduce the time required for information sharing among transmitters. From a more practical point of view, we investigate the BER performance of distributed beamforming with phase errors and in the presence of additive white Gaussian noise (AWGN).

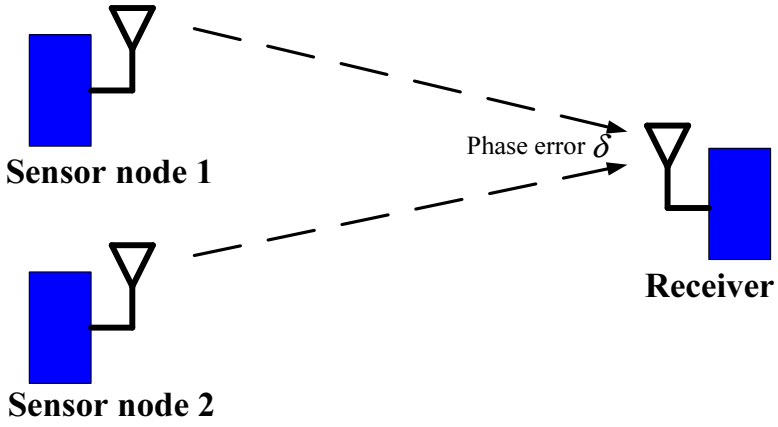



**Figure 3.1:** *A distributed beamforming system with two transmiters.*

The BER performance of beamforming has been well studied in the literature for various transmission techniques and over different channel models. Particularly, in a multiple-input single-

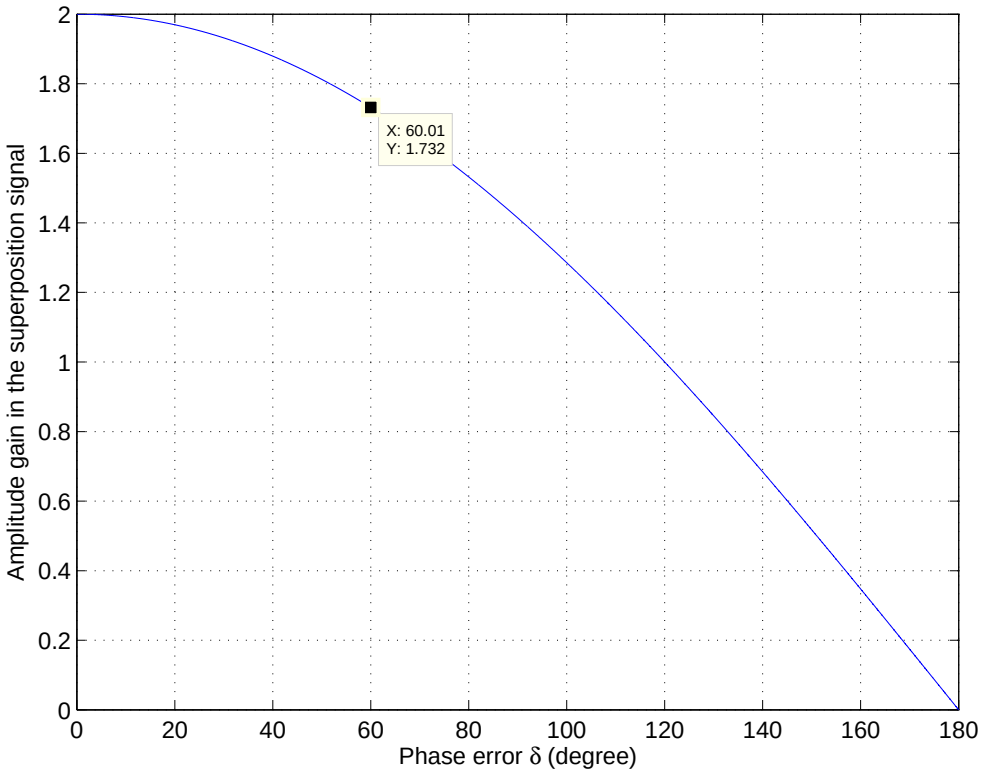



**Figure 3.2:** *Example of two equal power transmitters with a phase error $\delta$.*

output (MISO) system, diversity techniques such as MRT [40] and equal gain transmission (EGT) [41] are usually employed in transmit beamforming to obtain both diversity gain and array gain. Diversity gain represents that the signals can be transmitted through more than a single link between the transmitter and the receiver. The probability that a MISO communication system suffers from deep fading is much smaller than a single-input single-output system. Array gain represents the power gain obtained by using multiple antennas at the transmitter or the receiver. These techniques are analogous to the maximal ratio combining (MRC) and equal gain combining (EGC) used in a single-input multiple-output (SIMO) system [42]. With a constraint on the total transmit power, applying MRT at the transmitter side can maximize the received SNR at the receiver by weighting the signals transmitted from each channel in proportion to the channel gain. The BER performance of MRC in a SIMO system over different channel models, Rayleigh, Rician, and Nakagami fading, has been widely studied [43], [44], [45], [46]. However, applying MRT in a MISO system requires accurate CSI at the transmitter side, which may be obtained by using feedback and reciprocity schemes. The BER performance of MRT has been analyzed in [47], [48]. Unlike the classical transmit beamforming, distributed beamforming is performed in a distributed manner by a virtue antenna array composed of individual

sensor nodes, each of which is equipped with a single antenna. Although MRT provides the optimal performance in terms of power allocation, applying MRT to distributed beamforming requires abundant information exchange among the transmitters and the receiver due to the characteristics of distributed beamforming and is, therefore, difficult to achieve in practical implementations. More practically, sensor nodes may perform distributed beamforming with EGT by having all nodes transmit with equal power and adjust their phase settings to compensate for channel phase responses. EGT and EGC offers comparable performance and have a much simpler complexity and more modest requirements in practice compared to the MRT and MRC. The BER performance of EGC over different channel models has been analyzed in [49], [50], [51].

In this chapter, we investigate distributed beamforming with phase errors and focus on EGT. We derive the expression for BER as a function of the number of sensor nodes, phase errors and total transmit power for both small number of nodes and large number of nodes. The derivation for small number of nodes, denoted as Method 1, is based on expectation adjustment and variance compensation, and the BER expression for small number of nodes takes the form of a single dimensional integral solved by Hermite integration method. The derivation for large number of nodes, denoted as Method 2, is based on the CLT and moment matching approach, and the BER expression for large number of nodes is much simpler and computationally efficient compared to the one for small number of nodes. The accuracy of both methods is well examined by simulations where analytical results have a good prediction on the BER performance for various numbers of nodes and different levels of total transmit power. These analytical results can be extended to different modulation schemes and different phase error distributions. Practical issues, such as algorithm design for frequency and phase synchronization to reduce phase errors and assure phase alignment at the receiver, will be addressed in the following Chapter 4 and Chapter 5.

## 3.2 System model

We consider a wireless sensor network composed of $N$ sensor nodes collaboratively beamforming a narrowband message signal $m(t)$ to a distant receiver. This is performed in a distributed manner by each sensor node modulating $m(t)$ at the same carrier frequency, which are generated by independent local oscillators. Each sensor node pre-compensates the phase response of its channel to the receiver by adjusting its initial phase settings [3] in order to ensure phase

alignment at the receiver, as illustrated in Figure 3.3.

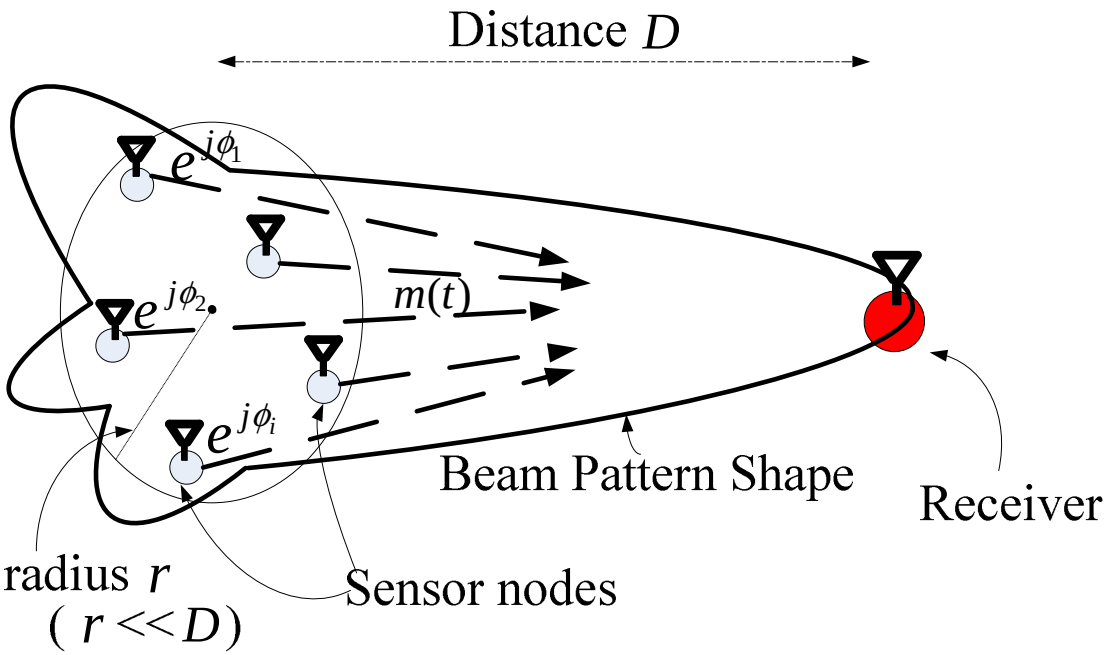



**Figure 3.3:** *System model for distributed beamforming.*

We assume that each sensor node and the receiver are equipped with one single ideal isotropic antenna. All sensor nodes are synchronized so that they can transmit at the same carrier frequency. This is a reasonable assumption which has been widely adopted in the literature [29], [35], [4]. Since the sensor nodes are located quite close to each other compared to their distance to the destination, frequency synchronization among the sensor nodes can be achieved by either employing a master-slave architecture [2], [3] or using a reference signal from the destination [37], [52]. Ideally signals transmitted from each sensor node will be added coherently at the receiver but phase errors cannot be avoided as discussed above. Considering a large number of sensor nodes, full CSI may be hard to obtain in practice. A study on the distributed beamforming performance with quantized CSI available at transmitters has been presented in [53], [54]. Techniques have been designed to pre-compensate the channel phase response to achieve phase alignment in [37], [4]. Thus, lack of full CSI and power limitation on the sensor nodes make MRT techniques unrealistic. Instead, more practically, we assume each sensor node transmits with equal power and applies channel phase compensation at the transmitter side. In order to reveal the fact that beamforming gain grows with the number of nodes $N$, we assume the overall power transmitted by all the nodes is fixed as $P$, where each node actually transmits with a power of $\frac{P}{N}$. This then permits us to model the BER improve-

ment with distributed beamforming gain. The complex baseband model of the received signal is given by:

$$r(t) = \sum_{i=1}^{N} |h_i(t)| e^{j\phi_i(t)} \sqrt{\frac{P}{N}} m(t) + n(t), \tag{3.1}$$

where $h_i(t)$ is the channel gain for sensor node $i$, $\phi_i(t)$ is the cumulative phase error of the carrier signal at the receiver for sensor node $i$, $n(t) \sim CN(0, \sigma_n^2)$ is AWGN. For simplicity, we assume all phase errors $\phi_i(t)$ are independently and uniformly distributed, bounded by $(-\phi_0, \phi_0)$, across time and across nodes, which is a common assumption adopted in previously reported investigations [2], [29], [34], [3]. Our BER analysis can be easily applied to other situations with a different phase error distribution to that discussed above in Section 3.1. We assume the signals experience slow fading channels, and the channel coefficients are independent circularly symmetric complex Gaussian distributed, denoted as $h_i(t) \sim CN(0, 2\sigma_c^2)$, which corresponds to non-line of sight channels.

Before the mathematical analysis and BER derivation, we want to point out that one of the major differences between the model in this chapter and other models for BER analysis in literature is the phase errors. The phase errors in distributed beamforming are bounded and uniformly distributed within $(-\phi_0, \phi_0)$ rather than $[0, 2\pi)$, where $\phi_0$ is usually expected to be less than $60^\circ$ in practice in order to achieve a reasonable beamforming gain [3]. As a result, the independent probability density function (pdf) of the magnitude gain or the power gain cannot be extracted easily from the joint pdf associated to the real and imaginary parts of the received signal, which is a key obstacle when deriving the BER as the BER performance mainly depends on the received SNR. This will be further discussed and justified by mathematical analysis in the following Section 3.4. The effects of phase errors on the received signal have been reported in [3] as a reduction in SNR gain and a fluctuation in the phase of the received signal. As illustrated in Figure 3.4, the received signal can be viewed as a sum of random vectors, whose magnitudes are Rayleigh distributed and the phases exhibit a bounded uniform distribution. We assume a coherent receiver which has the ability to retrieve the overall phase of the received signal. Thus, the effects of phase errors must be analyzed to determine their effect on reducing the SNR gain.

In the following derivation, we focus on the scenario of Rayleigh fading channels when describing our analysis methods. The BER in the scenario of static channels can be regarded as

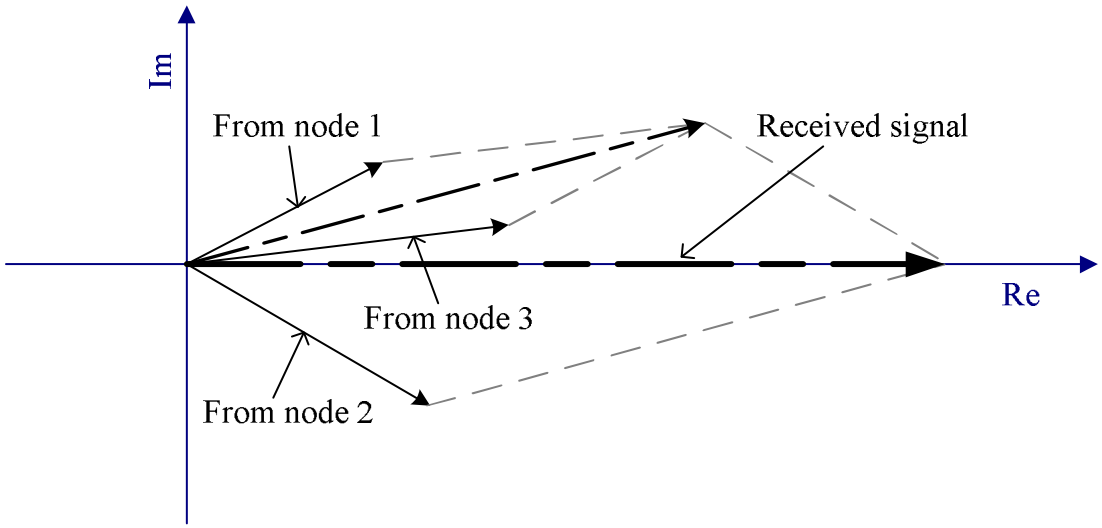



**Figure 3.4:** *The received signal is a sum of random vectors whose magnitudes are Rayleigh distributed and phase angles are bounded and uniformly distributed.*

a special case and can be easily derived in the same way by using our methods. After matched filter detection and analog-to-digital conversion, the decision variable for binary phase shift keying (BPSK) modulation can be expressed as:

$$\begin{aligned} r_D &= \pm\sqrt{\frac{P}{N}}\left|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\right| + \widehat{n} \\ &= s + \widehat{n}, \end{aligned} \tag{3.2}$$

and the corresponding decision rule is:

$$m(t) = \begin{cases} 1 & r_D > 0 \\ 0 & r_D < 0 \end{cases}, \tag{3.3}$$

where $\widehat{n}$ represents the noise, $n(t)$, projected onto the received signal vector. We focus on BPSK signalling as an example because of its simplicity. Our analysis methods can be easily extended to other modulation schemes as discussed at the end of Section 3.4.

## 3.3 BER for small number of nodes - Method 1

The BER for BPSK modulation over a fixed channel in the presence of AWGN is given in Chapter 5 in [55] as:

$$P_e(\gamma) = Q(\sqrt{2\gamma}), \tag{3.4}$$

where $\gamma$ is the received signal-to-noise ratio per bit and $Q(.)$ is the $Q$-function defined as $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$ ($x \geq 0$). When the channel gain is random, the average BER for BPSK over all values of $\gamma$ is given in Chapter 14 in [55] as:

$$P_e = \int_0^\infty P_e(\gamma) p(\gamma) d\gamma, \tag{3.5}$$

where:

$$\gamma = \frac{\frac{P}{N} \left| \sum_{i=1}^{N} \left| h_i \right| e^{j\phi_i} \right|^2}{\sigma_n^2}, \tag{3.6}$$

in our system model described above. The function $p(\gamma)$ denotes the pdf of $\gamma$. Due to the effects of phase errors, the distribution of $\gamma$ is unknown and the pdf expression of $\gamma$ is difficult to evaluate.

However, the probability of error for EGC at a multiple antenna receiver with $L$ independent receive branches over Rayleigh channels has been studied in [49], [50]. The decision variable for coherent BPSK in [49], [50] is expressed as:

$$r_d = \pm(x_1 + x_2 + \cdots + x_L) + \sum_{i=1}^{L} n_i, \tag{3.7}$$

where $x_i$ is the amplitude of the received signal at the output of $i$th branch with a Rayleigh distribution. The scalar $n_i$ is the complex baseband Gaussian noise at the output of $i$th branch.

Although [49], [50] are studying equal gain diversity receivers and their system models are different from ours, as shown above, the decision variable in (3.7) is identical to (3.2) when $L = N$ if we neglect the phase errors in our model and modify the noise component. The

noise in (3.7) comprises $L$-branch superimposed noise while in (3.2) there is only one AWGN component. By studying [49], [50] and modifying the coefficients of the noise, we can thus derive the BER expression for distributed beamforming *without* phase errors over Rayleigh channels as:

$$P_e \approx \frac{1}{2} - \frac{1}{\pi} \sum_{m=1}^{M} \omega_m G(z_m, \Omega, \sigma_n^2, N), \tag{3.8}$$

where:

$$G(z, \Omega, \sigma_n^2, N) = \mathrm{Im} \left\{ \left[ {}_1F_1(-\frac{1}{2}; \frac{1}{2}; \frac{\Omega z^2}{\sigma_n^2 + N\Omega}) + jz \sqrt{\frac{\pi\Omega}{\sigma_n^2 + N\Omega}} \right]^N \right\} z^{-1}, \tag{3.9}$$

and:

$$\Omega = \mathrm{E}\left[ \left( \sqrt{\frac{P}{N}} |h_i| \right)^2 \right] = \frac{2\sigma_c^2 P}{N} \tag{3.10}$$

is the average energy of a Rayleigh distributed variable in (3.2) and in the case of no phase errors, $\phi_i = 0$ in that equation. The function $\mathrm{E}[x]$ denotes the expectation of $x$ and $2M$ is the order of Hermite polynomials. The expression of the above confluent hypergeometric function, ${}_1F_1(a; b; x)$, is given by [56]:

$${}_1F_1(a; b; x) = \sum_{n=0}^{\infty} \frac{(a)_n x^n}{(b)_n n!}, \tag{3.11}$$

where $(a)_n = \frac{\Gamma(a+n)}{\Gamma(a)}$ and $(b)_n = \frac{\Gamma(b+n)}{\Gamma(b)}$.

Equation (3.8) refers to Hermite integration explained on page 890 in [1], and the values for $\omega_m$ and $z_m$ are given on page 924 in [1]. The validity of using the Hermite method of integration to compute the error probability for EGC has been fully justified in [50]. Equation (3.8) becomes more accurate when $M$ tends to infinity. However, it is shown in [50] that $M = 10$ is sufficient to ensure acceptable accuracy. For calculation convenience, the values of $\omega_m$ and $z_m$ with $M = 10$ are given in Table 3.1.

If there are phase errors, i.e. $\phi_0 \neq 0$, the power of the signal part, $s$, in (3.2) is reduced by phase errors, and the expectation of the received SNR becomes smaller than the case without phase

| $m$ | 1 | 2 | 3 |
|---|---|---|---|
| $\omega_m$ | $0.4622436696006$ | $0.2866755053628$ | $0.1090172060200$ |
| $z_m$ | 0.2453407083009 | 0.7374737285454 | 1.2340762153953 |
| $m$ | 4 | 5 | 6 |
| $\omega_m$ | $0.02481052088746$ | $3.243773342238 \times 10^{-3}$ | $2.283386360163 \times 10^{-4}$ |
| $z_m$ | 1.7385377121166 | 2.2549740020893 | 2.7888060584281 |
| $m$ | 7 | 8 | 9 |
| $\omega_m$ | $7.802556478532 \times 10^{-6}$ | $1.086069370769 \times 10^{-7}$ | $4.399340992273 \times 10^{-10}$ |
| $z_m$ | 3.3478545673832 | 3.9447640401156 | 4.6036824495507 |
| $m$ | 10 | | |
| $\omega_m$ | $2.229393645534 \times 10^{-13}$ | | |
| $z_m$ | 5.3874808900112 | | |

**Table 3.1:** *Values of $\omega_m$ and $z_m$ with $M = 10$ in Hermite integration [1]*

errors. In order to incorporate the effects of phase errors, we define a factor $\eta$. We multiply every single Rayleigh variable, $\sqrt{\frac{P}{N}}|h_i|$, with $\eta$ to make the expectation of the received SNR equal:

$$\mathrm{E}\Big[\frac{P}{N}\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|^2\Big] = \mathrm{E}\Big[\frac{P}{N}\Big(\sum_{i=1}^{N}\eta|h_i|\Big)^2\Big]. \tag{3.12}$$

Rearranging this equation, we have:

$$\eta^2 = \mathrm{E}\Big[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|^2\Big] \Big/ \mathrm{E}\Big[\Big(\sum_{i=1}^{N}|h_i|\Big)^2\Big]. \tag{3.13}$$

The expression of $\eta^2$ in terms of the number of nodes $N$ and the phase error range $\phi_0$ is derived in Appendix A.1. The average power of an adjusted Rayleigh variable, $\eta\sqrt{\frac{P}{N}}|h_i|$, becomes $\Omega' = \mathrm{E}\left[\left(\eta\sqrt{\frac{P}{N}}|h_i|\right)^2\right] = \eta^2\Omega$. We use $\Omega'$ to substitute for $\Omega$ in (3.9). The purpose of this is to use the distribution of a sum of $N$ Rayleigh variables to approximate the distribution of the signal, $s$, in (3.2) while keeping the expectation of the received SNR per bit $\mathrm{E}[\gamma]$ to be the same.

The expectation of $\gamma$ has been adjusted by introducing the factor $\eta$. There is still a difference

between the actual variance of the received signal and the variance after the expectation adjustment. Thus, we further define a variable, $\sigma_d^2$, to compensate for the residual variance between the two:

$$\sigma_d^2 = \frac{P}{N}\left(\text{Var}\left[\left|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\right|\right] - \text{Var}\left[\eta\Big(\sum_{i=1}^{N}|h_i|\Big)\right]\right), \tag{3.14}$$

where Var$[x]$ denotes the variance of $x$. The expression of $\sigma_d^2$ in terms of the number of nodes $N$ and the phase error range $\phi_0$ is derived in Appendix A.1. We treat this residual variance as a contribution to the receiver noise, and compute the total noise power as:

$$\widetilde{\sigma}_n^2 = \sigma_n^2 + \sigma_d^2. \tag{3.15}$$

By substituting $\Omega'$ for $\Omega$, $\widetilde{\sigma}_n^2$ for $\sigma_n^2$ into (3.9), the final BER expression for EGT in distributed beamforming with phase errors over Rayleigh channels is given by (3.8), while the function for computation becomes $G(z, \Omega', \widetilde{\sigma}_n^2, N)$. We use equation (3.8) and $G(z, \Omega', \widetilde{\sigma}_n^2, N)$ to compute the BER in the simulations of Section 3.5, and this is denoted as method 1. Method 1 is valid for any number of nodes, but it is proposed here to use method 1 only for small number of nodes due to its high computational complexity for large $N$. This will be justified and further explained in Section 3.5.

## 3.4 BER for large number of nodes - Method 2

In (3.6) we see that the distribution of $\gamma$ mainly depends on the distribution of $\left|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\right|^2$. Therefore, for simplicity, we define the concept of an equivalent channel, $H$, as:

$$H = \sum_{i=1}^{N}|h_i|e^{j\phi_i}, \tag{3.16}$$

and the system model in (3.1) becomes:

$$r(t) = \sqrt{\frac{P}{N}}H(t)m(t) + n(t). \tag{3.17}$$

Based on the CLT, with a large number of nodes $N$, and the independent identically distributed (i.i.d.) random variables, $h_i$, which are independent from the i.i.d. random variables $\phi_i$, the key element which determines the error probability can be expressed as:

$$\begin{aligned} \left|H\right|^2 &= \left|\sum_{i=1}^{N} \left|h_i\right| e^{j\phi_i}\right|^2 = \left|\sum_{i=1}^{N} \left|h_i\right| \cos\phi_i + j\sum_{i=1}^{N} \left|h_i\right| \sin\phi_i\right|^2 \\ &= |a+jb|^2 = a^2+b^2, \end{aligned} \tag{3.18}$$

where $a$ and $b$ are defined as:

$$a = \sum_{i=1}^{N} \left|h_i\right| \cos\phi_i \sim N(\mu_a, \sigma_a^2), \qquad b = \sum_{i=1}^{N} \left|h_i\right| \sin\phi_i \sim N(\mu_b, \sigma_b^2). \tag{3.19}$$

A similar analysis of the beamforming gain using the CLT has been presented in [3]. Since the channel coefficients $h_i \sim CN(0, 2\sigma_c^2)$, and the phase errors $\phi_i \sim \text{uniform}(-\phi_0, \phi_0)$, the expectations and variances of $a$ and $b$ can be obtained as follows:

$$\begin{aligned} \mu_a &= N \cdot \mathrm{E}\Big[\left|h_i\right| \cos\phi_i\Big] = N \cdot \mathrm{E}\Big[\left|h_i\right|\Big] \cdot \mathrm{E}\Big[\cos\phi_i\Big] \\ &= N \cdot (2\sigma_c^2)^{\frac{1}{2}}\Gamma(\frac{3}{2}) \cdot \frac{\sin\phi_0}{\phi_0} = \frac{\sqrt{2\pi}N\sigma_c \sin\phi_0}{2\phi_0}, \end{aligned} \tag{3.20}$$

$$\mu_b = 0, \tag{3.21}$$

$$\begin{aligned} \sigma_a^2 &= N\left(\mathrm{E}\Big[\Big(\left|h_i\right|\cos\phi_i\Big)^2\Big] - \Big(\mathrm{E}\Big[\left|h_i\right|\cos\phi_i\Big]\Big)^2\right) \\ &= N\left(\mathrm{E}\Big[\left|h_i\right|^2\Big] \cdot \mathrm{E}\Big[\cos^2\phi_i\Big] - \Big(\mathrm{E}\Big[\left|h_i\right|\cos\phi_i\Big]\Big)^2\right) \\ &= N\left(2\sigma_c^2\Gamma(2) \cdot (\frac{1}{2} + \frac{\sin 2\phi_0}{4\phi_0}) - (\frac{\sqrt{2\pi}\sigma_c \sin\phi_0}{2\phi_0})^2\right) \\ &= N\sigma_c^2\left(1 + \frac{\sin 2\phi_0}{2\phi_0} - \frac{\pi}{2}(\frac{\sin\phi_0}{\phi_0})^2\right), \end{aligned} \tag{3.22}$$

$$\begin{aligned}
\sigma_b^2 &= N\left(\mathrm{E}\left[\left(|h_i|\sin\phi_i\right)^2\right] - \left(\mathrm{E}\left[|h_i|\sin\phi_i\right]\right)^2\right) \\
&= N\left(\mathrm{E}\left[|h_i|^2\right]\cdot\mathrm{E}\left[\sin^2\phi_i\right] - \left(\mathrm{E}\left[|h_i|\sin\phi_i\right]\right)^2\right) \\
&= N\left(2\sigma_c^2\Gamma(2)\cdot(\frac{1}{2} - \frac{\sin 2\phi_0}{4\phi_0}) - 0\right) \\
&= N\sigma_c^2\left(1 - \frac{\sin 2\phi_0}{2\phi_0}\right). \qquad (3.23)
\end{aligned}$$

From (3.22) and (3.23) we see, for the equivalent channel, $H$, with most values of $\phi_0$ (i.e. $\phi_0 \neq 45°$), the variance of the real part $\sigma_a^2$ and the imaginary part $\sigma_b^2$ are not equal, which means the expression of the pdf of $|H|^2$ is difficult to compute. However, if we make the approximation that the variance of the real part and the variance of the imaginary part of $H$ are equal, the magnitude gain of the equivalent channel, $|H|$, follows a Rician distribution, and the channel gain, $|H|^2$, has a non-central chi-square distribution with 2 degrees of freedom. Therefore, we propose three ways, namely Rician Approx 1, Rician Approx 2 and Rician Approx 3, to use a Rician distribution to approximate the distribution of $|H|$.

**Rician Approx 1**: we generate a Rician distribution whose second and forth moments equal to $\mathrm{E}\left[|H|^2\right]$ and $\mathrm{E}\left[|H|^4\right]$. The square of the Rician distribution is a non-central chi-square distribution where the noncentrality parameter, $\lambda^2$, and the variance, $\sigma^2$, satisfy [55]:

$$\mathrm{E}\left[|H|^2\right] = 2\sigma^2 + \lambda^2, \qquad (3.24)$$

and:

$$\mathrm{Var}\left[|H|^2\right] = 4\sigma^4 + 4\sigma^2\lambda^2. \qquad (3.25)$$

Rearranging equations (3.24) and (3.25), we can derive the expressions for $\lambda^2$ and $\sigma^2$ as:

$$\lambda^2 = \sqrt{2\mathscr{A}^2 - \mathscr{B}}, \qquad (3.26)$$

$$\sigma^2 = \frac{\mathscr{A} - \sqrt{2\mathscr{A}^2 - \mathscr{B}}}{2}, \tag{3.27}$$

where $\mathscr{A} = \mathrm{E}\left[\left|H\right|^2\right]$ is the second moment of $|H|$, $\mathscr{B} = \mathrm{E}\left[\left|H\right|^4\right]$ is the fourth moment of $|H|$. The expressions for $\mathscr{A}$, $\mathscr{B}$ in terms of $N$ and $\phi_0$ are derived in Appendix A.2 by applying the CLT. The relationship of $\lambda^2$ and $\sigma^2$ to the Ricean shape parameter $K$, which represents the ratio between the power in the direct path and the power in other paths in Ricean fading, is $K = \frac{\lambda^2}{2\sigma^2}$.

A similar technique using a central chi-square distribution to approximate the distribution of a sum of independent chi-square distributed random variables through first and second order moment matching was introduced in G. E. P. Box's work [57], which is frequently an accurate approximation. Rician Approx 1, an approach of moment matching to a non-central chi-square distribution (the distribution of $|H|^2$), is inspired by [57] and [58] although there are technical differences in how the moment matching is implemented.

**Rician Approx 2**: since both $a$, $b$ in (3.18) are Gaussian random variables, we use the results derived in equations (3.20), (3.21), (3.22), and (3.23) to generate a Rician distribution. The parameters of the corresponding non-central chi-square distribution are computed as:

$$\lambda^2 = \mu_a^2 + \mu_b^2, \tag{3.28}$$

$$\sigma^2 = \max(\sigma_a^2, \sigma_b^2). \tag{3.29}$$

**Rician Approx 3**: The parameters of the non-central chi-square distribution are also obtained directly from (3.20), (3.21), (3.22), (3.23) and are computed as:

$$\lambda^2 = \mu_a^2 + \mu_b^2, \tag{3.30}$$

$$\sigma^2 = \frac{\sigma_a^2 + \sigma_b^2}{2}. \tag{3.31}$$

The BER for BPSK signalling in a Rician fading channel has been studied in [59], permitting

the closed-form expression for the BER of our model to be easily obtained as:

$$P_e = Q_1(u, w) - \frac{1}{2}\left(1 + \sqrt{\frac{d}{1+d}}\right)\exp\left(-\frac{u^2+w^2}{2}\right) I_0(uw), \tag{3.32}$$

where:

$$d = \frac{2\sigma^2 P}{\sigma_n^2 N}, \quad u = \sqrt{\frac{\lambda^2}{2\sigma^2}\cdot\frac{1+2d-2\sqrt{d(1+d)}}{2(1+d)}}, \quad w = \sqrt{\frac{\lambda^2}{2\sigma^2}\cdot\frac{1+2d+2\sqrt{d(1+d)}}{2(1+d)}}, \tag{3.33}$$

and $I_0(x)$ is the zeroth-order-modified Bessel function of the first kind, defined as [55]:

$$I_0(x) = \sum_{\kappa=0}^{\infty}\frac{(x/2)^{2\kappa}}{\kappa!\Gamma(\kappa+1)}, \qquad x \geq 0. \tag{3.34}$$

The function $Q_1(x, y)$ is the Marcum $Q$-function, defined as [55]:

$$Q_1(x, y) = \int_y^{\infty} z\cdot\exp\left(-\frac{z^2+x^2}{2}\right) I_0(xz)dz. \tag{3.35}$$

An approximation of $I_0(x)$ is given by [60] in Chapter 6:

$$I_0(x) \approx \frac{1}{\sqrt{2\pi x}}\exp(x), \qquad x \gg 0, \tag{3.36}$$

and after manipulation, (3.32) can be simplified to:

$$P_e \approx Q_1(u, w) - \frac{1}{2\sqrt{2\pi uw}}\left(1 + \sqrt{\frac{d}{1+d}}\right)\exp\left(-\frac{(u-w)^2}{2}\right), \qquad uw \gg 0. \tag{3.37}$$

To the best of the authors' knowledge, (3.37) is a new result which simplifies the BER expression.

By substituting the expressions of $\lambda^2$ and $\sigma^2$, either (3.26), (3.27) from Rician Approx 1, or (3.28), (3.29) from Rician Approx 2, or (3.30), (3.31) from Rician Approx 3, into (3.33) and

(3.37), we can obtain the final BER expression for EGT in distributed beamforming with phase errors for large number of nodes. The accuracy of the three methods, Rician Approx 1, Rician Approx 2 and Rician Approx 3, on predicting the BER performance are compared in Figure 3.5 in the following Section 3.5. It shows that the method Rician Approx 1 outperforms the other two regardless of the number of nodes and the phase error range. Therefore, we adopt Rician Approx 1 to generate a Rician distribution to approximate the distribution of $|H|$, use it to predict the BER performance for a large number of nodes, and we define this as method 2 in the following simulations.

Method 2 can be extended to analyze BER with MRT for distributed beamforming in a similar way. Although perfect CSI at transmitters is difficult to obtain and MRT is not feasible for distributed beamforming in practice, schemes may be designed to allocate more power to transmitters with better quality links through limited information exchange. For example, in [22] the authors proposed a power allocation scheme for distributed beamforming using a common power-scaling factor periodically broadcasted from the destination to the sensor nodes. In the study of such schemes for power allocation among sensor nodes, the BER with MRT may be considered as a lower bound on the BER performance to evaluate the fixed power transmission techniques. The system model described in Section 3.2 with MRT is given by:

$$r(t) = \sum_{i=1}^{N} |h_i(t)|^2 e^{j\phi_i(t)} \sqrt{\frac{P}{N}} m(t) + n(t), \tag{3.38}$$

and the corresponding equivalent channel $H_M$, which determines the received SNR, can be expressed as:

$$H_M = \sum_{i=1}^{N} \left|h_i\right|^2 e^{j\phi_i}. \tag{3.39}$$

Following the derivations for the case of EGT presented in Appendix A.2, we can easily derive the second and the fourth moment of $H_M$ for MRT. When $h_i \sim CN(0,1)$, the second moment of $H_M$ is expressed as:

$$\mathscr{C} = 2N + N(N-1)\left(\frac{\sin\phi_0}{\phi_0}\right)^2, \tag{3.40}$$

and the fourth moment of $H_M$ is expressed as:

$$\begin{aligned}\mathscr{D} &= 8N(N+2)+8N(N+1)(N-1)\left(\frac{\sin\phi_0}{\phi_0}\right)^2+4N(N-1)\left(\frac{\sin 2\phi_0}{2\phi_0}\right)^2 \\ &\quad +4N(N-1)(N-2)\frac{\sin 2\phi_0}{2\phi_0}\left(\frac{\sin\phi_0}{\phi_0}\right)^2+N(N-1)(N-2)(N-3)\left(\frac{\sin\phi_0}{\phi_0}\right)^4 \quad (3.41)\end{aligned}$$

Substituting $\mathscr{C}$ for $\mathscr{A}$, $\mathscr{D}$ for $\mathscr{B}$ into (3.26), (3.27), then taking (3.26), (3.27) into (3.33) and (3.37), we can obtain the BER expression for BPSK signalling with MRT in distributed beamforming with phase errors.

The BER expressions derived above may be extended to other modulation schemes by studying [61] and [62]. For example, the average BER of $M$-ary pulse-amplitude-modulated (PAM) signals over a fixed channel in the presence of AWGN can be expressed in the form of (see eq. 9 in [62] and eq. 54 in [63]):

$$P_{e\mathrm{PAM}}(\gamma)=\sum_{m=1}^{M}a_m\mathrm{erfc}(\sqrt{b_m\gamma}) \quad (3.42)$$

where $a_m$ and $b_m$ are coefficients depending on the constellation distance of each bit of each symbol. The average BER of $M$-ary PAM in our model over all values of $\gamma$ in (3.5) becomes:

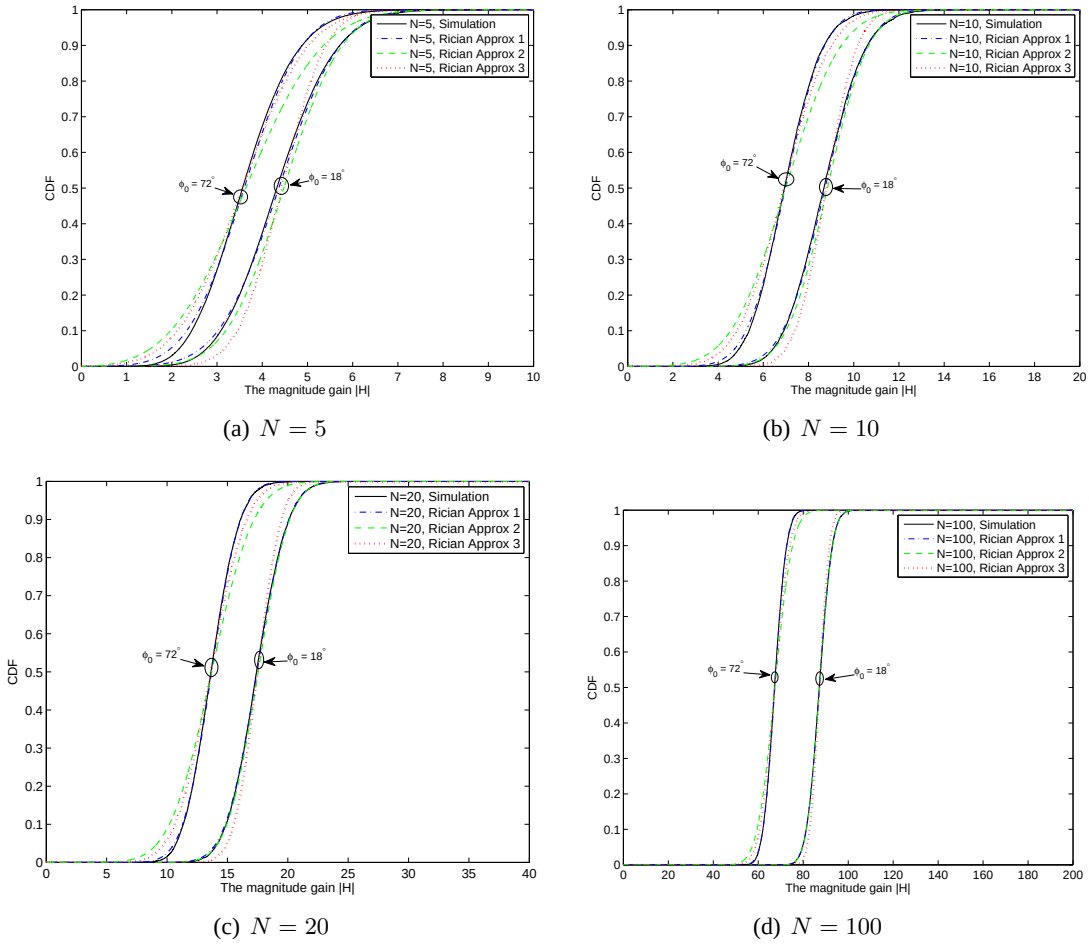
$$P_{e\mathrm{PAM}}=\int_0^\infty\sum_{m=1}^{M}a_m\mathrm{erfc}(\sqrt{b_m\gamma})p(\gamma)d\gamma \quad (3.43)$$

By switching the sum function and the integration, and applying (3.8) or (3.37), one can obtain the final expression of the BER for $M$-ary PAM with distributed beamforming with phase errors. These results can easily be extended to quadrature-amplitude-modulated (QAM) constellations as well.

## 3.5 Simulation results

In this section, we present some simulation results in accordance with our previous assumptions for distributed beamforming with phase errors over Rayleigh fading channels, and compare

them with the analytical results given by mathematical expressions derived in Section 3.3 and Section 3.4.




(a) $N = 5$ (b) $N = 10$

(c) $N = 20$ (d) $N = 100$

**Figure 3.5:** *Cumulative distribution function of $|H|$ with $N = 5, 10, 20, 100$ distributed sensor nodes, phase errors constrained within the range $\phi_0 = 18^\circ, 72^\circ$.*

In Section 3.4, we proposed three ways to use a Rician distribution to approximate the distribution of $|H|$. Figure 3.5 shows a comparison of the cumulative distribution function (CDF) of $|H|$ obtained from the three ways and via simulation. It is shown that the method Rician Approx 1 always performs better than Rician Approx 2 and Rician Approx 3 regardless of the number of nodes $N$ and the phase error range $\phi_0$. The CDF obtained from Rician Approx 1 can give a close match to the CDF obtained from simulation, especially for large $N$. Therefore, we adopt Rician Approx 1 as a solution to predict BER for a large number of nodes in the following simulations, and denote it as method 2. It is also shown in Figure 3.5 that all three methods become more accurate when $N$ increases. This is because, with a large $N$, the mean of $|H|^2$

is proportional to $N^2$ while its standard deviation is proportional to $N$ [3]. When $N$ increases, the magnitude gain $|H|$ becomes more concentrated around its mean value, as illustrated in the figures, and the variance approximation plays a less important role.

In the following, we examine the BER performance of EGT in distributed beamforming with phase errors for various numbers of nodes. We set the channel coefficients as $h_i(t) \sim CN(0,1)$ and the AWGN noise as $n(t) \sim CN(0,1)$. Given these assumptions, the value of the total transmit power $P$ in the figures can be viewed as normalized to the noise power at the receiver, or it can be viewed as the ratio of the transmit power over the noise power at the receiver. Therefore, there are no measurement units for $P$ in the following figures, i.e. $P$ is dimensionless. If the noise power is measured in watt, then the measurement unit of the transmit power $P$ is watt as well. For example, $P = 1$ implies that the total transmit power equals the noise power at the receiver. Given equation (3.6), with a perfect phase alignment at the receiver, $P = 1$ implies $\mathrm{E}[\gamma] \approx 6\mathrm{dB}$ when $N = 5$, $\mathrm{E}[\gamma] \approx 12\mathrm{dB}$ when $N = 20$. The simulation results for every point in the following figures are averaged over $10^6$ runs. As the received SNR cannot illustrate the advantages of beamforming gain and the effects of the number of nodes and phase errors, our simulation results and analytical results are plotted as BER vs fixed total transmit power $P$, which is one of the major concerns in practical design in wireless sensor networks. We have derived two expressions to predict the BER results for small number of nodes and large number of nodes separately. For simplicity, we denote equation (3.8) in Section 3.3 as method 1, while equation (3.37) in Section 3.4 is denoted as method 2.

Figure 3.6 shows the comparison of the simulation results with the analytical results based on method 1 for very small but different number of nodes $N = 3, 5$ and increasing phase error ranges $\phi_0 = 18^\circ, 36^\circ, 54^\circ$ and $72^\circ$. As can be seen, our analysis shows a good match with the simulation results for all values of $\phi_0$ up to $72^\circ$ with both $N = 3$ and $N = 5$. Because method 2 is based on the CLT it thus has a large deviation from the simulation results for a small $N$, we only present the results based on method 1 in Figure 3.6. (The accuracy of method 1 and method 2 when increasing $N$ from small numbers to large numbers are compared later in Figure 3.10.) From Figure 3.6 we see that increasing the number of nodes $N$ can dramatically reduce the transmit power requirement for the same BER performance, or from another point of view, it can significantly improve the BER performance with the same total transmit power. This is consistent with the conclusions presented in [6]. What is more, with a given number of nodes, increasing the phase error range $\phi_0$ will degrade the BER performance. It also shows

that with a fixed increment in $\phi_0$, the phase errors have a more significant effect on the BER performance at higher values of $\phi_0$. Taking the curves for $N = 5$ for example, subject to the same BER at $10^{-2}$, the performance loss when increasing from $\phi_0 = 54°$ to $\phi_0 = 72°$ is larger than the degradation when moving from $\phi_0 = 18°$ to $\phi_0 = 36°$. These observations agree with the discussion in Chapter 2 about the effects of the number of nodes and phase errors.

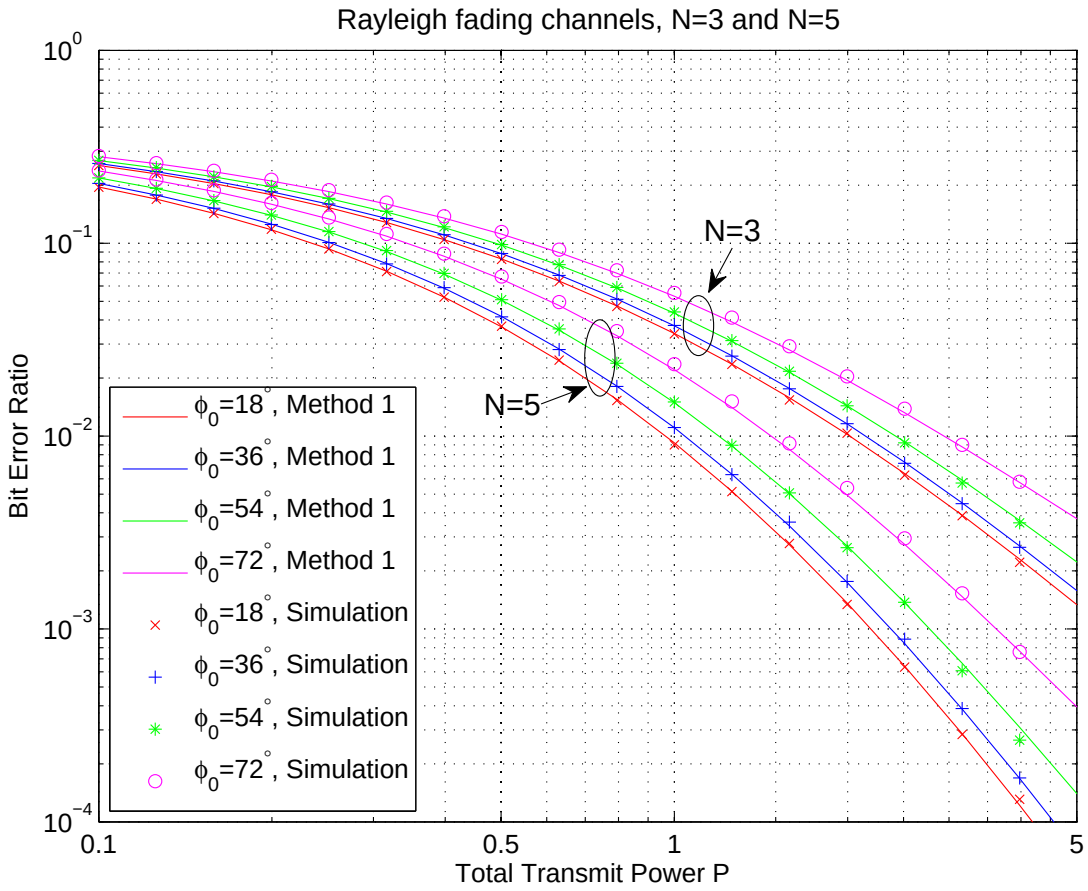



**Figure 3.6:** *Comparison of analytical results based on method 1 with simulation results of BER versus total transmit power with $N = 3, 5$ distributed sensor nodes, phase errors constrained within the range $\phi_0 = 18°, 36°, 54°, 72°$ relative to total transmit power $P = 1$.*

Figure 3.7 shows the comparison of the simulation results with the analytical results based on method 1 and method 2 for the same number of nodes $N = 20$, for the phase error ranges $\phi_0 = 18°, 36°, 54°$ and $72°$. It shows that both method 1 and method 2 have a good prediction on the BER results with $N = 20$ distributed sensor nodes. There exists a slight difference between the analytical results based on the two methods. Method 1 appears to be a little more

closer than method 2 to the simulation results for $\phi_0 = 36°, 54°$ while method 2 appears to be a little more closer than method 1 for $\phi_0 = 18°, 72°$.

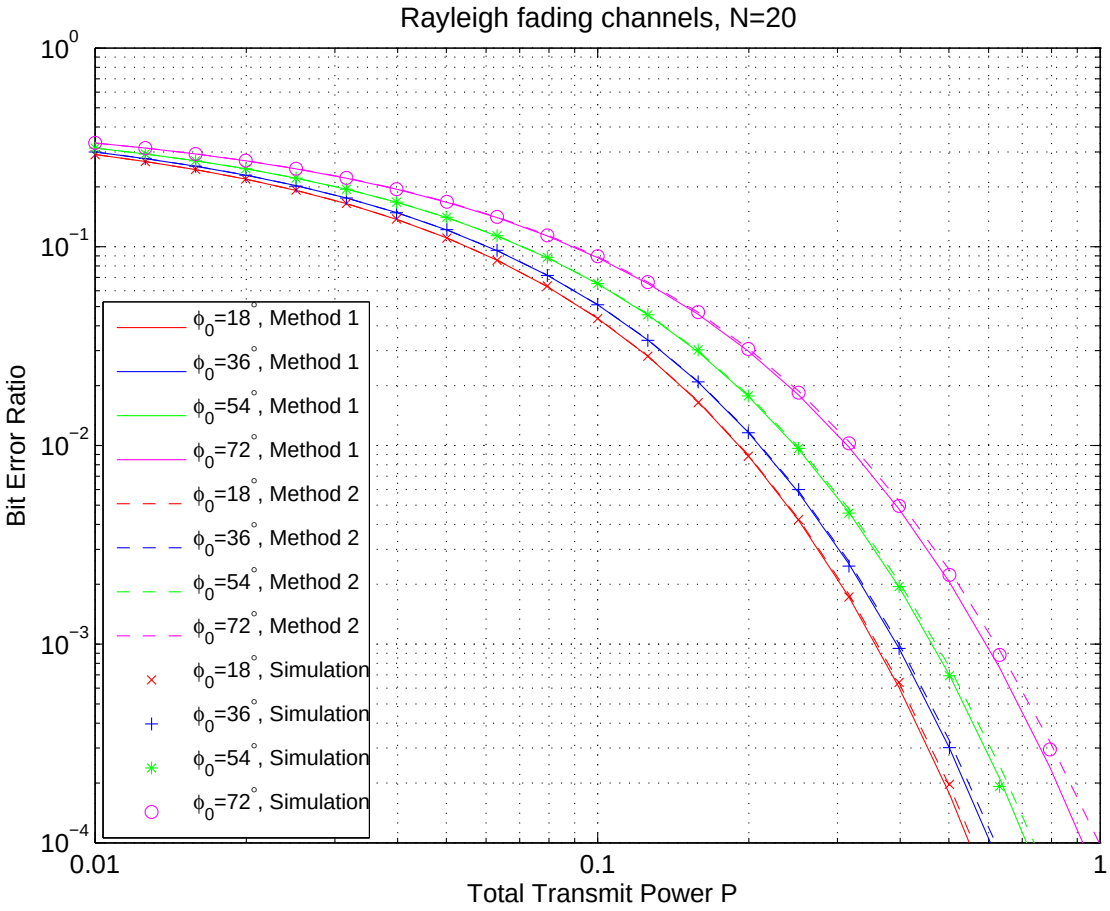



**Figure 3.7:** *Comparison of analytical results based on method 1 and method 2 with simulation results of BER versus total transmit power with* $N = 20$*, and* $\phi_0 = 18°, 36°, 54°, 72°$.

Figure 3.8 shows the comparison of the simulation results with the analytical results based on method 2 for large numbers of nodes $N = 40, 100$ for the same phase error ranges $\phi_0 = 18°, 36°, 54°$ and $72°$. As we see, for both $N = 40$ and $N = 100$ the simulation results and the analytical results show excellent agreement with each other. Method 1 still provides a good prediction for large $N$. However, with large $N$, method 1 has a high computational complexity, thus we only present the results based on method 2 in Figure 3.8. From Figure 3.8, we can draw the same conclusions about the effects of the number of nodes and the phase errors as from Figure 3.6 stated above. Comparing the two figures and considering the practical design, we have the conclusion that adding more nodes when $N$ is small, or minimizing the phase

errors when $\phi_0$ is large, can significantly improve the BER performance.

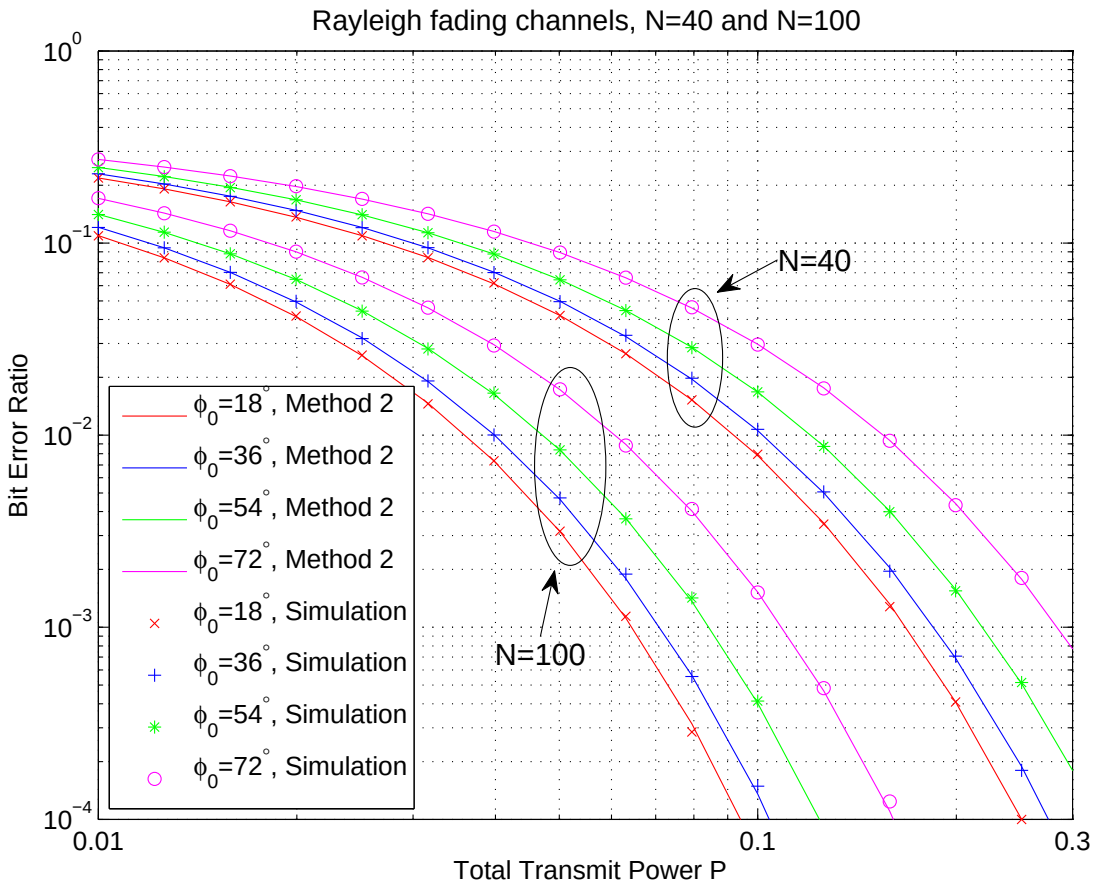
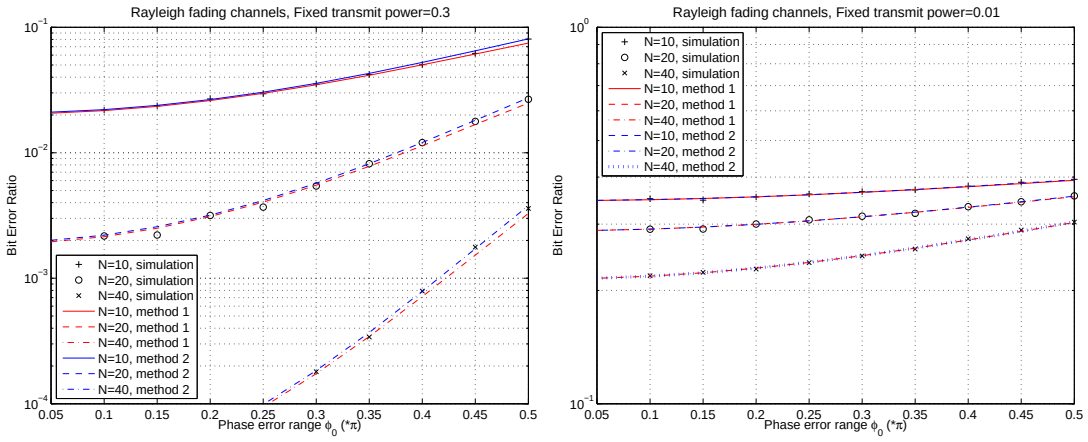



**Figure 3.8:** *Comparison of analytical results based on method 2 with simulation results of BER versus total transmit power with* $N = 40, 100$, *and* $\phi_0 = 18°, 36°, 54°, 72°$.

Figure 3.9 shows the BER over the $9°$ to $90°$ phase error range, $\phi_0$, at a fixed total transmit power to analyze the accuracy of methods 1 and 2 when $\phi_0$ grows larger. We realize that some plots in Figure 3.9 & 3.10 show unacceptably high BER but they are provided as further verification of the good match of our analysis and simulations. It can be seen in Figure 3.9 (a) that at a higher transmit power, there is a small gap between the two curves of method 1 and method 2 at large $\phi_0$, where method 1 has a more accurate prediction for the case $N = 10$, even up to $\phi_0 = 80°$. This is because method 2 is based on the CLT and is not so accurate for small $N$. Also, it can be seen that at a lower transmit power in Figure 3.9 (b), the two curves of method 1 and method 2 overlap each other and both of them match the simulation results accurately for all values of $\phi_0$ up to $90°$. However $90°$ may be considered too large and unacceptable a phase error range in most application scenarios for distributed beamforming as




(a) (b)

**Figure 3.9:** *Comparison of analytical results based on method 1 and method 2 with simulation results of BER versus phase error range* $9^\circ$ *to* $90^\circ$ *with* $N = 10, 20, 40$*, and (a) total transmit power* $P = 0.3$*, (b) total transmit power* $P = 0.01$*.*

in the plot the BER has a particularly high value.

Figure 3.10 shows the BER versus the number of nodes $N$ to analyze the accuracy of method 1 and method 2 when increasing $N$. In order to keep the received SNR roughly constant when increasing $N$, the total transmit power in Figure 3.10 is set to be inversely proportional to $N$, i.e. normalized by $N$, which is different to the simulations in previous figures. It can be seen that there is a gap between the two curves of method 1 and method 2 for small $N$, where method 1 provides a much more accurate prediction. Method 2 achieves progressively more accuracy as $N$ increases. This is because method 2 is based on the CLT and thus is not appropriate for small $N$. However, the solution given by method 1 takes the form of a single dimensional integral solved in our simulations by the Hermite integration method. The solution given by method 2 is much simpler and more computationally efficient compared to method 1. Therefore, it is preferable to use method 1 only for a small number of nodes (e.g. $N \leqslant 10$) and use method 2 for a large number of nodes (e.g. $N \geqslant 20$).

## 3.6 Summary

In this Chapter, we have derived BER expressions for BPSK signalling in distributed beamforming with phase errors. The simulation results show excellent agreement with analytical

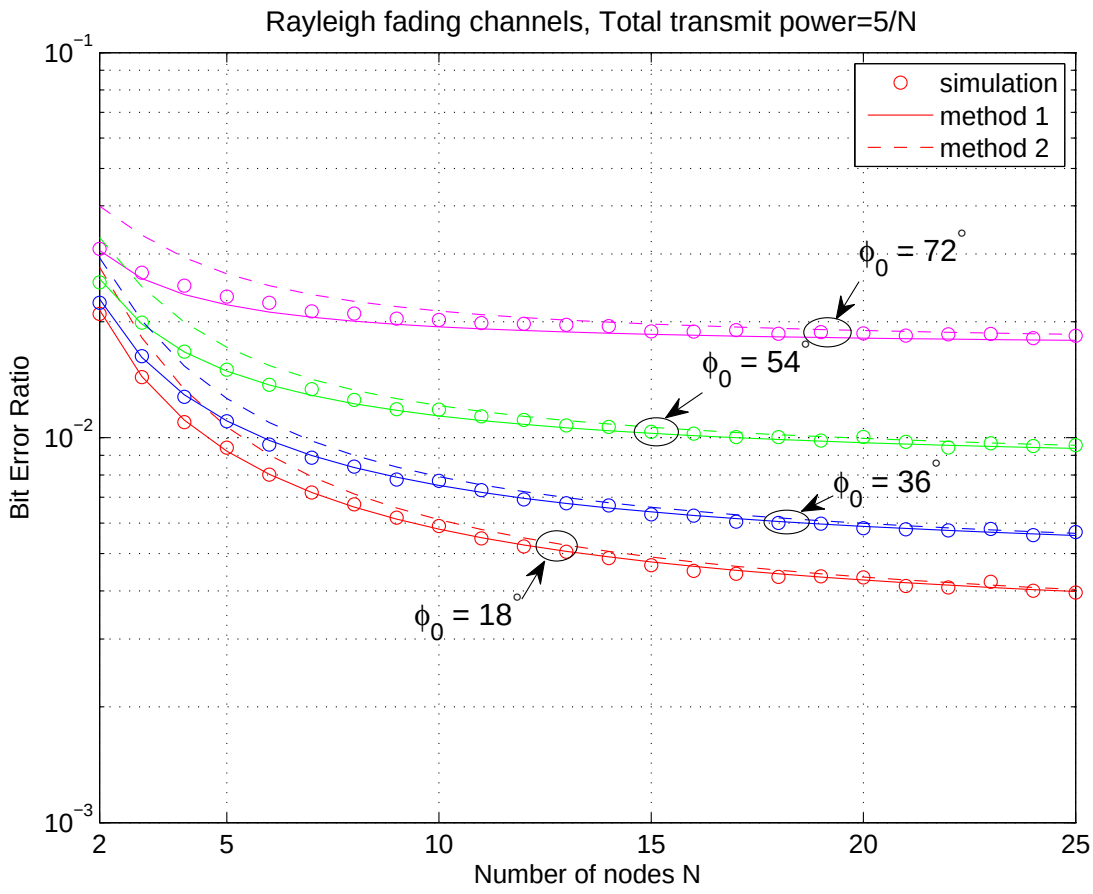



**Figure 3.10:** *Comparison of analytical results based on method 1 and method 2 with simulation results of BER versus number of nodes with* $\phi_0 = 18°, 36°, 54°, 72°$*, and total transmit power* $P = \frac{5}{N}$.

results. We analyzed the model from different approaches to approximate the distribution of the equivalent channel gain. It is suggested to use method 1, presented in Section 3.3, to predict BER for a small number of nodes (e.g. $N \leqslant 10$) and use method 2, presented in Section 3.4, for a large number of nodes (e.g. $N \geqslant 20$). We propose using method 2 here predominantly due to its reduced computational load for large $N$. The system performance has been analyzed for different numbers of nodes and different phase error ranges. With a given number of nodes and a defined transmit power constraint, one can use our BER expressions to bound the permissable phase errors. Alternatively, knowing the number of nodes and phase error range, one can calculate the energy requirement for each node. Our methods can be extended to analyze BER with MRT for distributed beamforming in a similar way. The analysis can also be applied to other phase error distributions in the literature discussed in Section 3.1, such as Gaussian and "exp-cosine". One can obtain the BER with other phase error distributions by simply substituting the corresponding pdfs into (A.7) and (A.8) to compute the second and the fourth moment of $\left|H\right|$. The theoretical analysis presented in this Chapter gives an accurate understanding of the impact of phase errors on the beamforming performance. In the next Chapter, we will probe into practical realizations of achieving phase alignment and minimizing phase errors at the receiver for distributed beamforming.

# Chapter 4
# A Reverse-Perturbation Scheme for Phase Alignment

A fundamental problem of realizing distributed beamforming in practice is to achieve phase alignment at the intended receiver. Signals transmitted from sensor nodes should be frequency synchronized and phase adjusted so that they can add coherently at the receiver and the accuracy of this coherence is critical to the beamformer performance. A simple iterative algorithm using one-bit feedback from the receiver in each iteration has been proposed in the literature which can achieve nearly perfect phase alignment after many iterations. In this Chapter, we propose an improved version of the one-bit feedback algorithm which has a faster convergence speed of phase synchronization and requires no extra hardware or information exchange. The advantage in the convergence speed is obtained by exploiting the one-bit feedback information in each iteration more efficiently.

## 4.1 Introduction

Distributed transmit beamforming can provide high SNR gains, extend the communication range, or reduce the energy requirement for each transmitter in signal transmission. However, these potential benefits rely on accurate phase alignment of the signals arriving at the receiver and phase alignment is critical to the beamformer performance. In Chapter 3, we have quantitatively studied the impact of phase errors on the BER performance of distributed beamforming. With a given number of sensor nodes and a constraint on the transmit power, phase errors among signals arriving at the receiver have to be minimized and contained within a certain range in order to maintain a BER performance. In Chapter 2, we have reviewed the challenges in practical realizations of distributed transmit beamforming. The most crucial part of realizing distributed transmit beamforming is carrier frequency and phase synchronization among all the transmitters to ensure that the signals can be added coherently at the receiver [6]. The frequency synchronization problem can be solved by employing a master-slave scheme

presented in [3], where the slave nodes lock their frequencies to a reference carrier signal periodically broadcasted by the master node. Alternatively, the frequency synchronization can be achieved by the destination node periodically broadcasting a reference signal to all transmitters. In addition to the frequency synchronization, how to achieve phase alignment or minimize phase errors in a distributed manner?

Several schemes have been proposed in the literature to achieve phase alignment or phase coherence at the receiver for distributed transmit beamforming. In [3], the authors proposed an open-loop scheme to achieve phase alignment where phase synchronization is first coordinated within the sensor network by employing a master-slave architecture, then achieved at the destination by pre-compensating the phase responses of node-destination channels based on channel reciprocity. The major problem of this scheme is that it requires accurate CSI at transmitters, and the beamforming performance is limited by several sources of estimation errors in the synchronization process. In [64], [65], [66], the authors proposed another scheme called the round-trip scheme to achieve phase alignment based on channel reciprocity. The basic idea is that the phase shift accumulated in a clockwise round-trip transmission is equal to a counter-clockwise round-trip transmission through a multi-hop chain of nodes. In [67], the authors first present a simple iterative algorithm (which we term the original algorithm) to adjust phase settings at transmitters, which can achieve nearly perfect phase alignment at the receiver in static channels after a large number of iterations. The algorithm is then comprehensively studied and mathematically analyzed in [4]. In the algorithm, the phase training process is performed by every transmitter making a random perturbation on its phase offset in each iteration. If the perturbation results in a positive feedback indicating a bigger beamforming gain, it will be adopted. Otherwise, it will be discarded. Such a training process can be reformulated as a random search algorithm [68], [69], [70] or associated to an ordinary differential equation (ODE) [71], [72], [73]. It can start with an arbitrary distribution of phase settings at transmitters and adjust transmitters' phase settings in a distributed manner to achieve phase alignment at the receiver. The key advantages of this algorithm is that it does not need channel state information and only relies on one-bit feedback in each iteration. Its simplicity in implementation and scalability to large number of transmitters make it a promising way to realize distributed transmit beamforming in practical applications. The shortcoming of the original algorithm is that the algorithm only converges upon positive feedback indicating successful perturbations and it takes a large number of iterations to achieve convergence. As discussed in Chapter 2, energy efficiency is one of the major concerns in wireless sensor networks and radio transmission

is one of the most energy-expensive operations [74]. Therefore, it is desirable to improve the convergence speed of the original algorithm while maintaining its key advantages.

The original algorithm has received wide attention in the literature and similar algorithms based on it using one-bit feedback were proposed for distributed beamforming. In [75], the original algorithm was further developed to account for carrier frequency errors, in addition to phase errors, among transmitters. In [35], the validity of the original algorithm was first verified by laboratory experiments, where expected performance results were obtained. The authors in [35] also made efforts in extending the algorithm to track time-varying channels. We will further discuss the problem of tracking time-varying channels in Chapter 5. The original algorithm is generalized to a multiuser context in [76]. Similar algorithms to achieve optimal power allocation in wireless relay networks were proposed in [77], [78]. In [79], the authors studied the convergence performance of a case where sensor nodes are restricted to sending binary phases rather than arbitrary, continuous valued phases. In [80], the authors studied a case where more channel phase information is fed back from the destination to the sensor nodes. A common feature of the original algorithm and these extended algorithms is that they only exploit positive feedback information and discard negative feedback information.

Recently, more related work on the one-bit feedback algorithm is presented in [81], [82], [83], [84], [85], [86]. In [81], the authors proposed a partitioned one-bit feedback algorithm where sensor nodes are divided into subsets for the phase synchronization process. Each subset performs the phase training process independently and simultaneously until it achieves a certain beamforming gain. Then the destination estimates and feeds back a beamforming weight for each subset to achieve phase synchronization across subsets. The partitioned algorithm has an advantage in the convergence time compared to the original algorithm at the cost of sending more feedback from the destination in each iteration. However, it does not save more energy compared to the original algorithm. In [82], the authors proposed a 3-bit feedback algorithm, where one bit is used as in the original algorithm and two bits are used to estimate the relative motion between transmitters and the receiver. However, this algorithm is not robust to random phase drifts and its convergence performance needs further investigation. In [83], [84], the authors presented more variations to the original algorithm by studying the impact of some influence factors, such as the network size, choice of nodes, and optimum perturbations. In [85], the authors studied the performance of the one-bit feedback algorithm with feedback bit errors. In [86], the authors presented a bio-inspired algorithm which can adaptively adjust perturbation

sizes and has a faster convergence speed under static channel conditions.

The idea of the one-bit feedback algorithm is similar to some nature-inspired random search algorithms in the area of swarm intelligence, such as the firefly algorithm [87], particle swarm optimization [88], ant colony algorithm [89], etc. Unlike the one-bit feedback algorithms, in the nature-inspired algorithms, each unit in a swarm takes into account the results obtained by other swarm units to compute possible solutions to an optimization problem. However, learning from adjacent nodes in wireless sensor networks is considered costly as it may require abundant information exchange among sensor nodes. All operations of the beamforming process have to be organized and implemented in a distributed manner. Therefore, the nature-inspired algorithms cannot be applied directly to perform distributed beamforming.

In this Chapter, we present a novel algorithm (namely the improved algorithm) based on the one-bit feedback algorithm described in [4] (namely the original algorithm) to achieve carrier phase alignment at the receiver in distributed transmit beamforming. The improved algorithm still requires only one-bit feedback from the receiver. It keeps all the benefits of the original algorithm, such as its simplicity and scalability, and requires no extra hardware. The improved algorithm is shown to have an advantage in the convergence speed. It requires fewer iterations, thus consumes less energy, to achieve a certain beamforming gain than the original algorithm by making use of the random perturbation obtained in each time slot more efficiently.

## 4.2 System model

We consider a distributed transmit beamforming system similar to the one described in Chapter 3. The system is composed of $N$ transmitters collaboratively beamforming a narrowband message signal to a distant receiver. This is performed in a distributed manner by each transmitter modulating the message signal at the same carrier frequency and adjusting its phase setting iteratively to achieve phase alignment at the receiver. The system model including phase components contributing to the phase of the received signal at the receiver is illustrated in Figure 4.1.

In order to compare the improved algorithm with the original algorithm easily and fairly, the assumptions made in this Chapter are all the same with the assumptions in [4]. We repeat some key assumptions below. For more details, please see the list of assumptions in [4]. The channel from each transmitter to the receiver, $h_i$, is assumed to be static during the phase

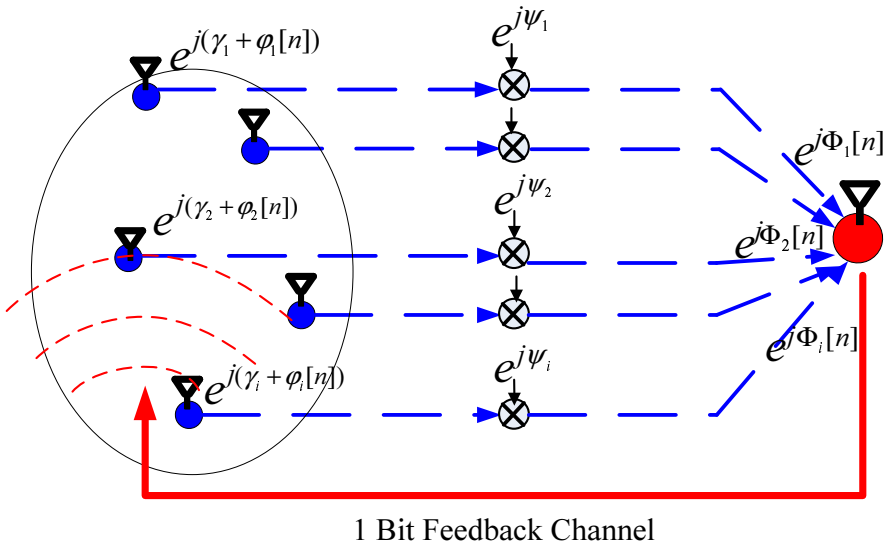



**Figure 4.1:** *System model for distributed transmit beamforming using one-bit feedback algorithms.*

synchronization process. For simplicity, we set $|h_i| = 1$. All transmitters are frequency-synchronized so that they only need to adjust their phase settings to achieve phase alignment at the receiver. The local carrier of each transmitter $i$ has an unknown phase offset $\gamma_i$ relative to the receiver's phase reference. As both algorithms considered here put emphasis on the phase synchronization process and the effect of phase difference on the beamforming gain, we assume unit transmit power for every transmitter. The phase training process in both algorithms are performed in a time-slotted fashion, and the phase of the received signal at the receiver from transmitter $i$ in time slot $n$ can be expressed as:

$$\Phi_i[n] = \gamma_i + \psi_i + \varphi_i[n] \tag{4.1}$$

where $\gamma_i$ is an unknown phase offset at transmitter $i$, $\psi_i$ is the channel phase response from transmitter $i$ to the receiver. Both $\gamma_i$ and $\psi_i$ are assumed to be static during the convergence process, uniformly distributed within $[0, 2\pi)$ over $i$ and unknown to both the transmitters and the receiver. The scalar $\varphi_i[n]$ is the adaptive component adjusted by transmitter $i$ in each time slot based on the one-bit feedback information from the receiver. We set $\varphi_i[0] = 0$ for both algorithms. The phase of the received signal, $\Phi_i[n]$, is related to the phase error among signals arriving at the receiver, $\phi_i$, defined in Chapter 3. The ideal phase alignment of distributed beamforming is that there are no phase differences among the signals arriving at the receiver,

i.e.:

$$\gamma_i + \psi_i + \varphi_i[n] \equiv \gamma_k + \psi_k + \varphi_k[n] \pmod{2\pi}, \quad i \neq k \quad \forall i, k = 1, 2, ..., N. \tag{4.2}$$

The objective of the algorithm design is to let each transmitter adjust its value $\varphi_i[n]$ based on the one-bit feedback information in each time slot to achieve nearly perfect phase alignment at the receiver as fast as possible.

The received signal strength (RSS), which determines the beamforming gain, in time slot $n$ is defined as:

$$R[n] = \left| \sum_{i=1}^{N} e^{j\Phi_i[n]} \right| \tag{4.3}$$

The noise power at the receiver is assumed to be fairly small compared to the signal power at the receiver, and the RSS in each time slot, $R[n]$, can be measured accurately by averaging the received signal over a certain time interval. Both algorithms use the RSS as a metric to measure the beamforming performance during the convergence process. In Chapter 3, we have analyzed the BER performance for distributed beamforming with phase errors over Rayleigh fading channels. Given equation (3.5) and (3.6), we see the BER performance mainly depends on the distribution of the RSS defined in (4.3). In the phase training process, it is more easier to measure the RSS rather than the BER at the receiver in practical implementations. Therefore, for simplicity, we use the RSS as a metric to measure the beamforming performance in the algorithm design described in this Chapter.

## 4.3 Original one-bit feedback algorithm

The original one-bit feedback algorithm to achieve phase alignment at the receiver for distributed beamforming introduced in [4] repeats the following steps.

1. At time slot $n$, each transmitter records its best known phase used for beamforming,

$\theta_i[n]$, in memory and adds a random perturbation, $\delta_i[n] = \pm\delta_0$ (with equal probability for "+" and "-"), to it. (We set $\theta_i[1] = 0$).

2. All transmitters use their new adaptive components, $\varphi_i[n] = \theta_i[n] + \delta_i[n]$, to perform transmit beamforming.

3. The receiver measures the RSS, $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$, and compares it with the best RSS in memory. The receiver updates the best RSS in memory and feeds back (error free) one-bit of information to all transmitters conveying whether the RSS has been improved or not.

4. If the RSS has been improved, all transmitters adopt their perturbed phases and update their best known phases to be $\theta_i[n+1] = \varphi_i[n] = \theta_i[n] + \delta_i[n]$ for the next time slot $(n+1)$. Otherwise, all transmitters discard the perturbed phases and keep the best known phases as before, $\theta_i[n+1] = \theta_i[n]$, for the next time slot $(n+1)$.

The adaptive component $\varphi_i[n]$ used for beamforming in time slot $n$ in the original algorithm is composed of two parts:

$$\varphi_i[n] = \theta_i[n] + \delta_i[n] \tag{4.4}$$

where $\theta_i[n]$ represents the best known phase of transmitter $i$ in time slot $n$. The scalar $\delta_i[n]$ is the random component applied to the best known phase in time slot $n$.

The original algorithm in [4] can be mathematically expressed as:

At the transmitter side:

$$\theta_i[n+1] = \begin{cases} \theta_i[n] + \delta_i[n] & R[n] > R_{\text{best}}[n] \\ \theta_i[n] & \text{otherwise} \end{cases} \tag{4.5}$$

At the receiver side:

$$R_{\text{best}}[n+1] = \max(R_{\text{best}}[n], R[n]) \tag{4.6}$$

where $R_{\text{best}}[n]$ is the best RSS in memory, or in other words, the maximal RSS in the past $n-1$ time slots. By inserting (4.4) into (4.1), the overall phase of the received signal at the receiver in time slot $(n+1)$ can be expressed as:

$$\begin{aligned} \Phi_i[n+1] &= \gamma_i + \psi_i + \varphi_i[n+1] \\ &= \gamma_i + \psi_i + \theta_i[n+1] + \delta_i[n+1] \end{aligned} \tag{4.7}$$

Given (4.5), when $R[n] > R_{\text{best}}[n]$, (4.7) becomes:

$$\Phi_i[n+1] = \gamma_i + \psi_i + \theta_i[n] + \delta_i[n] + \delta_i[n+1] \tag{4.8}$$

Otherwise, when $R[n] \leq R_{\text{best}}[n]$, (4.7) becomes:

$$\Phi_i[n+1] = \gamma_i + \psi_i + \theta_i[n] + \delta_i[n+1] \tag{4.9}$$

The original algorithm can achieve phase alignment after many iterations. Figure 4.2 shows simulation results for one instance of the original algorithm with $N = 100$, $\delta_0 = \frac{\pi}{50}$. It shows that the RSS is increasing gradually with increased time slots and a high beamforming gain close to the optimum value can be obtained after many iterations. For more details of the original algorithm including its advantages over other alternative approaches for distributed beamforming, see [4].

## 4.4 Reverse perturbation algorithm

The original algorithm can be viewed as a random search process in which each transmitter is trying to adjust its phase correctly based on the feedback information. Since the original algorithm only changes phase for positive feedback and discards other "failed" perturbations, it only makes use of the feedback information which indicates performance improvement. However, failure can also be used to obtain future success. Making use of the information contained within the failed perturbations which led to performance degradation is expected to be helpful

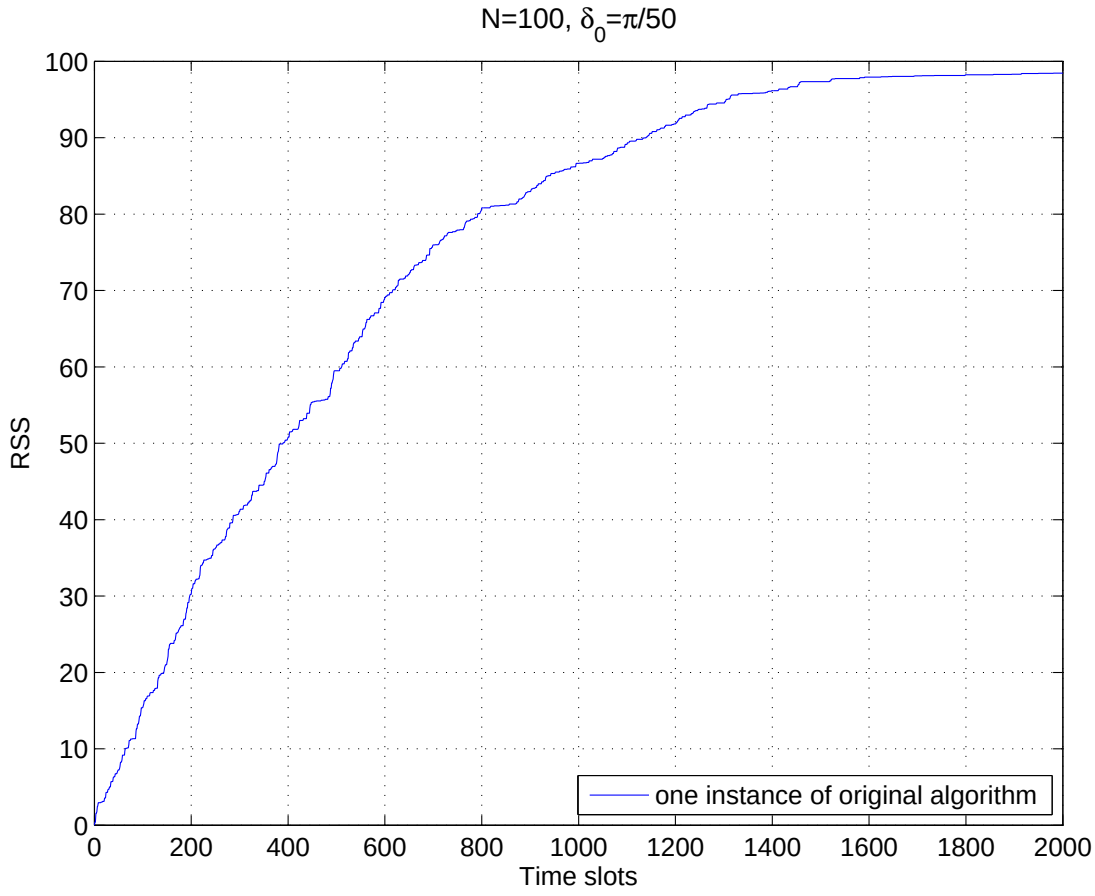



**Figure 4.2:** *Simulation results for one instance of the original algorithm on the received signal strength versus the number of time slots.*

in improving the convergence speed of phase alignment. Hereby, we propose a new algorithm based on the original algorithm summarized as follows.

1. At time slot $n$, each transmitter applies a random perturbation, $\delta_i[n] = \pm\delta_0$, to its best known carrier phase, $\theta_i[n]$, for beamforming. Meanwhile, each transmitter also adds a modifying factor, $\epsilon_i[n]$, to its best known carrier phase for beamforming. The function of $\epsilon_i[n]$ is to add an opposite value of $\delta_i[n-1]$ into the new adaptive component if $\delta_i[n-1]$ has led to performance degradation in the previous time slot. Otherwise, the value of $\epsilon_i[n]$ is set to be $0$.

2. All transmitters use their new adaptive components, $\varphi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$, to perform transmit beamforming.

3. The receiver measures the RSS, $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$, and compares it with the best RSS in memory. The receiver updates the best RSS in memory and feeds back (error free) one-bit of information to all transmitters conveying whether the RSS has been improved or not.

4. If the RSS has been improved, all transmitters adopt their perturbed phases and update their best known phases to be $\theta_i[n+1] = \varphi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$ for the next time slot $(n+1)$. The modifying factor for the next time slot is set to be $\epsilon_i[n+1] = 0$. Otherwise, all transmitters discard the perturbed phases and keep the best known phases as before, $\theta_i[n+1] = \theta_i[n]$, for the next time slot $(n+1)$. The modifying factor for the next time slot is set to be $\epsilon_i[n+1] = -\delta_i[n]$.

   The algorithm then repeats these four steps.

The adaptive component $\varphi_i[n]$ used for beamforming in time slot $n$ in the improved algorithm is composed of three parts:

$$\varphi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n] \tag{4.10}$$

where $\theta_i[n]$ represents the best known phase, $\epsilon_i[n]$ is the modifying factor and $\delta_i[n]$ is the random component.

The improved algorithm can be mathematically expressed as:

At the transmitter side:

$$\theta_i[n+1] = \begin{cases} \theta_i[n] + \epsilon_i[n] + \delta_i[n] & R[n] > R_{\text{best}}[n] \\ \theta_i[n] & \text{otherwise} \end{cases} \tag{4.11}$$

$$\epsilon_i[n+1] = \begin{cases} 0 & R[n] > R_{\text{best}}[n] \\ -\delta_i[n] & \text{otherwise} \end{cases} \tag{4.12}$$

At the receiver side:

$$R_{\text{best}}[n+1] = \max(R_{\text{best}}[n], R[n]) \tag{4.13}$$

By substituting (4.10) into (4.1), the overall phase of the received signal at the receiver in time slot $(n+1)$ can be expressed as:

$$\begin{aligned}\Phi_i[n+1] &= \gamma_i + \psi_i + \varphi_i[n+1] \\ &= \gamma_i + \psi_i + \theta_i[n+1] + \epsilon_i[n+1] + \delta_i[n+1] \end{aligned} \tag{4.14}$$

Given (4.11) and (4.12), when $R[n] > R_{\text{best}}[n]$, (4.14) becomes:

$$\Phi_i[n+1] = \gamma_i + \psi_i + \theta_i[n] + \epsilon_i[n] + \delta_i[n] + 0 + \delta_i[n+1] \tag{4.15}$$

Otherwise, when $R[n] \leq R_{\text{best}}[n]$, (4.14) becomes:

$$\Phi_i[n+1] = \gamma_i + \psi_i + \theta_i[n] - \delta_i[n] + \delta_i[n+1] \tag{4.16}$$

When the perturbation size $\delta_0$ is quite small compared to the phase differences at the receiver, a perturbation on the carrier phases would lead to either a reduction or an increment in phase differences at the receiver, thus yielding beamforming performance improvement or degradation. The basic idea behind the improved algorithm is that for a single transmitter in each time slot, if a positive perturbation on its carrier phase leads to performance degradation, usually, a negative perturbation on the same carrier phase will lead to performance improvement, and vice versa. Figure 4.3 shows an example of two transmitters.

By comparing (4.15) with (4.8) we see that in both algorithms, when an adaptive component $\varphi_i[n]$ leads to a bigger beamforming gain, it will be retained and be set as the best known phase for the next time slot, so $\theta_i[n+1] = \varphi_i[n]$. In the next time slot $(n+1)$, a random perturbation, $\delta_i[n+1]$, will be applied to this best known phase, $\theta_i[n+1]$, and there is no further modification apart from the random perturbation on $\theta_i[n+1]$ for beamforming. By comparing (4.16) with (4.9) we see that in both algorithms, when an adaptive component $\varphi_i[n]$ leads to a smaller beamforming gain, it will be discarded and the best known phase is kept unchanged for the next time slot, so $\theta_i[n+1] = \theta_i[n]$. In the next time slot $(n+1)$, the original algorithm will perform a random perturbation again based on the same $\theta_i[n]$ while the improved algorithm

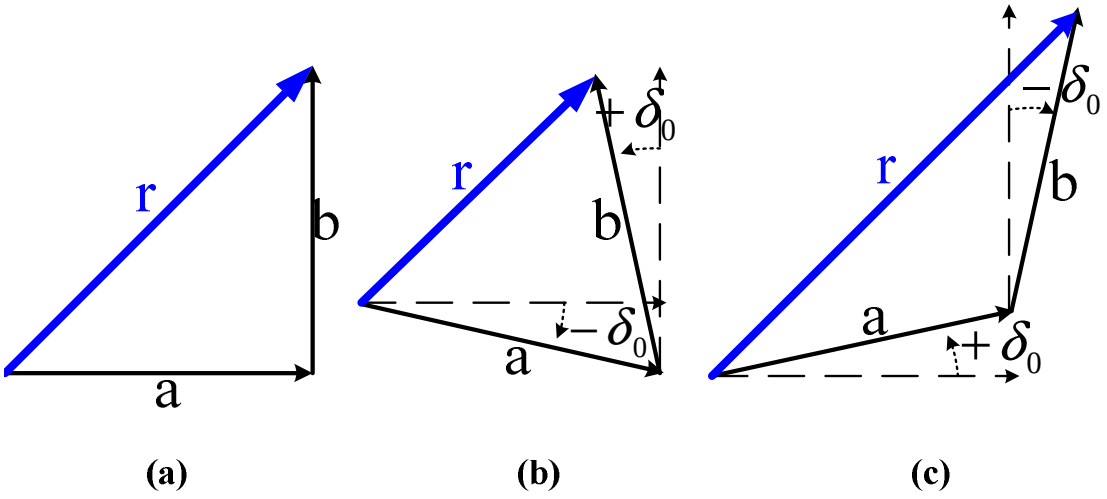



**Figure 4.3:** *Phase perturbation results in the case of two transmitters. If (b) a random perturbation leads to performance degradation, (c) an opposite perturbation will lead to performance improvement. Vector $a$ is the received signal from one transmitter, vector $b$ is the received signal from the other transmitter.*

will perform a random perturbation based on $\theta_i[n] - \delta_i[n]$, where $-\delta_i[n]$ is introduced by the modifying factor, $\epsilon_i[n]$. Consequently, both successful and failed perturbations in the improved algorithm contribute to the convergence speed.

One may ask why in the case of a failed perturbation in time slot $n$, should the algorithm not directly update the best known phase to be $\theta_i[n+1] = \theta_i[n] - \delta_i[n]$ for time slot $(n+1)$ rather than introducing the modifying factor, $\epsilon_i[n+1]$? In that case the random perturbation would be based on $\theta_i[n] - \delta_i[n]$ in time slot $(n+1)$. Such an idea can be mathematically expressed as:

$$\theta_i[n+1] = \begin{cases} \theta_i[n] + \delta_i[n] & R[n] > R_{\text{best}}[n] \\ \theta_i[n] - \delta_i[n] & \text{otherwise} \end{cases} \tag{4.17}$$

The reason for not doing so is because $(\theta_i[n] - \delta_i[n])$ does not always result in a better performance than $\theta_i[n]$. If not, the update equation $\theta_i[n+1] = \theta_i[n] - \delta_i[n]$ may drift off the best phase for beamforming corresponding to the best RSS in memory. Figure 4.4 shows simulation results for one instance of using (4.17) to update the best known phase. In contrast, the improved algorithm only updates the best known phase when positive feedback happens and also makes use of the negative feedback information in a single time slot to enhance the probability of generating better phase changes.

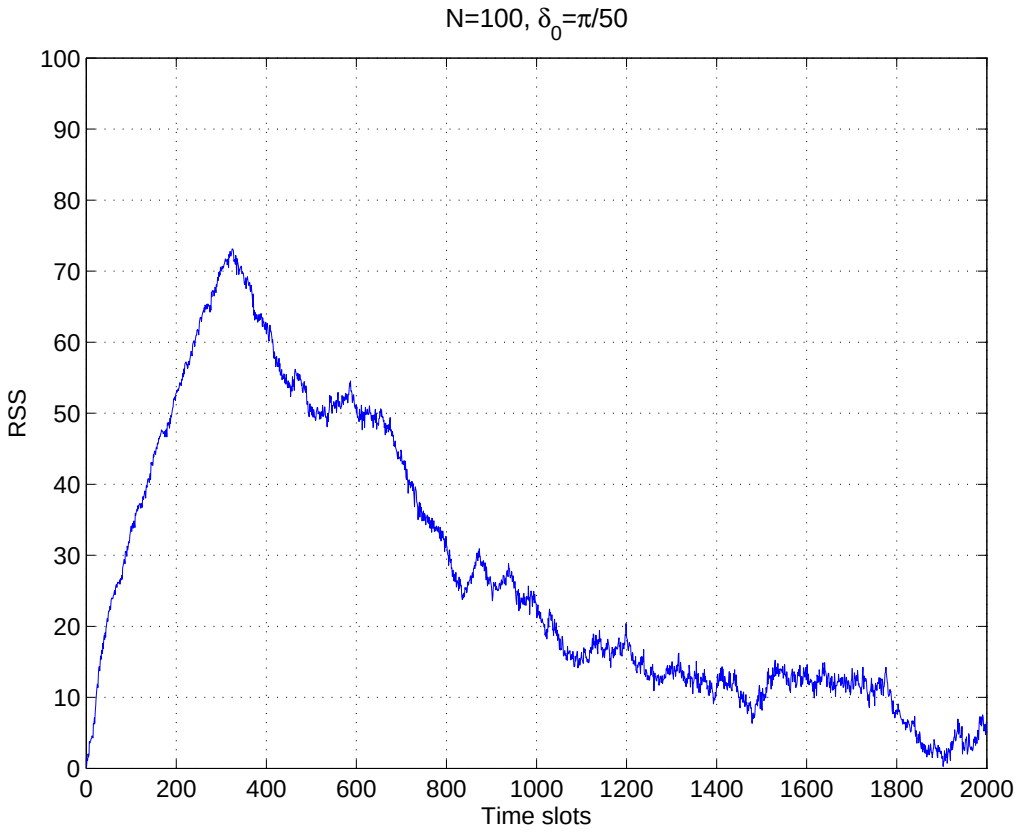



**Figure 4.4:** *Simulation results for one instance of using (4.17) to update the best known phase.*

Figure 4.5 shows an example of the evolution of the adaptive component for transmitter $i$ in several time slots using the improved algorithm, starting from time slot $n$ till time slot $(n+5)$. From Figure 4.5 we see, in the case of negative feedback, an opposite value of the perturbation in time slot $n$ will be added into the next adaptive component in time slot $(n+1)$, which enhances the probability of generating better phase changes. In the case of successive negative feedback steps, the values of the adaptive component $\varphi_i$ are always located around the best known phase $\theta_i$. This is because the value of $\theta_i$ is updated only in the case of positive feedback in order to prevent $\theta_i$ from drifting off its best value.

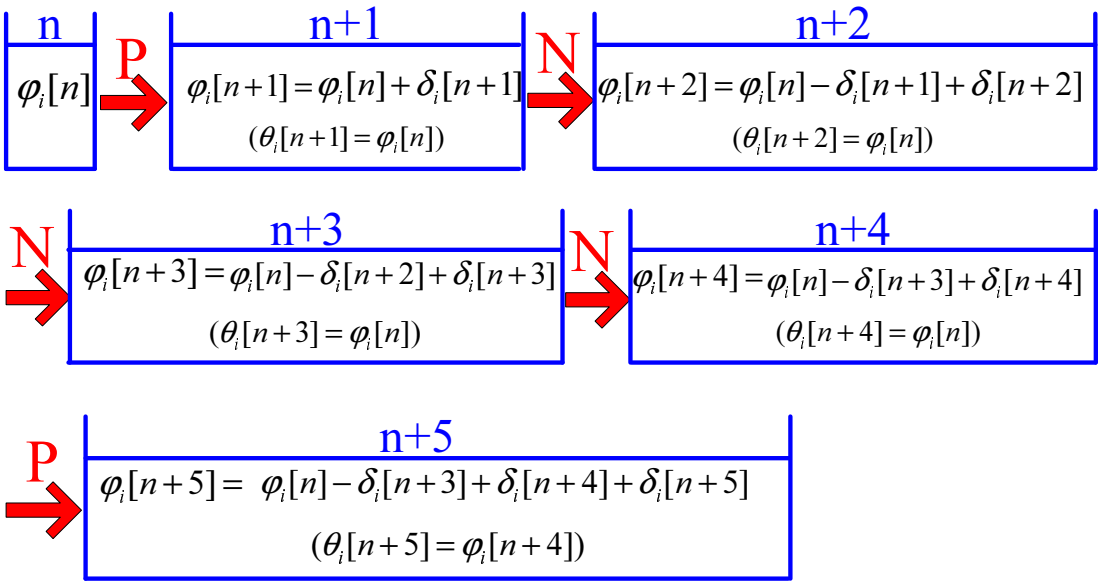



**Figure 4.5:** *Evolution of the adaptive component $\varphi_i$ for beamforming using the improved algorithm, starting from time slot $n$ till time slot $(n+5)$. "P" represents positive feedback while "N" represents negative feedback. $\theta_i$ is the best known phase and $\delta_i$ is the random perturbation.*

## 4.5 Analysis of the improved algorithm

We present some mathematical analysis of the improved algorithm and provide a close upper bound on its convergence speed. We begin our analysis by studying the original one-bit feedback algorithm. The original algorithm described above can be reformulated as:

$$\theta_i[n+1] = \theta_i[n] + \delta_i[n]\mathbf{1}_G \tag{4.18}$$

where the indicator function $\mathbf{1}_G$ equals 1 when the condition $G$ is satisfied and equals 0 otherwise. The condition $G = \{ \sum_{i=1}^{N} \cos(\hat{\Phi}_i[n] + \delta_i[n]) > \sum_{i=1}^{N} \cos(\hat{\Phi}_i[n]) \}$ and $\hat{\Phi}_i[n] = \gamma_i + \psi_i + \theta_i[n]$. The condition $G$ exists because with a large $N$, the RSS mainly depends on the cosines of the carrier phases and the contribution of sines can be discarded. In [71], the authors proved that the trajectories of (4.18) collapse to the solution of a certain ODE. For the readers' convenience, we first repeat some of the key results in deriving the ODE for the original algorithm. For details, please see [71]. We then derive an ODE that mimics the behavior of the improved algorithm in a similar way.

For a small perturbation size $\delta_0$, $\cos(\hat{\Phi}_i[n] + \delta_i[n]) \approx \cos(\hat{\Phi}_i[n]) - \delta_i[n] \sin(\hat{\Phi}_i[n])$. Therefore, the condition $G$ can be simplified to $G = \{\sum_{i=1}^{N} \delta_i[n] \sin(\hat{\Phi}_i[n]) < 0\}$. With large $N$, the summation of $(N-1)$ terms excluding $\delta_j[n] \sin(\hat{\Phi}_j[n])$ can be written as:

$$Z_j = \sum_{i=1(i \neq j)}^{N} \delta_i[n] \sin(\hat{\Phi}_i[n]) \tag{4.19}$$

which is a zero mean Gaussian variable according to the Lyapunov CLT, whose variance is $\mathrm{Var}(Z_j) = \delta_0^2 \sum_{i=1(i \neq j)}^{N} \sin^2(\hat{\Phi}_i[n])$. Therefore, the probability of condition $G$ being satisfied is:

$$\begin{aligned} \Pr(G) &= \Pr(Z_j + \delta_j[n] \sin(\hat{\Phi}_j[n]) < 0) \\ &= \frac{1}{2} - \frac{1}{2}\mathrm{erf}\left( \frac{\delta_j[n] \sin(\hat{\Phi}_j[n]) - \mathrm{E}(Z_j)}{\sqrt{2}\sqrt{\mathrm{Var}(Z_j)}} \right) \\ &\approx \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \cdot \frac{\delta_j[n] \sin(\hat{\Phi}_j[n])}{\delta_0 \sqrt{\sum_{i=1(i \neq j)}^{N} \sin^2(\hat{\Phi}_i[n])}}. \end{aligned} \tag{4.20}$$

where $\mathrm{erf}(\cdot)$ represents the Gaussian error function. The last approximation comes from the first term of the error function's Taylor series $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n!(2n+1)}$. Thus, the expectation of the random perturbation applied on phase settings for transmitter $j$ can be computed as:

$$\begin{aligned} \mathrm{E}(\delta_j[n]\mathbf{1}_G) &\approx \mathrm{E}(\delta_j[n]\mathrm{Pr}(G)) \\ &= \mathrm{E}\left(\delta_j[n]\left(\frac{1}{2} - \frac{1}{\sqrt{2\pi}} \cdot \frac{\delta_j[n]\sin(\hat{\Phi}_j[n])}{\delta_0\sqrt{\sum_{i=1(i\neq j)}^{N}\sin^2(\hat{\Phi}_i[n])}}\right)\right) \\ &= -\frac{\delta_0\sin(\hat{\Phi}_j[n])}{\sqrt{2\pi}\sqrt{\sum_{i=1(i\neq j)}^{N}\sin^2(\hat{\Phi}_i[n])}}. \end{aligned} \tag{4.21}$$

The convergence of the best known phases $\theta_i$ to their correct settings is equivalent to the convergence of $\hat{\Phi}_i$ to zero. The ODE corresponding to equation (4.18) which mimics the behavior of the original algorithm can be obtained as:

$$\frac{d\hat{\Phi}_j(t)}{dt} = -\frac{\delta_0\sin(\hat{\Phi}_j[n])}{\sqrt{2\pi}\sqrt{\sum_{i=1(i\neq j)}^{N}\sin^2(\hat{\Phi}_i[n])}}. \tag{4.22}$$

In the original algorithm, the decision on the perturbation $\delta_i[n]$ only depends on $R[n]$ and the corresponding feedback in time slot $n$. However, in the improved algorithm, the decision on the perturbation $\delta_i[n]$ not only depends on the feedback in time slot $n$, but also the feedback in time slots $(n+1)$ and $(n-1)$. A flowchart of the adaptive component for transmitter $j$ in the improved algorithm is shown in Figure 4.6. The conditions ConA, ConB in Figure 4.6 are defined mathematically as follows:

$$\mathrm{ConA}(\delta_i[n]) = \left\{ \sum_{i=1}^{N}\cos(\hat{\Phi}_i[n] + \delta_i[n]) > \sum_{i=1}^{N}\cos(\hat{\Phi}_i[n]) \right\}. \tag{4.23}$$

$$\mathrm{ConB}(\delta_i[n-1], \delta_i[n]) = \left\{ \sum_{i=1}^{N}\cos(\hat{\Phi}_i[n-1] - \delta_i[n-1] + \delta_i[n]) > \sum_{i=1}^{N}\cos(\hat{\Phi}_i[n-1]) \right\}. \tag{4.24}$$

The condition ConA is the same as condition $G$ in the original algorithm. Therefore, its probability is given by

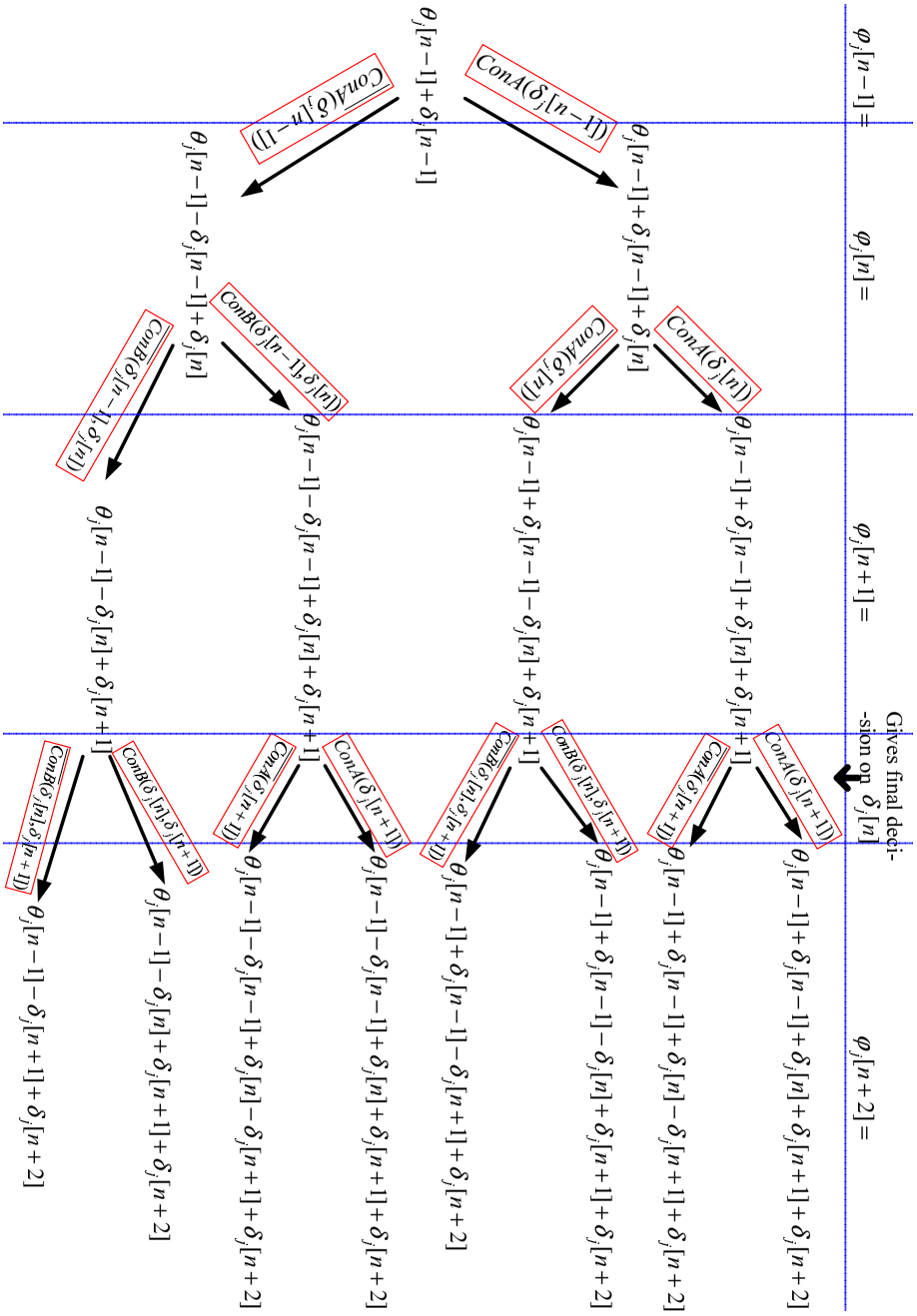



**Figure 4.6:** *A flowchart of the adaptive component $\varphi_j$ in the improved algorithm.*

$$\Pr(\mathrm{ConA}(\delta_j[n])) \quad = \quad \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \cdot \frac{\delta_j[n]\eta_j[n]}{\delta_0}. \tag{4.25}$$

where $\eta_j[n] = \frac{\sin(\hat{\Phi}_j[n])}{\sqrt{\sum_{i=1(i\neq j)}^{N} \sin^2(\hat{\Phi}_i[n])}}$. The probability of condition ConB can be derived in a similar way and is expressed as:

$$\Pr(\mathrm{ConB}(\delta_j[n-1], \delta_j[n])) \quad = \quad \frac{1}{2} - \frac{1}{\sqrt{2}\sqrt{2\pi}} \cdot \frac{(-\delta_j[n-1] + \delta_j[n])\eta_j[n-1]}{\delta_0}. \tag{4.26}$$

The conditions $\overline{\mathrm{ConA}}$, $\overline{\mathrm{ConB}}$ are the negations of $\mathrm{ConA}$, $\mathrm{ConB}$, whose probabilities can be calculated using the equations:

$$\Pr(\mathrm{ConA}) + \Pr(\overline{\mathrm{ConA}}) = \Pr(\mathrm{ConB}) + \Pr(\overline{\mathrm{ConB}}) = 1. \tag{4.27}$$

From Figure 4.6 we have:

$$\theta_i[n+1] = \begin{cases} \theta_i[n] + \delta_i[n]\mathbf{1}_{G^+} \\ \theta_i[n] - \delta_i[n]\mathbf{1}_{G^-} \end{cases} \tag{4.28}$$

where

$$G^+ = \left\{ \mathrm{ConA}(\delta_i[n-1]) \cdot \mathrm{ConA}(\delta_i[n]) + \overline{\mathrm{ConA}}(\delta_i[n-1]) \cdot \mathrm{ConB}(\delta_i[n-1], \delta_i[n]) \right\}, \tag{4.29}$$

$$\begin{aligned} G^- \quad = \quad & \Big\{ \mathrm{ConA}(\delta_i[n-1]) \cdot \overline{\mathrm{ConA}}(\delta_i[n]) \cdot \mathrm{ConB}(\delta_i[n], \delta_i[n+1]) \\ & + \overline{\mathrm{ConA}}(\delta_i[n-1]) \cdot \overline{\mathrm{ConB}}(\delta_i[n-1], \delta_i[n]) \cdot \mathrm{ConB}(\delta_i[n], \delta_i[n+1]) \Big\}. \end{aligned} \tag{4.30}$$

Thus,

$$\mathrm{E}(\delta_j[n]\mathbf{1}_{G^+} - \delta_j[n]\mathbf{1}_{G^-}) \;=\; \mathrm{E}\left[\delta_j[n]\cdot\left(\Pr(G^+) - \Pr(G^-)\right)\right] \tag{4.31}$$

For a small perturbation size $\delta_0$, $\eta_j[n-1] \approx \eta_j[n] \approx \eta_j[n+1] = \eta_j$. Substituting (4.25), (4.26), (4.27), (4.29) and (4.30) into (4.31), we have:

$$\begin{aligned}\mathrm{E}(\delta_j[n]\mathbf{1}_{G^+} - \delta_j[n]\mathbf{1}_{G^-}) \;&=\; \mathrm{E}\Bigg[-\Big(\frac{1}{2}\cdot\frac{\delta_j[n]^2\eta_j}{\sqrt{2\pi}\delta_0} + \frac{1}{2}\cdot\frac{\delta_j[n]^2\eta_j}{\sqrt{2}\sqrt{2\pi}\delta_0} \\ &\quad +(\frac{1}{4}+\frac{1}{4\sqrt{2}})\cdot\frac{\delta_j[n]^2\eta_j}{\sqrt{2\pi}\delta_0} + \frac{1}{2\sqrt{2}}\cdot\frac{\delta_j[n]^2\eta_j}{\sqrt{2\pi}\delta_0} - \frac{\delta_j[n]^2\eta_j^3}{4\sqrt{2}\pi^{\frac{3}{2}}\delta_0}\Big)\Bigg] \\ &>\; -\frac{5\sqrt{2}+6}{8}\cdot\frac{\delta_0\eta_j}{\sqrt{2\pi}}.\end{aligned} \tag{4.32}$$

The ODE corresponding to equation (4.28) which mimics the behavior of the improved algorithm can be obtained as:

$$\frac{d\hat{\Phi}_j(t)}{dt} = -\frac{5\sqrt{2}+6}{8}\cdot\frac{\delta_0\sin(\hat{\Phi}_j[n])}{\sqrt{2\pi}\sqrt{\sum_{i=1(i\neq j)}^{N}\sin^2(\hat{\Phi}_i[n])}}. \tag{4.33}$$

Comparing (4.33) with (4.22), we see that the improved algorithm has a faster convergence speed of $\frac{5\sqrt{2}+6}{8} \approx 1.634$ compared to the original algorithm. The accuracy of (4.33) will be justified by simulation results in Section 4.6.

## 4.6 Simulation results and comparisons

In this section, we present some Monte Carlo simulation results in accordance with our previous assumptions.

Figure 4.7 shows the comparison of the trajectories of the phases $\hat{\Phi}$ obtained from simulation with the trajectories of the ODE in (4.33) for the improved algorithm with 20 sensor nodes. The initial values of the phases $\hat{\Phi}$ are set as uniformly distributed within $(-\pi, \pi)$. It shows that the ODE in (4.33) can give a good prediction on the behavior of the phase alignment process under

the improved algorithm.

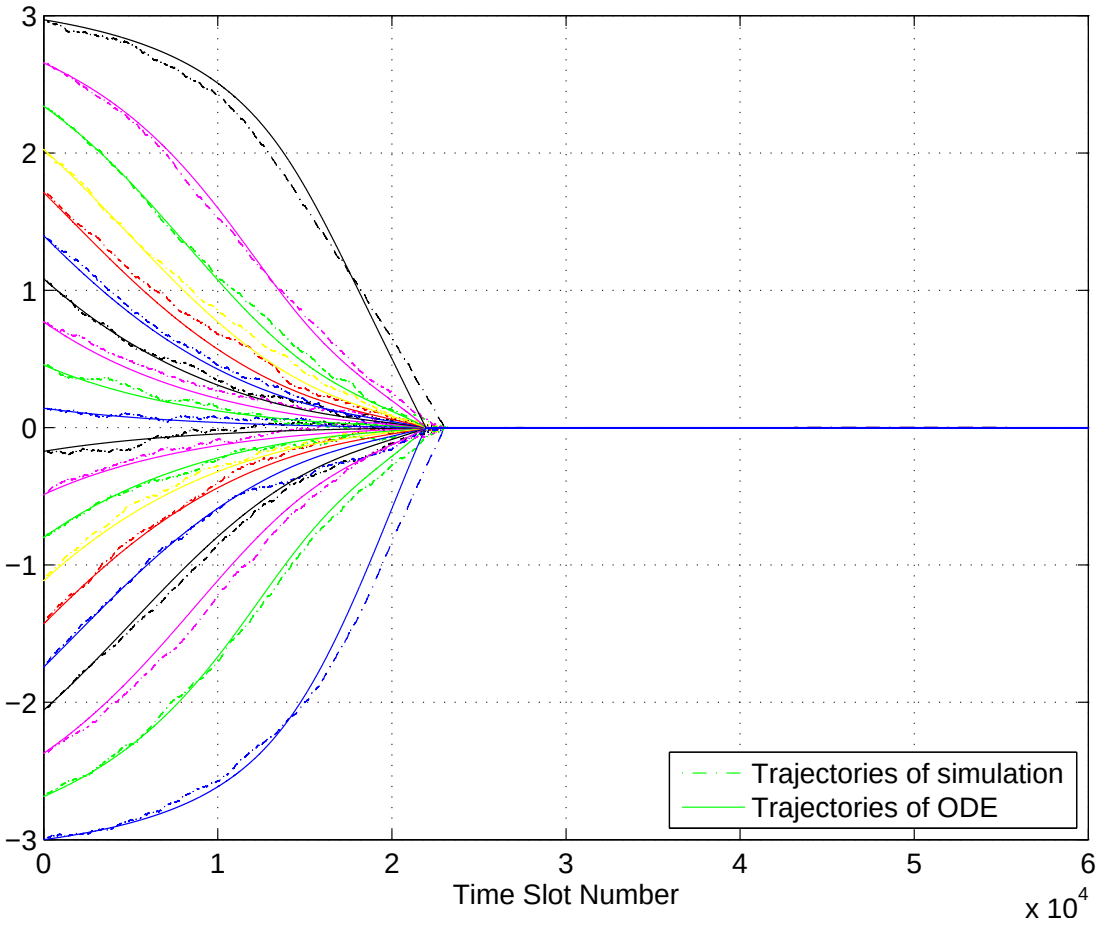



**Figure 4.7:** *Comparison of the trajectories of the phases $\hat{\Phi}$ obtained from simulation (dashed lines) with the trajectories of the ODE (solid lines) in (4.33) for the improved algorithm with $N = 20$, $\delta_0 = 6 \times 10^{-4}$. The convergence of $\hat{\Phi}$ to zero is equivalent to the convergence of the phase alignment process.*

Figure 4.8 shows the comparison of the RSS calculated using the ODE in (4.33) with the simulation results of RSS versus number of time slots with different numbers of transmitters $N = 20, 50$, and $100$. As we can see, the analytical results provide a close upper bound on the convergence speed and yield a good match with the simulation results for most of the convergence process.

Although the analytical results presented in Section 4.5 proved that the improved algorithm has a faster convergence speed of 1.634 compared to the original algorithm. However, the analysis is only valid for a small perturbation size. For a more comprehensive study, below

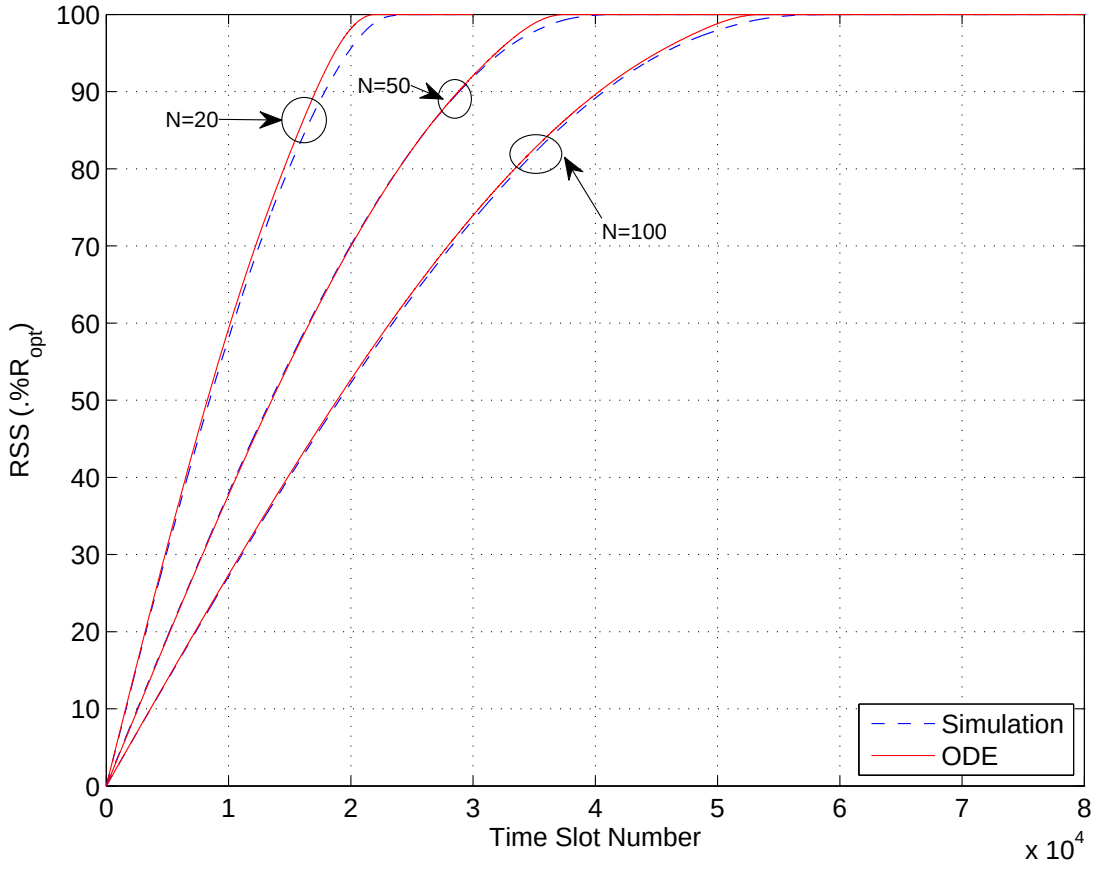



**Figure 4.8:** *Comparison of the simulation results (dashed lines) with the results obtained from the ODE (solid lines) in (4.33) for the RSS versus number of time slots with* $N = 20, 50, 100$ *and* $\delta_0 = 6 \times 10^{-4}$.

we compare the improved algorithm with the original algorithm in terms of the convergence time required to achieve a certain beamforming gain. In order to compare the two algorithms fairly and effectively, we use the same sequences of pseudo random values of $\gamma_i$ and $\psi_i$ for both algorithms and set $\varphi_i[1] = 0$.

Figure 4.9 shows the comparison of the original algorithm and the improved algorithm using the average RSS versus the number of time slots up to 2000 with $N = 100$ and $\delta_0 = \frac{\pi}{100}, \frac{\pi}{50}, \frac{\pi}{25}$. The simulation results for every curve in Figure 4.9 are averaged over $10^3$ instances. It shows that with the same value of $\delta_0$, the improved algorithm converges faster than the original algorithm at initial stages, which is consistent with our expectation in Section 4.4. However, it also shows that with the same value of $\delta_0$, the original algorithm results in a bigger RSS than the improved algorithm after a lot of iterations when the RSS gets closer to its optimum value. This

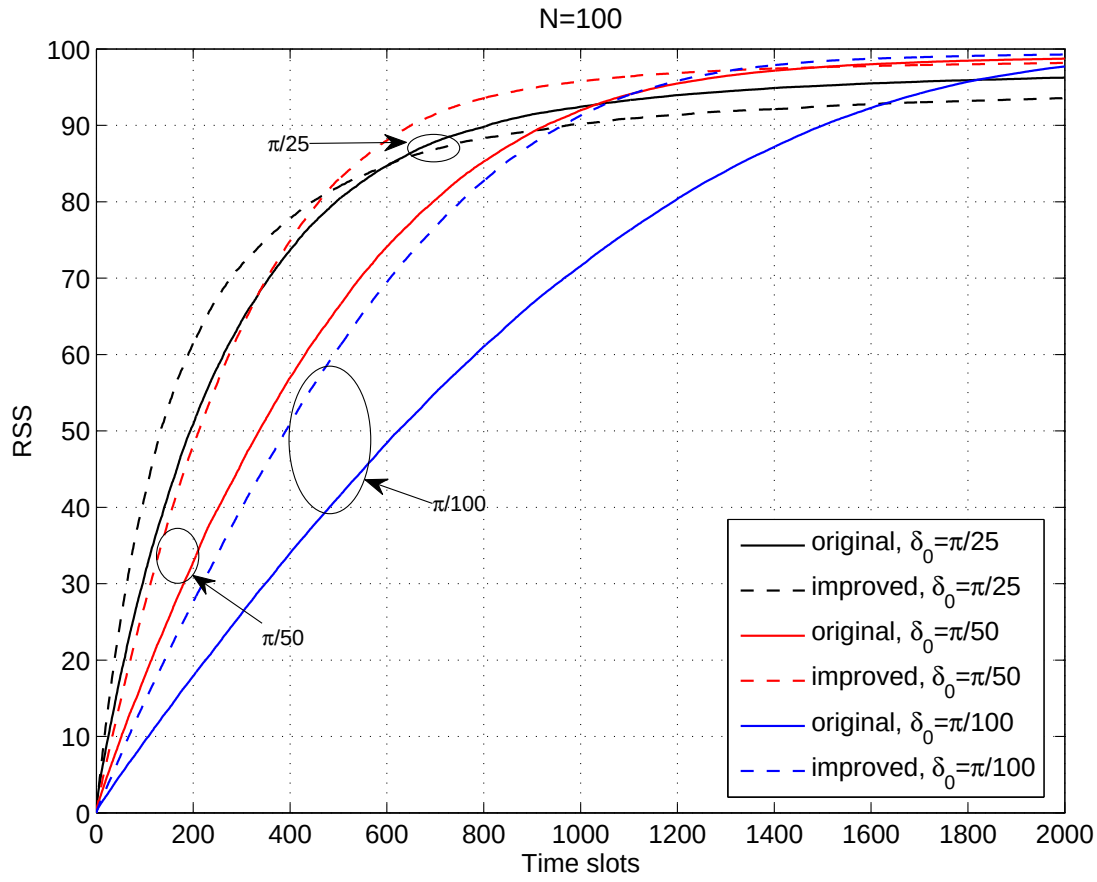



**Figure 4.9:** *Comparison of the original algorithm and the improved algorithm on the average RSS versus the number of time slots with* $N = 100$ *and* $\delta_0 = \frac{\pi}{100}, \frac{\pi}{50}, \frac{\pi}{25}$.

is because the original algorithm performs better when the phase differences among the signals arriving at the receiver become on the same order as $\delta_0$. For instance, Figure 4.10 shows the case of two transmitters from which the received signals at the receiver has a phase difference $\Delta\Phi$ smaller than $\delta_0$. When the phase difference between the two signal vectors, $\Delta\Phi$, is no bigger than the perturbation size, $\delta_0$, there leaves no space for a reduction in the phase difference when the iterations evolve. In this situation, the original algorithm keeps the phase difference unchanged while the improved algorithm results in a bigger phase difference. Accordingly, the original algorithm performs better when the RSS gets closer to its optimum value.

Figure 4.11 shows the probability of the improved algorithm performing better than the original algorithm versus the number of time slots up to 2000 with $N = 100$ and $\delta_0 = \frac{\pi}{100}, \frac{\pi}{50}, \frac{\pi}{25}$. The probability in time slot $n$ is calculated for $10^5$ instances, the number of instances that the improved algorithm leads to a bigger RSS than the original algorithm in time slot $n$ when

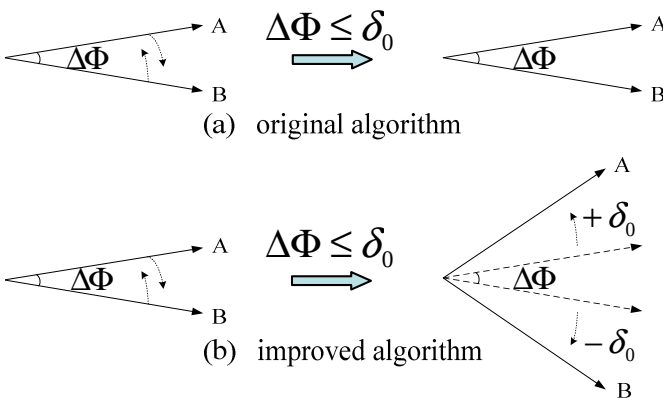



**Figure 4.10:** *Perturbation results in the case of two transmitters when $\Delta\Phi \leq \delta_0$ ($\Delta\Phi$ denotes the phase difference at the receiver). (a) applying the original algorithm; (b) applying the improved algorithm.*

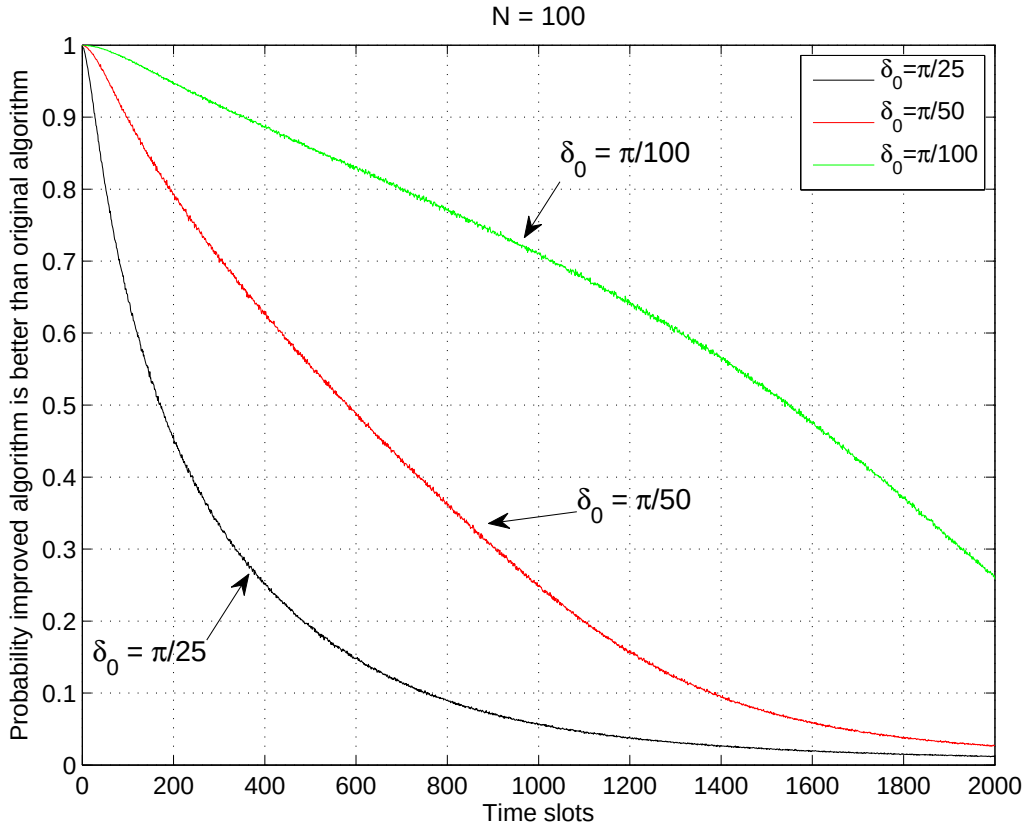



**Figure 4.11:** *Probability of improved algorithm leading to a bigger RSS than original algorithm versus the number of time slots.*

a failed perturbation happened in time slot $(n-1)$. This is divided by the total number of instances that a failed perturbation happened in time slot $(n-1)$. From Figure 4.11 we see that the probability decreases when the number of time slots increases and the probability with a bigger $\delta_0$ decreases faster than the case with a smaller $\delta_0$. These findings are consistent with our explanation above.

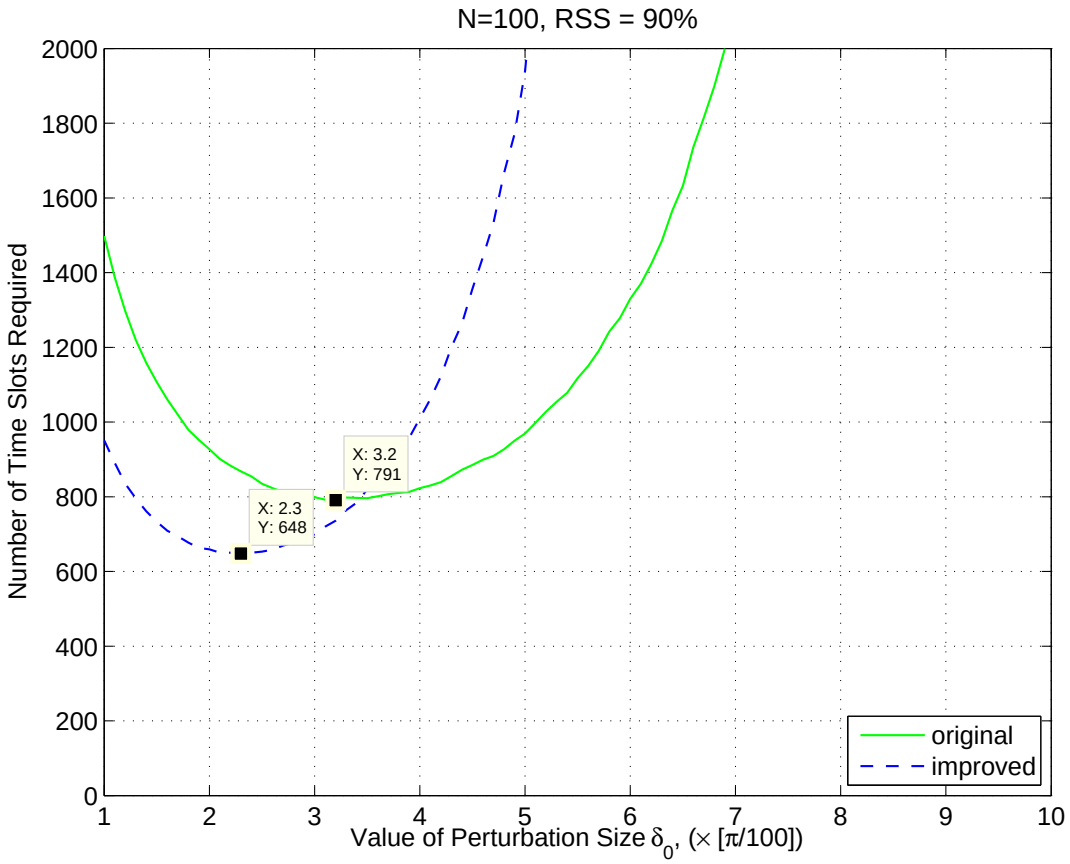



**Figure 4.12:** *Comparison of the original algorithm and the improved algorithm on the number of time slots required to achieve an RSS of* $90\%$ *of maximum with different values of* $\delta_0$.

From Figure 4.9 we see that with the same $\delta_0 = \frac{\pi}{100}$ the improved algorithm converges faster than the original algorithm, the original algorithm with $\delta_0 = \frac{\pi}{50}$ converges even much faster than both algorithms with $\delta_0 = \frac{\pi}{100}$. How can one compare the convergence speed of the two algorithms more quantitatively? Based on the average RSS versus the number of time slots, the number of time slots required to achieve an RSS of $90$ with different values of $\delta_0$ are plotted in Figure 4.12 for both algorithms. It shows that to achieve an RSS of 90, both the original algorithm and the improved algorithm have an optimum $\delta_0$ corresponding to the

minimum number of time slots. From the simulation results we see that the minimum number of time slots required for the original algorithm is $791$, while the minimum number of time slots required for the improved algorithm is $648$. This implies that the improved algorithm can converge faster than the original algorithm to achieve an RSS value of $90\%$ of maximum.

Figure 4.13 shows the minimum number of time slots required to achieve different values of RSS for both algorithms and Figure 4.14 shows the corresponding values of $\delta_0$ which result in the minimum number of time slots versus the value of RSS.

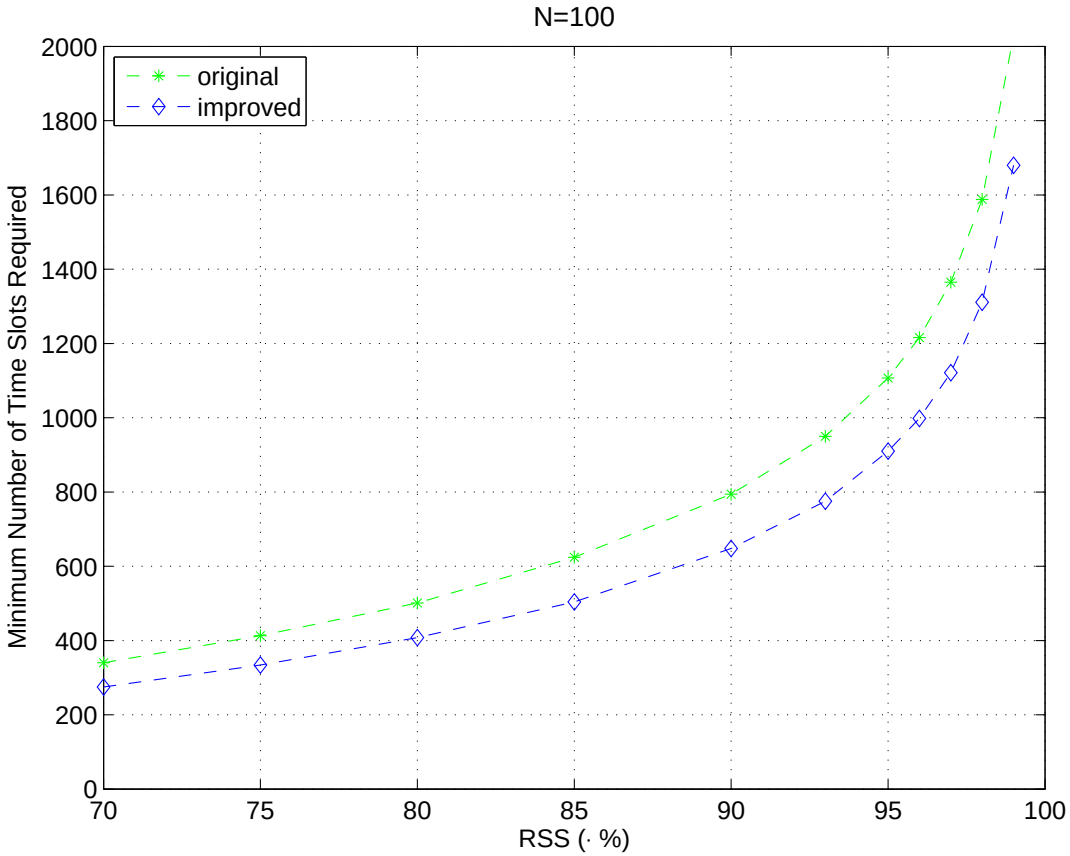



**Figure 4.13:** *Comparison of the original algorithm and the improved algorithm for the minimum number of time slots required to achieve different RSS values.*

If we denote $\delta_0 = \delta_1$ for the original algorithm, and $\delta_0 = \delta_2$ for the improved algorithm, the number of time slots $n_1$ used to achieve a certain value of RSS for the original algorithm is a function of $\delta_1$ and $R$: $n_1 = f(\delta_1, R)$. Similarly, for the improved algorithm the number of time slots $n_2 = g(\delta_2, R)$. From Figure 4.12 and Figure 4.13 we have: for any given $R$, there always exists a $\delta_2$ satisfying:

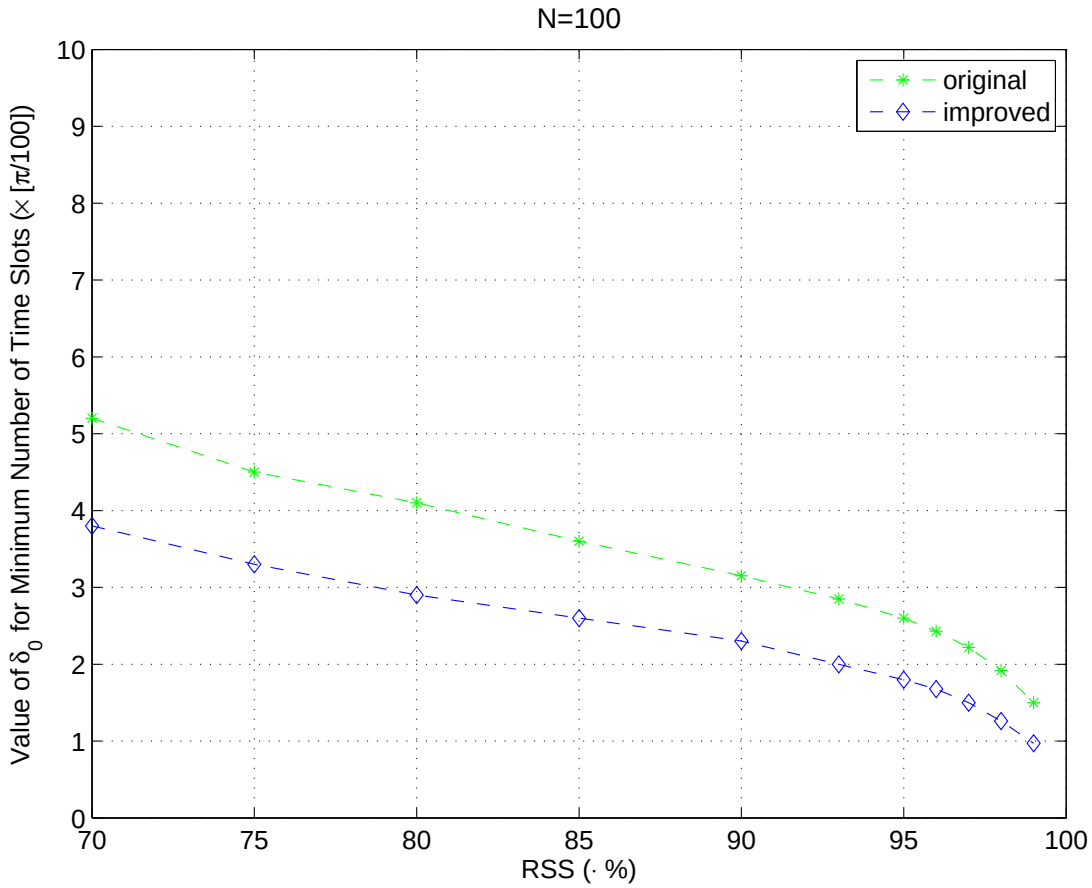



**Figure 4.14:** *Value of perturbation size $\delta_0$ which results in the minimum number of time slots to achieve different RSS values.*

$$n_2 = g(\delta_2, R) < n_1 = f(\delta_1, R), \quad \forall \delta_1 \tag{4.34}$$

It shows in Figure 4.13 that the gap between the minimum number of time slots required by the original algorithm and the improved algorithm increases with the value of RSS. For the original algorithm, we define the convergence speed to achieve an RSS value of $R$ to be inversely proportional to the minimum number of time slots required, which is expressed as:

$$v_1(R) \propto \frac{1}{\hat{n}_1(R)} \tag{4.35}$$

where $\hat{n}_1(R) = \min(n_1 = f(\delta_1, R)), \forall \delta_1$ is the minimum number of time slots required

to achieve $R$ by the original algorithm. The improvement in the convergence speed of the improved algorithm compared to the original algorithm can be expressed as:

$$\beta(R) = \frac{v_2(R) - v_1(R)}{v_1(R)} = \frac{\hat{n}_1(R) - \hat{n}_2(R)}{\hat{n}_2(R)} \tag{4.36}$$

where $v_2(R)$ is the convergence speed for the improved algorithm and $\hat{n}_2(R)$ is the minimum number of time slots required by the improved algorithm. The improvement in the convergence speed to achieve different values of RSS as a percentage are given in Table 4.1, where $\hat{n}_1(R)$ and $\hat{n}_2(R)$ are obtained from the results plotted in Figure 4.13. It shows that to achieve a certain RSS between $70\%$ and $99\%$, the improved algorithm converges at least $20\%$ faster than the original algorithm.

| RSS | 70 | 75 | 80 | 85 | 90 | 93 |
|---|---|---|---|---|---|---|
| $\beta$ | $23.64\%$ | $23.65\%$ | $22.79\%$ | $23.81\%$ | $22.53\%$ | $22.58\%$ |
| RSS | 95 | 96 | 97 | 98 | 99 | |
| $\beta$ | $21.65\%$ | $21.84\%$ | $21.77\%$ | $21.13\%$ | $20.77\%$ | |

**Table 4.1:** *Improvement in convergence speed, $\beta$ from equation (4.36), to achieve different RSS of the improved algorithm compared to the original algorithm.*

## 4.7 Summary

We proposed an improved algorithm for distributed transmit beamforming based on the original one-bit feedback algorithm presented in [4]. The improved algorithm yields a $20\%$ faster convergence speed compared to the original algorithm in static channels. It makes use of the negative feedback information in a single time slot to enhance the probability of generating better phase changes. It does not require any more information exchange or hardware support than the original algorithm. Also, it keeps all the benefits of the original algorithm, such as the simplicity and scalability. Simulation results confirm the potential of the improved algorithm in improving the convergence speed and show the minimum number of time slots required to achieve a certain beamforming gain and the corresponding value of perturbation size used. In the next Chapter, we will further explore the negative feedback information in successive time slots to improve the convergence speed of phase alignment and extend the improved algorithm into time-varying channels.

# Chapter 5
# A Hybrid Algorithm for Phase Alignment in Slowly Time-varying Channels

We continue to improve the one-bit feedback algorithm for achieving phase alignment at the intended receiver in distributed transmit beamforming. Besides the reverse perturbation scheme discussed in Chapter 4, intuitively, adjusting perturbation sizes during the convergence process would also help improve the convergence speed of phase alignment. The question is how to implement it in practice based on only one-bit feedback information. In Chapter 4, we studied the convergence performance of the one-bit feedback algorithms in the ideal channels which have static phase responses. However, in practice, channel phase responses change in time due to moving scatters or obstructions in the propagation environment. Also, sensor nodes experience phase drifts in signal generation due to oscillator internal noise. Therefore, the one-bit feedback algorithm must be modified to be robust to random phase drifts before its practical implementations. In this Chapter, we will further exploit the negative feedback information to improve the one-bit feedback algorithm and address the above issues.

## 5.1 Introduction

In Chapter 4, we have proposed an improved algorithm based on the original algorithm in the literature to achieve phase alignment at the receiver in static channels for distributed transmit beamforming. The improved algorithm provides a superior performance in the convergence speed compared to the original algorithm while maintaining all of its advantages, such as simplicity in implementation and scalability to a large number of nodes [4]. The benefit in the convergence speed is obtained by making use of the negative feedback information in a single time slot and taking a reverse-perturbation scheme to generate better phase changes at transmitters. However, both the original algorithm and the improved algorithm are using a fixed perturbation size across time slots. When the phase differences among signals arriving at the

receiver are diminished to the same order of the perturbation size, the probability of generating better phase changes decreases as the algorithm converges, which can be inferred from Figure 4.10 and Figure 4.11. This implies that a decreasing perturbation size may be used to improve the convergence speed. This is similar to the idea of variable step-size least mean square (LMS) algorithms in the literature. It shows in [90] that there exists a tradeoff between the steady state performance and the speed of adaptation in the LMS algorithms with a fixed step size. The issue of optimization of step size was studied in [91], [92], [93] to improve the performance of LMS algorithms, which can provide fast convergence at early stages while obtaining small final misadjustment errors. In a similar way to the variable step-size LMS algorithms in the literature, the one-bit feedback algorithm can use a large step size at early stages to speed up the convergence process. When the algorithms get close to the optimum solution, a smaller step size can yield a better steady state performance.

In Chapter 4, we proposed an improved version of the one-bit feedback algorithm which has a faster convergence speed under static channel conditions. Particularly, in the system model expressed in equation (4.1), it is assumed that the phase offset at transmitters, $\gamma_i$, and the channel phase responses, $\psi_i$, are static during the convergence process. However, in practice the assumption of static channel phase responses does not hold when either the receiver, surrounding obstructions or scatters are in relative motion to the transmitters [94]. In addition, phase drifts in the phase settings at transmitters are introduced by oscillator internal noise or phase noise [95] which cannot be eliminated. Therefore, the one-bit feedback algorithm for phase synchronization must be modified to track time-varying channels or to be robust to phase drifts before possible practical implementation is viable. It is well known that the performance of transmit beamforming is very sensitive to the phase changes in time-varying channels [96]. In conventional beamforming, CSI is measured at the receiver and periodically conveyed to the transmitter through a feedback link. The transmitter then computes an antenna weighting scheme corresponding to the available CSI, which can result in a good beamforming gain in a slow fading environment [97]. In distributed beamforming, CSI may be measured and periodically fed back to transmitters in the same way. But applying CSI at transmitters in distributed beamforming requires a lot of information exchange and coordination among sensor nodes, which brings unacceptably high overhead, especially with a large number of nodes.

In this Chapter, we further exploit the negative feedback information to improve the convergence performance of the original one-bit feedback algorithm for achieving phase alignment

while retaining the advantages of the original algorithm. The system model considered in this Chapter is the same as the one described in Section 4.2 in Chapter 4. We first propose a decreasing perturbation-size scheme based on the original algorithm which still requires only one-bit feedback in each iteration. The scheme makes use of the negative feedback information in successive time slots to adjust perturbation sizes and has the potential to improve the convergence speed with a wide range of parameter selections. Then, we show that the decreasing perturbation-size scheme proposed in this Chapter and the reverse-perturbation scheme proposed in Chapter 4 can be combined to generate a hybrid algorithm, which can provide over $40\%$ faster convergence speed compared to the original algorithm. Further, we modify the hybrid algorithm to track time-varying channels without the knowledge of phase drift speed. The modified hybrid algorithm has the ability to detect variations in the speed of channel phase changes and adjust perturbation sizes adaptively according to the speed, which enhances the robustness of the one-bit feedback algorithm in practical implementations.

## 5.2 A decreasing perturbation-size scheme

In Chapter 4, we studied the performance of the original algorithm with a fixed perturbation size across time slots. Intuitively, the original algorithm can have a faster convergence speed by adopting a bigger perturbation size at initial stages of the convergence process and requires a smaller perturbation size when the resulted beamforming gain approaches its optimum value. When the phase differences among signals arriving at the receiver are large, a bigger perturbation size can accelerate the convergence speed. When the phase differences become smaller, a bigger perturbation size will decelerate the convergence speed or even cease the convergence process. In [4], the authors derived an analytical formula of the optimal perturbation size in each time slot for the original algorithm. The optimal perturbation size $\Delta_0$ in time slot $(n+1)$ is expressed as a function of $(R[n]/R_{\rm opt})$, where $R[n]$ is the RSS in time slot $n$, defined in (4.3), $R_{\rm opt}$ represents the RSS with perfect phase alignment. Following the derivations in [4], the numerically computed $\Delta_0$ for the original algorithm with $N = 100$ transmitters can be easily obtained, as plotted in Figure 5.1. In each time slot, all sensor nodes adjust their perturbation sizes to the same optimal value shown in Figure 5.1. The analysis in [4] gives a fundamental understanding of the original algorithm, and can be used as a good metric for comparison and algorithm design. However, the value of $R_{\rm opt}$ is hard to obtain in practice before the phase training process converges and feedback of the optimal value requires several bits instead of

one. A practical version of the algorithm using a variable perturbation size is required.

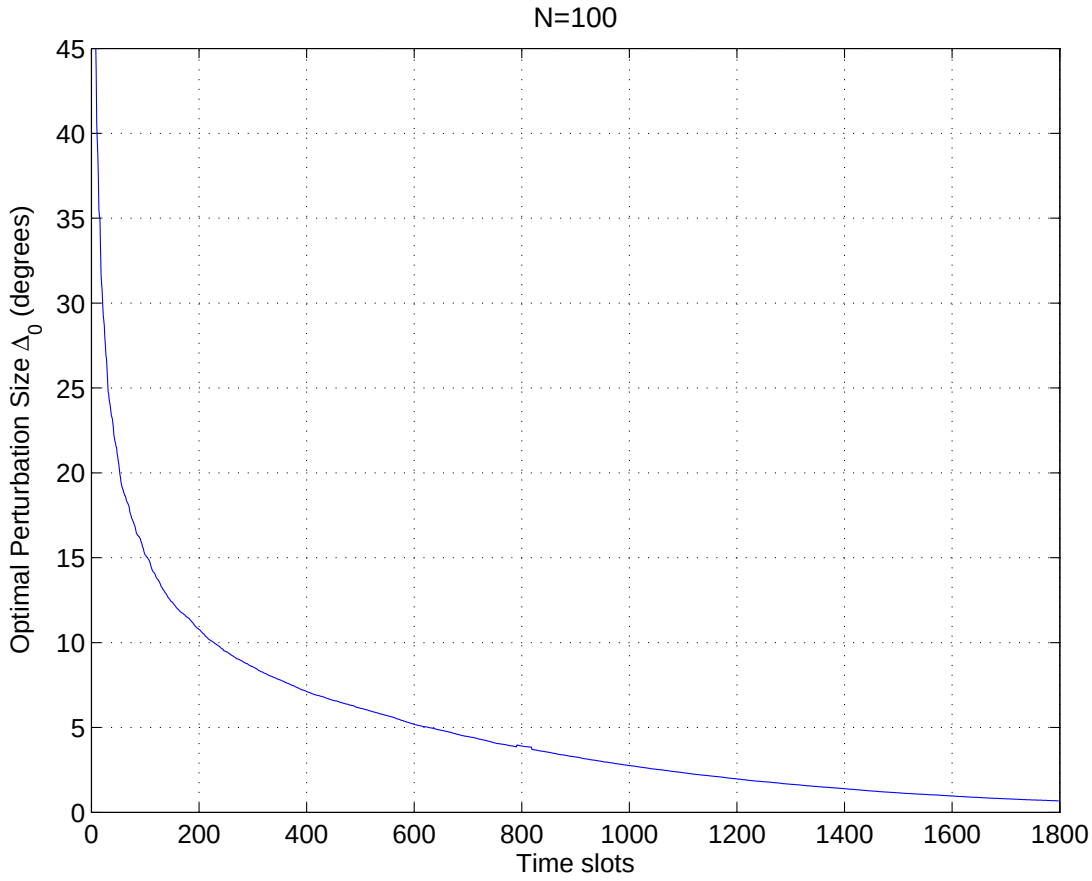



**Figure 5.1:** *Optimal perturbation size $\Delta_0$ versus the number of time slots for the original one-bit feedback algorithm with $N = 100$.*

### 5.2.1 Algorithm description

From Figure 5.1 we see that the optimal value of perturbation size decreases as the number of iterations increases. Based on this point, we adopt a decreasing size for $\delta_0$ in our practical design. The transmitters will adopt a smaller $\delta_0$ when the number of successive negative feedback steps $C_N$ meets a certain threshold $C_T$. The decreasing perturbation-size scheme is described as follows.

1. At time slot $n$, each transmitter records its best known phase used for beamforming, $\theta_i[n]$, in memory and adds a random perturbation, $\delta_i[n] = \pm\delta_0$ (with equal probability for "+" and "-"), to it. (We set $\theta_i[1] = 0$).

2. All transmitters use their new adaptive components, $\varphi_i[n] = \theta_i[n] + \delta_i[n]$, to perform transmit beamforming.

3. The receiver measures the RSS, $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$, and compares it with the best RSS in memory. The receiver updates the best RSS in memory and feeds back (error free) one-bit of information to all transmitters conveying whether the RSS has been improved or not.

4. If the RSS has been improved, all transmitters adopt their perturbed phases and update their best known phases to be $\theta_i[n+1] = \varphi_i[n] = \theta_i[n] + \delta_i[n]$ for the next time slot $(n+1)$. Otherwise, all transmitters discard the perturbed phases and keep the best known phases as before, $\theta_i[n+1] = \theta_i[n]$, for the next time slot $(n+1)$. Meanwhile, the transmitters record the number of successive failed perturbations with a counting variable $C_N$. If it is a positive feedback indicating a successful perturbation, $C_N$ will be cleared to zero. Otherwise, the value of $C_N$ will be increased by 1 until it surpasses a certain threshold $C_T$. When $C_N \geq C_T$, $C_N$ is cleared to zero and all transmitters adopt a new perturbation size $\delta_0 = \delta_0 \cdot R_D$ $(0 < R_D < 1)$, where $R_D$ is a fixed decreasing ratio of perturbation size.

   The algorithm then repeats these four steps.

The mathematical expressions of the decreasing perturbation-size scheme are the same as the original algorithm expressed in (4.5), (4.6) except adding the following:

$$C_N = \begin{cases} 0 & R[n] > R_{\text{best}}[n] \\ C_N + 1 & \text{otherwise} \end{cases} \tag{5.1}$$

$$\delta_0 = \begin{cases} \delta_0 & C_N < C_T \\ \delta_0 \cdot R_D & C_N \geq C_T \end{cases} \tag{5.2}$$

The decreasing perturbation-size scheme makes use of the negative feedback information in successive time slots to adjust the perturbation size. It still requires only one-bit feedback per iteration and requires no extra hardware or information exchange. It is a simple but effective scheme which can be easily applied into practical implementations.

### 5.2.2 Simulation results

We present some simulation results to study the performance of the decreasing perturbation-size scheme as a function of two parameters: the threshold for successive negative feedback steps $C_T$, and the decreasing ratio of perturbation size $R_D$. The simulation results for every point plotted in the following figures are averaged over 800 instances.

Figure 5.2 shows the average number of time slots required to achieve an RSS of $90\%R_{\text{opt}}$ with different values of $C_T$ and $R_D$ for the decreasing perturbation-size scheme. There exists an optimum value for the parameter selection which can result in the minimum number of time slots. From the simulation results we see that the minimum number of time slots required to achieve an RSS of $90\%R_{\text{opt}}$ is 688 time slots, which is obtained with $C_T = 11$ and $R_D = 0.75$.

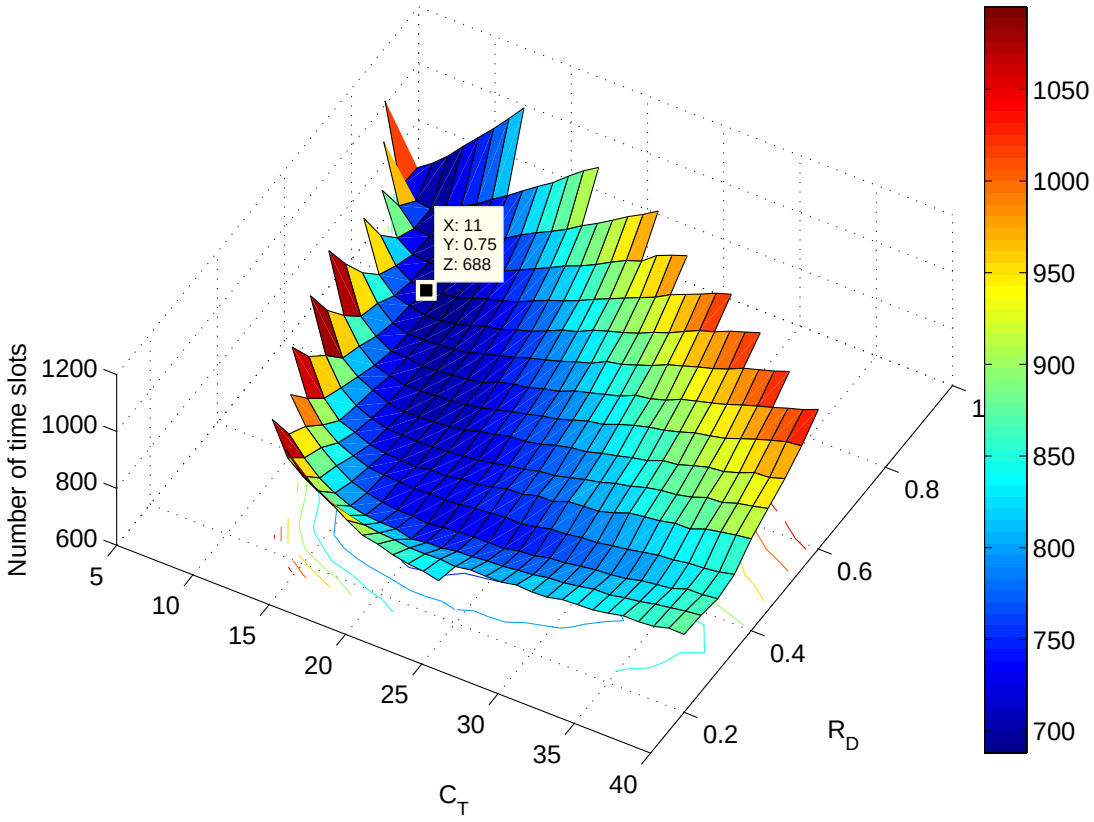



**Figure 5.2:** *Simulation results for the decreasing perturbation-size scheme showing the average number of time slots required to achieve an RSS of $90\%R_{\text{opt}}$ with different values of $C_T$ and $R_D$.*

In Chapter 4, we have studied the performance of the original algorithm in a similar way, where

the minimum number of time slots required to achieve an RSS of $90\%R_{\text{opt}}$ by the original algorithm was 791 time slots. Figure 5.3 shows the contour plot of Figure 5.2. It shows that the decreasing perturbation-size scheme can achieve an RSS of $90\%R_{\text{opt}}$ within 790 time slots with a wide range of parameter selections. This shows the robustness of the algorithm to small mismatches in parameter settings.

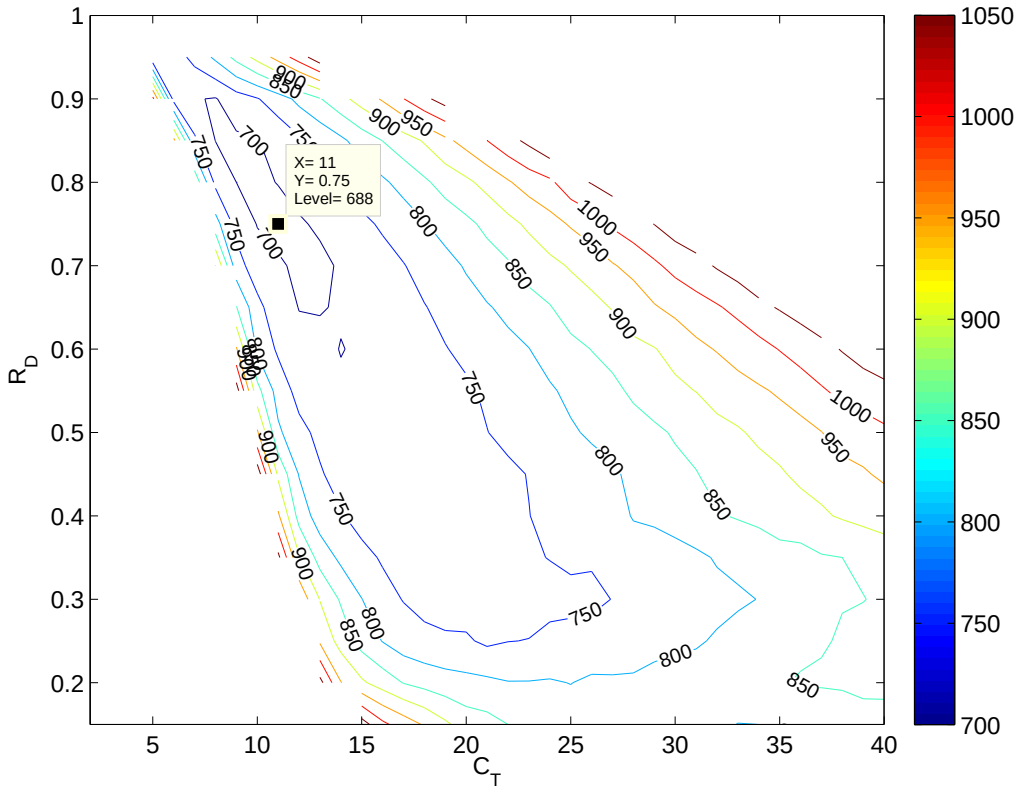



**Figure 5.3:** *Contour plot of the average number of time slots required to achieve an RSS of $90\%R_{\text{opt}}$ with different values of $C_T$ and $R_D$ for the decreasing perturbation-size scheme.*

## 5.3 Hybrid algorithm

In Section 4.4, we proposed a reverse perturbation scheme to improve the convergence speed of the original algorithm by exploiting negative feedback information in a single time slot. In Section 5.2, we proposed a decreasing perturbation-size scheme to improve the convergence speed of the original algorithm by exploiting negative feedback information in successive time slots. For simplicity, we denote the reverse-perturbation scheme as Scheme 1 and the decreasing perturbation-size scheme as Scheme 2 in the following sections. Both schemes are using only

one-bit feedback in each iteration and can be easily applied into practical implementations. In this Section, we show that the two schemes can be combined to provide a significant advantage in the convergence speed compared to the original algorithm.

### 5.3.1 Algorithm description

Scheme 1 and Scheme 2 speed up the convergence process based on different approaches. Combining Scheme 1 and Scheme 2 yields a hybrid algorithm which can provide a significant improvement in the convergence speed in the phase training process. The hybrid algorithm is summarized in Table 5.1.

| |
|---|
| **Initialization**: $C_N = 0$; $\delta_0 = \frac{\pi}{4}$; $\theta_i[1] = 0$; $\epsilon_i[1] = 0$; $R_{\text{best}}[1] = 0$. |
| **Iterate**: |
| 1. Set $\delta_i[n] = \pm\delta_0$ ("+" or "-" with equal probability). |
| 2. Use $\varphi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$ to perform beamforming. |
| 3. Estimate $R[n] = \left\lvert \sum_{i=1}^{N} e^{j\Phi_i[n]} \right\rvert$; |
| Update $R_{\text{best}}[n+1] = \max(R_{\text{best}}[n], R[n])$. |
| $\longrightarrow$ (*One bit feedback.*) |
| 4. If $R[n] > R_{\text{best}}[n]$ |
| $\theta_i[n+1] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$; $\epsilon_i[n+1] = 0$; $C_N = 0$; |
| else |
| $\theta_i[n+1] = \theta_i[n]$; $\epsilon_i[n+1] = -\delta_i[n]$; $C_N = C_N + 1$; |
| if $C_N \geq C_T$ |
| $\delta_0 = \delta_0 \cdot R_D$; $C_N = 0$; |
| end |
| end |

**Table 5.1:** *Summary of the Hybrid Algorithm*

### 5.3.2 Simulation results

We present some simulation results to study the convergence performance of the hybrid algorithm over static channels, and compare it with the performance of the original algorithm. The simulation results also reveal the advantages of Scheme 1 (the reverse-perturbation scheme) and Scheme 2 (the decreasing perturbation-size scheme). To make a fair comparison, the hybrid algorithm does not need any more information exchange compared than the original algorithm, and in each iteration there is only one phase setting used for beamforming and one-bit feedback from the receiver which match to the original algorithm.

Figure 5.4 shows the average number of time slots required to achieve an RSS of $90\%R_{\text{opt}}$ with $N = 100$ transmitters for the hybrid algorithm. It shows that the hybrid algorithm can achieve $90\%R_{\text{opt}}$ within 700 time slots over a wide range of parameter selections, while the minimum number of time slots is 549 obtained with $C_T = 8$, $R_D = 0.7$. The number of time slots in the 3D plot has a fairly flat surface. This reveals the robustness of the hybrid algorithm to small mismatches in parameter settings.

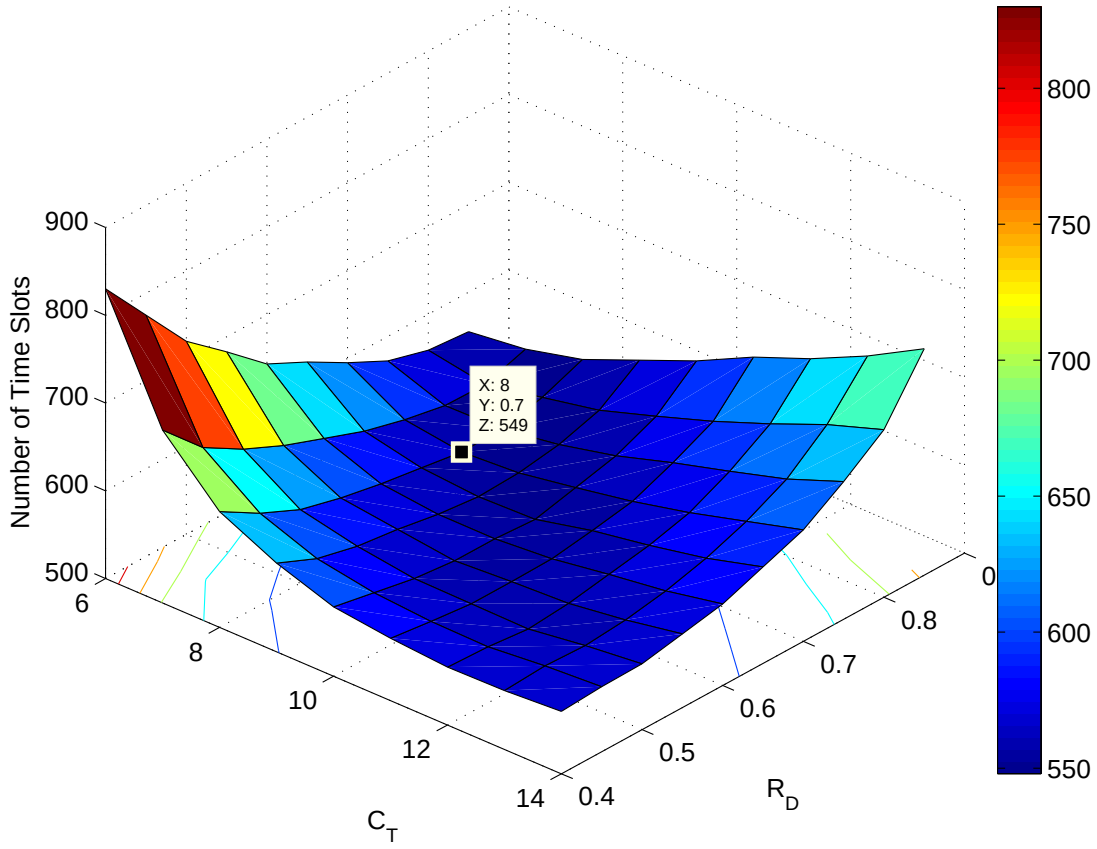



**Figure 5.4:** *The average number of time slots required to achieve $90\%R_{\text{opt}}$ with $N = 100$ for the hybrid algorithm, where $C_T$ is the threshold for successive negative feedback steps and $R_D$ is the decreasing ratio of perturbation size.*

In Figure 5.5, we compare the convergence speed of four algorithms for $N = 100$: the hybrid algorithm, Scheme 2, the original algorithm with optimal perturbation size and the original algorithm with a static perturbation size. The curve of the hybrid algorithm is plotted with $C_T = 8$, $R_D = 0.7$. The curve of the original algorithm with optimal perturbation size $\Delta_0$ for each time slot is plotted based on the analysis in [4], and the value of $\Delta_0$ versus the number of time slots was given in Figure 5.1. The parameter settings for the original algorithm with a

static perturbation size and Scheme 2 are the optimal settings obtained from Figure 4.12 and Figure 5.2. From Figure 5.5 we see the hybrid algorithm has the best performance among the four, and that Scheme 2 can achieve performance close to the original algorithm with optimal perturbation sizes. Comparing the hybrid algorithm with Scheme 2, we see that they have a similar convergence speed in their initial stages and the hybrid algorithm has a better performance due to the contribution of Scheme 1. In achieving an RSS of $90\%R_{\text{opt}}$, there is a big gap of $791-549 = 242$ time slots between the hybrid algorithm and the original algorithm with a static perturbation size or, in other words, the hybrid algorithm has a $\left(\frac{1}{549} - \frac{1}{791}\right) \Big/ \left(\frac{1}{791}\right) \approx 44\%$ faster convergence speed compared to the practical original algorithm. This gain in the convergence speed is obtained by exploiting negative feedback information in the iterations.

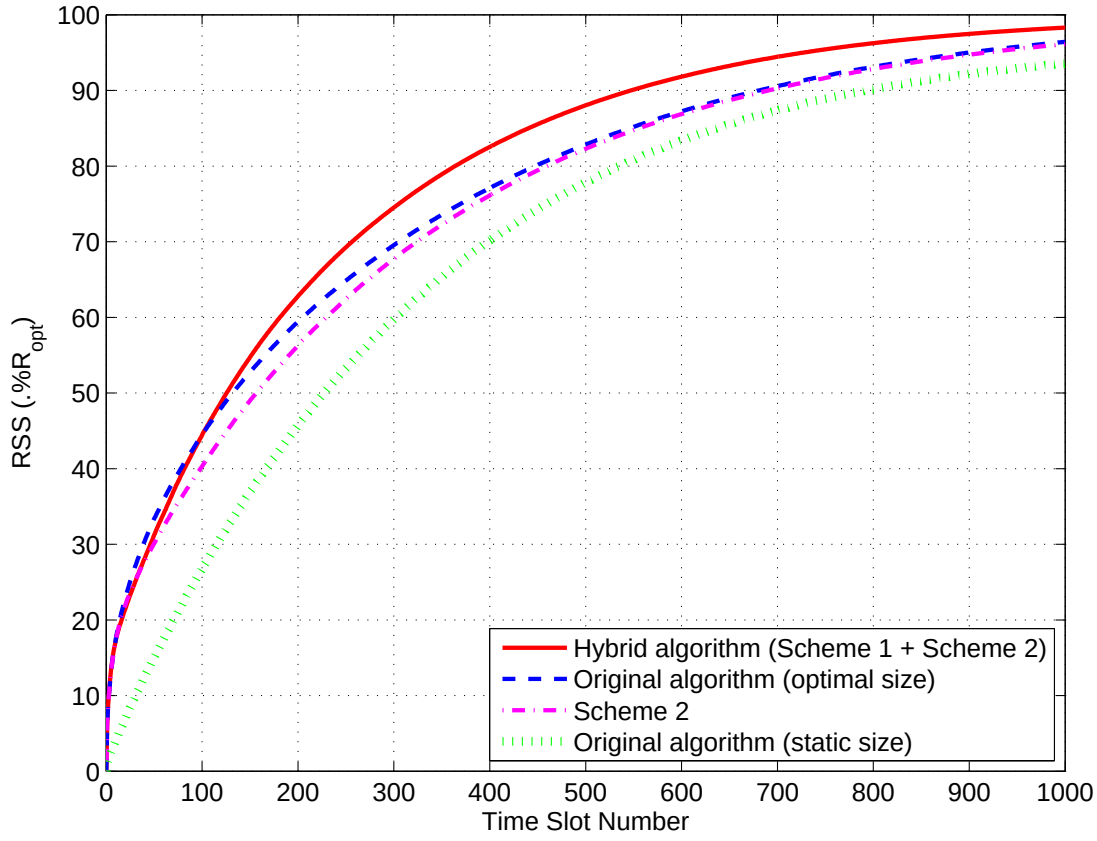



**Figure 5.5:** *Comparison of the hybrid algorithm with the original algorithm on the received signal strength versus time slots for $N = 100$.*

## 5.4 Tracking time-varying channels

In this section, we show that the hybrid algorithm proposed above can be modified to track time-varying channels while maintaining its fast convergence speed. What is more, the modified hybrid algorithm has the ability to detect variations in the speed of channel phase changes and adjust perturbation sizes adaptively according to the speed. In order to focus on the effect of changes in channel phase responses, we still assume unit channel power gain from each transmitter to the receiver, but model the channel phase response from transmitter $i$ to the receiver as $\Psi_i[n] = \psi_i + \lambda_i[n]$, where $\psi_i \sim \mathrm{uniform}[0, 2\pi)$ are static during the convergence process as assumed in Section 4.2. The phase drift components $\lambda_i[n]$ are assumed to be independent, identically distributed across transmitters and uncorrelated in time slots with a uniform distribution $\lambda_i[n] \sim [-\Lambda_0, \Lambda_0]$ as in [35], where $\Lambda_0$ is termed as the phase drift speed. The variations in phase offset at transmitters due to the oscillator internal noise can be modeled in the same way.

Most work in the literature on the one-bit feedback algorithms is focused on static channel conditions. Few of them extended the algorithms to time-varying channels apart from [35]. In [35], the authors modified the original one-bit feedback algorithm to track time-varying channels by lowering the criterion at the receiver every time it encounters a negative feedback step:

$$R_{\mathrm{best}}[n+1] = \begin{cases} R[n] & R[n] > R_{\mathrm{best}}[n] \\ R_{\mathrm{best}}[n] \cdot \rho & \mathrm{otherwise} \end{cases} \qquad (0 < \rho < 1) \tag{5.3}$$

where $\rho$ is the discounting factor which reflect the expected deterioration due to channel variations. The reason for lowering the criterion is because phase drifts in time-varying channels cause reductions in the RSS, which make it hard or even impossible to achieve an $R[n]$ greater than $R_{\mathrm{best}}[n]$ if the received phases at the receiver become highly coherent in a previous time slot $(n-1)$. Even with right perturbations which can reduce phase differences in static channels, in time-varying channels the resulted $R[n]$ may be less than $R_{\mathrm{best}}[n]$ due to the effect of phase drifts. If the receiver still compare $R[n]$ with $R_{\mathrm{best}}[n]$ as in the original algorithm, the perturbations will be discarded and the phase differences become larger. The achievable RSS with random perturbations keeps decreasing in the following time slots due to phase drifts and becomes less than $R_{\mathrm{best}}[n]$. No perturbations will be retained and the received phases will lose coherence. By lowering the criterion of achievable RSS, the algorithm in [35] can achieve a

reasonable beamforming gain in time-varying channels but it requires accurate knowledge of the phase drift speed $\Lambda_0$ in channel variations in order to set appropriate values for $\rho$ and the perturbation size $\delta_0$. For more details of the algorithm, please see [35].

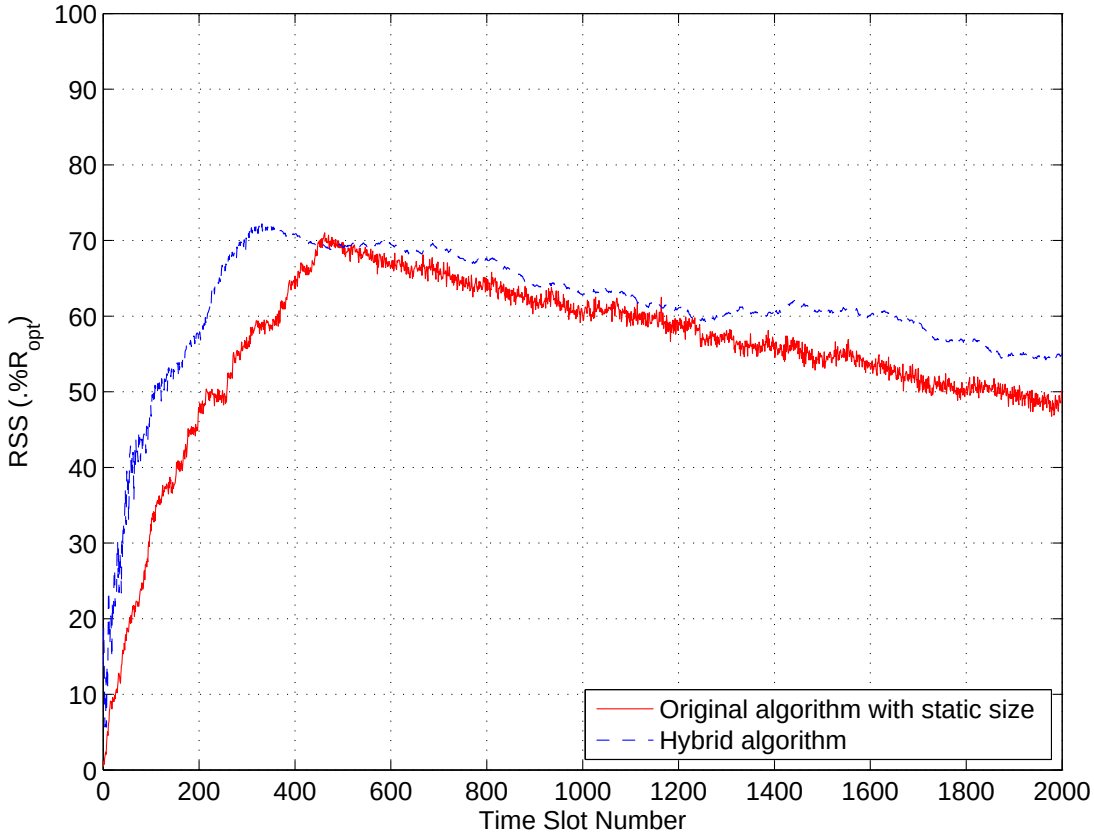



**Figure 5.6:** *One simulated instance of the original algorithm ($\delta_0 = \frac{3.2\pi}{100}$) and the hybrid algorithm ($C_T = 8$, $R_D = 0.7$) in time-varying channels with channel phase drift speed $\lambda_i[n] \sim [-\frac{\pi}{100}, \frac{\pi}{100}]$ for $N = 100$.*

We meet the following difficulties in the algorithm design under time-varying channel conditions. First, if the received phases at the receiver $\Phi_i[n]$ become highly coherent in time slot $n$, the corresponding RSS value cannot be surpassed by subsequent perturbations as the RSS will reduce again due to channel variations. Therefore, the RSS judgement rule $R[n] > R_{\text{best}}[n]$ at the receiver is not sufficient in time-varying channels. Figure 5.6 shows one simulated instance of the original algorithm and the hybrid algorithm in time-varying channels. As we can see, both algorithms are not reliable in time-varying channels and the RSS decreases after a certain time point. Second, since the hybrid algorithm described above in Section 5.3 keeps reducing the perturbation size, obviously, it cannot track time-varying channels when the perturbation

size becomes smaller than the phase drift speed. Third, in static channels, successive negative feedback steps only suggest that the perturbation size is too big to converge. However, in time-varying channels, this may also result from the effect of channel variations, which in contrast may require a bigger perturbation size. Fourth, successive positive feedback steps are not available to aid the design.

A straightforward solution to overcome these difficulties is to apply the hybrid algorithm described in Section 5.3 to the initial stages of the convergence process. When the phase drift speed $\Lambda_0$ is fairly small compared to the perturbation size $\delta_0$, the effect of channel variations on the RSS is negligible. Therefore, the initial stages of the convergence process can be viewed as under static channel conditions. When the perturbation size falls to a certain value close to the phase drift speed $\Lambda_0$, the transmitters and the receiver then change to follow the algorithm described in [35]. Such a solution can improve the convergence speed and provide a reliable beamforming gain under the time-varying channel conditions. However, the transmitters still requires accurate knowledge of the phase drift speed in channel variations.

### 5.4.1 A modified hybrid algorithm

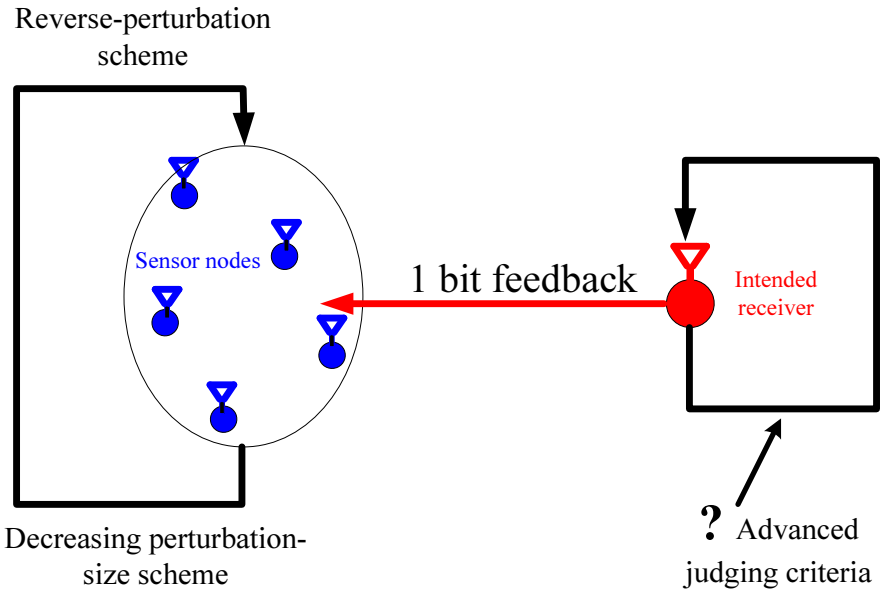



**Figure 5.7:** *Diagram of blocks in the one-bit feedback algorithm design.*

Figure 5.7 shows a diagram of blocks in the one-bit feedback algorithm for design purposes. In the above sections, we have proposed a reverse-perturbation scheme and a decreasing perturbation-size scheme, both of which are focusing on the transmitter side to improve the algorithm

performance. Under the time-varying channel conditions, the simple judgement rule $R[n] > R_{\text{best}}[n]$ at the receiver is not sufficient and more advanced judging criteria are required to cooperate with the random search process at transmitters.

We present a modified hybrid algorithm which can track time-varying channels without the knowledge of the phase drift speed $\Lambda_0$. It is summarized in Table 5.2 and explained as follows.

**Initialization**: $C_N = 0$; $\delta_0 = \frac{\pi}{4}$; $\theta_i[1] = 0$; $\epsilon_i[1] = 0$; $R_{\text{best}}[1] = 0$; $R_{\min}[1] = 0$.

**Normal mode, iterate**:

1. Set $\delta_i[n] = \pm\delta_0$ ("+" or "-" with equal probability).
2. Use $\varphi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$ to perform beamforming.
3. Estimate $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$;
   Update $R_{\text{best}}[n+1] = \max(R_{\text{best}}[n], R[n])$;
   Update $R_{\min}[n+1] = \min(R_{\min}[n], R[n])$.

$\longrightarrow$ (*One bit feedback.*)

4. If $R[n] > R_{\text{best}}[n]$
   $\theta_i[n+1] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$; $\epsilon_i[n+1] = 0$; $C_N = 0$;
   else
   $\theta_i[n+1] = \theta_i[n]$; $\epsilon_i[n+1] = -\delta_i[n]$; $C_N = C_N + 1$;
   if $C_N \geq C_T$
   $\longrightarrow$ (*Enter the testing mode in the next time slot.*)
   end
   end

**Testing mode (one time slot)**:

1>. Use $\varphi_i[n] = \theta_i[n]$ to perform beamforming.

2>. Estimate $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$;
   Update $R_{\text{best}}[n+1] = R_{\min}[n+1] = R[n]$; (reactivation)
   Compute $S_C = |R_{\text{best}}[n] - R[n]|$; (estimation of the channel drift speed)
   Compute $S_P = R_{\text{best}}[n] - R_{\min}[n]$. (estimation of the perturbation 'catch-up' speed)

$\longrightarrow$ (*One bit feedback.*)

3>. If $S_P > 2S_C$
   $\delta_0 = \delta_0 \cdot R_D$; (decrease the perturbation size)
   else
   $\delta_0 = \delta_0 / R_D$; (increase the perturbation size)
   end

4>. Set $C_N = 0$; $\theta_i[n+1] = \theta_i[n]$; $\epsilon_i[n+1] = \epsilon_i[n]$.

$\longrightarrow$ (*Exit the testing mode.*)

**Table 5.2:** *Summary of the Modified Hybrid Algorithm to Track Time-varying Channels*

The modified hybrid algorithm operates in two modes, the *normal mode* and the *testing mode*. Operations in the normal mode are similar to the operations presented in Section 5.3 except for the following. Besides updating the best RSS in memory at the receiver, in step 3 the receiver

also records the minimum RSS in memory, $R_{\min}[n+1] = \min(R_{\min}[n], R[n])$. In step 4, when the number of successive negative feedback steps $C_N$ meets the threshold $C_T$, instead of decreasing the perturbation size $\delta_0$, the transmitters and the receiver enter the testing mode for one time slot.

We define the successive time slots taking the same perturbation size as a *size period*. In the testing mode in time slot $n$, instead of performing phase perturbations, the transmitters use their best known phases to perform transmit beamforming, $\varphi_i[n] = \theta_i[n]$. The receiver measures the corresponding RSS $R[n]$, set it as the new best RSS, $R_{\text{best}}[n+1]$, and the new minimum RSS, $R_{\min}[n+1]$ in memory. Therefore, the variables $R_{\text{best}}$ and $R_{\min}$ actually record the maximum and the minimum RSS within a size period. This prevents the algorithm from operating in local rather than globally optimum phase solutions, which may be caused by the first of the difficulties listed above. The receiver then computes the absolute difference of $R_{\text{best}}[n]$ and $R[n]$:

$$S_C = \left| R_{\text{best}}[n] - R[n] \right| \tag{5.4}$$

and the difference of $R_{\text{best}}[n]$ and $R_{\min}[n]$:

$$S_P = R_{\text{best}}[n] - R_{\min}[n] \tag{5.5}$$

Since both $R_{\text{best}}[n]$ and $R[n]$ are obtained with the same adaptive component $\varphi_i[n] = \theta_i[n]$, the scalar $S_C$ can be viewed as an estimation of the channel drift speed within one size period. The scalar $S_P$ can be viewed as an estimation of the perturbation 'catch-up' speed within the same size period. By comparing $S_C$ and $S_P$, the receiver makes a judgement on the perturbation size and feeds back one bit of information telling all transmitters to adopt a larger or smaller perturbation size in the next size period. Results in [35] conclude that the perturbation 'catch-up' speed should be faster than the channel drift speed, and the perturbation size should not be too large to avoid large fluctuations in the steady-state RSS. We adopt a coefficient of 2 in comparing $S_P$ with $S_C$ in the following simulations. The modified hybrid algorithm is summarized in Table 5.2. By inserting only one time slot between two size periods, the modified hybrid algorithm has the ability to track time-varying channels and adjust perturbation sizes adaptively according to the rates of phase drift. The overhead of implementing this solution is

very low.

### 5.4.2 Simulation results

We present some simulation results to study the performance of the modified hybrid algorithm and verify its ability to track time-varying channels. We show that the modified algorithm can not only achieve phase alignment in time-varying channels which have a constant phase drift speed over time, it is also robust to time-varying channels which have variable rates of phase drift.

Figure 5.8 shows the performance of the modified hybrid algorithm with $N = 100$ transmitters, $C_T = 8$, $R_D = 0.7$, in time-varying channels with different phase drift speeds. As we see, the modified hybrid algorithm can achieve phase coherence and provide a good beamforming gain in time-varying channels without the knowledge of channel state information. It also shows that with a relative small phase drift speed $\Lambda_0 = \frac{\pi}{180}$ the algorithm on average achieves an RSS

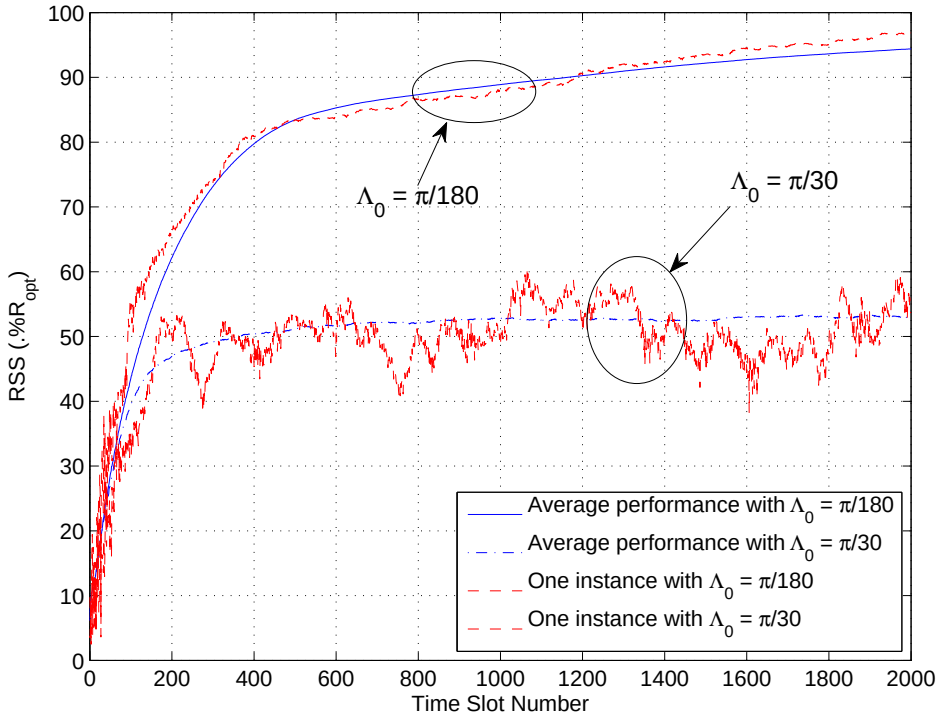



**Figure 5.8:** *Performance of the modified hybrid algorithm in time-varying channels with different channel phase drift speeds* $\lambda_i[n] \sim \text{uniform}[-\frac{\pi}{180}, \frac{\pi}{180}]$, $\lambda_i[n] \sim \text{uniform}[-\frac{\pi}{30}, \frac{\pi}{30}]$ *for* $N = 100$.

of $80\%R_{\text{opt}}$ in 400 time slots, which is close to the performance achieved in static channels (364 time slots) shown in Figure 5.5. This confirms that the modified hybrid algorithm still maintains a fast convergence speed in the initial stages of the convergence process under time-varying channel conditions.

Figure 5.9 shows one simulated instance of the modified hybrid algorithm in time-varying channels when the channel phase drift speed $\Lambda_0$ is changing. As we see, the modified hybrid algorithm has the ability to detect variations in the speed of channel phase changes and adjust perturbation sizes $\delta_0$ adaptively according to the speed. When $\Lambda_0$ becomes larger, it will lock the perturbation size to a bigger value to track the changes in channel phase responses. When $\Lambda_0$ becomes smaller, it will shift the perturbation size to a smaller value to obtain a superior beamforming gain. This ability makes the one-bit feedback algorithm much more robust to channel variations in practical implementations.

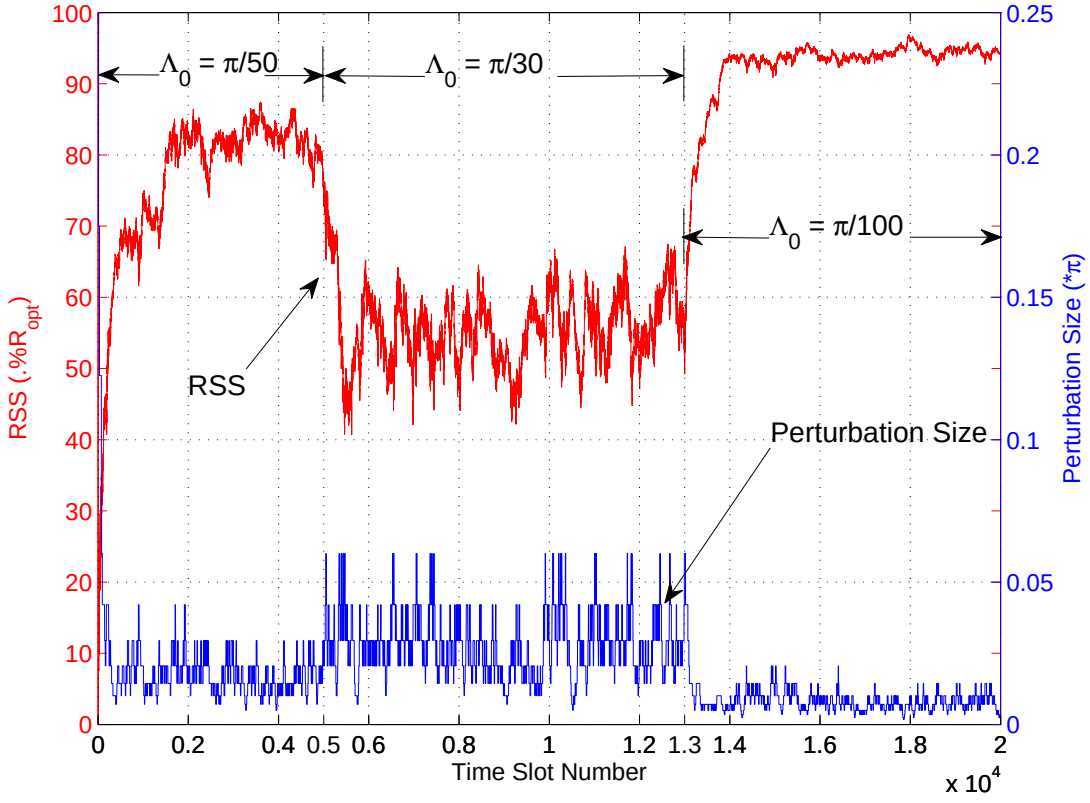



**Figure 5.9:** *One simulated instance of the modified hybrid algorithm in time-varying channels with variable phase drift speeds $\Lambda_0$ for $N = 100$. The red curve at the top shows the RSS versus time slots. The blue curve at the bottom shows the perturbation sizes versus time slots.*

Figure 5.10 shows the average performance of the modified hybrid algorithm in time-varying channels with different error rates in the one-bit feedback channel. The parameter settings for the modified hybrid algorithm are $C_T = 8$, $R_D = 0.7$, and the channel phase drift speed is $\lambda_i[n] \sim \text{uniform}[-\frac{\pi}{180}, \frac{\pi}{180}]$. It shows that the performance of the algorithm is sensitive to feedback error rates. This is mainly because the adjustment of perturbation sizes depends on the number of successive negative feedback steps. In order to achieve a good beamforming gain, it is suggested to maintain a feedback error rate under $0.01$ in the one-bit feedback channel in practice.

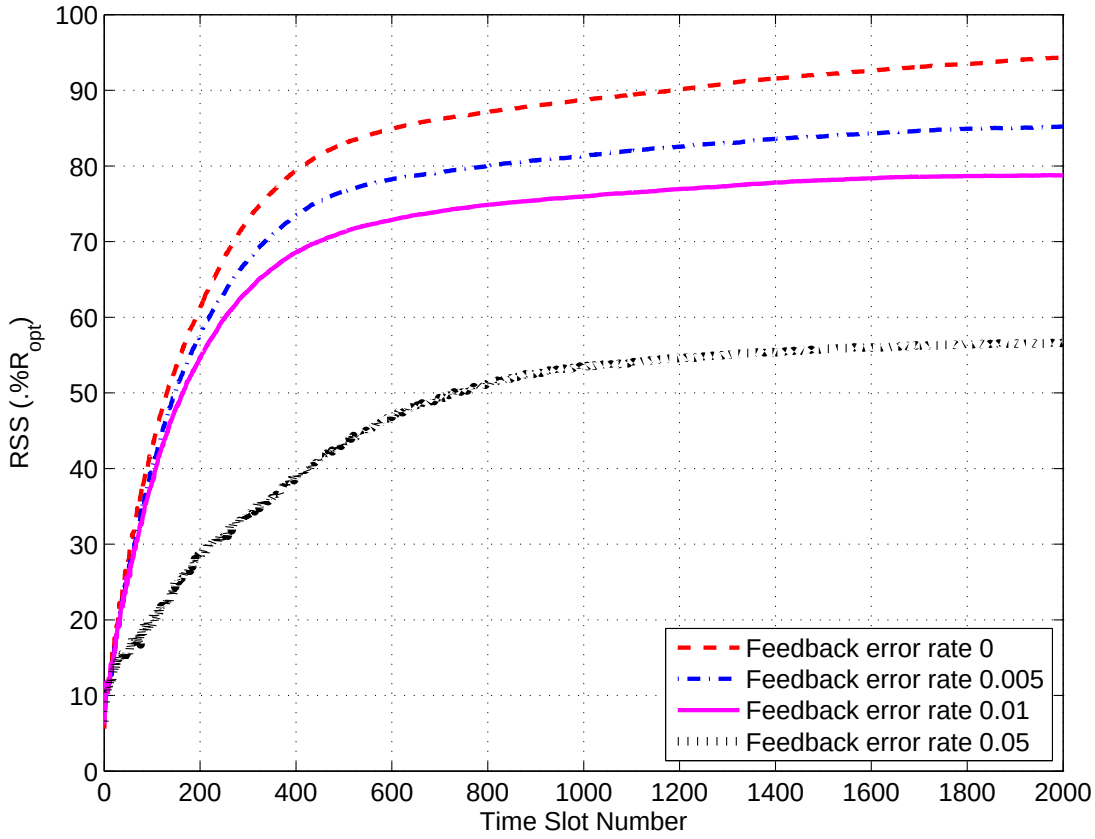



**Figure 5.10:** *Average performance of the modified hybrid algorithm in time-varying channels with $N = 100$, $\Lambda_0 = \frac{\pi}{180}$, and different error rates in the one-bit feedback channel.*

## 5.5 Summary

We proposed a decreasing perturbation-size scheme based on the original algorithm in the literature to achieve carrier phase alignment for distributed transmit beamforming. We show that the decreasing perturbation-size scheme and the previously proposed reverse-perturbation scheme can be combined to generate a hybrid algorithm. The hybrid algorithm can be easily applied into practical implementations and does not require any more information exchange or hardware changes. By exploiting negative feedback information in the iterations, the hybrid algorithm can largely enhance the convergence speed of phase alignment by over 40% compared to the original algorithm. By adding one time slot per size period, the hybrid algorithm can be modified to track time-varying channels without the knowledge of channel state information. The modified hybrid algorithm has the ability to adjust perturbation sizes adaptively according to the rate of phase drift in channel variations.

# Chapter 6
# Conclusions and Future Work

This thesis contributed to the performance analysis of distributed transmit beamforming with phase errors and algorithm design to achieve carrier phase alignment at the destination, which is critical for the practical realization of distributed beamforming. This Chapter will give key conclusions based on the results in previous chapters. We will also discuss limitations of our work and possible extensions for future work.

## 6.1 Conclusions

This thesis is concerned with distributed transmit beamforming in the context of wireless sensor networks. We consider the application scenarios that the destination is located far away from the sensor network and signal transmission from the sensor network to the destination cannot be realized by a single node due to node power constraints. Since traditional transmission techniques used for within-network communications, such as direct transmission and multi-hop transmission, have limited communication ranges, it came up naturally to consider transmit beamforming as a good candidate for long-range communications. However, realizing distributed beamforming faces severe challenges. Among the challenges brought to researchers, the most difficult one is to achieve phase alignment at the destination. Due to oscillator internal noise, errors in position estimation, channel estimation and timing synchronization, phase errors among signals arriving at the destination can only be minimized but cannot be removed. This is also the major difference between distributed beamforming and conventional beamforming. Then, we probed into the issue of phase errors from both the theoretical and practical aspects.

### 6.1.1 Performance analysis

In Chapter 3 we investigated the BER performance of distributed beamforming with phase errors. In particular, we derived the expressions for the BER performance of BPSK modulation

with EGT over Rayleigh fading channels. We assumed the bounded uniform distribution of phase errors for the analysis, which is a common assumption adopted in the literature. The BER performance can be expressed as a function of the number of nodes, phase errors and total transmit power. As explained in Chapter 3, the pdf of the beamforming gain cannot be extracted easily from the joint pdf associated to the real and imaginary parts of the received signal. Therefore, the exact BER expression is difficult to obtain. Instead, we provided two methods to give an accurate approximation. Method 1 was based on expectation adjustment and variance compensation of the received SNR. Method 2 was based on CLT and moment matching of distributions. Simulation results showed excellent agreement with analytical results. Method 1 is valid for any number of nodes but has a high computational complexity for a large number of nodes, while the solution given by method 2 is much simpler but valid for large number of nodes only. It is suggested to use method 1 to predict BER for a small number of nodes and use method 2 for a large number of nodes. Our methods can be extended to analyze BER with MRT in a similar way and the analysis can also be applied to other phase error distributions and signal modulations. The system performance was analyzed for different number of nodes and different phase error ranges. It shows that increasing the number of nodes can dramatically reduce the power cost of each node subject to the same BER performance. It also shows that adding more nodes when $N$ is small or minimizing phase errors when $\phi_0$ is large can significantly improve the BER performance. Depending on the practical constraints and targeted performance, the system engineer can use our expressions to explore the trade-offs among the number of nodes, phase errors and transmit power.

### 6.1.2 Algorithm development

The theoretical analysis gave us a good understanding of the impact of phase errors on the beamforming performance. However, minimizing phase errors and achieving phase alignment at the destination is a crucial problem in practice. Besides the knowledge of CSI, it requires sensor nodes to coordinate with each other and adjust their phase settings in a distributed manner to ensure that signals transmitted from different nodes can add coherently at the destination. In Chapter 4, we reviewed a simple iterative algorithm (the original algorithm) in the literature, which can achieve nearly perfect phase alignment after many iterations. The original algorithm does not require CSI, relies on only one-bit feedback in each iteration and has many other advantages, such as its simplicity in practical implementation and scalability to large number of nodes. The shortcoming of the original algorithm is that it discards negative feedback steps

indicating failed perturbations and only converges upon positive feedback steps indicating successful perturbations. Therefore, it takes a large number of iterations to achieve convergence. Based on this point, we proposed a reverse-perturbation scheme which exploits both positive and negative feedback information to improve the convergence speed. The reverse-perturbation scheme makes use of negative feedback in a single time slot to enhance the probability of generating better phase changes. Then in Chapter 5, we proposed another novel scheme, decreasing the perturbation-size scheme, to further improve the convergence speed of phase alignment. The decreasing perturbation-size scheme makes use of negative feedback information in successive time slots to adjust the perturbation size. The two schemes use different mechanisms to improve the convergence speed and can be combined to generate an advanced algorithm, the so-called hybrid algorithm. It shows that in static channels, the hybrid algorithm has an over $40\%$ faster convergence speed compared to the original algorithm. The hybrid algorithm does not require extra hardware or information exchange compared to the original algorithm. It still requires only one-bit feedback in each iteration and keeps all the benefits of the original algorithm. Its advantages in the convergence speed completely result from the information contained within negative feedback steps. Although the type of one-bit feedback algorithm received wide attention in the literature, few papers discussed the issue of distributed beamforming in time-varying channels. In Chapter 5, we show that the hybrid algorithm can be modified to track time-varying channels without CSI while maintaining its fast convergence speed. By switching between two operation modes, the modified hybrid algorithm has the ability to detect variations in the speed of channel phase changes and adjust perturbation sizes adaptively according to the speed. Its robustness against channel variations makes it a much more attractive candidate in practical implementations.

## 6.2 Limitations

The work presented in this thesis has its limitations. In Chapter 3, we adopted a Rayleigh fading model to analyze the BER performance, which reflects the effect of multipath fading and is a reasonable model when the destination moves around in short distances. However, if the destination moves over a long distance, we must consider the effect of shadowing, for which a log-normal distribution model is more appropriate. Moreover, we assumed perfect information sharing among sensor nodes and only considered the effect of phase errors in our analysis. For a more comprehensive view of the BER performance, one may include the effect

of errors in the information sharing process.

In Chapter 4, we presented some theoretical analysis of the reverse-perturbation scheme to predict its convergence speed, which mathematically proved that the scheme can yield a faster convergence speed compared to the original algorithm. However, our analysis is only valid for a small perturbation size.

## 6.3 Future work

In Chapter 3, we analyzed the BER performance of distributed beamforming with phase errors. The analysis was based on the assumption of a common signal transmitted from all sensor nodes, which requires perfect information sharing ahead. However, in practice, errors exist in the information sharing process, which may introduce differences among signals decoded at different sensor nodes. As discussed in Section 2.2, the information sharing process may be viewed as the first phase of a relaying process. Therefore, in order to give a more accurate prediction on the BER performance, one may include the impact of errors among signals transmitted from different nodes, and study the BER performance with different schemes for information sharing or different relaying strategies. What is more, while increasing the number of nodes can dramatically enhance the beamforming gain, both the overhead of information sharing and the convergence time of the one-bit feedback algorithms grow with the number of nodes. There may exist an optimum value for the number of nodes constituting a distributed beamformer.

In Chapter 4, we proposed a reverse-perturbation scheme which exploits the negative feedback information in a single time slot and has a faster convergence speed of phase alignment compared to the original one-bit feedback algorithm. Although its advantages in the convergence speed have been well supported by numerical results, deriving analytical results may lead to a better understanding of the scheme and further improve its performance. In our proposed scheme, the value of the modifying factor, $\epsilon_i[n]$, is set equal to the opposite value of $\delta_i[n-1]$ if $\delta_i[n-1]$ has led to performance degradation in the previous time slot. What would be the performance if $\epsilon_i[n]$ is set equal to $-0.8 \times \delta_i[n-1]$ or $-1.5 \times \delta_i[n-1]$ instead of $-\delta_i[n-1]$? There may exist an optimum quantity for the reverse operation.

In Chapter 5, we proposed a modified hybrid algorithm which can achieve reasonable beamforming gains in time-varying channels. It does not require CSI and keeps all the benefits of

the original algorithm. However, it shows that its performance is sensitive to feedback error rates. This is because the adjustment of perturbation size partly depends on the number of successive negative feedback steps. Therefore, the algorithm's robustness to feedback error rates may be improved by substituting the ratio of negative feedback steps for the number of successive negative feedback steps. In Chapter 5, we show that making use of information contained within failed perturbations can result in an over $40\%$ faster convergence speed compared to the original algorithm in static channels. Besides the negative feedback information, the convergence speed may be further improved by exploiting information contained within each sensor node's own perturbation experience. Each sensor node may adjust its perturbation size and make phase changes based on both its own experience of perturbations and the common feedback information.

# Appendix A
# Derivations for BER Analysis in Chapter 3

## A.1 The factor $\eta$ and the residual variance $\sigma_d^2$

We derive the factor $\eta$ defined in (3.13) and the residual variance, $\sigma_d^2$, defined in (3.14) in Section 3.3. For simplicity, we again suppress the time variable $\lambda$ in this section. The expression of $\eta^2$ in (3.13) can then be written as:

$$\eta^2 = \frac{\mathrm{E}\left[\left|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\right|^2\right]}{\mathrm{E}\left[\left(\sum_{i=1}^{N}|h_i|\right)^2\right]} = \frac{\mathscr{A}}{\mathscr{A}(\phi_0=0)} = \frac{1+\frac{\pi}{4}(N-1)\left(\frac{\sin\phi_0}{\phi_0}\right)^2}{1+\frac{\pi}{4}(N-1)}, \tag{A.1}$$

where the expression of $\mathscr{A}$ is derived in Appendix A.2.

We rewrite (3.14) as follows:

$$\begin{aligned}\sigma_d^2 &= \frac{P}{N}\left(\mathrm{Var}\left[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|\right] - \mathrm{Var}\left[\eta\Big(\sum_{i=1}^{N}|h_i|\Big)\right]\right)\\ &= \frac{P}{N}\left(\mathrm{E}\left[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|^2\right] - \left(\mathrm{E}\left[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|\right]\right)^2 - \eta^2\mathrm{E}\left[\Big(\sum_{i=1}^{N}|h_i|\Big)^2\right]\right.\\ &\quad \left.+\eta^2\left(\mathrm{E}\Big[\sum_{i=1}^{N}|h_i|\Big]\right)^2\right).\end{aligned} \tag{A.2}$$

Substituting (3.13) into (A.2), it yields:

$$\sigma_d^2 = \frac{P}{N}\left(\eta^2\left(N\cdot\mathrm{E}\Big[|h_i|\Big]\right)^2 - \left(\mathrm{E}\Big[|H|\Big]\right)^2\right), \tag{A.3}$$

where the expressions of $\eta^2$, $\mathrm{E}\big[|h_i|\big]$ are given by (A.1), (A.9) separately, and the definition of $H$ is given by (3.16).

The second moment and the fourth moment of $\big|H\big|$ in terms of $N$ and $\phi_0$ are derived in Appendix A.2. However, the pdf of $\big|H\big|$ is unknown and the first moment of $\big|H\big|$ is hard to compute. Instead, we use the Nakagami $m$-distribution [98] to give an approximate expression for the first moment. One of the characteristics of the Nakagami $m$-distribution is that it has great flexibility and can approximate many other distributions modeling fading environments. The $\beta$th moment of Nakagami $m$-distributed $\big|H\big|$ is given by [55]:

$$\mathrm{E}\Big[\big|H\big|^{\beta}\Big] = \frac{\Gamma(m+\frac{1}{2}\beta)}{\Gamma(m)}\left(\frac{\mathrm{E}\big[|H|^2\big]}{m}\right)^{\frac{\beta}{2}}, \tag{A.4}$$

where the parameter $m$ is the ratio of moments:

$$m = \frac{\Big(\mathrm{E}\Big[\big|H\big|^2\Big]\Big)^2}{\mathrm{Var}\Big[\big|H\big|^2\Big]} = \frac{\mathscr{A}^2}{\mathscr{B}-\mathscr{A}^2}, \tag{A.5}$$

and $\mathscr{A}$, $\mathscr{B}$ in terms of $N$ and $\phi_0$ are given in (A.7), (A.8). By taking (A.7), (A.8) into (A.4), (A.5) the first moment of $\big|H\big|$ can be easily obtained as:

$$\mathrm{E}\Big[\big|H\big|\Big] = \frac{\Gamma(m+\frac{1}{2})}{\Gamma(m)}\left(\frac{\mathscr{A}}{m}\right)^{\frac{1}{2}}. \tag{A.6}$$

By substituting (A.6) into (A.3), one can obtain the final expression of the residual variance $\sigma_d^2$ in terms of $N$ and $\phi_0$.

## A.2 The second and the fourth moment of $\big|H\big|$

We derive the second and the fourth moment of $|H|$ used in (3.26), (3.27) in terms of $N$ and $\phi_0$ based on the assumption that both $h_i$ and $\phi_i$ are independent i.i.d. variables, where $h_i \sim CN(0, 2\sigma_c^2)$ and $\phi_i \sim (-\phi_0, \phi_0)$.

The second moment of $\big|H\big|$ in (3.26), (3.27) is expressed as:

$$\begin{aligned}
\mathscr{A} &= \mathrm{E}\Big[\big|H\big|^2\Big] \\
&= \mathrm{E}\Big[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|^2\Big] \\
&= \mathrm{E}\Big[\sum_{i=1}^{N}|h_i|e^{j\phi_i}\cdot\sum_{l=1}^{N}|h_l|e^{-j\phi_l}\Big] \\
&= \underbrace{N\cdot\mathrm{E}\big[|h_i|^2\big]}_{i=l}+\underbrace{N(N-1)\cdot\mathrm{E}\big[|h_i||h_l|e^{j(\phi_i-\phi_l)}\big]}_{i\neq l} \\
&= N\cdot\mathrm{E}\big[|h_i|^2\big]+N(N-1)\cdot\mathrm{E}\big[|h_i||h_l|\cos(\phi_i-\phi_l)\big] \\
&= N\cdot\mathrm{E}\big[|h_i|^2\big]+N(N-1)\Big(\mathrm{E}\big[|h_i|\big]\Big)^2\left(\frac{\sin\phi_0}{\phi_0}\right)^2. \qquad \text{(A.7)}
\end{aligned}$$

The fourth moment of $\big|H\big|$ is expressed as (A.8).

Since $h_i \sim CN(0, 2\sigma_c^2)$, the moments of $|h_i|$ in above derivations are given by [55]:

$$\mathrm{E}\big[|h_i|^\alpha\big] = (2\sigma_c^2)^{\frac{\alpha}{2}}\Gamma(1+\frac{\alpha}{2}). \qquad \text{(A.9)}$$

In particular, when $h_i \sim CN(0,1)$, $\mathscr{A}$ and $\mathscr{B}$ become:

$$\mathscr{A} = N+\frac{\pi}{4}N(N-1)\left(\frac{\sin\phi_0}{\phi_0}\right)^2, \qquad \text{(A.10)}$$

$$\begin{aligned}
\mathscr{B} &= 2N^2+\pi N(N-1)(N-\frac{1}{2})\left(\frac{\sin\phi_0}{\phi_0}\right)^2+N(N-1)\left(\frac{\sin 2\phi_0}{2\phi_0}\right)^2 \\
&\quad +\frac{\pi}{2}N(N-1)(N-2)\frac{\sin 2\phi_0}{2\phi_0}\left(\frac{\sin\phi_0}{\phi_0}\right)^2 \\
&\quad +\frac{\pi^2}{16}N(N-1)(N-2)(N-3)\left(\frac{\sin\phi_0}{\phi_0}\right)^4. \qquad \text{(A.11)}
\end{aligned}$$

$$
\begin{aligned}
\mathscr{B} &= \mathrm{E}\Big[\big|H\big|^4\Big] \\
&= \mathrm{E}\Big[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|^4\Big] \\
&= \mathrm{E}\Big[\sum_{i=1}^{N}|h_i|e^{j\phi_i}\cdot\sum_{k=1}^{N}|h_k|e^{j\phi_k}\cdot\sum_{l=1}^{N}|h_l|e^{-j\phi_l}\cdot\sum_{m=1}^{N}|h_m|e^{-j\phi_m}\Big] \\
&= \underbrace{N\cdot\mathrm{E}\big[|h_i|^4\big]}_{i=k=l=m} + \underbrace{4N(N-1)\cdot\mathrm{E}\big[|h_i|^3|h_m|e^{j(\phi_i-\phi_m)}\big]}_{\substack{i=k=l\neq m\\ i=k=m\neq l\\ i=l=m\neq k\\ k=l=m\neq i}} + \underbrace{N(N-1)\cdot\mathrm{E}\big[|h_i|^2|h_l|^2e^{j(2\phi_i-2\phi_l)}\big]}_{(i=k)\neq(l=m)} \\
&\quad + \underbrace{2N(N-1)\cdot\mathrm{E}\big[|h_i|^2|h_m|^2\big]}_{\substack{(i=l)\neq(k=m)\\ (i=m)\neq(k=l)}} + \underbrace{2N(N-1)(N-2)\cdot\mathrm{E}\big[|h_i|^2|h_m||h_l|e^{j(2\phi_i-\phi_m-\phi_l)}\big]}_{\substack{(i=k)\neq l\neq m\\ (l=m)\neq i\neq k}} \\
&\quad + \underbrace{4N(N-1)(N-2)\cdot\mathrm{E}\big[|h_i|^2|h_k||h_m|e^{j(\phi_k-\phi_m)}\big]}_{\substack{(i=l)\neq k\neq m\\ (i=m)\neq k\neq l\\ (k=l)\neq i\neq m\\ (k=m)\neq i\neq l}} \\
&\quad + \underbrace{N(N-1)(N-2)(N-3)\cdot\mathrm{E}\big[|h_i||h_k||h_l||h_m|e^{j(\phi_i+\phi_k-\phi_l-\phi_m)}\big]}_{i\neq k\neq l\neq m} \\
&= N\cdot\mathrm{E}\big[|h_i|^4\big] + 4N(N-1)\cdot\mathrm{E}\big[|h_i|^3\big]\cdot\mathrm{E}\big[|h_i|\big]\left(\frac{\sin\phi_0}{\phi_0}\right)^2 + N(N-1)\Big(\mathrm{E}\big[|h_i|^2\big]\Big)^2\left(\frac{\sin 2\phi_0}{2\phi_0}\right)^2 \\
&\quad + 2N(N-1)\Big(\mathrm{E}\big[|h_i|^2\big]\Big)^2 + 2N(N-1)(N-2)\cdot\mathrm{E}\big[|h_i|^2\big]\Big(\mathrm{E}\big[|h_i|\big]\Big)^2\frac{\sin 2\phi_0}{2\phi_0}\left(\frac{\sin\phi_0}{\phi_0}\right)^2 \\
&\quad + 4N(N-1)(N-2)\cdot\mathrm{E}\big[|h_i|^2\big]\Big(\mathrm{E}\big[|h_i|\big]\Big)^2\left(\frac{\sin\phi_0}{\phi_0}\right)^2 \\
&\quad + N(N-1)(N-2)(N-3)\Big(\mathrm{E}\big[|h_i|\big]\Big)^4\left(\frac{\sin\phi_0}{\phi_0}\right)^4. \qquad \text{(A.8)}
\end{aligned}
$$

# Appendix B
# **Publications**

## B.1 Journal Papers

- Shuo Song, John S. Thompson, Pei-Jung Chung, and Peter M. Grant, "Exploiting negative feedback information for one-bit feedback beamforming algorithm," *IEEE Transactions on Wireless Communications*, accepted for publication.

- Shuo Song, John S. Thompson, Pei-Jung Chung, and Peter M. Grant, "BER analysis for distributed beamforming with phase errors," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 4169-4174, 2010.

## B.2 Conference Papers

- Shuo Song, and John S. Thompson, "One-bit feedback algorithm with decreasing step size for distributed beamforming," in *Proceeding of UKIWCWS 2010 International Workshop*, IIT Delhi, December 2010, pp. 1-5.

- Shuo Song, John S. Thompson, Pei-Jung Chung, and Peter M. Grant, "Improving the one-bit feedback algorithm for distributed beamforming," in *Proceeding of WCNC 2010 International Conference*, Sydney, April 2010, pp. 1-6.

- Shuo Song, John S. Thompson, Pei-Jung Chung, and Peter M. Grant, "Probability of error for BPSK modulation in distributed beamforming with phase errors," in *Proceeding of International ITG Workshop on Smart Antennas*, Berlin, February 2009, pp. 149-154.

# Exploiting Negative Feedback Information for One-bit Feedback Beamforming Algorithm

Shuo Song, John S. Thompson, *Member, IEEE,* Pei-Jung Chung, *Member, IEEE,* and Peter M. Grant, *Fellow, IEEE*

**Abstract**

In this paper a hybrid one-bit feedback algorithm is proposed to achieve carrier phase alignment at the receiver for distributed transmit beamforming. The proposed iterative algorithm employs two schemes to speed up the convergence process, which exploit *negative feedback information* in a single time slot (Scheme 1) and in successive time slots (Scheme 2) respectively, whereas previously proposed algorithms in the literature discard this information. We show that the proposed algorithm yields a *significant* improvement in the convergence speed compared to the original algorithm. Furthermore, we modify the proposed algorithm to be capable of tracking time-varying channels which have variable rates of phase drift. The modified hybrid algorithm has the ability to adjust perturbation sizes adaptively without the knowledge of channel state information and is suited for practical implementations.

**Index Terms**

Distributed beamforming, feedback communication, adaptive algorithms.

## I. INTRODUCTION

Distributed transmit beamforming is a promising form of transmission in some sensor network application scenarios because it can provide significant benefits in energy efficiency, communication range, security, etc. It is performed by a virtual antenna array composed of randomly

The authors are with the Institute for Digital Communications, University of Edinburgh, EH9 3JL, UK (e-mail: {S.Song, John.Thompson, P.Chung, Peter.Grant}@ed.ac.uk).

Shuo Song would like to thank China Scholarship Council/University of Edinburgh Joint Scholarship Program for supporting his PhD studies. The authors would like to thank Prof. Vahid Tarokh from Harvard University for commenting on an early draft and discussing this work.

located sensor nodes, each of which has a single antenna and an independent oscillator. Unlike conventional beamforming, phase errors among the signals arriving at the receiver cannot be avoided in distributed beamforming and phase synchronization is critical to the beamformer performance [1]. In [2], the authors studied the average far-field beampattern of a random array and showed that a desirable beampattern with a narrow mainlobe and relatively small sidelobes can be achieved in theory for distributed beamforming. In [3], the authors analyzed the achievable bit error ratio performance of distributed beamforming with phase errors. Besides the theoretical performance, practical realization of distributed beamforming requires carrier frequency synchronization among transmitters and phase alignment at the receiver [1], where the frequency synchronization problem can be solved by employing a master-slave scheme presented in [4]. In [5], the authors proposed an iterative algorithm (which we term the original algorithm) to adjust phase settings at transmitters, which can achieve nearly perfect phase alignment at the receiver under the assumption of static channels. The training process is performed by each transmitter adding a random perturbation to its phase offset in each iteration. If the perturbation results in a positive feedback indicating a bigger received signal strength (RSS), it will be adopted. Otherwise, it will be discarded. The key advantages of this algorithm is that it does not need channel state information and only relies on one-bit feedback in each iteration. The original algorithm has received wide attention in the literature and similar algorithms also using one-bit feedback were proposed in [6], [7], [8], [9] for distributed beamforming. A common feature of these algorithms is that they only exploit positive feedback information and discard negative feedback information.

The major issue considered in this paper is to improve the convergence performance of the original algorithm by exploiting negative feedback information which indicates failed perturbations while still retaining its advantages. We propose a novel algorithm (which we term the hybrid algorithm) which has a faster convergence speed and is robust to time-varying channels with variable rates of phase drift. To make a fair comparison, the proposed algorithm does not need any more information exchange compared to the original algorithm, and in each iteration there is only one phase setting used for beamforming and one-bit feedback from the receiver which matches the original algorithm. The proposed algorithm can be seen as an extension of the original algorithm for efficient application to time-varying channels. The rest of the paper is organized as follows. In Section II, we introduce the system model and briefly review the

original algorithm in the literature. In Section III, we present our proposed algorithm which employs two schemes to improve the convergence speed. In Section IV, the performance of the proposed algorithm and the original algorithm in static channels are evaluated by computer simulations. In Section V, the algorithm is modified to track time-varying channels which have variable rates of phase drift. Section VI presents conclusions to the paper.

## II. Original One-bit Feedback Algorithm

We consider a random array composed of $N$ frequency-synchronized transmitters collaboratively beamforming a common signal to a distant receiver. This is performed by each transmitter adjusting its phase offset independently and iteratively to achieve carrier phase alignment at the receiver. The phase of the received signal at the receiver from transmitter $i$ in time slot $n$ is expressed as:

$$\Phi_i[n] = \gamma_i + \psi_i + \phi_i[n] \tag{1}$$

where $\gamma_i$ is an unknown phase offset at transmitter $i$ and $\psi_i$ is the channel phase response from transmitter $i$ to the receiver. Both $\gamma_i$ and $\psi_i$ are assumed to be static during the convergence process, uniformly distributed within $[0, 2\pi)$ over $i$ and unknown to both the transmitters and the receiver. The scalar $\phi_i[n]$ is the adaptive component adjusted by transmitter $i$ in time slot $n$ based on the one-bit feedback information from the receiver. We set $\phi_i[0] = 0$. Since the objective of the algorithm is to achieve phase alignment, we assume unit transmit power for every transmitter and unit channel power gain from each transmitter to the receiver [5]. The RSS in time slot $n$, tested at the receiver, is defined as:

$$R[n] = \left| \sum_{i=1}^{N} e^{j\Phi_i[n]} \right| \tag{2}$$

The original algorithm presented in [5] repeats the following steps:

1) Each transmitter records its best known phase used for beamforming, $\theta_i[n]$, in memory and adds a random perturbation, $\delta_i[n] = \pm\delta_0$, to it. (We set $\theta_i[1] = 0$).
2) All transmitters use their new adaptive components, $\phi_i[n] = \theta_i[n] + \delta_i[n]$, to perform transmit beamforming.
3) The receiver measures the new RSS $R[n]$, and updates the best RSS in memory, $R_{best}[n+1] = \max\Big(R_{best}[n], R[n]\Big)$. It then feeds back one-bit of information (error free) to all transmitters conveying whether the RSS has been improved or not.

4) The transmitters update their best known phases as $\theta_i[n+1] = \begin{cases} \theta_i[n] + \delta_i[n], & R[n] > R_{best}[n] \\ \theta_i[n], & \text{otherwise} \end{cases}$

The original algorithm can achieve phase alignment after many iterations, but it only changes phase for positive feedback when $R[n] > R_{best}[n]$. For more details of the original algorithm and its advantages over other alternative approaches for distributed beamforming, see [5].

## III. Hybrid One-bit Feedback Algorithm

We propose two schemes which do not require any more information exchange or hardware changes compared to the original algorithm, and therefore keep all of its advantages. We show that the two schemes can also be combined (denoted as the hybrid algorithm) to provide a significant improvement in the convergence speed in the phase training process.

### *A. Scheme 1*

For a single transmitter in a single time slot, if a positive perturbation leads to performance degradation, usually, a negative perturbation on the same phase offset will lead to performance improvement, and vice versa. Therefore, we introduce a modifying factor $\epsilon_i[n]$ into the adaptive component used for beamforming in step 2):

$$\phi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n] \tag{3}$$

The function of $\epsilon_i[n]$ is to add an opposite value of $\delta_i[n-1]$ into the new adaptive component if $\delta_i[n-1]$ has led to performance degradation in the previous time slot. In step 4), we update the modifying factor $\epsilon_i[n]$ and the best known phase $\theta_i[n]$ as follows:

$$\epsilon_i[n+1] = \begin{cases} 0, & R[n] > R_{best}[n] \\ -\delta_i[n], & \text{otherwise} \end{cases} \tag{4}$$

$$\theta_i[n+1] = \begin{cases} \theta_i[n] + \epsilon_i[n] + \delta_i[n], & R[n] > R_{best}[n] \\ \theta_i[n], & \text{otherwise} \end{cases} \tag{5}$$

The other steps are the same as in the original algorithm.

Scheme 1 makes use of the negative feedback information in a single time slot to enhance the probability of generating better phase changes. In the case of negative feedback, an opposite value of the perturbation in time slot $n$ will be added into the next adaptive component in time slot $n+1$, which enhances the probability of generating better phase changes. In the case of

successive negative feedback steps, the values of the adaptive component $\phi_i$ are always located around the best known phase $\theta_i$. This is because the value of $\theta_i$ is updated only in the case of positive feedback in order to prevent $\theta_i$ from drifting off its best value.

The performance of Scheme 1 and its capability of improving the convergence speed have been well studied and verified by simulation results in [10]. Below we present some mathematical analysis of Scheme 1 and provide a close upper bound on its convergence speed. We begin our analysis by studying the original one-bit feedback algorithm. The original algorithm described above can be reformulated as:

$$\theta_i[n+1] = \theta_i[n] + \delta_i[n]\mathbf{1}_G \tag{6}$$

where the indicator function $\mathbf{1}_G$ equals 1 when the condition $G$ is satisfied and equals 0 otherwise. The condition $G = \{ \sum_{i=1}^{N} \cos(\hat{\Phi}_i[n] + \delta_i[n]) > \sum_{i=1}^{N} \cos(\hat{\Phi}_i[n]) \}$ and $\hat{\Phi}_i[n] = \gamma_i + \psi_i + \theta_i[n]$. The condition $G$ exists because with a large $N$, the RSS mainly depends on the cosines of the carrier phases and the contribution of sines can be discarded. This has been verified in [3] by applying the central limit theorem (CLT). In [11], the authors proved that the trajectories of (6) collapse to the solution of a certain limiting ordinary differential equation (ODE). For the readers' convenience, we first repeat some of the key results in deriving the ODE for the original algorithm. For details, please see [11]. We then derive an ODE that mimics the behavior of Scheme 1 in a similar way.

For a small perturbation size $\delta_0$, $\cos(\hat{\Phi}_i[n] + \delta_i[n]) \approx \cos(\hat{\Phi}_i[n]) - \delta_i[n]\sin(\hat{\Phi}_i[n])$. Therefore, the condition $G$ can be simplified to $G = \{\sum_{i=1}^{N} \delta_i[n]\sin(\hat{\Phi}_i[n]) < 0\}$. With large $N$, the summation of $(N-1)$ terms excluding $\delta_j[n]\sin(\hat{\Phi}_j[n])$ can be written as:

$$Z_j = \sum_{i=1(i\neq j)}^{N} \delta_i[n]\sin(\hat{\Phi}_i[n]) \tag{7}$$

which is a zero mean Gaussian variable according to the Lyapunov CLT, whose variance is $\mathrm{Var}(Z_j) = \delta_0^2 \sum_{i=1(i\neq j)}^{N} \sin^2(\hat{\Phi}_i[n])$. Therefore, the probability of condition $G$ being satisfied is:

$$\begin{aligned}\Pr(G) &= \Pr(Z_j + \delta_j[n]\sin(\hat{\Phi}_j[n]) < 0) \\ &= \frac{1}{2} - \frac{1}{2}\mathrm{erf}\left(\frac{\delta_j[n]\sin(\hat{\Phi}_j[n]) - \mathrm{E}(Z_j)}{\sqrt{2}\sqrt{\mathrm{Var}(Z_j)}}\right) \\ &\approx \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \cdot \frac{\delta_j[n]\sin(\hat{\Phi}_j[n])}{\delta_0\sqrt{\sum_{i=1(i\neq j)}^{N}\sin^2(\hat{\Phi}_i[n])}}.\end{aligned} \tag{8}$$

where $\mathrm{erf}(\cdot)$ represents the Gaussian error function. The last approximation comes from the first term of the error function's Taylor series $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-1)^n x^{2n+1}}{n!(2n+1)}$. Thus, the expectation of the random perturbation applied on phase settings for transmitter $j$ can be computed as:

$$\begin{aligned}\mathrm{E}(\delta_j[n]\mathbf{1}_G) &\approx \mathrm{E}(\delta_j[n]\Pr(G)) \\ &= \mathrm{E}\left(\delta_j[n]\left(\frac{1}{2} - \frac{1}{\sqrt{2\pi}} \cdot \frac{\delta_j[n]\sin(\hat{\Phi}_j[n])}{\delta_0\sqrt{\sum_{i=1(i\neq j)}^{N}\sin^2(\hat{\Phi}_i[n])}}\right)\right) \\ &= -\frac{\delta_0\sin(\hat{\Phi}_j[n])}{\sqrt{2\pi}\sqrt{\sum_{i=1(i\neq j)}^{N}\sin^2(\hat{\Phi}_i[n])}}.\end{aligned} \tag{9}$$

The convergence of the best known phases $\theta_i$ to their correct settings is equivalent to the convergence of $\hat{\Phi}_i$ to zero. The ODE corresponding to equation (6) which mimics the behavior of the original algorithm can be obtained as:

$$\frac{d\hat{\Phi}_j(t)}{dt} = -\frac{\delta_0\sin(\hat{\Phi}_j[n])}{\sqrt{2\pi}\sqrt{\sum_{i=1(i\neq j)}^{N}\sin^2(\hat{\Phi}_i[n])}}. \tag{10}$$

In the original algorithm, the decision on the perturbation $\delta_i[n]$ only depends on $R[n]$ and the corresponding feedback in time slot $n$. However, in Scheme 1, the decision on the perturbation $\delta_i[n]$ not only depends on the feedback in time slot $n$, but also the feedback in time slots $(n+1)$ and $(n-1)$. A flowchart of the adaptive component for transmitter $j$ under Scheme 1 is shown in Fig. 1. The conditions ConA, ConB in Fig. 1 are defined mathematically as follows:

$$\mathrm{ConA}(\delta_i[n]) = \left\{ \sum_{i=1}^{N}\cos(\hat{\Phi}_i[n] + \delta_i[n]) > \sum_{i=1}^{N}\cos(\hat{\Phi}_i[n]) \right\}. \tag{11}$$

$$\mathrm{ConB}(\delta_i[n-1], \delta_i[n]) = \left\{ \sum_{i=1}^{N}\cos(\hat{\Phi}_i[n-1] - \delta_i[n-1] + \delta_i[n]) > \sum_{i=1}^{N}\cos(\hat{\Phi}_i[n-1]) \right\}. \tag{12}$$

The condition ConA is the same as condition $G$ in the original algorithm. Therefore, its probability is given by

$$\Pr(\mathrm{ConA}(\delta_j[n])) \;=\; \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \cdot \frac{\delta_j[n]\eta_j[n]}{\delta_0}. \tag{13}$$

where $\eta_j[n] = \frac{\sin(\hat{\Phi}_j[n])}{\sqrt{\sum_{i=1(i\neq j)}^{N} \sin^2(\hat{\Phi}_i[n])}}$. The probability of condition ConB can be derived in a similar way and is expressed as:

$$\Pr(\mathrm{ConB}(\delta_j[n-1], \delta_j[n])) \;=\; \frac{1}{2} - \frac{1}{\sqrt{2}\sqrt{2\pi}} \cdot \frac{(-\delta_j[n-1] + \delta_j[n])\eta_j[n-1]}{\delta_0}. \tag{14}$$

The conditions $\overline{\mathrm{ConA}}$, $\overline{\mathrm{ConB}}$ are the negations of $\mathrm{ConA}$, $\mathrm{ConB}$, whose probabilities can be calculated using the equations:

$$\Pr(\mathrm{ConA}) + \Pr(\overline{\mathrm{ConA}}) = \Pr(\mathrm{ConB}) + \Pr(\overline{\mathrm{ConB}}) = 1. \tag{15}$$

From Fig. 1 we have:

$$\theta_i[n+1] = \begin{cases} \theta_i[n] + \delta_i[n]\mathbf{1}_{G^+} \\ \theta_i[n] - \delta_i[n]\mathbf{1}_{G^-} \end{cases} \tag{16}$$

where

$$G^+ = \left\{ \mathrm{ConA}(\delta_i[n-1]) \cdot \mathrm{ConA}(\delta_i[n]) + \overline{\mathrm{ConA}}(\delta_i[n-1]) \cdot \mathrm{ConB}(\delta_i[n-1], \delta_i[n]) \right\}, \tag{17}$$

$$\begin{aligned} G^- \;=\; & \Big\{ \mathrm{ConA}(\delta_i[n-1]) \cdot \overline{\mathrm{ConA}}(\delta_i[n]) \cdot \mathrm{ConB}(\delta_i[n], \delta_i[n+1]) \\ & + \overline{\mathrm{ConA}}(\delta_i[n-1]) \cdot \overline{\mathrm{ConB}}(\delta_i[n-1], \delta_i[n]) \cdot \mathrm{ConB}(\delta_i[n], \delta_i[n+1]) \Big\}. \end{aligned} \tag{18}$$

Thus,

$$\mathrm{E}(\delta_j[n]\mathbf{1}_{G^+} - \delta_j[n]\mathbf{1}_{G^-}) \;=\; \mathrm{E}\left[\delta_j[n] \cdot \left(\Pr(G^+) - \Pr(G^-)\right)\right] \tag{19}$$

For a small perturbation size $\delta_0$, $\eta_j[n-1] \approx \eta_j[n] \approx \eta_j[n+1] = \eta_j$. Substituting (13), (14), (15), (17) and (18) into (19), we have:

$$\begin{aligned}\mathrm{E}(\delta_j[n]\mathbf{1}_{G^+} - \delta_j[n]\mathbf{1}_{G^-}) &= \mathrm{E}\Bigg[ -\Bigg(\frac{1}{2}\cdot\frac{\delta_j[n]^2\eta_j}{\sqrt{2\pi}\delta_0} + \frac{1}{2}\cdot\frac{\delta_j[n]^2\eta_j}{\sqrt{2}\sqrt{2\pi}\delta_0} + (\frac{1}{4}+\frac{1}{4\sqrt{2}})\cdot\frac{\delta_j[n]^2\eta_j}{\sqrt{2\pi}\delta_0} \\ &\quad + \frac{1}{2\sqrt{2}}\cdot\frac{\delta_j[n]^2\eta_j}{\sqrt{2\pi}\delta_0} - \frac{\delta_j[n]^2\eta_j^3}{4\sqrt{2}\pi^{\frac{3}{2}}\delta_0}\Bigg)\Bigg] \\ &> -\frac{5\sqrt{2}+6}{8}\cdot\frac{\delta_0\eta_j}{\sqrt{2\pi}}.\end{aligned} \tag{20}$$

The ODE corresponding to equation (16) which mimics the behavior of Scheme 1 can be obtained as:

$$\frac{d\hat{\Phi}_j(t)}{dt} = -\frac{5\sqrt{2}+6}{8}\cdot\frac{\delta_0\sin(\hat{\Phi}_j[n])}{\sqrt{2\pi}\sqrt{\sum_{i=1(i\neq j)}^{N}\sin^2(\hat{\Phi}_i[n])}}. \tag{21}$$

Comparing (21) with (10), we see that Scheme 1 has a faster convergence speed of $\frac{5\sqrt{2}+6}{8} \approx 1.634$ compared to the original algorithm. The accuracy of (21) will be justified by simulation results in Section IV.

### *B. Scheme 2*

In [5], the authors derived an analytical formula for the optimal perturbation size in each time slot for the original algorithm. The optimal perturbation size $\Delta_0$ in time slot $n+1$ is expressed as a function of $(R[n]/R_{\mathrm{opt}})$, where $R_{\mathrm{opt}}$ represents the RSS with perfect phase alignment. The analysis in [5] gives a fundamental understanding of the original algorithm, and can be used as a good metric for comparison and algorithm design. However, the value of $R_{\mathrm{opt}}$ is hard to obtain in practice before the phase training process converges and feedback of the optimal value requires several bits instead of one. The results presented in [5] show that the optimal value of perturbation size decreases as the number of iterations increases. Based on this point, we adopt a decreasing size for $\delta_0$ in our practical design. The transmitters will adopt a smaller $\delta_0$ when the number of successive negative feedback steps $C_N$ meets a certain threshold $C_T$. In step 4), the counter $C_N$ and the perturbation size $\delta_0$ are updated as follows:

$$C_N = \begin{cases} 0 & R[n] > R_{best}[n] \\ C_N + 1 & \text{otherwise} \end{cases} \tag{22}$$

$$\delta_0[n+1] = \begin{cases} \delta_0[n] & C_N < C_T \\ \delta_0[n] \cdot R_D & C_N \geq C_T \end{cases} \tag{23}$$

where $R_D$ $(0 < R_D < 1)$ is the decreasing ratio of the perturbation size. The number of successive negative feedback steps $C_N$ is reset to zero every time the perturbation size is adjusted. The other steps are the same as in the original algorithm.

Scheme 2 makes use of the negative feedback information in successive time slots to adjust the perturbation size. It is a simple but effective scheme which can be easily applied into practical implementations. Simulation results in Section IV will show that Scheme 2 can achieve close performance to the method with optimal perturbation sizes derived in [5].

Both Scheme 1 and Scheme 2 speed up the convergence process by exploiting the negative feedback information which indicates failed perturbations. Combining Scheme 1 and Scheme 2 gives us the hybrid algorithm which repeats the following steps.

1) Each transmitter adds a random perturbation $\delta_i[n] = \pm\delta_0$ to its phase setting.
2) All transmitters use $\phi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$ to perform transmit beamforming.
3) The receiver measures the new RSS $R[n]$, updates $R_{best}[n+1] = \max\Big(R_{best}[n], R[n]\Big)$, and feeds back one-bit of information.
4) The transmitters update their settings as

If $R[n] > R_{best}[n]$

$\theta_i[n+1] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]; \quad \epsilon_i[n+1] = 0; \quad C_N = 0;$

else

$\theta_i[n+1] = \theta_i[n]; \quad \epsilon_i[n+1] = -\delta_i[n]; \quad C_N = C_N + 1;$

if $C_N \geq C_T$

$\delta_0 = \delta_0 \cdot R_D; \quad C_N = 0;$

end

end

## IV. Simulation Results in Static Channels

We present some simulation results in accordance with our previous assumptions to study the convergence performance of the hybrid algorithm over static channels, and compare it with the performance of the original algorithm. The simulation results also reveal the advantages of Scheme 1 and Scheme 2.

Fig. 2 shows the comparison of the trajectories of the phases $\hat{\Phi}$ obtained from simulation with the trajectories of the ODE in (21) for Scheme 1 with 20 transmitters. The initial values of the

phases $\hat{\Phi}$ are set as uniformly distributed within $(-\pi, \pi)$. It shows that the ODE in (21) can give a good prediction on the behavior of the phase alignment process under Scheme 1.

Fig. 3 shows the comparison of the RSS calculated using the ODE in (21) with the simulation results of RSS versus number of time slots with different numbers of transmitters $N = 20, 50$, and $100$. As we can see, the analytical results provide a close upper bound on the convergence speed and yield a good match with the simulation results for most of the convergence process.

Fig. 4 shows the number of time slots required to achieve an RSS of $90\% \cdot R_{\text{opt}}$, averaged over 4000 runs, with $N = 100$ transmitters for the hybrid algorithm. It shows that the hybrid algorithm can achieve $90\% \cdot R_{\text{opt}}$ within 700 time slots over a wide range of parameter selections, while the minimum number of time slots is 550 obtained with $C_T = 7$, $R_D = 0.75$. The number of time slots in the 3D plot has a fairly flat surface. This reveals the robustness of the hybrid algorithm to small mismatches in parameter settings.

Table II shows the average number of time slots required to achieve an RSS of $90\% \cdot R_{\text{opt}}$ with $N = 20, 100$ and $500$ transmitters for the hybrid algorithm. We define the convergence speed to be inversely proportional to the required number of time slots, which is expressed as $v(C_T, R_D) \propto \frac{1}{n(C_T, R_D)}$. The convergence speed with a pair of $C_T$ and $R_D$ over the convergence speed with optimal values of $C_T$ and $R_D$ can be calculated as $\frac{v(C_T, R_D)}{v_{max}} = \frac{n_{min}}{n(C_T, R_D)}$ and is shown in the parenthesis in Table II, where $v_{max}$ and $n_{min}$ are the convergence speed and the required number of time slots obtained with optimal values of $C_T$ and $R_D$. Table II again reveals the robustness of the hybrid algorithm to small mismatches in parameter settings, especially with a large number of transmitters. For example, the optimal values of $C_T$ and $R_D$ for $N = 500$ are $C_T = 12$, $R_D = 0.75$, which result in a time slot number $n(12, 0.75) = 2738$. The hybrid algorithm with $C_T = 10$ and $R_D = 0.75$, which has an error of 2 in the threshold for successive negative feedback steps, can still result in a time slot number $n(10, 0.75) = 2763$ and a convergence speed of $99.1\% \cdot v_{max}$. Table II also shows that the optimal convergence speeds for $N = 20, 100$ and $500$ all result from $R_D = 0.75$. We studied the performance of the hybrid algorithm with other numbers of transmitters, and extensive simulation results show that $R_D = 0.75$ is the optimal setting for the decreasing ratio of the perturbation size regardless of the number of transmitters. The optimal values of $C_T$, the threshold for successive negative feedback steps, versus the number of transmitters are plotted in Fig. 5. It shows that the optimal value of $C_T$ increases with the number of transmitters, but the slope of the curve decreases as the number

of transmitters increases.

Fig. 6 shows the average number of time slots required to achieve an RSS of $90\% \cdot R_{\text{opt}}$ with optimal $C_T$ and $R_D$ versus the number of transmitters for the hybrid algorithm. As we can see, the required number of time slots with optimal parameter settings grows linearly with the number of transmitters. This is consistent with the analytical results presented in [5] and [7] that the number of time slots required to converge to a given fraction, e.g. $90\%$ of the perfect alignment increases with the number of transmitters, $N$, but no faster than linearly with $N$.

In Fig. 7, we compare the convergence speed for five algorithms with $N = 100$. Each curve is an average over $10^3$ runs. The curve of the hybrid algorithm is plotted with $C_T = 7$, $R_D = 0.75$. The curve of the original algorithm with optimal perturbation size $\Delta_0$ for each time slot is plotted based on the analysis in [5]. The parameter settings for Scheme 1, Scheme 2 and the original algorithm with a static perturbation size are the optimal settings obtained from extensive simulations. From Fig. 7 we see the hybrid algorithm has the best performance among the five, and Scheme 2 can achieve performance close to the original algorithm with optimal perturbation sizes in [5]. Comparing the hybrid algorithm with Scheme 2, the hybrid algorithm has a better performance due to the contribution of Scheme 1. In achieving an RSS of $90\% \cdot R_{\text{opt}}$, there is a big gap of $791 - 550 = 241$ time slots between the hybrid algorithm and the original algorithm with a static perturbation size or, in other words, the hybrid algorithm has a $\left(\frac{1}{550} - \frac{1}{791}\right) \Big/ \left(\frac{1}{791}\right) \approx 44\%$ faster convergence speed compared to the original algorithm. This gain in the convergence speed is obtained by exploiting negative feedback information in the iterations.

## V. Tracking Time-varying Channels

It is well known that the performance of beamforming is very sensitive to the phase changes in time-varying channels. In this section, we show that the hybrid algorithm proposed above can be modified to track time-varying channels while maintaining its fast convergence speed. What is more, the modified hybrid algorithm has the ability to detect variations in the speed of channel phase changes and adjust perturbation sizes adaptively according to the speed. This further enhances the robustness of the one-bit feedback algorithm in practical implementations. In order to focus on the effect of changes in channel phase responses, we still assume unit channel power gain from each transmitter to the receiver, but model the channel phase response from transmitter $i$ to the receiver as $\Psi_i[n] = \psi_i + \lambda_i[n]$, where the phases $\psi_i \sim \text{uniform}[0, 2\pi)$

are static during the convergence process as assumed in Section II. The phase drift components $\lambda_i[n]$ are assumed to be independent, identically distributed across transmitters and uncorrelated in time slots with a uniform distribution $\lambda_i[n] \sim [-\Lambda_0, \Lambda_0]$ [7], where $\Lambda_0$ is termed as the phase drift speed. The variations in phase offset at transmitters due to the oscillator internal phase noise can be modeled in the same way.

Most work in the literature on the one-bit feedback algorithms is focused on static channel conditions. Few of them extended the algorithms to time-varying channels apart from [7]. We meet the following difficulties in the algorithm design under time-varying channel conditions. First, if the received phases at the receiver $\Phi_i[n]$ become highly coherent in time slot $n$, the corresponding RSS value cannot be surpassed by subsequent perturbations as the RSS will reduce again due to channel variations. Therefore, the RSS judgement rule $R[n] > R_{best}[n]$ at the receiver is not sufficient in time-varying channels. Second, since the hybrid algorithm described above in Section III keeps reducing the perturbation size, obviously, it cannot track time-varying channels when the perturbation size becomes smaller than the phase drift speed. Third, in static channels, successive negative feedback steps only suggest that the perturbation size is too big to converge. However, in time-varying channels, this may also result from the effect of channel variations, which in contrast may require a bigger perturbation size. Fourth, successive positive feedback steps are not available to aid the design. Below we give two solutions to overcome these difficulties and apply the proposed hybrid algorithm to time-varying channels.

### *A. Solution 1*

When the phase drift speed $\Lambda_0$ is fairly small compared to the perturbation size $\delta_0$, the effect of channel variations on the RSS is negligible. Therefore, the initial stages of the convergence process can be viewed as under static channel conditions. In [7], the authors modified the original algorithm to track time-varying channels by proposing the criterion $R_{best}[n+1] = R_{best}[n] \cdot \rho$ $(0 < \rho < 1)$ every time it encounters a negative feedback step. This algorithm requires knowledge of the phase drift speed $\Lambda_0$ in order to set the value of $\rho$ and the value of perturbation size $\delta_0$. For details of the algorithm, please see [7]. A straightforward solution (Solution 1) is to apply our hybrid algorithm to the initial stages of the convergence process. When the perturbation size falls to a certain value close to the phase drift speed $\Lambda_0$, the transmitters and the receiver change to the algorithm in [7].

### B. Solution 2

Solution 2 can track time-varying channels without the knowledge of the phase drift speed $\Lambda_0$. It is summarized in Table I and explained as follows. Solution 2 operates in two modes, the *normal mode* and the *testing mode*. Operations in the normal mode are similar to the operations presented in Section III except the following. Besides updating the best RSS in memory at the receiver, in step 3) the receiver also records the minimum RSS in memory, $R_{min}[n+1] = \min(R_{min}[n], R[n])$. In step 4) when the number of successive negative feedback steps $C_N$ meets the threshold $C_T$, instead of decreasing the perturbation size $\delta_0$, the transmitters and the receiver enter the testing mode for one time slot.

We define the successive time slots taking the same perturbation size as a *size period*. In the testing mode in time slot $n$, instead of performing phase perturbations, the transmitters use their best known phases to perform transmit beamforming, $\phi_i[n] = \theta_i[n]$. The receiver measures the corresponding RSS $R[n]$, set it as the new best RSS, $R_{best}[n+1]$, and the new minimum RSS, $R_{min}[n+1]$ in memory. Therefore, the variables $R_{best}$ and $R_{min}$ actually record the maximum and the minimum RSS within a size period. This prevents the algorithm from operating in local rather than globally optimum phase solutions, which may be caused by the first of the difficulties listed above. The receiver then computes the absolute difference of $R_{best}[n]$ and $R[n]$:

$$S_C = \Big| R_{best}[n] - R[n] \Big| \tag{24}$$

and the difference of $R_{best}[n]$ and $R_{min}[n]$:

$$S_P = R_{best}[n] - R_{min}[n] \tag{25}$$

Since both $R_{best}[n]$ and $R[n]$ are obtained with the same adaptive component $\phi_i[n] = \theta_i[n]$, the scalar $S_C$ can be viewed as an estimation of the channel drift speed, $\Lambda_0$, within one size period. The scalar $S_P$ can be viewed as an estimation of the perturbation 'catch-up' speed within the same size period. By comparing $S_C$ and $S_P$, the receiver makes a judgement on the perturbation size and feeds back one bit of information telling all transmitters to adopt a larger or smaller perturbation size in the next size period. Results in [7] conclude that the perturbation 'catch-up' speed should be faster than the channel drift speed, and the perturbation size should not be too large to avoid large fluctuations in the steady-state RSS. We adopt a coefficient of $a$ in comparing $S_P$ with $S_C$ in the following simulations. Solution 2 is summarized in Table I. By

inserting only one time slot between two size periods, the modified hybrid algorithm has the ability to track time-varying channels and adjust perturbation sizes adaptively according to the rates of phase drift. The overhead of implementing this solution is very low as it does not require the knowledge of the phase drift speed and add only a few more time slots compared to the hybrid algorithm for static channels.

Fig. 8 shows the performance of Solution 2 with $N = 100$ transmitters, $C_T = 7$, $R_D = 0.75$, $a = 2$ in time-varying channels with different phase drift speeds. As we see, the modified hybrid algorithm can achieve phase coherence and provide a good beamforming gain in time-varying channels without the knowledge of channel state information. It also shows that with a relatively small phase drift speed $\Lambda_0 = \frac{\pi}{180}$ the algorithm on average achieves an RSS of $80\% \cdot R_{\text{opt}}$ in 400 time slots, which is close to the performance achieved in static channels (364 time slots) shown in Fig. 7. This confirms that the modified hybrid algorithm still maintains a fast convergence speed in the initial stages of the convergence process under time-varying channel conditions.

Fig. 9 shows one simulated instance of Solution 2 in time-varying channels when the channel phase drift speed $\Lambda_0$ is changing. As we see, Solution 2 has the ability to detect variations in the speed of channel phase changes and adjust perturbation sizes $\delta_0$ adaptively according to the speed. When $\Lambda_0$ becomes larger, it will lock the perturbation size to a bigger value to track the changes in channel phase responses. When $\Lambda_0$ becomes smaller, it will shift the perturbation size to a smaller value to obtain a superior beamforming gain. This ability makes the one-bit feedback algorithm much more robust to channel variations in practical implementations.

Below we study the performance of Solution 2 with different values of the coefficient $a$ in Table I, which is used for comparing the perturbation 'catch-up' speed, $S_P$, with the channel drift speed, $S_C$. Fig. 10 shows both the averaged performance and one simulated instance of Solution 2 in time-varying channels with $N = 100$, $\Lambda_0 = \frac{\pi}{50}$, $C_T = 7$, $R_D = 0.75$, but different values of $a = 1, 4$ and 7. As expected, if the coefficient is too small, $a = 1$, the algorithm cannot effectively track the time-varying channels. If the coefficient is too big, $a = 7$, the perturbation size in the steady state is kept around a large value. Consequently, the steady-state RSS has a bigger fluctuation and a lower average value compared to the one with a smaller coefficient, $a = 4$. We examine the performance of the average steady-state RSS with different values of $a$. Extensive simulation results show that the algorithm cannot track time-varying channels when $a < 1$. The optimal steady-state RSS can be obtained with $a = 2.6$ regardless of the channel

drift speed. After the point of 2.6, the average steady-state RSS decreases when the value of $a$ increases.

## VI. CONCLUSIONS

We proposed a hybrid one-bit feedback algorithm based on the original algorithm in the literature to achieve carrier phase alignment for distributed transmit beamforming. The hybrid algorithm can be easily applied into practical implementations and does not require any more information exchange or hardware changes. By exploiting negative feedback information in the iterations, the proposed algorithm can enhance the convergence speed of phase alignment by over 40% compared to the original algorithm. By adding one time slot per size period, the hybrid algorithm can be modified to track time-varying channels without the knowledge of channel state information. The modified hybrid algorithm has the ability to adjust perturbation sizes adaptively according to the rate of phase drift in channel variations.

## REFERENCES

[1] R. Mudumbai, D. R. Brown, U. Madhow, and H. V. Poor, "Distributed transmit beamforming: challenges and recent progress," *IEEE Commun. Mag.*, vol. 47, no. 2, pp. 102–110, 2009.

[2] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Trans. Signal Processing*, vol. 53, no. 11, pp. 4110–4124, 2005.

[3] S. Song, J. S. Thompson, P. J. Chung, and P. M. Grant, "BER analysis for distributed beamforming with phase errors," *IEEE Trans. Veh. Technol.*, vol. 59, no. 8, pp. 4169–4174, 2010.

[4] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1754–1763, 2007.

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

[6] M. Seo, M. Rodwell, and U. Madhow, "A feedback-based distributed phased array technique and its application to 60-GHz wireless sensor network," in *Proc. IEEE MTT-S International Microwave Symposium*, Atlanta, GA, June 2008, pp. 683–686.

[7] R. Mudumbai, B. Wild, U. Madhow, and K. Ramchandran, "Distributed beamforming using 1 bit feedback: from concept to realization," in *Proc. 44th Allerton Conf. Commun., Control, Comp.*, Monticello, IL, Sept. 2006, pp. 1020–1027.

[8] J. Thukral and H. Bolcskei, "Distributed spatial multiplexing with 1-bit feedback," in *Proc. 45th Allerton Conf. Commun., Control, Comp.*, Monticello, IL, Sept. 2007, pp. 502–509.

[9] P. Fertl, A. Hottinen, and G. Matz, "Perturbation-based distributed beamforming for wireless relay networks," in *Proc. IEEE Global Telecommunications Conference*, New Orleans, LO, Nov. 2008, pp. 1–5.

[10] S. Song, J. S. Thompson, P. J. Chung, and P. M. Grant, "Improving the one-bit feedback algorithm for distributed beamforming," in *Proc. IEEE Wireless Communications and Networking Conference*, Sydney, Australia, April 2010, pp. 1–6.

[11] J. A. Bucklew and W. A. Sethares, "Convergence of a class of decentralized beamforming algorithms," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2280–2288, 2008.

TABLE I

SUMMARY OF THE MODIFIED HYBRID ALGORITHM TO TRACK TIME-VARYING CHANNELS (SOLUTION 2)

**Initialization**: $C_N = 0$; $\delta_0 = \frac{\pi}{4}$; $\theta_i[1] = 0$; $\epsilon_i[1] = 0$; $R_{best}[1] = 0$; $R_{min}[1] = 0$.

**Normal mode, iterate**:

1. Set $\delta_i[n] = \pm\delta_0$ ("+" or "-" with equal probability).
2. Use $\phi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$ to perform beamforming.
3. Estimate $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$;
   Update $R_{best}[n+1] = \max(R_{best}[n], R[n])$;
   Update $R_{min}[n+1] = \min(R_{min}[n], R[n])$.

$\longrightarrow$ (*One bit feedback*.)

4. If $R[n] > R_{best}[n]$
   $\theta_i[n+1] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$; $\epsilon_i[n+1] = 0$; $C_N = 0$;
   else
   $\theta_i[n+1] = \theta_i[n]$; $\epsilon_i[n+1] = -\delta_i[n]$; $C_N = C_N + 1$;
   if $C_N \geq C_T$
   $\longrightarrow$ (*Enter the testing mode in the next time slot*.)
   end
   end

**Testing mode (one time slot)**:

1>. Use $\phi_i[n] = \theta_i[n]$ to perform beamforming.

2>. Estimate $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$;
   Update $R_{best}[n+1] = R_{min}[n+1] = R[n]$; (reactivation)
   Compute $S_C = |R_{best}[n] - R[n]|$; (estimation of the channel drift speed)
   Compute $S_P = R_{best}[n] - R_{min}[n]$. (estimation of the perturbation 'catch-up' speed)

$\longrightarrow$ (*One bit feedback*.)

3>. If $S_P > a \cdot S_C$
   $\delta_0 = \delta_0 \cdot R_D$; (decrease the perturbation size)
   else
   $\delta_0 = \delta_0 / R_D$; (increase the perturbation size)
   end

4>. Set $C_N = 0$; $\theta_i[n+1] = \theta_i[n]$; $\epsilon_i[n+1] = \epsilon_i[n]$.

$\longrightarrow$ (*Exit the testing mode*.)

(a) $N = 20$

| $R_D$ \ $C_T$ | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| 0.85 | 102 (99.0%) | 108 (93.5%) | 117 (86.3 %) | 129 (78.3%) |
| 0.8 | 101 (100%) | 102 (99.0%) | 109 (92.7%) | 118 (85.6%) |
| 0.75 | 109 (92.7%) | **101** *(Optimal)* | 104 (97.1%) | 111 (91.0%) |
| 0.7 | 121 (83.5 %) | 101 (100 %) | 102 (99.0%) | 106 (95.3 %) |
| 0.65 | 140 (72.1 %) | 106 (95.3%) | 103 (98.1%) | 104 (97.1%) |

(b) $N = 100$

| $R_D$ \ $C_T$ | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|
| 0.85 | 570 (96.5%) | 555 (99.1%) | 562 (97.9 %) | 581 (94.7%) | 603 (91.2%) |
| 0.8 | 602 (91.4%) | 559 (98.4%) | 551 (99.8%) | 558 (98.6%) | 574 (95.8%) |
| 0.75 | 641 (85.8%) | 573 (96.0%) | **550** *(Optimal)* | 551 (99.8%) | 561 (98.0%) |
| 0.7 | 695 (79.1 %) | 592 (92.9 %) | 560 (98.2%) | 551 (99.8 %) | 553 (99.5%) |
| 0.65 | 760 (72.4 %) | 619 (88.9%) | 570 (96.5%) | 552 (99.6%) | 551(99.8%) |

(c) $N = 500$

| $R_D$ \ $C_T$ | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|
| 0.85 | 2756 (99.3%) | 2775 (98.7%) | 2815 (97.3 %) | 2864 (95.6%) | 2930 (93.4%) |
| 0.8 | 2743 (99.8%) | 2746 (99.7%) | 2755 (99.4%) | 2787 (98.2%) | 2832 (96.7%) |
| 0.75 | 2763 (99.1%) | 2748 (99.6%) | **2738** *(Optimal)* | 2757 (99.3%) | 2785 (98.3%) |
| 0.7 | 2816 (97.2 %) | 2770 (98.8 %) | 2740 (99.9%) | 2745 (99.7 %) | 2768 (98.9%) |
| 0.65 | 2877 (95.2 %) | 2786 (98.3%) | 2755 (99.4%) | 2747 (99.7%) | 2755(99.4%) |

TABLE II

THE AVERAGE NUMBER OF TIME SLOTS REQUIRED TO ACHIEVE $90\% \cdot R_{\text{opt}}$ WITH $N = 20, 100$ AND $500$ FOR THE HYBRID ALGORITHM, WHERE $C_T$ IS THE THRESHOLD FOR SUCCESSIVE NEGATIVE FEEDBACK STEPS AND $R_D$ IS THE DECREASING RATIO OF THE PERTURBATION SIZE. THE PERCENTAGE VALUE IN EACH PARENTHESIS REPRESENTS THE CORRESPONDING CONVERGENCE SPEED DIVIDED BY THE OPTIMAL CONVERGENCE SPEED.

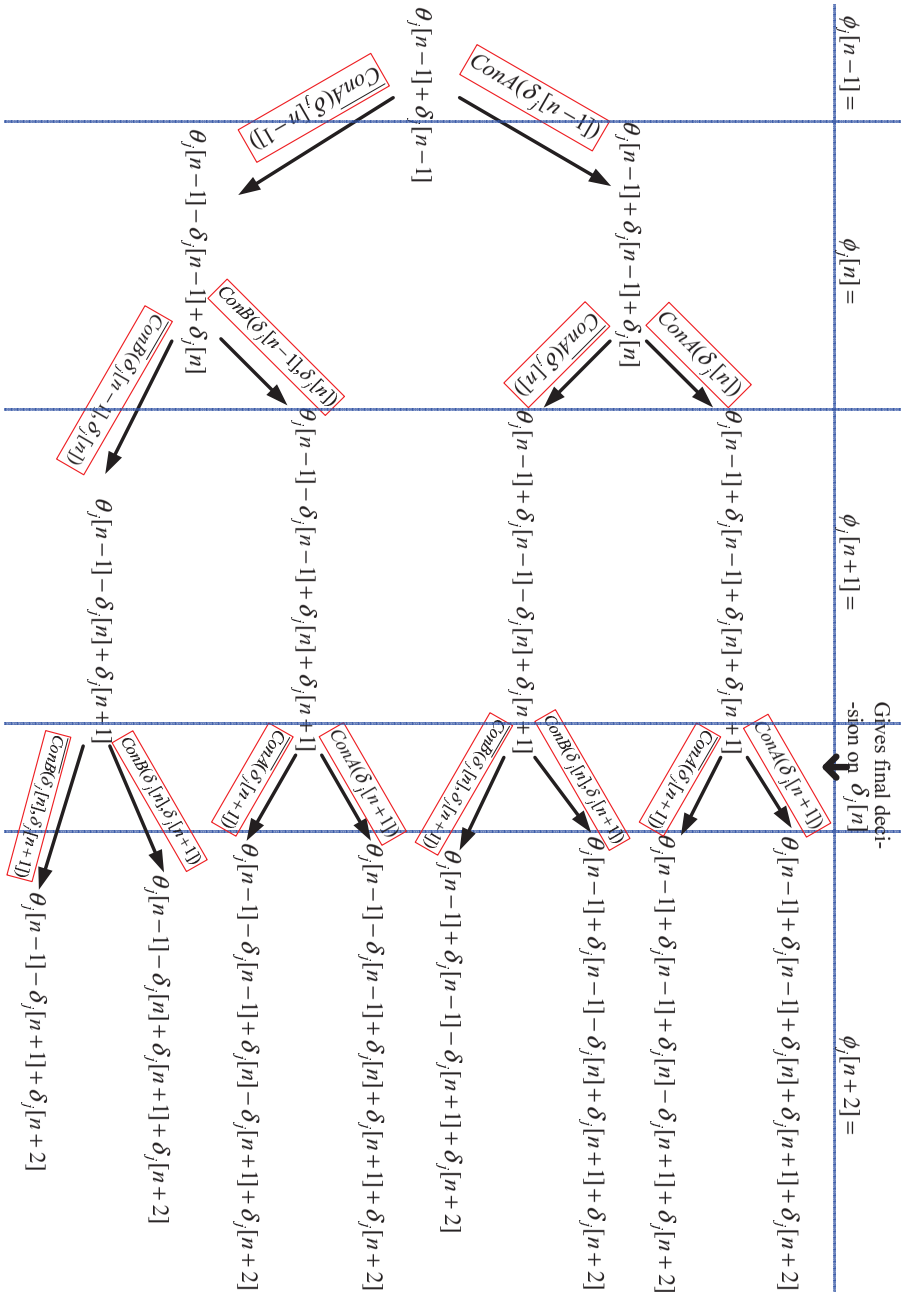



Fig. 1. A flowchart of the adaptive component $\phi_j$ under Scheme 1.

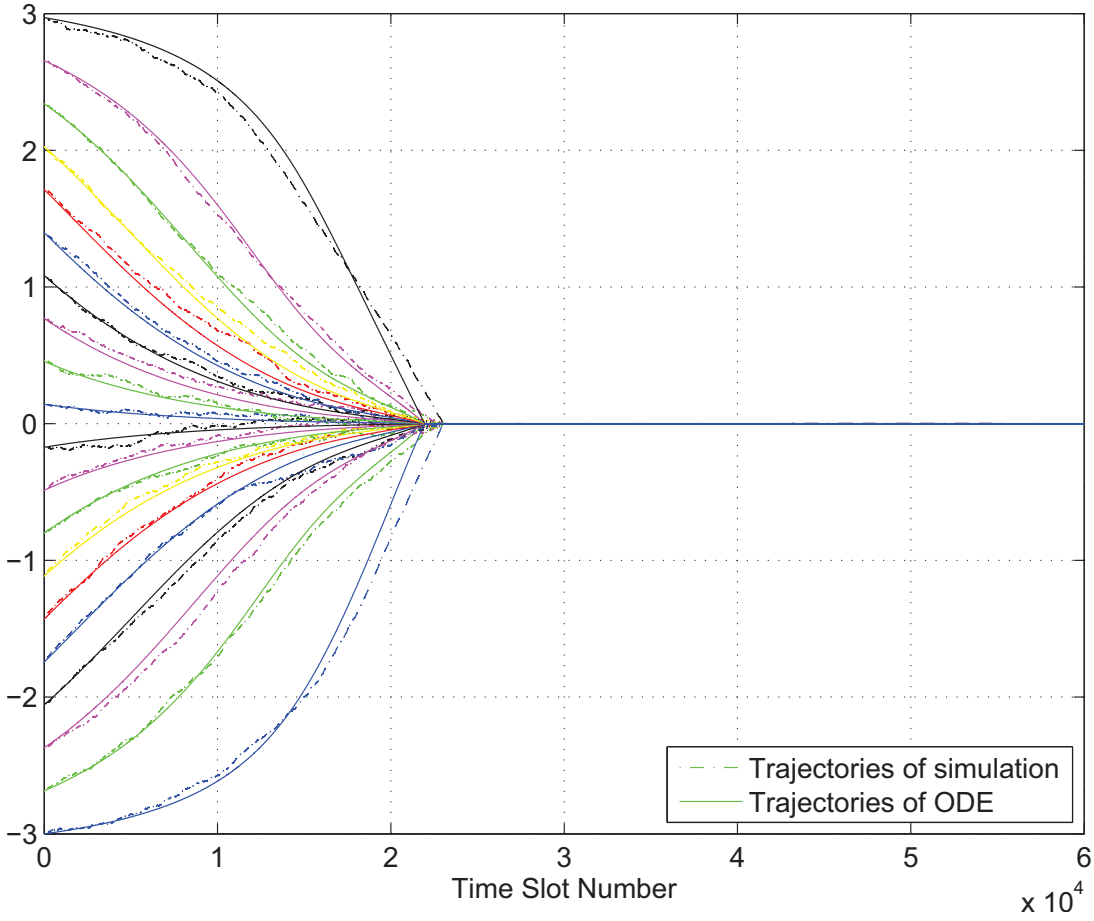



Fig. 2. Comparison of the trajectories of the phases $\hat{\Phi}$ obtained from simulation (dashed lines) with the trajectories of the ODE (solid lines) in (21) for Scheme 1 with $N = 20$, $\delta_0 = 6 \times 10^{-4}$. The convergence of $\hat{\Phi}$ to zero is equivalent to the convergence of the phase alignment process.

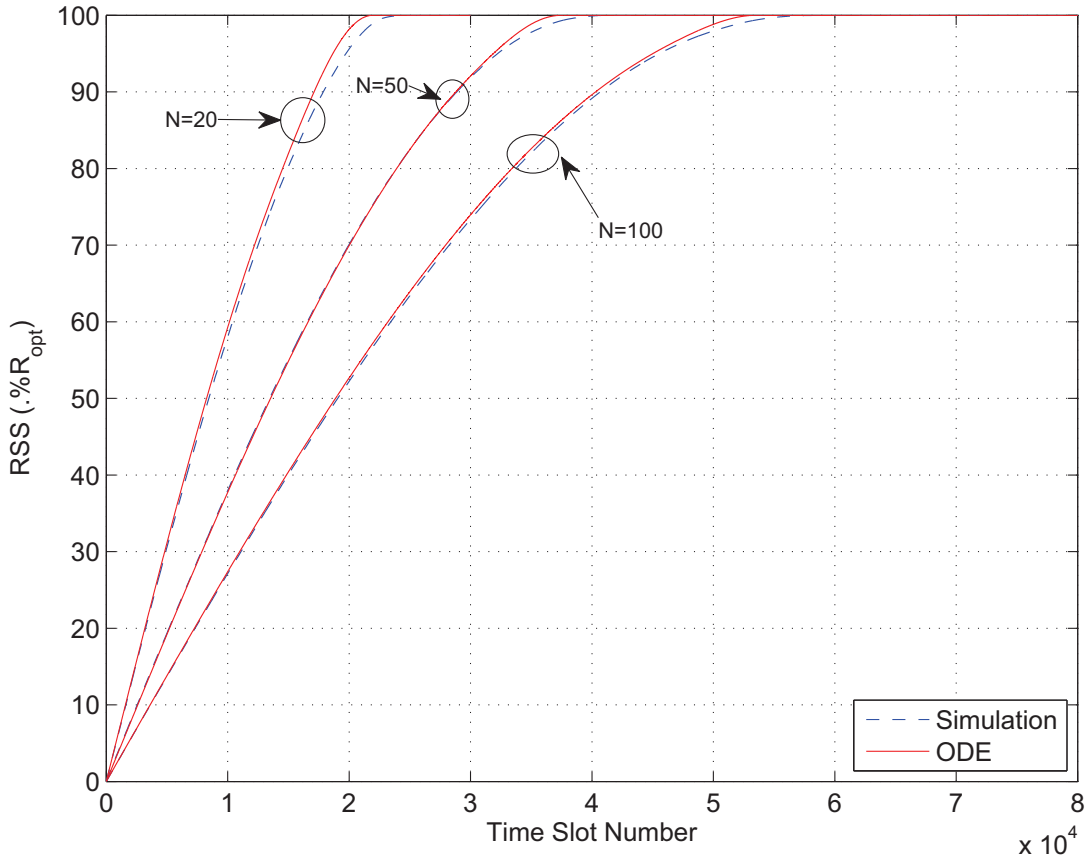



Fig. 3. Comparison of the simulation results (dashed lines) with the results obtained from the ODE (solid lines) in (21) for the RSS versus number of time slots with $N = 20, 50, 100$ and $\delta_0 = 6 \times 10^{-4}$.

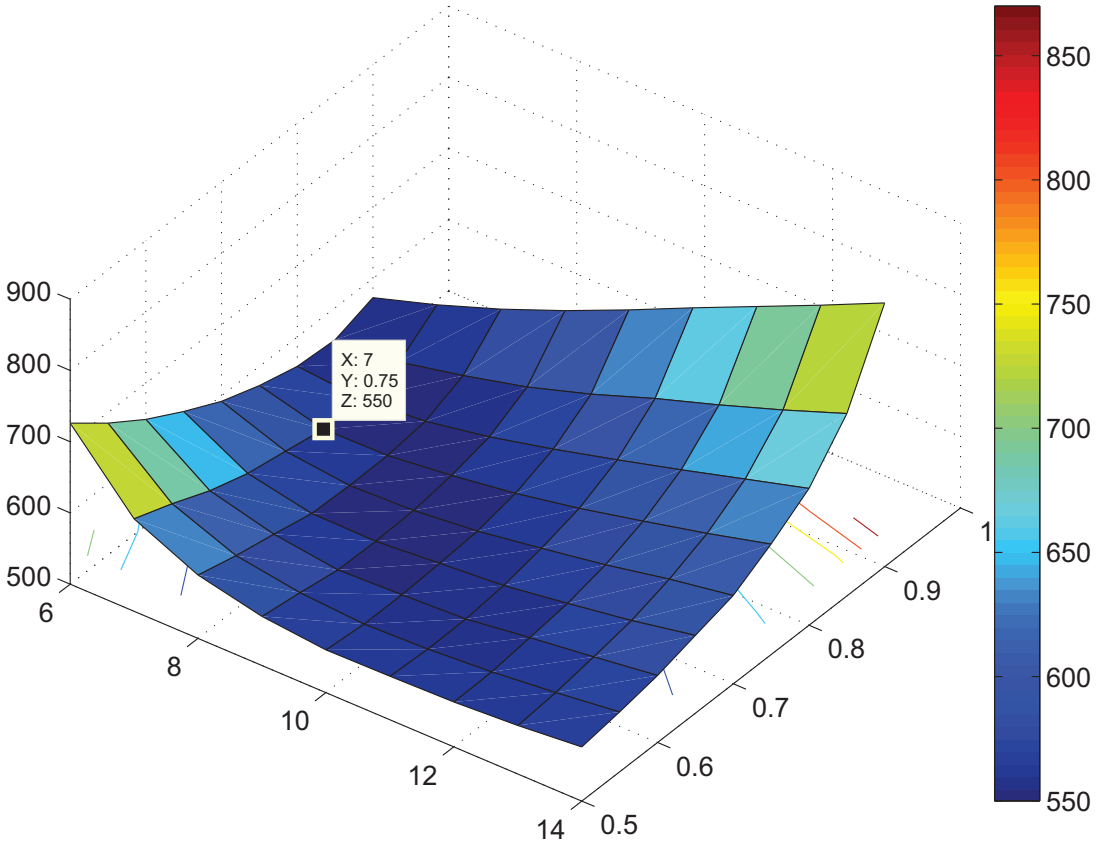



Fig. 4. The average number of time slots required to achieve $90\% \cdot R_{\text{opt}}$ with $N = 100$ for the hybrid algorithm, where $C_T$ is the threshold for successive negative feedback steps and $R_D$ is the decreasing ratio of the perturbation size.

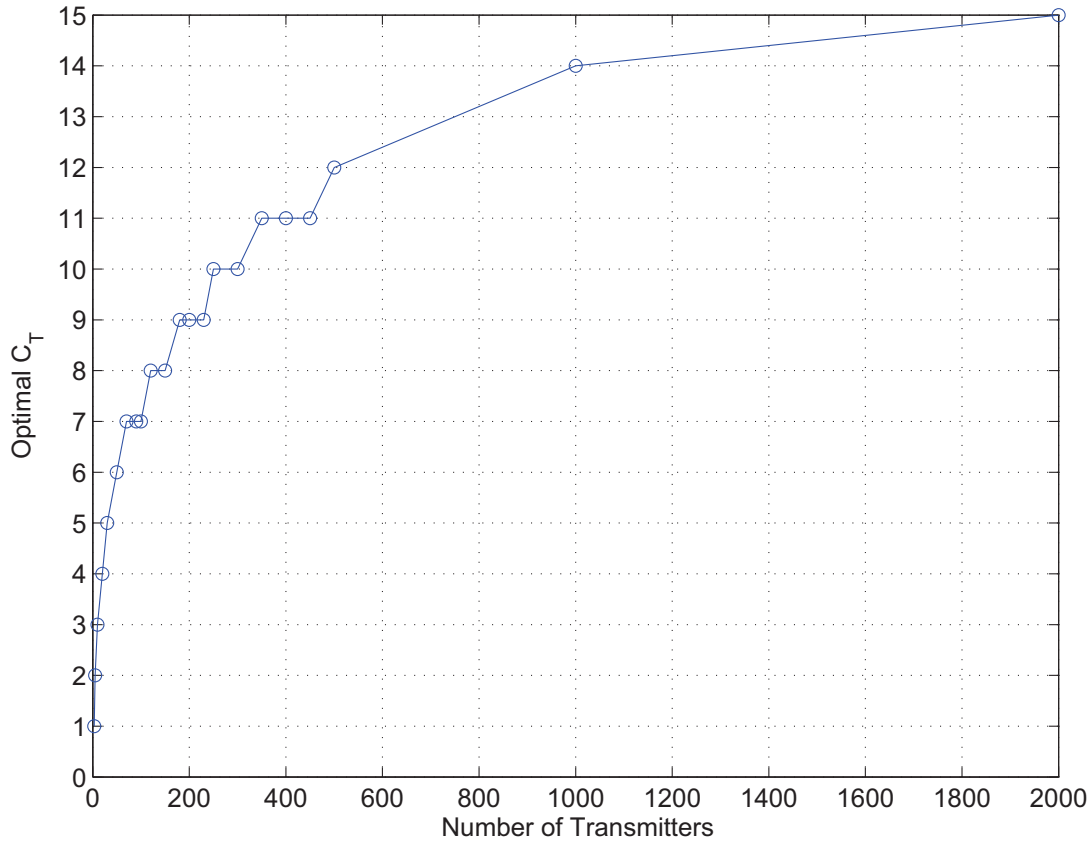



Fig. 5. The optimal $C_T$, the threshold for successive negative feedback steps, versus the number of transmitters for the hybrid algorithm.

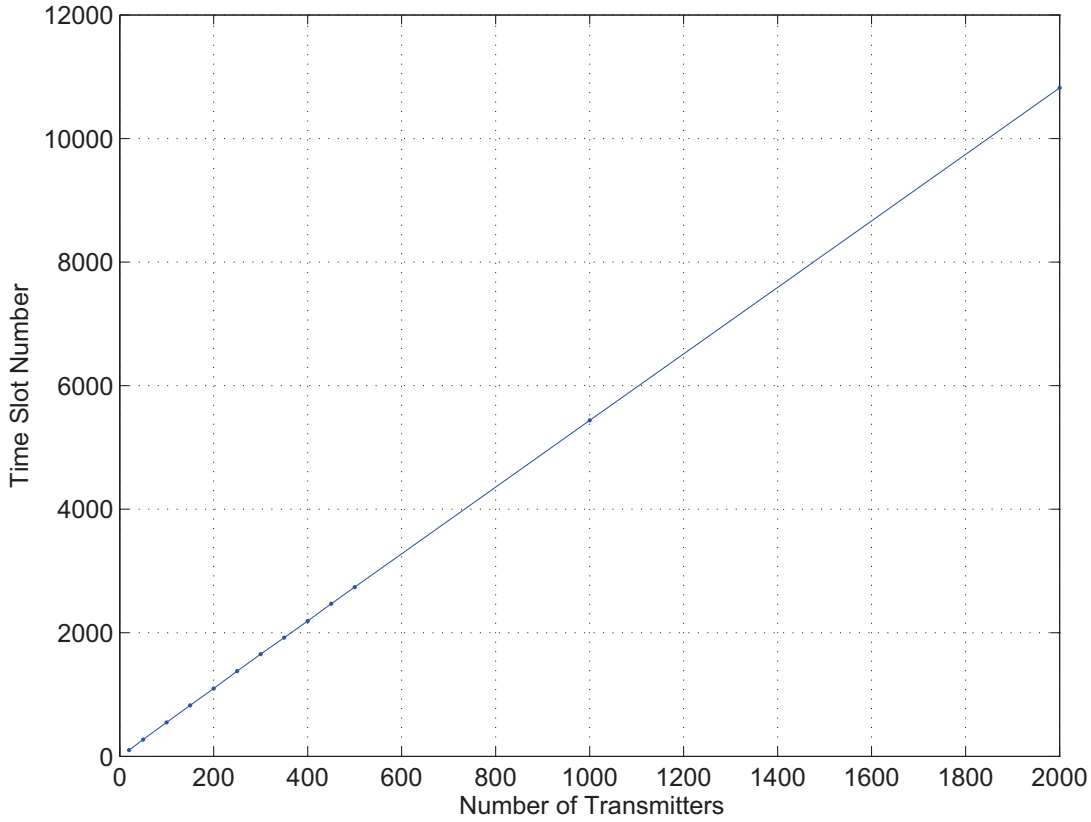



Fig. 6. The average number of time slots required to achieve $90\% \cdot R_{\mathrm{opt}}$ versus the number of transmitters for the hybrid algorithm.

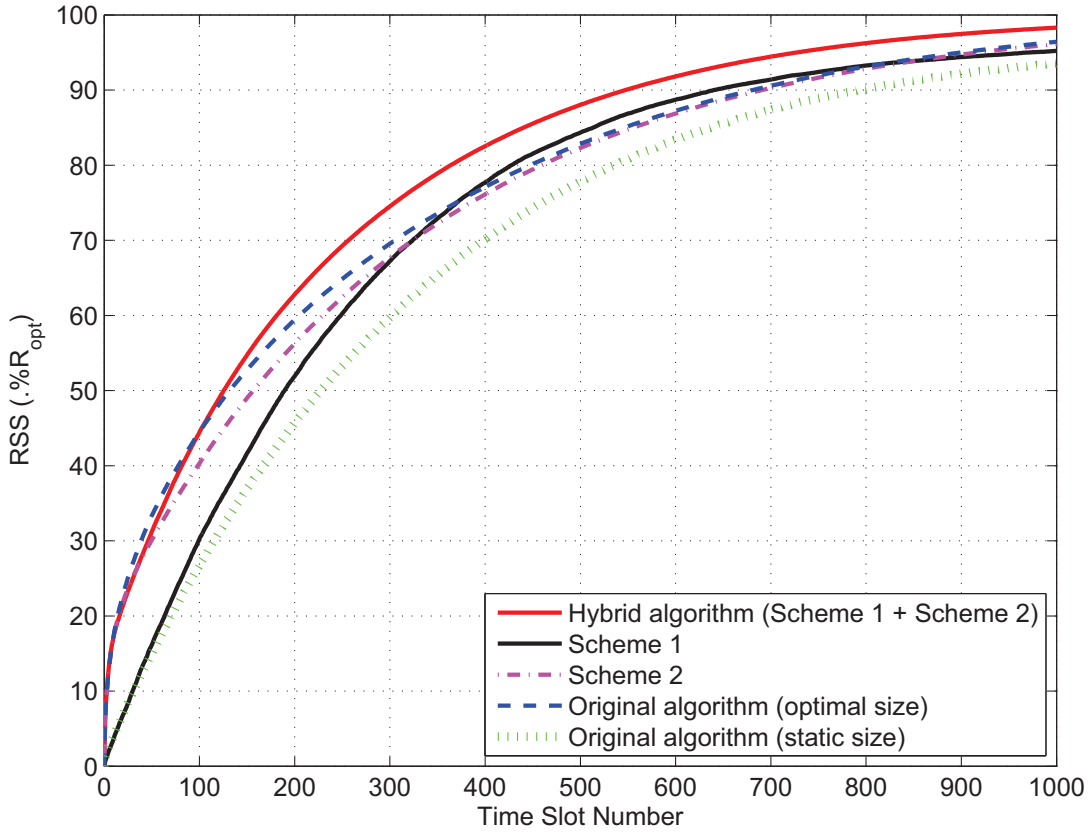



Fig. 7. Comparison of the hybrid algorithm with the original algorithm for the received signal strength versus time slots for $N = 100$.

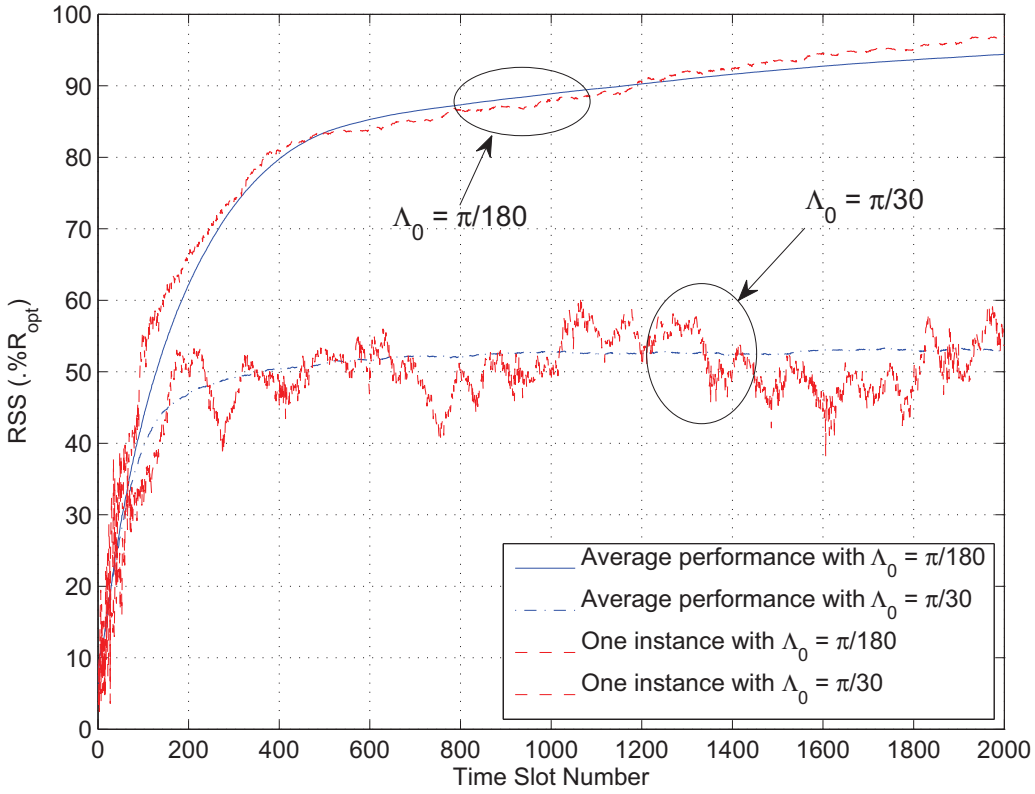



Fig. 8. Performance of the modified hybrid algorithm (Solution 2) in time-varying channels with different channel phase drift speeds $\lambda_i[n] \sim [-\frac{\pi}{180}, \frac{\pi}{180}]$, $\lambda_i[n] \sim [-\frac{\pi}{30}, \frac{\pi}{30}]$ for $N = 100$.

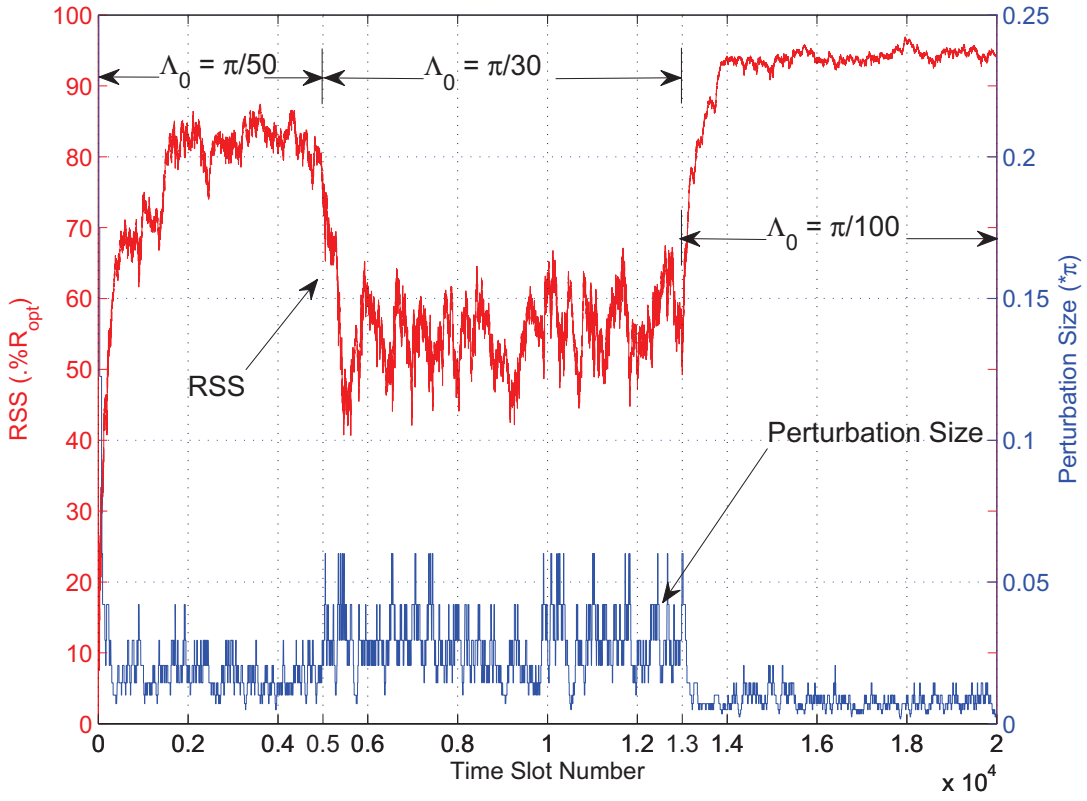



Fig. 9. One simulated instance of the modified hybrid algorithm (Solution 2) in time-varying channels with variable phase drift speeds $\Lambda_0$ for $N = 100$. The red curve at the top shows the RSS versus time slots. The blue curve at the bottom shows the perturbation sizes versus time slots.

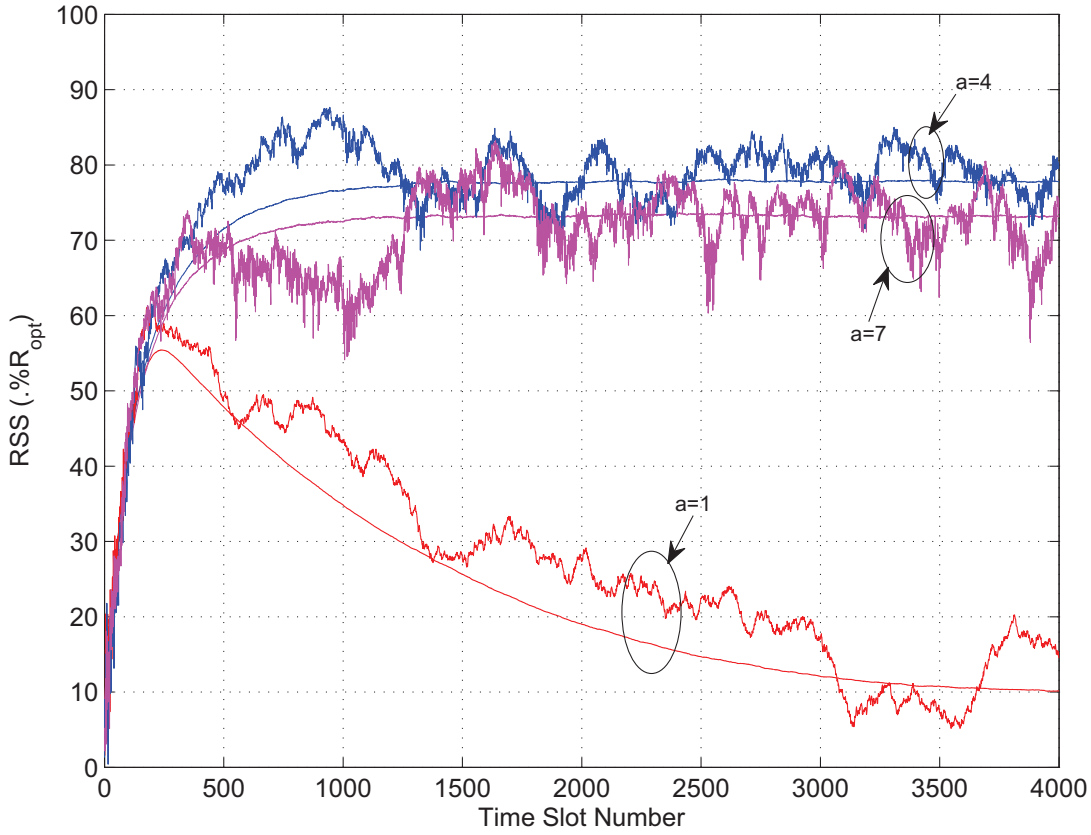



Fig. 10. Performance of the modified hybrid algorithm (Solution 2) in time-varying channels with $N = 100$ transmitters, channel drift speed $\Lambda_0 = \frac{\pi}{50}$, and different values of the coefficient $a = 1, 4, 7$.

# BER Analysis for Distributed Beamforming with Phase Errors

Shuo Song, *Student Member, IEEE,* John S. Thompson, *Member, IEEE,* Pei-Jung Chung, *Member, IEEE,* and Peter M. Grant, *Fellow, IEEE*

***Abstract*—The probability of error for distributed transmit beamforming with phase errors is *not* available in closed form in the literature. This paper presents an investigation into the bit error ratio of distributed transmit beamforming with phase errors for equal power transmission in the context of wireless sensor networks. We derive two distinct formulae to approximate the error probability performance for binary phase shift keying over Rayleigh fading channels corresponding to small numbers of nodes (e.g. $N \leqslant 10$) and large numbers of nodes (e.g. $N \geqslant 20$) respectively. Simulation results show a good match with the analytical results. The effects of the phase errors on the beamforming performance are examined for various numbers of nodes and different levels of total transmit power.**

***Index Terms*—Distributed beamforming, sensor networks, equal gain combining, bit error ratio performance.**

## I. INTRODUCTION

In the context of wireless sensor networks, sensor nodes are usually randomly located in the sensing area to collect information on demand, either for on-line data collection, e.g. periodic sampling of a parameter of interest, or for alarm triggering, e.g. abnormal parameter variation in the monitored environment. Then the sensor nodes are intended to send or report the information to a destination, which may be far away from the sensor network in some application scenarios. Due to hardware constraints and low-cost configurations, each sensor node is usually equipped with one single antenna and energy-limited batteries which cannot easily be replaced. These conditions make coherent cooperative transmission, or in other words, transmit beamforming a very promising form of transmission. This is sometimes called distributed beamforming or collaborative beamforming in the literature. The motivation for applying beamforming techniques in wireless sensor networks is to reduce the energy requirement for each sensor node in signal transmission and to extend the communication range to a far field receiver. Recently there have been several papers discussing the practical problems of realizing distributed beamforming [1] and describing its potential benefits in applications [2].

The principle behind the transmit beamforming technique is that the signals transmitted from each antenna should be frequency-synchronized and phase-adjusted so that the signals can add coherently at the receiver. While conventional beamforming is implemented on a device with a centralized antenna array, distributed beamforming is performed by a virtual antenna array composed of randomly located sensor nodes, each of which has an independent carrier oscillator. Unlike conventional beamforming, phase errors among the signals arriving at the receiver cannot be avoided in distributed beamforming. This may arise from the noise in individual carrier oscillators [1], node position errors [3], or timing synchronization errors [1]. To measure the beamforming performance, the bit error ratio (BER) expression of distributed beamforming with phase errors is both theoretically and practically important but not available to date.

The phase error effect on the average beamforming gain has been initially studied in [1], while its effect on the far-field beam pattern has been comprehensively studied in [3]. In [4], we have studied the error probability of maximal ratio transmission (MRT) in distributed beamforming with phase errors, where the analysis provides a good prediction on the achievable BER only for large number of nodes. In this paper, we investigate the error probability for the more realistic case of equal power transmission in distributed beamforming with phase errors. We derive expressions for the BER performance of binary phase shift keying (BPSK) modulation as a function of the number of nodes, phase errors and total transmit power for both small number of nodes and large number of nodes.

## II. SYSTEM MODEL

We consider a wireless sensor network composed of $N$ sensor nodes which collaboratively beamform a narrowband message signal $m(t)$ to a distant receiver. This is performed in a distributed manner by each node modulating $m(t)$ at the same carrier frequency. Each sensor node pre-compensates the phase response of its channel to the receiver by adjusting its initial phase settings [1] in order to ensure phase alignment at the receiver, as illustrated in Fig. 1.

Considering a large number of sensor nodes, full channel state information (CSI) may be hard to obtain in practice. Techniques have been designed to pre-compensate the channel phase response to achieve phase alignment [5], [6]. The lack of full CSI and power limitation on the sensor nodes make MRT techniques unrealistic. Instead, more practically, we assume each sensor node transmits with equal power and applies channel phase compensation at the transmitter side. In order to reveal the fact that beamforming gain grows with the number of nodes $N$, we assume the overall power transmitted by all



The authors are with the Institute for Digital Communications, University of Edinburgh, Edinburgh EH9 3JL, UK (e-mail: {S.Song, John.Thompson, P.Chung, Peter.Grant}@ed.ac.uk).

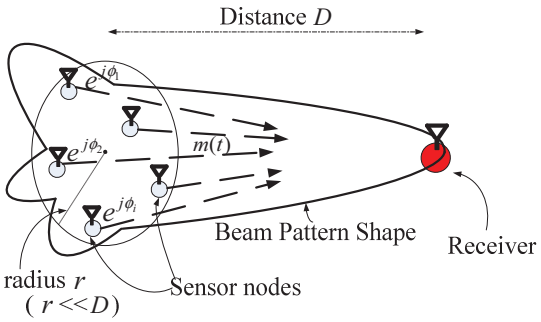



Fig. 1. System model for distributed beamforming

the nodes is fixed as $P$, where each node actually transmits with a power of $\frac{P}{N}$. This then permits us to model the BER improvement with distributed beamforming gain. The complex baseband model of the received signal is given by:

$$r(t) = \sum_{i=1}^{N} |h_i(t)| e^{j\phi_i(t)} \sqrt{\frac{P}{N}} m(t) + n(t), \qquad (1)$$

where $h_i(t)$ is the channel gain for sensor node $i$, $\phi_i(t)$ is the cumulative phase error of the carrier signal at the receiver for sensor node $i$, $n(t) \sim CN(0, \sigma_n^2)$ is additive white Gaussian noise (AWGN). For simplicity, we assume all phase errors $\phi_i(t)$ are independently and uniformly distributed, bounded by $(-\phi_0, \phi_0)$, across time and across nodes, which is a common assumption adopted in previously reported investigations [1], [3], [7]. The scalar $\phi_0$ is usually expected to be less than $60^\circ$ in practice in order to achieve a reasonable beamforming gain [1]. We assume the signals experience slow fading channels, and the channel coefficients are independent, circularly symmetric, complex Gaussian distributed, denoted as $h_i(t) \sim CN(0, 2\sigma_c^2)$.

We focus on the scenario of Rayleigh fading channels where the BER for static channels can be regarded as a special case and can be easily derived. After matched filter detection and analog-to-digital conversion, the decision variable for BPSK modulation can be expressed as:

$$r_D = \pm\sqrt{\frac{P}{N}} \left| \sum_{i=1}^{N} |h_i| e^{j\phi_i} \right| + \widehat{n} = s + \widehat{n}, \qquad (2)$$

and the corresponding decision rule is $m(t) = \begin{cases} 1 & r_D > 0 \\ 0 & r_D < 0 \end{cases}$, where $\widehat{n}$ represents the noise, $n(t)$, projected onto the received signal vector.

## III. BER FOR SMALL NUMBER OF NODES - METHOD 1

The BER for BPSK over a fixed channel in the presence of AWGN is given by $P_e(\gamma) = \frac{1}{2}\mathrm{erfc}(\sqrt{\gamma})$, where $\gamma$ is the received signal-to-noise ratio (SNR) per bit, and $\mathrm{erfc}(.)$ is the complementary error function. When the channel gain is random, the average BER over all values of $\gamma$ is given by [8]:

$$P_e = \int_0^\infty P_e(\gamma) p(\gamma) d\gamma, \qquad \gamma = \frac{\frac{P}{N} \left| \sum_{i=1}^{N} |h_i| e^{j\phi_i} \right|^2}{\sigma_n^2}, \qquad (3)$$

where $p(\gamma)$ denotes the probability density function (pdf) of $\gamma$. Due to the effect of phase errors, the distribution of $\gamma$ is unknown and the pdf expression of $\gamma$ is difficult to evaluate.

However, the probability of error for equal gain combiners with $L$ independent receive branches over Rayleigh channels has been studied in [9]. The decision variable for coherent BPSK in [9] is expressed as:

$$r_d = \pm(x_1 + x_2 + \cdots + x_L) + \sum_{i=1}^{L} n_i, \qquad (4)$$

where $x_i$ is the amplitude of the received signal at the output of the $i$th branch with a Rayleigh distribution. The scalar $n_i$ is the complex baseband Gaussian noise at the output of the $i$th branch.

Although [9] investigates equal gain diversity receivers and their system models are different from ours, as shown above, the decision variable in (4) is identical to (2) when $L = N$ if we neglect the phase errors in our model and modify the noise component. The noise in (4) comprises $L$-branch superimposed noise while in (2) there is only one AWGN component. By studying [9] and modifying the coefficients of the noise, we can thus derive the BER expression for distributed beamforming *without* phase errors over Rayleigh channels as:

$$P_e \approx \frac{1}{2} - \frac{1}{\pi} \sum_{m=1}^{M} \omega_m G(z_m, \Omega, \sigma_n^2, N), \qquad (5)$$

where:

$$\begin{aligned} G(z, \Omega, \sigma_n^2, N) &= \mathrm{Im}\left\{ \left[ {}_1F_1(-\tfrac{1}{2}; \tfrac{1}{2}; \tfrac{\Omega z^2}{\sigma_n^2 + N\Omega}) \right. \right. \\ &\quad \left. \left. + jz\sqrt{\tfrac{\pi\Omega}{\sigma_n^2 + N\Omega}} \right]^N \right\} z^{-1}, \end{aligned} \qquad (6)$$

and $\Omega = \mathrm{E}\left[ \left( \sqrt{\frac{P}{N}} |h_i| \right)^2 \right] = \frac{2\sigma_c^2 P}{N}$ is the average energy of a Rayleigh distributed variable in (2) and in the case of no phase errors, $\phi_i = 0$ in that equation. The function $\mathrm{E}[x]$ denotes the expectation of $x$ and $2M$ is the order of Hermite polynomials. The definition of the above confluent hypergeometric function, ${}_1F_1(a; b; x)$, is given in [10].

Equation (5) refers to Hermite integration explained on page 890 in [11], and the values for $\omega_m$ and $z_m$ are given on page 924 in [11]. The validity of using the Hermite method of integration to compute the error probability for equal gain combiners has been fully justified in [9]. Equation (5) becomes more accurate when $M$ tends to infinity. However, it is shown in [9] that $M = 10$ is sufficient to ensure acceptable accuracy.

If there are phase errors, i.e. $\phi_0 \neq 0$, the power of the signal part, $s$, in (2) is reduced by phase errors, and the expectation of the received SNR becomes smaller than the case without phase errors. In order to incorporate the effects of phase errors, we

define a factor $\eta$. We multiply every single Rayleigh variable, $\sqrt{\frac{P}{N}}|h_i|$, with $\eta$ to make the expectation of the received SNR equal: $\mathrm{E}\Big[\frac{P}{N}\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|^2\Big] = \mathrm{E}\Big[\frac{P}{N}\Big(\sum_{i=1}^{N}\eta|h_i|\Big)^2\Big]$. Re-arranging this equation, we have:

$$\eta^2 = \mathrm{E}\Big[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|^2\Big] \Big/ \mathrm{E}\Big[\Big(\sum_{i=1}^{N}|h_i|\Big)^2\Big]. \quad (7)$$

The expression of $\eta^2$ in terms of the number of nodes $N$ and the phase error range $\phi_0$ is derived in Appendix A. The average power of an adjusted Rayleigh variable, $\eta\sqrt{\frac{P}{N}}|h_i|$, becomes $\Omega' = \mathrm{E}\Big[\Big(\eta\sqrt{\frac{P}{N}}|h_i|\Big)^2\Big] = \eta^2\Omega$. We use $\Omega'$ to substitute for $\Omega$ in (6). The purpose of this is to use the distribution of a sum of $N$ Rayleigh variables to approximate the distribution of the signal, $s$, in (2) while keeping the expectation of the received SNR per bit $\mathrm{E}[\gamma]$ to be the same.

The expectation of $\gamma$ has been adjusted by introducing $\eta$. There is still a difference between the actual variance of the received signal and the variance after the expectation adjustment. Thus, we further define a variable, $\sigma_d^2$, to compensate for the residual variance between the two:

$$\sigma_d^2 = \frac{P}{N}\Bigg(\mathrm{Var}\Big[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|\Big] - \mathrm{Var}\Big[\eta\Big(\sum_{i=1}^{N}|h_i|\Big)\Big]\Bigg), \quad (8)$$

where $\mathrm{Var}[x]$ denotes the variance of $x$. The expression of $\sigma_d^2$ in terms of the number of nodes $N$ and the phase error range $\phi_0$ is derived in Appendix A. We treat this residual variance as a contribution to the receiver noise, and compute the total noise power as $\widetilde{\sigma}_n^2 = \sigma_n^2 + \sigma_d^2$.

By substituting $\Omega'$ for $\Omega$, $\widetilde{\sigma}_n^2$ for $\sigma_n^2$ into (6), the final BER expression for distributed beamforming with phase errors over Rayleigh channels is given by (5), while the function for computation becomes $G(z, \Omega', \widetilde{\sigma}_n^2, N)$. We use equation (5) and $G(z, \Omega', \widetilde{\sigma}_n^2, N)$ to compute the BER in the simulations of Section V, and this is denoted as method 1. Method 1 is valid for any number of nodes, but it is proposed here to use method 1 only for small number of nodes due to its high computational complexity for large $N$. This will be justified and further explained in Section V.

## IV. BER FOR LARGE NUMBER OF NODES - METHOD 2

In (3) we see that the distribution of $\gamma$ mainly depends on the distribution of $\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|^2$. Therefore, for simplicity, we define the concept of an equivalent channel, $H$, as:

$$H = \sum_{i=1}^{N}|h_i|e^{j\phi_i}. \quad (9)$$

Based on the central limit theorem (CLT), with a large number of nodes $N$, and the independent identically distributed (i.i.d.) random variables, $h_i$, which are independent from the i.i.d. random variables $\phi_i$, the key element which determines the error probability can be expressed as:

$$\begin{aligned}|H|^2 &= \Big|\sum_{i=1}^{N}|h_i|\cos\phi_i + j\sum_{i=1}^{N}|h_i|\sin\phi_i\Big|^2 \\ &= |a+jb|^2, \end{aligned} \quad (10)$$

where $a = \sum_{i=1}^{N}|h_i|\cos\phi_i \sim N(\mu_a, \sigma_a^2)$, $b = \sum_{i=1}^{N}|h_i|\sin\phi_i \sim N(\mu_b, \sigma_b^2)$. A similar analysis of the beamforming gain using the CLT has been presented in [1]. Since $h_i \sim CN(0, 2\sigma_c^2)$, and $\phi_i \sim (-\phi_0, \phi_0)$, the expectations and variances of $a$ and $b$ can be obtained as follows:

$$\begin{aligned}\mu_a &= N\cdot\mathrm{E}\Big[|h_i|\cos\phi_i\Big] = N\cdot\mathrm{E}\Big[|h_i|\Big]\cdot\mathrm{E}\Big[\cos\phi_i\Big] \\ &= \frac{\sqrt{2\pi}N\sigma_c\sin\phi_0}{2\phi_0}, \qquad \mu_b = 0, \end{aligned} \quad (11)$$

$$\begin{aligned}\sigma_a^2 &= N\Bigg(\mathrm{E}\Big[\Big(|h_i|\cos\phi_i\Big)^2\Big] - \Big(\mathrm{E}\Big[|h_i|\cos\phi_i\Big]\Big)^2\Bigg) \\ &= N\sigma_c^2\left(1 + \frac{\sin 2\phi_0}{2\phi_0} - \frac{\pi}{2}\Big(\frac{\sin\phi_0}{\phi_0}\Big)^2\right), \end{aligned} \quad (12)$$

$$\begin{aligned}\sigma_b^2 &= N\Bigg(\mathrm{E}\Big[\Big(|h_i|\sin\phi_i\Big)^2\Big] - \Big(\mathrm{E}\Big[|h_i|\sin\phi_i\Big]\Big)^2\Bigg) \\ &= N\sigma_c^2\left(1 - \frac{\sin 2\phi_0}{2\phi_0}\right). \end{aligned} \quad (13)$$

From (12) and (13) we see, for the equivalent channel, $H$, with most values of $\phi_0$ (i.e. $\phi_0 \neq 45°$), the variance of the real part $\sigma_a^2$ and the imaginary part $\sigma_b^2$ are not equal, which means the expression of the pdf of $|H|^2$ is difficult to compute. However, if we make the approximation that the variance of the real part and the variance of the imaginary part of $H$ are equal, the magnitude gain of the equivalent channel, $|H|$, follows a Rician distribution, and the channel gain, $|H|^2$, has a non-central chi-square distribution with 2 degrees of freedom, where the noncentrality parameter, $\lambda^2$, and the variance, $\sigma^2$, satisfy [8]:

$$\mathrm{E}\Big[|H|^2\Big] = 2\sigma^2 + \lambda^2, \qquad \mathrm{Var}\Big[|H|^2\Big] = 4\sigma^4 + 4\sigma^2\lambda^2. \quad (14)$$

From (14), we can derive the expressions for $\lambda^2$ and $\sigma^2$ as:

$$\lambda^2 = \sqrt{2\mathscr{A}^2 - \mathscr{B}}, \qquad \sigma^2 = \frac{\mathscr{A} - \sqrt{2\mathscr{A}^2 - \mathscr{B}}}{2}, \quad (15)$$

where $\mathscr{A} = \mathrm{E}\Big[|H|^2\Big]$, $\mathscr{B} = \mathrm{E}\Big[|H|^4\Big]$. The expressions for $\mathscr{A}$, $\mathscr{B}$ in terms of $N$ and $\phi_0$ are derived in Appendix B.

The BER for BPSK signalling in a Rician fading channel has been studied in [12], permitting the closed-form expression for the BER of our model to be easily obtained as:

$$P_e = Q_1(u,w) - \frac{1}{2}\left(1 + \sqrt{\frac{d}{1+d}}\right)\exp\Big(-\frac{u^2+w^2}{2}\Big)I_0(uw), \quad (16)$$

where:

$$d = \frac{2\sigma^2 P}{\sigma_n^2 N}, \quad u = \sqrt{\frac{\lambda^2}{2\sigma^2}\cdot\frac{1+2d-2\sqrt{d(1+d)}}{2(1+d)}},$$
$$w = \sqrt{\frac{\lambda^2}{2\sigma^2}\cdot\frac{1+2d+2\sqrt{d(1+d)}}{2(1+d)}}, \tag{17}$$

and $I_0(x)$ is the zeroth-order-modified Bessel function of the first kind, $Q_1(x,y)$ is the Marcum $Q$-function, both defined in [8]. An approximation of $I_0(x)$ is given by [13] in Chapter 6 as $I_0(x) \approx \frac{1}{\sqrt{2\pi x}}\exp(x)$ $(x \gg 0)$ and after manipulation, (16) can be simplified to:

$$P_e \approx Q_1(u,w) - \frac{1}{2\sqrt{2\pi uw}}\left(1+\sqrt{\frac{d}{1+d}}\right) \exp\left(-\frac{(u-w)^2}{2}\right), \quad uw \gg 0. \tag{18}$$

To the best of the authors' knowledge, (18) is a new result which simplifies the BER expression.

By substituting (15) into (17) and (18), we can obtain the final BER expression for BPSK signalling in distributed beamforming for large number of nodes and we define this as method 2 in the following simulations. By doing so, we actually use Rician distribution to approximate the distribution of $|H|$ while keeping the second and fourth moments of $|H|$ unchanged. These expressions may be extended to other modulation schemes by studying [14].

## V. Results and Discussions

In this section, we present some simulation results in accordance with our previous assumptions for distributed beamforming with phase errors over Rayleigh fading channels, and compare them with the analytical results given by mathematical expressions derived in Section III and Section IV. We set the channel coefficients as $h_i(t) \sim CN(0,1)$ and the AWGN noise as $n(t) \sim CN(0,1)$, so the value of the total transmit power $P$ in the figures can be viewed as normalized to the noise power at the receiver. Given equation (3), with a perfect phase alignment at the receiver, $P = 1$ implies $\mathrm{E}[\gamma] \approx 6\,\mathrm{dB}$ when $N = 5$, $\mathrm{E}[\gamma] \approx 12\,\mathrm{dB}$ when $N = 20$. The simulation results for every point in the following figures are averaged over $10^6$ runs. As the received SNR cannot illustrate the advantages of beamforming gain and the effects of the number of nodes and phase errors, our simulation results and analytical results are plotted as BER vs fixed total transmit power $P$, which is one of the major concerns in practical design in wireless sensor networks. We have derived two expressions to predict the BER results for small number of nodes and large number of nodes separately. For simplicity, we denote equation (5) in Section III as method 1, while equation (18) in Section IV is denoted as method 2.

Fig. 2 shows the comparison of the simulation results with the analytical results based on method 1 for very small but different numbers of nodes $N = 3, 5$ and increasing phase error ranges $\phi_0 = 18°, 36°, 54°$ and $72°$. As can be seen, our analysis shows a good match with the simulation results for all values of $\phi_0$ up to $72°$ with both $N = 3$ and $N = 5$. Because method 2 is based on the CLT it thus has a large deviation from the simulation results for a small $N$, we only present the results based on method 1 in Fig. 2. (The accuracy of method 1 and method 2 when increasing $N$ from small numbers to large numbers are compared later in Fig. 4.) From Fig. 2 we see that increasing the number of nodes $N$ can dramatically reduce the transmit power requirement for the same BER performance. It also shows that with a fixed increment in $\phi_0$, the phase errors have a more significant effect on the BER performance at higher values of $\phi_0$. Taking the curves for $N = 5$ for example, subject to the same BER at $10^{-2}$, the performance loss when increasing from $\phi_0 = 54°$ to $72°$ is larger than the degradation when moving from $\phi_0 = 18°$ to $36°$.

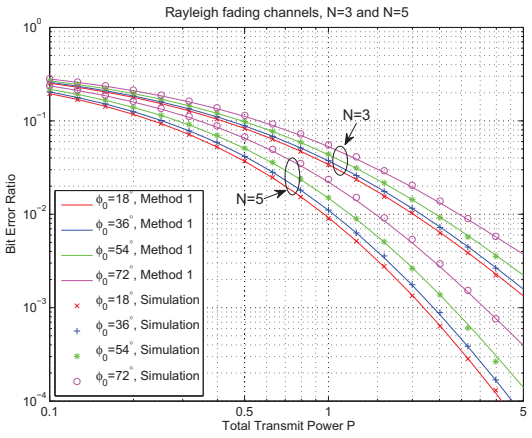



Fig. 2. Comparison of analytical results based on method 1 with simulation results of BER versus total transmit power with $N = 3, 5$ distributed sensor nodes, phase errors constrained within the range $\phi_0 = 18°, 36°, 54°, 72°$ relative to total transmit power $P = 1$.

Fig. 3 shows the comparison of the simulation results with the analytical results based on method 2 for large numbers of nodes $N = 40, 100$ for the same phase error ranges $\phi_0 = 18°, 36°, 54°$ and $72°$. As we see, for both $N = 40$ and $N = 100$ the simulation results and the analytical results show excellent agreement with each other. Method 1 still provides a good prediction for large $N$. However, with large $N$, method 1 has a high computational complexity, thus we only present the results based on method 2 in Fig. 3. From Fig. 3, we can draw the same conclusions about the effects of the number of nodes and the phase errors as from Fig. 2.

Fig. 4 shows the BER versus the number of nodes $N$ to analyze the accuracy of method 1 and method 2 when increasing $N$. In order to keep the received SNR approximately constant when increasing $N$, the total transmit power in Fig. 4 is set to be inversely proportional to $N$, which is different to the simulations in previous figures. It can be seen here that there is a gap between the two curves of method 1 and method 2 for small $N$, where method 1 provides a much more accurate prediction. Method 2 achieves progressively more accuracy as $N$ increases. This is because method 2 is based on the CLT

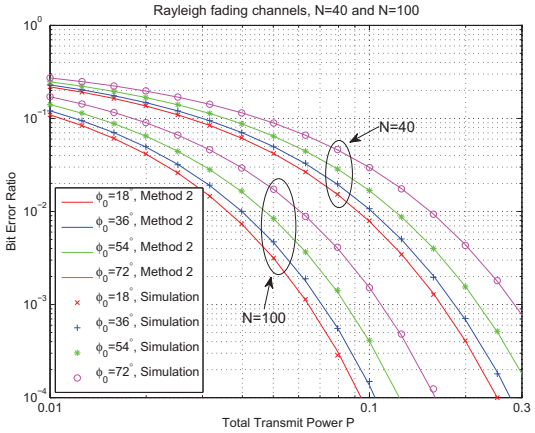



Fig. 3. Comparison of analytical results based on method 2 with simulation results of BER versus total transmit power with $N = 40, 100$, and $\phi_0 = 18°, 36°, 54°, 72°$.

and thus is not appropriate for small $N$. The solution given by method 1 takes the form of a single dimensional integral solved in our simulations by the Hermite integration method while method 2 is a much simpler and more computationally efficient approach. Therefore, it is preferable to use method 1 only for a small number of nodes and use method 2 for a large number of nodes.

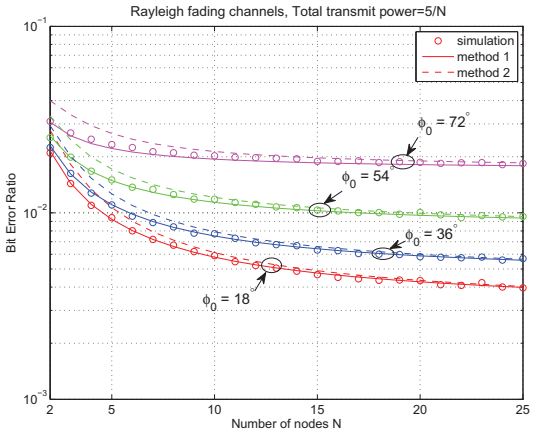



Fig. 4. Comparison of analytical results based on method 1 and method 2 with simulation results of BER versus number of nodes with $\phi_0 = 18°, 36°, 54°, 72°$, and total transmit power $P = \frac{5}{N}$.

## VI. CONCLUSIONS

We have derived BER expressions for BPSK with equal power transmission for distributed beamforming with phase errors. The simulation results show excellent agreement with analytical results. We analyzed the model from different approaches to approximate the distribution of the equivalent channel gain. It is suggested to use method 1, presented in Section III, to predict BER for a small number of nodes (e.g. $N \leqslant 10$) and use method 2, presented in Section IV, for a large number of nodes (e.g. $N \geqslant 20$). We propose using method 2 here predominantly due to its reduced computational load for large $N$. The system performance has been analyzed for different numbers of nodes and different phase error ranges.

## APPENDIX A
## THE FACTOR $\eta$ AND THE RESIDUAL VARIANCE $\sigma_d^2$

We derive the factor $\eta$ defined in (7) and the residual variance, $\sigma_d^2$, defined in (8) in Section III. The expression of $\eta^2$ in (7) can then be written as:

$$\eta^2 = \frac{\mathscr{A}}{\mathscr{A}(\phi_0 = 0)} = \frac{1 + \frac{\pi}{4}(N-1)\left(\frac{\sin\phi_0}{\phi_0}\right)^2}{1 + \frac{\pi}{4}(N-1)}, \quad (19)$$

where the expression of $\mathscr{A}$ is derived in Appendix B.

We rewrite (8) as follows:

$$\sigma_d^2 = \frac{P}{N}\left(\mathrm{E}\Big[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|^2\Big] - \left(\mathrm{E}\Big[\Big|\sum_{i=1}^{N}|h_i|e^{j\phi_i}\Big|\Big]\right)^2 - \eta^2\mathrm{E}\Big[\Big(\sum_{i=1}^{N}|h_i|\Big)^2\Big] + \eta^2\left(\mathrm{E}\Big[\sum_{i=1}^{N}|h_i|\Big]\right)^2\right). \quad (20)$$

Substituting (7) into (20), it yields:

$$\sigma_d^2 = \frac{P}{N}\left(\eta^2\left(N \cdot \mathrm{E}\Big[|h_i|\Big]\right)^2 - \left(\mathrm{E}\Big[|H|\Big]\right)^2\right). \quad (21)$$

The second moment and the fourth moment of $|H|$ in terms of $N$ and $\phi_0$ are derived in Appendix B. However, the pdf of $|H|$ is unknown and the first moment of $|H|$ is hard to compute. Instead, we use the Nakagami $m$-distribution [15] to give an approximate expression for the first moment. One of the characteristics of the Nakagami $m$-distribution is that it has great flexibility and can approximate many other distributions modeling fading environments. The $\beta$th moment of Nakagami $m$-distributed $|H|$ is given by [8]:

$$\mathrm{E}\Big[|H|^\beta\Big] = \frac{\Gamma(m + \frac{1}{2}\beta)}{\Gamma(m)}\left(\frac{\mathrm{E}[|H|^2]}{m}\right)^{\frac{\beta}{2}}. \quad (22)$$

By taking (24), (25) into (22), the first moment of $|H|$ can be easily obtained as:

$$\mathrm{E}\Big[|H|\Big] = \frac{\Gamma(m + \frac{1}{2})}{\Gamma(m)}\left(\frac{\mathscr{A}}{m}\right)^{\frac{1}{2}}, \quad (23)$$

where $m = \frac{\mathscr{A}^2}{\mathscr{B} - \mathscr{A}^2}$ and $\mathscr{A}$, $\mathscr{B}$ are given in (24), (25). By substituting (23) into (21), one can obtain the final expression of the residual variance $\sigma_d^2$ in terms of $N$ and $\phi_0$.

$$
\begin{aligned}
\mathscr{B} &= \mathrm{E}\Big[\sum_{i=1}^{N}|h_i|e^{j\phi_i}\cdot\sum_{k=1}^{N}|h_k|e^{j\phi_k}\cdot\sum_{l=1}^{N}|h_l|e^{-j\phi_l}\cdot\sum_{m=1}^{N}|h_m|e^{-j\phi_m}\Big] \\
&= \underbrace{N\cdot\mathrm{E}\big[|h_i|^4\big]}_{i=k=l=m} + \underbrace{4N(N-1)\cdot\mathrm{E}\big[|h_i|^3|h_m|e^{j(\phi_i-\phi_m)}\big]}_{\substack{i=k=l\neq m\\ i=k=m\neq l\\ i=l=m\neq k\\ k=l=m\neq i}} + \underbrace{N(N-1)\cdot\mathrm{E}\big[|h_i|^2|h_l|^2e^{j(2\phi_i-2\phi_l)}\big]}_{(i=k)\neq(l=m)} \\
&\quad + \underbrace{2N(N-1)\cdot\mathrm{E}\big[|h_i|^2|h_m|^2\big]}_{\substack{(i=l)\neq(k=m)\\(i=m)\neq(k=l)}} + \underbrace{2N(N-1)(N-2)\cdot\mathrm{E}\big[|h_i|^2|h_m||h_l|e^{j(2\phi_i-\phi_m-\phi_l)}\big]}_{\substack{(i=k)\neq l\neq m\\(l=m)\neq i\neq k}} \\
&\quad + \underbrace{4N(N-1)(N-2)\cdot\mathrm{E}\big[|h_i|^2|h_k||h_m|e^{j(\phi_k-\phi_m)}\big]}_{\substack{(i=l)\neq k\neq m\\(i=m)\neq k\neq l\\(k=l)\neq i\neq m\\(k=m)\neq i\neq l}} + \underbrace{N(N-1)(N-2)(N-3)\cdot\mathrm{E}\big[|h_i||h_k||h_l||h_m|e^{j(\phi_i+\phi_k-\phi_l-\phi_m)}\big]}_{i\neq k\neq l\neq m} \\
&= N\cdot\mathrm{E}\big[|h_i|^4\big] + 4N(N-1)\cdot\mathrm{E}\big[|h_i|^3\big]\cdot\mathrm{E}\big[|h_i|\big]\left(\frac{\sin\phi_0}{\phi_0}\right)^2 + N(N-1)\Big(\mathrm{E}\big[|h_i|^2\big]\Big)^2\left(\frac{\sin 2\phi_0}{2\phi_0}\right)^2 \\
&\quad + 2N(N-1)\Big(\mathrm{E}\big[|h_i|^2\big]\Big)^2 + 2N(N-1)(N-2)\cdot\mathrm{E}\big[|h_i|^2\big]\Big(\mathrm{E}\big[|h_i|\big]\Big)^2\frac{\sin 2\phi_0}{2\phi_0}\left(\frac{\sin\phi_0}{\phi_0}\right)^2 \\
&\quad + 4N(N-1)(N-2)\cdot\mathrm{E}\big[|h_i|^2\big]\Big(\mathrm{E}\big[|h_i|\big]\Big)^2\left(\frac{\sin\phi_0}{\phi_0}\right)^2 + N(N-1)(N-2)(N-3)\Big(\mathrm{E}\big[|h_i|\big]\Big)^4\left(\frac{\sin\phi_0}{\phi_0}\right)^4. \qquad (25)
\end{aligned}
$$

## APPENDIX B
## THE SECOND AND THE FOURTH MOMENT OF $\big|H\big|$

We derive the second and the fourth moment of $|H|$ used in (15) in terms of $N$ and $\phi_0$ based on the assumption that both $h_i$ and $\phi_i$ are independent i.i.d. variables, where $h_i \sim CN(0, 2\sigma_c^2)$ and $\phi_i \sim (-\phi_0, \phi_0)$.

The second moment of $\big|H\big|$ in (15) is expressed as:

$$
\begin{aligned}
\mathscr{A} &= \mathrm{E}\Big[\sum_{i=1}^{N}|h_i|e^{j\phi_i}\cdot\sum_{l=1}^{N}|h_l|e^{-j\phi_l}\Big] \\
&= \underbrace{N\cdot\mathrm{E}\big[|h_i|^2\big]}_{i=l} + \underbrace{N(N-1)\cdot\mathrm{E}\big[|h_i||h_l|e^{j(\phi_i-\phi_l)}\big]}_{i\neq l} \\
&= N\cdot\mathrm{E}\big[|h_i|^2\big] + N(N-1)\Big(\mathrm{E}\big[|h_i|\big]\Big)^2\left(\frac{\sin\phi_0}{\phi_0}\right)^2 \qquad (24)
\end{aligned}
$$

The fourth moment of $\big|H\big|$ in (15) is expressed as (25). Since $h_i \sim CN(0, 2\sigma_c^2)$, the moments of $|h_i|$ are given by $\mathrm{E}\big[|h_i|^\alpha\big] = (2\sigma_c^2)^{\frac{\alpha}{2}}\Gamma(1+\frac{\alpha}{2})$ [8]. In particular, when $h_i \sim CN(0,1)$, $\mathscr{A}$ and $\mathscr{B}$ become:

$$
\mathscr{A} = N + \frac{\pi}{4}N(N-1)\left(\frac{\sin\phi_0}{\phi_0}\right)^2, \qquad (26)
$$

$$
\begin{aligned}
\mathscr{B} &= 2N^2 + \pi N(N-1)(N-\frac{1}{2})\left(\frac{\sin\phi_0}{\phi_0}\right)^2 \\
&\quad + N(N-1)\left(\frac{\sin 2\phi_0}{2\phi_0}\right)^2 \\
&\quad + \frac{\pi}{2}N(N-1)(N-2)\frac{\sin 2\phi_0}{2\phi_0}\left(\frac{\sin\phi_0}{\phi_0}\right)^2 \\
&\quad + \frac{\pi^2}{16}N(N-1)(N-2)(N-3)\left(\frac{\sin\phi_0}{\phi_0}\right)^4 \qquad (27)
\end{aligned}
$$

## ACKNOWLEDGMENT

Shuo Song would like to thank China Scholarship Council/University of Edinburgh Joint Scholarship Program for supporting his PhD studies.

## REFERENCES

[1] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1754–1763, 2007.

[2] Z. Ding, W. H. Chin, and K. K. Leung, "Distributed beamforming and power allocation for cooperative networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1817–1822, 2008.

[3] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4110–4124, 2005.

[4] S. Song, J. S. Thompson, P. J. Chung, and P. M. Grant, "Probability of error for BPSK modulation in distributed beamforming with phase errors," in *Proc. International ITG Workshop on Smart Antennas*, Berlin, Feb. 2009, pp. 149–154.

[5] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Distributed transmit beamforming using feedback control," *IEEE Trans. Inf. Theory*, in review.

[6] D. R. Brown and H. V. Poor, "Time-slotted round-trip carrier synchronization for distributed beamforming," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5630–5643, 2008.

[7] E. Koyuncu, Y. Jing, and H. Jafarkhani, "Distributed beamforming in wireless relay networks with quantized feedback," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 8, pp. 1429–1439, 2008.

[8] J. G. Proakis, *Digital Communications*. 4th ed. New York: McGraw-Hill, 2001.

[9] Q. T. Zhang, "A simple approach to probability of error for equal gain combiners over Rayleigh channels," *IEEE Trans. Veh. Technol.*, vol. 48, no. 4, pp. 1151–1154, 1999.

[10] L. J. Slater, *Confluent Hypergeometric Functions*. Cambridge University Press, 1960.

[11] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1965.

[12] W. Lindsey, "Error probabilities for Rician fading multichannel reception of binary and $N$-ary signals," *IEEE Trans. Inf. Theory*, vol. 10, no. 4, pp. 339–350, 1964.

[13] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*. Cambridge University Press, 1992.
[14] J. S. Thompson and A. Smokvarski, "Bit error ratio performance of a receiver diversity scheme with channel estimation," *IET Commun.*, vol. 1, no. 1, pp. 92–100, 2007.
[15] M. Nakagami, "The $m$-distribution–A general formula of intensity distribution of rapid fading," *Statistical Methods in Radio Wave Propagation*, W. C. Hofman, Ed., New York: Pergamon, 1960.

# One-bit Feedback Algorithm with Decreasing Step Size for Distributed Beamforming

Shuo Song, John S. Thompson
Institute for Digital Communications, Joint Research Institute for Signal & Image Processing
School of Engineering, University of Edinburgh

***Abstract*—An iterative algorithm with a decreasing step size is proposed for distributed transmit beamforming to achieve carrier phase alignment at the receiver. The transmitters apply random perturbations on their phase offsets and adjust them based on one bit feedback from the receiver in each iteration. The perturbation step size becomes smaller when the phase angles at the receiver get closer to coherence. The step size is decreased by a specified ratio every time the number of successive failed perturbations surpasses a certain threshold. The proposed algorithm has an improvement in the convergence speed of phase alignment compared to the original iterative algorithm in the literature which has a fixed step size during the convergence process.**

## I. INTRODUCTION

Recently there has been great interest in applying transmit beamforming techniques into wireless sensor networks [1], [2], [3]. Since sensor nodes are working collaboratively in a distributed manner to perform beamforming transmission, this technique is called distributed beamforming [4] or collaborative beamforming [5] in the literature. The motivation of applying distributed beamforming is to reduce the energy requirement for each sensor node in signal transmission and extend the communication range to a far field receiver. Unlike conventional beamforming, realizing distributed beamforming faces a set of new challenges in several aspects, such as information sharing, frequency synchronization and phase alignment at the receiver, among which the most crucial part in practical implementation is achieving phase alignment [3].

Shuo Song thanks China Scholarship Council/University of Edinburgh Joint Scholarship Program for supporting his PhD studies.

We acknowledge the support of the Scottish Funding Council for the Joint Research Institute with the Heriot-Watt University which is a part of the Edinburgh Research Partnership.

A simple one-bit feedback iterative algorithm, a promising way to achieve carrier phase alignment at the receiver, was first proposed in R. Mudumbai et al.'s work [6]. This training process to achieve phase alignment at the receiver is performed by each transmitter introducing a random perturbation on its phase offset in each time slot. If the random perturbations introduced by all transmitters result in a bigger beamforming gain, they will be adopted by the transmitters; otherwise, they will be discarded. A detailed analysis of this algorithm including its benefits was presented in [7]. The validity of this type of one-bit feedback algorithm was verified by laboratory experiments presented in [8], where the expected performance results were obtained. Later, this algorithm was developed to account for carrier frequency errors among transmitters in [9]. Also, an improved algorithm which can yield a faster convergence speed by making use of both successful and failed perturbation results was presented in [10].

In this paper, we propose a new one-bit feedback algorithm which has a decreasing step size in the convergence process. The new algorithm still requires only one-bit feedback in each iteration, and results in a faster convergence speed to achieve carrier phase alignment at the receiver compared to the original algorithm presented in [7]. Simulation results show that the new algorithm has the potential to improve the convergence speed with a wide range of parameter selections. The rest of the paper is organized as follows. Section II describes the system model. Section III briefly reviews the original algorithm presented in [7]. In Section IV we describe the new one-bit feedback algorithm with a decreasing step size. Section V then presents some simulation results confirming the superior

performance of the new algorithm over the original algorithm and Section VI gives conclusions.

## II. SYSTEM MODEL

We consider a wireless communication system composed of $N$ transmitters collaboratively beamforming a narrowband message signal $m[n]$ to a distant receiver. This is performed in a distributed manner by each transmitter modulating $m[n]$ at the same carrier frequency and adjusting its phase offset to achieve phase alignment at the receiver, as illustrated in Fig. 1.

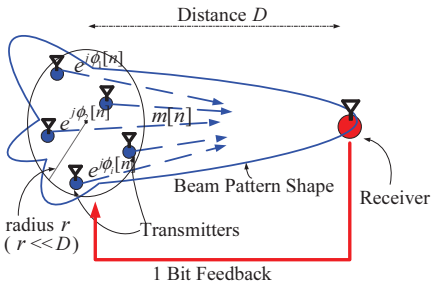



Fig. 1. System model for distributed transmit beamforming.

In order to compare the new algorithm with the original algorithm easily and fairly, the assumptions of our system model made in this paper are all the same with the assumptions in [7]. We repeat some key assumptions below. For more details, please see the list of assumptions in [7]. The channel from each transmitter to the receiver, $h_i$, is assumed to be static during the phase synchronization process. For simplicity, $|h_i| = 1$. All transmitters are frequency-synchronized so that they only need to adjust their phase offsets to achieve phase alignment at the receiver. The local carrier of each transmitter $i$ has an unknown phase offset $\gamma_i$ relative to the receiver's phase reference. All values of $\gamma_i$ are assumed to be uniformly distributed over $[0, 2\pi)$. All transmitters transmit with equal power $P$. As both algorithms considered here put emphasis on the phase synchronization process and the effect of phase difference on the beamforming gain, we set $P = 1$. The phase of the received signal at the receiver from transmitter $i$ in time slot $n$ can be expressed as:

$$\Phi_i[n] = \gamma_i + \psi_i + \phi_i[n] \tag{1}$$

where $\psi_i$ is the channel phase response from transmitter $i$ to the receiver, which is assumed to be static during the convergence process, and again all values of $\psi_i$ are assumed to be uniformly distributed over $[0, 2\pi)$. Both the values of $\gamma_i$ and $\psi_i$ are unknown to the transmitters and the receiver. The scalar $\phi_i$ is the adaptive component implemented by transmitter $i$, which is set to be zero at the start for both algorithms. The ideal phase alignment of distributed beamforming is that there are no phase differences among the signals arriving at the receiver, i.e.:

$$\gamma_i + \psi_i + \phi_i[n] \equiv \gamma_k + \psi_k + \phi_k[n] \ (\mathrm{mod}\ 2\pi), \quad i \neq k \quad \forall i, k = 1, 2, ..., N. \tag{2}$$

The objective of the algorithm design is to let each transmitter adjust its adaptive component, $\phi_i$, based on the one-bit feedback information in each time slot to achieve nearly perfect phase alignment at the receiver as fast as possible.

The complex baseband model of the received signal at the receiver is given by:

$$r[n] = \sum_{i=1}^{N} e^{j\Phi_i[n]} m[n] + n[n] \tag{3}$$

where $n[n]$ is additive white Gaussian noise of zero mean and variance $\sigma_n^2$. The received signal strength (RSS), which determines the beamforming gain, in time slot $n$ is defined as:

$$R[n] = \left| \sum_{i=1}^{N} e^{j\Phi_i[n]} \right| \tag{4}$$

We assume that the noise power at the receiver is fairly small compared to the signal power at the receiver. The RSS in each time slot, $R[n]$, can be measured accurately by averaging the received signal over a certain time interval.

## III. ORIGINAL ONE-BIT FEEDBACK ALGORITHM

The original one-bit feedback algorithm for distributed beamforming introduced in [7] can be briefly summarized as follows.

1) At time slot $n$, each transmitter applies a random perturbation, $\delta_i[n]$, to its best known carrier phase, $\theta_i[n]$, for beamforming, where

$i$ denotes the $i$th transmitter. (Without loss of generality, we set the distribution of $\delta_i[n]$ as $\delta_i[n] = \pm\delta_0$ in the simulations in Section V.)

2) All transmitters use their new adaptive components, $\phi_i[n] = \theta_i[n] + \delta_i[n]$, to perform transmit beamforming.
3) The receiver measures the corresponding RSS, $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$, and compares it with the best RSS in memory. The receiver updates the best RSS in memory and feeds back (error free) one-bit of information to all transmitters conveying whether the RSS has been improved or not.
4) If the RSS has been improved, all transmitters adopt their perturbed phases and update their best known phases to be $\theta_i[n+1] = \phi_i[n] = \theta_i[n] + \delta_i[n]$ for the next time slot $(n+1)$. Otherwise, all transmitters discard the perturbed phases and keep the best known phases as $\theta_i[n+1] = \theta_i[n]$ for the next time slot $(n+1)$. The algorithm then repeats these four steps.

The original algorithm can be viewed as a random search process in which each transmitter is trying to adjust its phase correctly based on the feedback information. The original algorithm has the ability to achieve nearly perfect phase alignment at the receiver after a lot of iterations. Its asymptotic convergence properties and convergence speed are well proved and analyzed in [7]. For more details of the original algorithm, please see [7].

Communication remains the most energy-consuming operation for sensor nodes compared to others, such as sensing, data processing, etc [11]. Since in most application scenarios sensor nodes are supplied by energy-limited batteries which can not be easily replaced, the phase alignment process for beamforming with a faster convergence speed is desired in practice. The faster the algorithm converges, the less energy it consumes. Recalling the objective of this kind of algorithm design is to achieve phase alignment as fast as possible, the original algorithm leaves us some space for improvement in the convergence speed.

## IV. NEW ONE-BIT FEEDBACK ALGORITHM WITH DECREASING STEP SIZE

Intuitively, the original algorithm could have a bigger perturbation step size at initial stages to converge faster and requires a smaller step size when the beamforming gain gets closer to its optimum value. Therefore, we propose a simple but effective algorithm with a decreasing perturbation step size to improve the convergence speed of phase alignment, which leads to a slight modification to the original algorithm. The new algorithm still requires only one-bit feedback in each time slot and can be easily implemented in practice. The new one-bit feedback algorithm with a decreasing step size is described as follows.

1) At time slot $n$, each transmitter applies a random perturbation, $\delta_i[n] = \pm\delta_0$, to its best known carrier phase, $\theta_i[n]$, for beamforming, where $i$ denotes the $i$th transmitter.
2) All transmitters use their new adaptive components, $\phi_i[n] = \theta_i[n] + \delta_i[n]$, to perform transmit beamforming.
3) The receiver measures the corresponding RSS, $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$, and compares it with the best RSS in memory. The receiver updates the best RSS in memory and feeds back (error free) one-bit of information to all transmitters conveying whether the RSS has been improved or not.
4) If the RSS has been improved, all transmitters adopt their perturbed phases and update their best known phases to be $\theta_i[n+1] = \phi_i[n] = \theta_i[n] + \delta_i[n]$ for the next time slot $(n+1)$. Otherwise, all transmitters discard the perturbed phases and keep the best known phases as before, $\theta_i[n+1] = \theta_i[n]$, for the next time slot $(n+1)$. Meanwhile, the transmitters record the number of successive failed perturbations with a counting variable $C_f$. If it is a positive feedback indicating a successful perturbation, $C_f$ will be cleared to zero. Otherwise, the value of $C_f$ will be increased by 1 until it surpasses a certain threshold $C_T$. When $C_f \geq C_T$, $C_f$ is cleared to zero and all transmitters adopt a new perturbation step size $\delta_0 = \delta_0 \cdot R_D$ $(0 < R_D < 1)$, where $R_D$ is the decreasing ratio of step size.
The algorithm then repeats these four steps.

The new algorithm adjusts the perturbation step size based on only one-bit feedback in each time slot. It makes use of the information contained

within the experience of successive failed perturbations which span several time slots. When the phase differences at the receiver are large, a bigger step size can accelerate the convergence speed. However, when the phase differences at the receiver become smaller, a bigger step size will decelerate the convergence speed or even cease the convergence process, and a smaller step size is required. Fig. 2 shows an example of two transmitters.

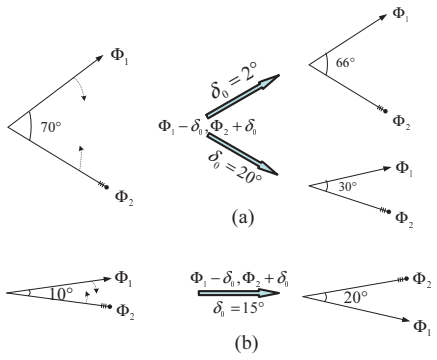



Fig. 2. Phase perturbation results in the case of two transmitters. (a) The phase difference at the receiver is large (This corresponds to the initial stages of the convergence process). (b) The phase difference at the receiver is smaller compared to the perturbation step size.

## V. RESULTS AND DISCUSSIONS

In this section, we present some Monte Carlo simulation results in accordance with our previous assumptions. We investigate the performance of the new algorithm as a function of two parameters: the threshold for successive failed perturbations $C_T$, and the decreasing ratio of the perturbation step size $R_D$. We then compare the new algorithm with the original algorithm in terms of the convergence time required to achieve a certain beamforming gain. We set the number of nodes as $N = 100$, the initial perturbation step size as $\delta_0 = \frac{\pi}{4}$. The simulation results for every point in the following figures are averaged over 800 instances.

Fig. 3 shows the average number of time slots required to achieve an RSS of 90 with different values of $C_T$ and $R_D$ for the new algorithm. There exists an optimum value for the parameter selection which can result in the minimum number of time slots. From the simulation results we see that the minimum number of time slots required to achieve an RSS of 90 for the new algorithm is 688 time slots, which is obtained with $C_T = 11$ and $R_D = 0.75$.

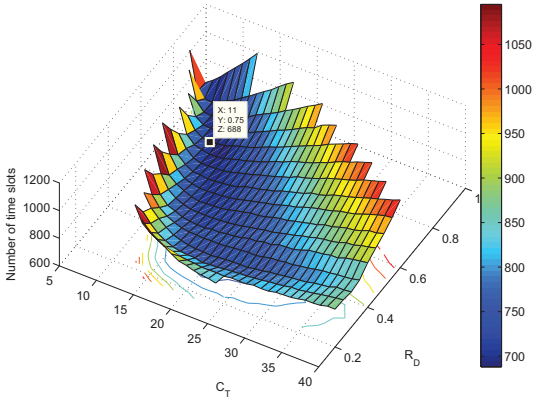



Fig. 3. Simulation results for the new algorithm showing the average number of time slots required to achieve an RSS of 90 with different values of $C_T$ and $R_D$, where $C_T$ is the threshold for successive failed perturbations and $R_D$ is the decreasing ratio of the perturbation step size.

In [10], we have studied the performance of the original algorithm in a similar way. It shows that the minimum number of time slots required to achieve an RSS of 90 with $N = 100$ by the original algorithm is 791 time slots. Fig. 4 shows the contour plot of Fig. 3. It shows that the new algorithm can achieve an RSS of 90 within 790 time slots, or in other words achieve as good performance as the original algorithm, with a wide range of parameter selections. This shows the robustness of this algorithm to small mismatches in parameter settings.

The new algorithm with a decreasing step size may be modified to suit a time-varying channel environment. However, several concerns have to be considered in this modification. In the case of time-varying channels, successive failed perturbations may imply that a smaller perturbation step size is required as in the case of static channels. But it may also be caused by the shift of channel phase responses which makes the new RSSs unable to surpass the best RSS in memory. Therefore, we have to revise the feedback mechanism at the receiver side. Fig. 5 shows that the whole system can be

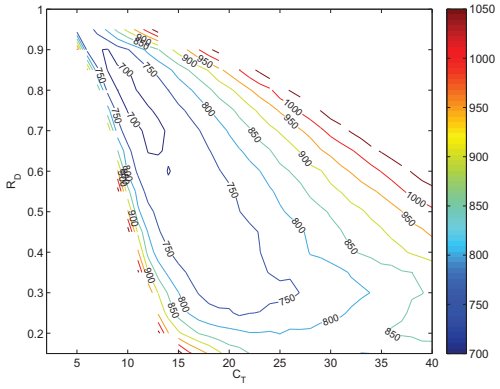



Fig. 4. Contour plot of the average number of time slots required to achieve an RSS of 90 with different values of $C_T$ and $R_D$ for the new algorithm.

divided into three blocks for design purpose.

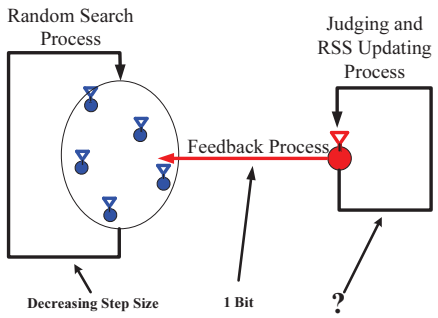



Fig. 5. Blocks of the feedback system for design purpose.

Since communication, either transmitting or receiving, remains the most energy-consuming operation for sensor nodes, it is important to have only one bit feedback and reduce the iterations in the convergence process. Random search techniques targeting at the sensor node side have been designed to reduce iterations. Intelligent algorithms with an advanced judging rule at the receiver side may be designed to better cooperate with the process at the sensor node side. For example, the receiver can exploit more information contained within the values of RSS obtained in successive time slots and employ an advanced rule of updating the best RSS. Advanced feedback mechanisms at the receiver side are left for future work.

## VI. CONCLUSIONS

We have proposed a new algorithm for distributed transmit beamforming to achieve carrier phase alignment at the receiver. The new algorithm has a faster convergence speed of phase alignment compared to the original algorithm in the literature. This improvement on the convergence speed is obtained by applying a decreasing perturbation step size rather than having a fixed one during the convergence process. The new algorithm employs a scheme to adjust the perturbation step size based on the number of successive failed perturbations. The new algorithm still requires only one-bit feedback in each time slot. Therefore, it keeps all the benefits (listed in [7]) of the original algorithm and can be easily implemented in practice.

## REFERENCES

[1] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4110–4124, 2005.

[2] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1754–1763, 2007.

[3] R. Mudumbai, D. R. Brown, U. Madhow, and H. V. Poor, "Distributed transmit beamforming: challenges and recent progress," *IEEE Commun. Mag.*, vol. 47, no. 2, pp. 102–110, 2009.

[4] G. Barriac, R. Mudumbai, and U. Madhow, "Distributed beamforming for information transfer in sensor networks," in *Proc. Third International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, Apr. 2004, pp. 81–88.

[5] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming in ad hoc networks," in *Proc. IEEE Information Theory Workshop*, San Antonio, TX, Oct. 2004, pp. 396–401.

[6] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Scalable feedback control for distributed beamforming in sensor networks," in *Proc. International Symposium on Information Theory*, Adelaide, SA, Sept. 2005, pp. 137–141.

[7] ——, "Distributed transmit beamforming using feedback control," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 411–426, 2010.

[8] R. Mudumbai, B. Wild, U. Madhow, and K. Ramchandran, "Distributed beamforming using 1 bit feedback: from concept to realization," in *Proc. 44th Allerton Conf. Commun., Control, Comp.*, Monticello, IL, Sept. 2006, pp. 1020–1027.

[9] M. Seo, M. Rodwell, and U. Madhow, "A feedback-based distributed phased array technique and its application to 60-GHz wireless sensor network," in *Proc. IEEE MTT-S International Microwave Symposium*, Atlanta, GA, June 2008, pp. 683–686.

[10] S. Song, J. S. Thompson, P. J. Chung, and P. M. Grant, "Improving the one-bit feedback algorithm for distributed beamforming," in *Proc. IEEE Wireless Communications and Networking Conference*, Australia, April 2010, pp. 1–6.

[11] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.

# Improving the One-bit Feedback Algorithm for Distributed Beamforming

Shuo Song, John S. Thompson, Pei-Jung Chung, Peter M. Grant
Institute for Digital Communications
Joint Research Institute for Signal & Image Processing
School of Engineering and electronics
University of Edinburgh, Edinburgh, EH9 3JL, UK
Email: {s.song, john.thompson, p.chung, peter.grant}@ed.ac.uk

***Abstract*—In this paper an improved iterative algorithm is proposed for distributed transmit beamforming to achieve carrier phase alignment at the receiver. The transmitters adjust their phase offsets based on one-bit feedback from the receiver in each time slot. The proposed algorithm has an improvement in the convergence speed of phase alignment compared to a previously proposed algorithm in the literature by exploiting one-bit feedback information more efficiently. Simulation results show that the improved algorithm on average has a 20% faster convergence speed. The minimum number of time slots required to achieve specified beamforming gains and the corresponding perturbation steps are obtained by Monte Carlo simulations.**

## I. INTRODUCTION

Recently there have been several papers discussing the practical problems of realizing distributed transmit beamforming [1] and describing its potential benefits in applications [2], [3]. Distributed transmit beamforming can provide high signal-to-noise ratio gains, extend the communication range, or reduce the energy requirement for each transmitter in signal transmission. The most crucial part of realizing distributed transmit beamforming is carrier frequency and phase synchronization among all the transmitters to ensure that the signals can be added coherently at the receiver [1]. A *master-slave* open-loop scheme was proposed in [4] to tackle this problem. Another open-loop scheme to solve this problem called the *round-trip* scheme was described in [5]. In [6], [7], the authors first present a simple *one-bit feedback* algorithm for distributed beamforming which does not need channel state information and can achieve nearly perfect carrier phase alignment at the receiver after a large number of iterations. This training process to achieve phase alignment at the receiver is performed by each transmitter introducing a random perturbation on its phase offset in each time slot. If the random perturbations result in a bigger beamforming gain, they will be adopted by the transmitters; otherwise, they will be discarded. Later, this algorithm was developed to account for carrier frequency errors among transmitters in [8]. The validity of this type of one-bit feedback algorithm was verified by laboratory experiments presented in [8] and [9], where the expected performance results were obtained. Its simplicity in implementation and scalability to large number of transmitters make it a promising way to realize distributed transmit beamforming in practical applications.

In this paper, we propose a new algorithm (namely the improved algorithm) based on the one-bit feedback algorithm described in [7] (namely the original algorithm) to achieve carrier phase alignment at the receiver in distributed transmit beamforming. The improved algorithm still requires only one-bit feedback from the receiver. It keeps all the benefits of the original algorithm, such as its simplicity and scalability, and requires no extra hardware. The improved algorithm is shown to have an advantage in the convergence speed. It requires fewer time slots, thus consumes less energy, to achieve a certain beamforming gain than the original algorithm by making use of the random perturbation obtained in each time slot more efficiently.

## II. SYSTEM MODEL

We consider a wireless communication system composed of $N$ transmitters collaboratively beamforming a narrowband message signal $m(t)$ to a distant receiver. This is performed in a distributed manner by each transmitter modulating $m(t)$ at the same carrier frequency and adjusting its phase offset to achieve phase alignment at the receiver, as illustrated in Fig. 1.

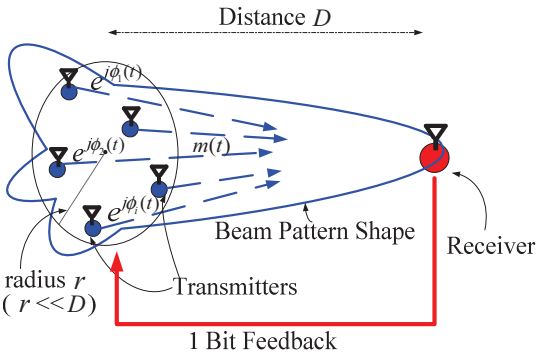



Fig. 1. System model for distributed transmit beamforming.

In order to compare the improved algorithm with the original algorithm easily and fairly, the assumptions made in this paper are all the same with the assumptions in [7]. We repeat some key assumptions below. For more details, please see the list of assumptions in [7]. The channel from each transmitter to the receiver, $h_i$, is assumed to be static during the phase synchronization process. For simplicity, $|h_i| = 1$. All transmitters are frequency-synchronized so that they only need to adjust their phase offsets to achieve phase alignment at the receiver. The local carrier of each transmitter $i$ has an unknown phase offset $\gamma_i$ relative to the receiver's phase reference. All values of $\gamma_i$ are assumed to be uniformly distributed over $[0, 2\pi)$. All transmitters transmit with equal power $P$. As both algorithms considered here put emphasis on the phase synchronization process and the effect of phase difference on the beamforming gain, we set $P = 1$. The phase of the received signal at the receiver from transmitter $i$ at time $t$ can be expressed as:

$$\Phi_i(t) = \gamma_i + \psi_i + \phi_i(t) \tag{1}$$

where $\psi_i$ is the channel phase response from transmitter $i$ to the receiver and again all values of $\psi_i$ are assumed to be uniformly distributed over $[0, 2\pi)$. Both the values of $\gamma_i$ and $\psi_i$ are unknown to the transmitters and the receiver. The scalar $\phi_i(t)$ is the adaptive component implemented by transmitter $i$, which is set to be zero at the start for both algorithms. The ideal phase alignment of distributed beamforming is that there are no phase differences among the signals arriving at the receiver, i.e.:

$$\gamma_i + \psi_i + \phi_i(t) \equiv \gamma_k + \psi_k + \phi_k(t) \pmod{2\pi}, \quad i \neq k \quad \forall i, k = 1, 2, ..., N. \tag{2}$$

The objective of the algorithm design is to let each transmitter adjust its value $\phi_i(t)$ based on the one-bit feedback information in each time slot to achieve nearly perfect phase alignment at the receiver as fast as possible.

The complex baseband model of the received signal at the receiver is given by:

$$r(t) = \sum_{i=1}^{N} e^{j\Phi_i(t)} m(t) + n(t) \tag{3}$$

where $n(t)$ is additive white Gaussian noise of zero mean and variance $\sigma_n^2$. The received signal strength (RSS), which determines the beamforming gain, at time $t$ is defined as:

$$R(t) = \left| \sum_{i=1}^{N} e^{j\Phi_i(t)} \right| \tag{4}$$

We assume that the noise power at the receiver is fairly small compared to the signal power at the receiver. The RSS in each time slot, $R[n]$, can be measured accurately by averaging the received signal, $r(t)$, over a certain time interval.

## III. ORIGINAL ONE-BIT FEEDBACK ALGORITHM

The original one-bit feedback algorithm for distributed beamforming introduced in [7] can be briefly summarized as follows.

1) At time slot $n$, each transmitter applies a random perturbation, $\delta_i[n]$, to its best known carrier phase, $\theta_i[n]$, for beamforming. There are two simple distributions for the perturbation step $\delta_i[n]$: the two valued distribution where $\delta_i[n] = \pm\delta_0$ and the uniform distribution where $\delta_i[n] \sim [-\delta_0, \delta_0]$, where $i$ denotes the $i$th transmitter.
2) All transmitters use their new adaptive phases, $\phi_i[n] = \theta_i[n] + \delta_i[n]$, to perform transmit beamforming.
3) The receiver measures the RSS, $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$, and compares it with the best RSS in memory. The receiver updates the best RSS in memory and feeds back (error free) one-bit of information to all transmitters conveying whether the RSS has been improved or not.
4) If the RSS has been improved, all transmitters adopt their perturbed phases and update their best known phases to be $\theta_i[n + 1] = \phi_i[n] = \theta_i[n] + \delta_i[n]$ for the next time slot $(n+1)$. Otherwise, all transmitters discard the perturbed phases and keep the best known phases as before, $\theta_i[n + 1] = \theta_i[n]$, for the next time slot $(n + 1)$. The algorithm then repeats these four steps.

The adaptive component $\phi_i[n]$ used for beamforming in time slot $n$ in the original algorithm is composed of two parts:

$$\phi_i[n] = \theta_i[n] + \delta_i[n] \tag{5}$$

where $\theta_i[n]$ represents the best known phase of transmitter $i$ in time slot $n$. The scalar $\delta_i[n]$ is the random component applied to the best known phase in time slot $n$.

The original algorithm in [7] can be mathematically expressed as:
At the transmitter side:

$$\theta_i[n + 1] = \begin{cases} \theta_i[n] + \delta_i[n] & R[n] > R_{best}[n] \\ \theta_i[n] & \text{otherwise} \end{cases} \tag{6}$$

At the receiver side:

$$R_{best}[n + 1] = \max(R_{best}[n], R[n]) \tag{7}$$

where $R_{best}[n]$ is the best RSS in memory, or in other words, the maximal RSS in the past $n - 1$ time slots. By inserting (5) into (1), the overall phase of the received signal at the receiver in time slot $(n + 1)$ can be expressed as:

$$\begin{aligned} \Phi_i[n + 1] &= \gamma_i + \psi_i + \phi_i[n + 1] \\ &= \gamma_i + \psi_i + \theta_i[n + 1] + \delta_i[n + 1] \end{aligned} \tag{8}$$

Given (6), when $R[n] > R_{best}[n]$, (8) becomes:

$$\Phi_i[n + 1] = \gamma_i + \psi_i + \theta_i[n] + \delta_i[n] + \delta_i[n + 1] \tag{9}$$

Otherwise, when $R[n] \leq R_{best}[n]$, (8) becomes:

$$\Phi_i[n + 1] = \gamma_i + \psi_i + \theta_i[n] + \delta_i[n + 1] \tag{10}$$

## IV. IMPROVED ONE-BIT FEEDBACK ALGORITHM

The original algorithm can be viewed as a random search process in which each transmitter is trying to adjust its phase correctly based on the feedback information. Since the original algorithm only adopts a phase perturbation when it yields a performance improvement and discards other "failed" perturbations, it only makes use of the feedback information which indicates performance improvement. However, failure can also be used to obtain future success. Making use of the information contained within the failed perturbations which led to performance degradation is expected to be helpful in improving the convergence speed of phase alignment. Hereby, we propose a new algorithm based on the original algorithm summarized as follows.

1) At time slot $n$, each transmitter applies a random perturbation, $\delta_i[n]$, to its best known carrier phase, $\theta_i[n]$, for beamforming. Meanwhile, each transmitter also adds an modifying factor, $\epsilon_i[n]$, to its best known carrier phase for beamforming. This modifying factor is introduced to add a minus $\delta_i[n-1]$ to the adaptive phase for beamforming in time slot $n$ if the random component in time slot $(n-1)$, $\delta_i[n-1]$, has led to a failed perturbation in time slot $(n-1)$. Otherwise, the value of $\epsilon_i[n]$ is set to be 0.
2) All transmitters use their new adaptive phases, $\phi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$, to perform transmit beamforming.
3) The receiver measures the RSS, $R[n] = \left|\sum_{i=1}^{N} e^{j\Phi_i[n]}\right|$, and compares it with the best RSS in memory. The receiver updates the best RSS in memory and feeds back (error free) one-bit of information to all transmitters conveying whether the RSS has been improved or not.
4) If the RSS has been improved, all transmitters adopt their perturbed phases and update their best known phases to be $\theta_i[n+1] = \phi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n]$ for the next time slot $(n+1)$. The modifying factor for the next time slot is set to be $\epsilon_i[n+1] = 0$. Otherwise, all transmitters discard the perturbed phases and keep the best known phases as before, $\theta_i[n+1] = \theta_i[n]$, for the next time slot $(n+1)$. The modifying factor for the next time slot is set to be $\epsilon_i[n+1] = -\delta_i[n]$.
   The algorithm then repeats these four steps.

The adaptive component $\phi_i[n]$ used for beamforming in time slot $n$ in the improved algorithm is composed of three parts:

$$\phi_i[n] = \theta_i[n] + \epsilon_i[n] + \delta_i[n] \tag{11}$$

where $\theta_i[n]$ represents the best known phase, $\epsilon_i[n]$ is the modifying factor and $\delta_i[n]$ is the random component.

The improved algorithm can be mathematically expressed as:

At the transmitter side:

$$\theta_i[n+1] = \begin{cases} \theta_i[n] + \epsilon_i[n] + \delta_i[n] & R[n] > R_{best}[n] \\ \theta_i[n] & \text{otherwise} \end{cases} \tag{12}$$

$$\epsilon_i[n+1] = \begin{cases} 0 & R[n] > R_{best}[n] \\ -\delta_i[n] & \text{otherwise} \end{cases} \tag{13}$$

At the receiver side:

$$R_{best}[n+1] = \max(R_{best}[n], R[n]) \tag{14}$$

By substituting (11) into (1), the overall phase of the received signal at the receiver in time slot $(n+1)$ can be expressed as:

$$\begin{aligned} \Phi_i[n+1] &= \gamma_i + \psi_i + \phi_i[n+1] \\ &= \gamma_i + \psi_i + \theta_i[n+1] + \epsilon_i[n+1] + \delta_i[n+1] \end{aligned} \tag{15}$$

Given (12) and (13), when $R[n] > R_{best}[n]$, (15) becomes:

$$\Phi_i[n+1] = \gamma_i + \psi_i + \theta_i[n] + \epsilon_i[n] + \delta_i[n] + 0 + \delta_i[n+1] \tag{16}$$

Otherwise, when $R[n] \le R_{best}[n]$, (15) becomes:

$$\Phi_i[n+1] = \gamma_i + \psi_i + \theta_i[n] - \delta_i[n] + \delta_i[n+1] \tag{17}$$

When the perturbation step $\delta_i$ is quite small compared to the phase differences at the receiver, a perturbation on the carrier phases would lead to either a reduction or an increment in phase differences at the receiver, thus yielding beamforming performance improvement or degradation. The basic idea behind the improved algorithm is that for a single transmitter in each time slot, if a positive perturbation on its carrier phase leads to performance degradation, usually, a negative perturbation on the same carrier phase will lead to performance improvement, and vice versa. Fig. 2 shows an example of two transmitters.

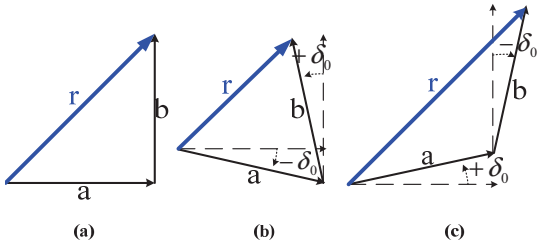



Fig. 2. Phase perturbation results in the case of two transmitters. If (b) a random perturbation leads to performance degradation, (c) an opposite perturbation will lead to performance improvement. Vector $a$ is the received signal from one transmitter, vector $b$ is the received signal from the other transmitter.

By comparing (16) with (9) we see that in both algorithms, when an adaptive component $\phi_i[n]$ leads to a bigger beamforming gain, it will be retained and be set as the best known phase for the next time slot, so $\theta_i[n+1] = \phi_i[n]$. In the next time slot $(n+1)$, a random perturbation, $\delta_i[n+1]$, will be applied to this best known phase, $\theta_i[n+1]$, and there is no further modification apart from the random perturbation on $\theta_i[n+1]$ for beamforming. By comparing (17) with (10) we see that in both algorithms, when an adaptive component $\phi_i[n]$ leads to a smaller beamforming gain, it will be discarded and the best known phase is kept unchanged for the next time slot, so $\theta_i[n+1] = \theta_i[n]$. In the next time slot $(n+1)$, the original

algorithm will perform a random perturbation again based on the same $\theta_i[n]$ while the improved algorithm will perform a random perturbation based on $\theta_i[n] - \delta_i[n]$, where $-\delta_i[n]$ is introduced by the modifying factor, $\epsilon_i[n]$. Consequently, both successful and failed perturbations in the improved algorithm contribute to the convergence speed.

One may ask why in the case of a failed perturbation in time slot $n$, why not update the best known phase to be $\theta_i[n+1] = \theta_i[n] - \delta_i[n]$ for time slot $(n+1)$ rather than introducing the modifying factor, $\epsilon_i[n+1]$? In that case the random perturbation would be based on $\theta_i[n] - \delta_i[n]$ in time slot $(n+1)$. This is because $\theta_i[n] - \delta_i[n]$ does not always result in a better performance than $\theta_i[n]$. If not, the update equation $\theta_i[n+1] = \theta_i[n] - \delta_i[n]$ may drift off the best phase for beamforming corresponding to the best RSS in memory.

The basic idea behind the improved algorithm is related to the signed algorithm proposed in [10], which also aims to make use of failed perturbations more efficiently. In [10], the authors have mathematically proved that making use of the failed perturbations can improve the convergence speed. However, the work in [10] mainly focuses on the convergence analysis of phase errors themselves rather than the resulting beamforming gain and does not provide the details on how to implement the algorithm. Rather than introducing a modifying factor, in the case of a failed perturbation, the signed algorithm directly adds the opposite perturbation to the best known phase, which is different from our improved algorithm. What is more, the update process of the best known phase for beamforming in [10] is based on the comparison between the RSS after perturbation and the RSS before perturbation in the same time slot. This implies that in the case of a failed perturbation, $\theta_i[n] + \delta_i[n]$, performed in time slot $n$, it requires the beamforming process and the measurement of RSS to be performed twice in time slot $(n+1)$, corresponding to the phases $\theta_i[n] - \delta_i[n]$ and $\theta_i[n] - \delta_i[n] + \delta_i[n+1]$. Or this operation may be counted as two time slots rather than one. This consumes more time and more energy. Moreover, the signed algorithm requires two bits feedback in each time slot. For details of the signed algorithm, please see [10]. A detailed comparison between our improved algorithm and the signed algorithm is ongoing work.

## V. RESULTS AND DISCUSSIONS

In this section, we present some Monte Carlo simulation results in accordance with our previous assumptions. We compare the improved algorithm with the original algorithm in terms of the convergence time required to achieve a certain beamforming gain. Between the two distributions for the perturbation step, $\delta_i$, given in [7], the original algorithm with $\delta_i[n] = \pm\delta_0$ converges faster at initial stages, while the original algorithm with $\delta_i[n] \sim [-\delta_0, \delta_0]$ results in a bigger RSS close to its optimum value after a lot of iterations. Since the following simulation results will reveal that the improved algorithm has an improvement in the convergence speed, we select the first kind of distribution, $\delta_i[n] = \pm\delta_0$, for the perturbation steps for both the original and the improved algorithm. In order to compare the two algorithms fairly and effectively, we use the same sequences of pseudo random values of $\gamma_i$ and $\psi_i$ for both algorithms and set $\phi_i[1] = 0$.

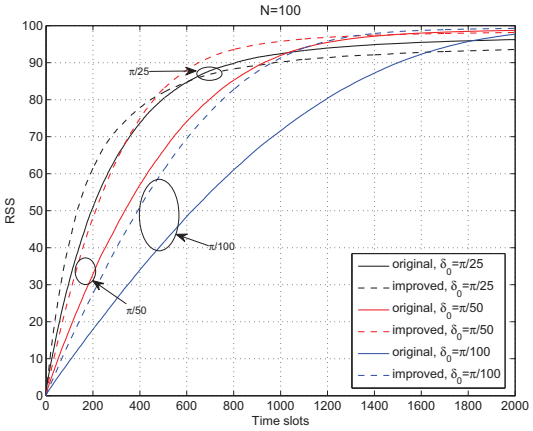



Fig. 3. Comparison of the original algorithm and the improved algorithm on the average RSS versus the number of time slots with $N = 100$ and $\delta_0 = \frac{\pi}{100}, \frac{\pi}{50}, \frac{\pi}{25}$.

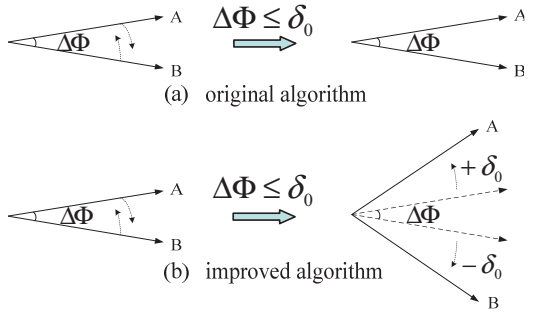



Fig. 4. Perturbation results in the case of two transmitters when $\Delta\Phi \le \delta_0$ ($\Delta\Phi$ denotes the phase difference at the receiver). (a) applying the original algorithm; (b) applying the improved algorithm.

Fig. 3 shows the comparison of the original algorithm and the improved algorithm using the average RSS versus the number of time slots up to 2000 with $N = 100$ and $\delta_0 = \frac{\pi}{100}, \frac{\pi}{50}, \frac{\pi}{25}$. The simulation results for every curve in Fig. 3 are averaged over $10^3$ instances. It shows that with the same value of $\delta_0$, the improved algorithm converges faster than the original algorithm at initial stages, which is consistent with our expectation in Section IV. However, it also shows that with the same value of $\delta_0$, the original algorithm results in a bigger RSS than the improved algorithm after a lot of iterations when the RSS gets closer to its optimum value. This is because the original algorithm performs better when the phase differences among the signals arriving at the receiver become on the same order as $\delta_0$. For instance, Fig. 4 shows

the case of two transmitters from which the received signals at the receiver has a phase difference $\Delta\Phi$ smaller than $\delta_0$. When the phase difference between the two signal vectors, $\Delta\Phi$, is no bigger than the perturbation step, $\delta_0$, there leaves no space for a reduction in the phase difference when the iterations evolve. In this situation, the original algorithm keeps the phase difference unchanged while the improved algorithm results in a bigger phase difference. Accordingly, the original algorithm performs better when the RSS gets closer to its optimum value.

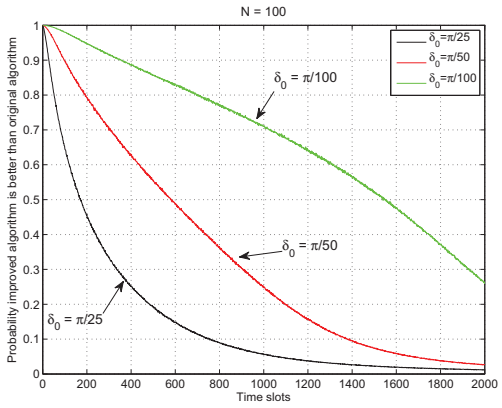



Fig. 5. Probability of improved algorithm leading to a bigger RSS than original algorithm versus the number of time slots.

Fig. 5 shows the probability of the improved algorithm performing better than the original algorithm versus the number of time slots up to 2000 with $N = 100$ and $\delta_0 = \frac{\pi}{100}, \frac{\pi}{50}, \frac{\pi}{25}$. The probability in time slot $n$ is calculated for $10^5$ instances, the number of instances that the improved algorithm leads to a bigger RSS than the original algorithm in time slot $n$ when a failed perturbation happened in time slot $(n-1)$. This is divided by the total number of instances that a failed perturbation happened in time slot $(n-1)$. From Fig. 5 we see that the probability decreases when the number of time slots increases and the probability with a bigger $\delta_0$ decreases faster than the case with a smaller $\delta_0$. These findings are consistent with our explanation above.

From Fig. 3 we see that with the same $\delta_0 = \frac{\pi}{100}$ the improved algorithm converges faster than the original algorithm, the original algorithm with $\delta_0 = \frac{\pi}{50}$ converges even much faster than both algorithms with $\delta_0 = \frac{\pi}{100}$. How can one compare the convergence speed of the two algorithms more quantitatively? Based on the average RSS versus the number of time slots, the number of time slots required to achieve an RSS of 90 with different values of $\delta_0$ are plotted in Fig. 6 for both algorithms. It shows that to achieve an RSS of 90, both the original algorithm and the improved algorithm have an optimum $\delta_0$ corresponding to the minimum number of time slots. From the simulation results we see that the minimum number of time slots required for the original algorithm is 791, while

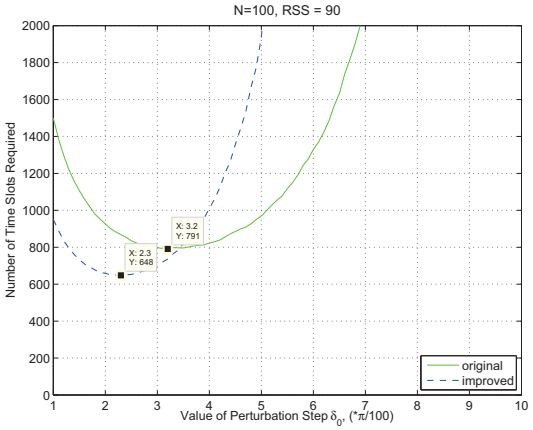



Fig. 6. Comparison of the original algorithm and the improved algorithm on the number of time slots required to achieve an RSS of 90 with different values of $\delta_0$.

the minimum number of time slots required for the improved algorithm is 648. This implies that the improved algorithm can converge faster than the original algorithm to achieve an RSS value of 90.

Fig. 7 shows the minimum number of time slots required to achieve different values of RSS for both algorithms and Fig. 8 shows the corresponding values of $\delta_0$ which result in the minimum number of time slots versus the value of RSS. If we denote $\delta_0 = \delta_1$ for the original algorithm, and $\delta_0 = \delta_2$ for the improved algorithm, the number of time slots $n_1$ used to achieve a certain value of RSS for the original algorithm is a function of $\delta_1$ and $R$: $n_1 = f(\delta_1, R)$. Similarly, for the improved algorithm the number of time slots $n_2 = g(\delta_2, R)$. From Fig. 6 and Fig. 7 we have: for any given $R$, there always exists a $\delta_2$ satisfying:

$$n_2 = g(\delta_2, R) < n_1 = f(\delta_1, R), \quad \forall \delta_1 \in [0, 2\pi) \tag{18}$$

It shows in Fig. 7 that the gap between the minimum number of time slots required by the original algorithm and the improved algorithm increases with the value of RSS. For the original algorithm, we define the convergence speed to achieve an RSS value of $R$ to be inversely proportional to the minimum number of time slots required, which is expressed as:

$$v_1(R) \propto \frac{1}{\hat{n}_1(R)} \tag{19}$$

where $\hat{n}_1(R) = \min(n_1 = f(\delta_1, R)), \forall \delta_1$ is the minimum number of time slots required to achieve $R$ by the original algorithm. The improvement in the convergence speed of the improved algorithm compared to the original algorithm can be expressed as:

$$\rho(R) = \frac{v_2(R) - v_1(R)}{v_1(R)} = \frac{\hat{n}_1(R) - \hat{n}_2(R)}{\hat{n}_2(R)} \tag{20}$$

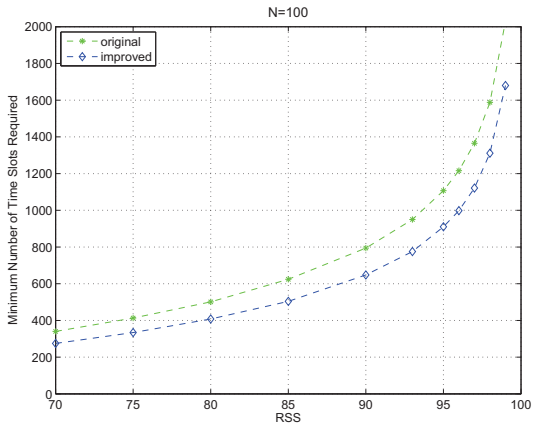



Fig. 7. Comparison of the original algorithm and the improved algorithm for the minimum number of time slots required to achieve different RSS values.

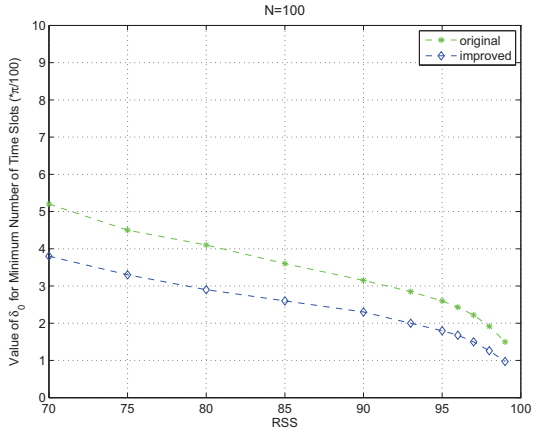



Fig. 8. Value of perturbation step $\delta_0$ which results in the minimum number of time slots to achieve different RSS values.

where $v_2(R)$ is the convergence speed for the improved algorithm and $\hat{n}_2(R)$ is the minimum number of time slots required by the improved algorithm. The improvement in the convergence speed to achieve different values of RSS as a percentage are given in Table I, where $\hat{n}_1(R)$ and $\hat{n}_2(R)$ are obtained from the results plotted in Fig. 7. It shows that to achieve a certain RSS between 70 and 99, the improved algorithm converges at least 20% faster than the original algorithm.

## VI. CONCLUSIONS

We have proposed a new algorithm for distributed transmit beamforming based on the original one-bit feedback algorithm presented in [7]. The improved algorithm yields a 20% faster convergence speed by making use of the one-bit feedback information more efficiently. It does not require any more information exchange or hardware support than the original algorithm. Also, it keeps all the benefits of the original algorithm, such as the simplicity and scalability. Simulation results confirm the potential of the improved algorithm in improving the convergence speed and show the minimum number of time slots required to achieve a certain beamforming gain and the corresponding value of perturbation step used. Obtaining a closed form expression for choosing the optimum value of perturbation step is the subject of ongoing work.

TABLE I
IMPROVEMENT IN CONVERGENCE SPEED TO ACHIEVE DIFFERENT RSS OF THE IMPROVED ALGORITHM COMPARED TO THE ORIGINAL ALGORITHM.

| $R$ | 70 | 75 | 80 | 85 | 90 | 93 |
|---|---|---|---|---|---|---|
| $\rho$ | 23.64% | 23.65% | 22.79% | 23.81% | 22.53% | 22.58% |
| $R$ | 95 | 96 | 97 | 98 | 99 | |
| $\rho$ | 21.65% | 21.84% | 21.77% | 21.13% | 20.77% | |

## ACKNOWLEDGMENT

Shuo Song would like to thank China Scholarship Council/University of Edinburgh Joint Scholarship Program for supporting his PhD studies. We acknowledge the support of the Scottish Funding Council for the Joint Research Institute with the Heriot-Watt University which is a part of the Edinburgh Research Partnership.

## REFERENCES

[1] R. Mudumbai, D. R. Brown, U. Madhow, and H. V. Poor, "Distributed transmit beamforming: challenges and recent progress," *IEEE Commun. Mag.*, vol. 47, no. 2, pp. 102–110, 2009.

[2] Z. Ding, W. H. Chin, and K. K. Leung, "Distributed beamforming and power allocation for cooperative networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1817–1822, 2008.

[3] P. Fertl, A. Hottinen, and G. Matz, "Perturbation-based distributed beamforming for wireless relay networks," in *Proc. IEEE Global Telecommunications Conference*, New Orleans, LO, Nov. 2008, pp. 1–5.

[4] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1754–1763, 2007.

[5] D. R. Brown and H. V. Poor, "Time-slotted round-trip carrier synchronization for distributed beamforming," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5630–5643, 2008.

[6] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Scalable feedback control for distributed beamforming in sensor networks," in *Proc. International Symposium on Information Theory*, Adelaide, SA, Sept. 2005, pp. 137–141.

[7] ——, "Distributed transmit beamforming using feedback control," *IEEE Trans. Inf. Theory*, in review.

[8] M. Seo, M. Rodwell, and U. Madhow, "A feedback-based distributed phased array technique and its application to 60-GHz wireless sensor network," in *Proc. IEEE MTT-S International Microwave Symposium*, Atlanta, GA, June 2008, pp. 683–686.

[9] R. Mudumbai, B. Wild, U. Madhow, and K. Ramchandran, "Distributed beamforming using 1 bit feedback: from concept to realization," in *Proc. 44th Allerton Conf. Commun., Control, Comp.*, Monticello, IL, Sept. 2006, pp. 1020–1027.

[10] J. A. Bucklew and W. A. Sethares, "Convergence of a class of decentralized beamforming algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2280–2288, 2008.

# PROBABILITY OF ERROR FOR BPSK MODULATION IN DISTRIBUTED BEAMFORMING WITH PHASE ERRORS

*Shuo Song, John S. Thompson, Pei-Jung Chung, Peter M. Grant*

Institute for Digital Communications
Joint Research Institute for Signal & Image Processing
School of Engineering and electronics
University of Edinburgh
{s.song, john.thompson, p.chung, peter.grant}@ed.ac.uk

**ABSTRACT**

This paper presents an investigation into the error probability performance for binary phase-shift keying modulation in distributed beamforming with phase errors. The effects of the number of nodes on the beamforming performance are examined as well as the influences of the cumulative phase errors and the total transmit power. Simulation results show a good match with the mathematical analysis of error probability in both static and time-varying channels.

## 1. INTRODUCTION

Recently, there has been interest in applying beamforming techniques into wireless sensor networks. The motivation is to reduce the energy requirement for each sensor node in signal transmission, and extend the communication range to a far field receiver. The individual sensor nodes share the collected information and transmit it in such a way that the signals add coherently at the destination. Transmit beamforming requires accurate synchronization in frequency and phase among sensors, and accurate channel estimation between each sensor node and the receiver. Although certain techniques have been designed in [1], [2], [3] to minimize the phase errors among sensor nodes, phase errors cannot be eliminated due to hardware constraints. Minimizing total transmit power using quantized channel state information has been studied in [4]. The beam pattern performance of distributed beamforming has been studied in [5] and [6] with synchronous phase errors among sensor nodes. From a more practical view, in this paper, we investigate the probability of error for binary phase-shift keying (BPSK) modulation in distributed beamforming with synchronous phase errors and noise.

Shuo Song thanks China Scholarship Council/University of Edinburgh Joint Scholarship Program for supporting his PhD studies.

We acknowledge the support of the Scottish Funding Council for the Joint Research Institute with the Heriot-Watt University which is a part of the Edinburgh Research Partnership.

The rest of the paper is organized as follows. Section 2 introduces the system model. In Section 3 we give an equivalent channel concept to simplify the whole beamforming process. In Section 4 the mathematical analysis of the average bit error ratio (BER) for BPSK in both static and time-varying channels are presented. In Section 5 we analyze the beamforming gain with constant total transmit power. Section 6 then presents simulation results to compare with the theoretical analysis and Section 7 draws conclusions for the paper.

## 2. SYSTEM MODEL

We consider a system of $N$ sensor nodes collaboratively beamforming a narrowband message signal $s(t) = A \cdot m(t)$ to a distant coherent receiver, where $A$ is the amplitude of the message signal. This is performed in a distributed manner by each sensor node modulating $s(t)$ with a RF carrier signal, as illustrated in Fig. 1.

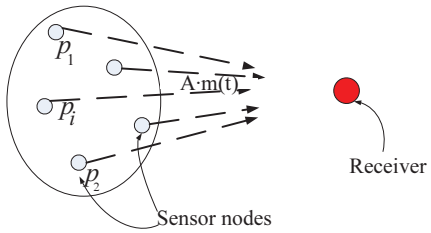



**Fig. 1**. System model for distributed beamforming

We assume that each sensor node and the receiver are equipped with one single ideal omnidirectional antenna, and there are no mutual coupling effects among the antennas. The receiver has the ability to retrieve the overall channel phase from the received signal. All sensor nodes are synchronized so that they can transmit at the same carrier frequency, and

signals transmitted from each sensor node will be added coherently at the receiver. The complex baseband model of the received signal is given by

$$r(t) = \sum_{i=1}^{N} |h_i(t)p_i(t)|e^{j\phi_i(t)}s(t) + n(t) \quad (1)$$

where $p_i(t)$ is the amplification factor and $h_i(t)$ is the channel gain for sensor node $i$, $\phi_i(t)$ is the cumulative phase error of the carrier signal from the synchronization process among sensor nodes and the estimation of the channel gain for sensor node $i$, $n(t) \sim CN(0, \sigma_n^2)$ is additive white Gaussian noise. We assume all phase errors $\phi_i(t)$ are independently and uniformly distributed within the range $(-\phi_0, \phi_0)$, which is the assumption adopted in previously reported investigations [1], [2].

## A. Static Channel

In a static channel scenario, $h_i(t)$ is set equal to a constant. For simplicity, we set coefficients $h_i(t)$, $p_i(t)$ to be unity. Then the system model is expressed as:

$$r(t) = \sum_{i=1}^{N} e^{j\phi_i(t)}s(t) + n(t) \quad (2)$$

## B. Time-Varying Channel

In our time-varying model, the channel coefficients are independent circularly symmetric complex Gaussian distributed, denoted as $h_i(t) \sim CN(0, 1)$, which corresponds to non-line of sight or Rayleigh fading channels. By applying maximal ratio combining, where the pre-amplification gain of each channel is made proportional to the received signal level, we set $|p_i(t)| = |h_i(t)|$ and the system model is then expressed as:

$$r(t) = \sum_{i=1}^{N} |h_i(t)|^2 e^{j\phi_i(t)}s(t) + n(t) \quad (3)$$

### 3. ANALYSIS OF THE EQUIVALENT CHANNEL

If we view the whole beamforming process as an equivalent channel, denoted as $H(t)$, the system model becomes:

$$r(t) = H(t)s(t) + n(t) \quad (4)$$

where $H(t) = \sum_{i=1}^{N} e^{j\phi_i(t)}$ for the static channel scenario, and $H(t) = \sum_{i=1}^{N} |h_i(t)|^2 e^{j\phi_i(t)}$ for the Rayleigh fading channel scenario. With a coherent receiver, the signal-to-noise ratio (SNR) gain, $\|H(t)\|^2$, is the key element deciding the error probability for distributed beamforming and the communication range for power limited sensor networks.

## A. Static Channel

By the central limit theorem, with a large number of sensor nodes $N$, and the independent identically distributed (i.i.d.) random variables $\phi_i(t)$, we have:

$$\begin{aligned} \|H(t)\|^2 &= \left\| \sum_{i=1}^{N} e^{j\phi_i(t)} \right\|^2 \\ &= \left\| \sum_{i=1}^{N} \cos\phi_i(t) + j \sum_{i=1}^{N} \sin\phi_i(t) \right\|^2 \\ &= \|a_S + jb_S\|^2 \\ &= a_S^2 + b_S^2 \end{aligned} \quad (5)$$

where $a_S = \sum_{i=1}^{N} \cos\phi_i(t) \sim N(\mu_{a_S}, \sigma_{a_S}^2)$, $b_S = \sum_{i=1}^{N} \sin\phi_i(t) \sim N(\mu_{b_S}, \sigma_{b_S}^2)$, using the subscript $S$ for the static channels.

Since the variables $\phi_i(t)$ are independently and uniformly distributed within the range $(-\phi_0, \phi_0)$, the means and variances of $a_S$ and $b_S$ can be obtained as:

$$\begin{aligned} \mu_{a_S} &= N \cdot E[\cos\phi_i(t)] \\ &= N\frac{\sin\phi_0}{\phi_0} \end{aligned} \quad (6)$$

$$\mu_{b_S} = 0 \quad (7)$$

$$\begin{aligned} \sigma_{a_S}^2 &= N\Big( E[\cos^2\phi_i(t)] - (E[\cos\phi_i(t)])^2 \Big) \\ &= N\left(\frac{1}{2} + \frac{\sin 2\phi_0}{4\phi_0} - (\frac{\sin\phi_0}{\phi_0})^2\right) \end{aligned} \quad (8)$$

$$\begin{aligned} \sigma_{b_S}^2 &= N\Big( E[\sin^2\phi_i(t)] - (E[\sin\phi_i(t)])^2 \Big) \\ &= N\left(\frac{1}{2} - \frac{\sin 2\phi_0}{4\phi_0}\right) \end{aligned} \quad (9)$$

From (8) and (9), we see, for the equivalent channel $H(t)$, the variance of the real part $\sigma_{a_S}^2$ and the variance of the imaginary part $\sigma_{b_S}^2$ are not equal, which means that the probability density function (PDF) of $\|H(t)\|^2$ is not easily obtained from the joint PDF of $H(t)$, $p(a_S, b_S)$.

## B. Rayleigh Fading Channel

For the Rayleigh fading channels, similarly, with a large number of sensor nodes $N$, and the i.i.d. random variables $h_i(t)$ which are independent from the i.i.d. random variables $\phi_i(t)$, we have:

$$\begin{aligned}\|H(t)\|^2 &= \left\|\sum_{i=1}^{N}|h_i(t)|^2 e^{j\phi_i(t)}\right\|^2 \\ &= \left\|\sum_{i=1}^{N}|h_i(t)|^2\cos\phi_i(t) + j\sum_{i=1}^{N}|h_i(t)|^2\sin\phi_i(t)\right\|^2 \\ &= \|a_R + jb_R\|^2 \\ &= a_R^2 + b_R^2 \end{aligned} \quad (10)$$

where $a_R = \sum_{i=1}^{N}|h_i(t)|^2\cos\phi_i(t) \sim N(\mu_{a_R}, \sigma_{a_R}^2)$, and $b_R = \sum_{i=1}^{N}|h_i(t)|^2\sin\phi_i(t) \sim N(\mu_{b_R}, \sigma_{b_R}^2)$, using the subscript $R$ for the Rayleigh fading channels.

Based on the previous assumptions that the channel coefficients $h_i(t)$ are independent circularly symmetric complex Gaussian distributed $h_i(t) \sim CN(0,1)$, and $\phi_i(t) \sim (-\phi_0, \phi_0)$, we derived the means and variances of $a_R$ and $b_R$ as follows:

$$\begin{aligned}\mu_{a_R} &= N\cdot E[|h_i(t)|^2\cos\phi_i(t)] \\ &= N\cdot E[|h_i(t)|^2]\cdot E[\cos\phi_i(t)] \\ &= N\frac{\sin\phi_0}{\phi_0} \end{aligned} \quad (11)$$

$$\mu_{b_R} = 0 \quad (12)$$

$$\begin{aligned}\sigma_{a_R}^2 &= N\Big(\ E[(|h_i(t)|^2\cos\phi_i(t))^2] - (E[|h_i(t)|^2\cos\phi_i(t)])^2\ \Big) \\ &= N\Big(E[|h_i(t)|^4]\cdot E[\cos^2\phi_i(t)] - (E[|h_i(t)|^2\cos\phi_i(t)])^2\Big) \\ &= N\left(1 + \frac{\sin 2\phi_0}{2\phi_0} - (\frac{\sin\phi_0}{\phi_0})^2\right) \end{aligned} \quad (13)$$

$$\begin{aligned}\sigma_{b_R}^2 &= N\Big(\ E[(|h_i(t)|^2\sin\phi_i(t))^2] - (E[|h_i(t)|^2\sin\phi_i(t)])^2\ \Big) \\ &= N\Big(E[|h_i(t)|^4]\cdot E[\sin^2\phi_i(t)] - (E[|h_i(t)|^2\sin\phi_i(t)])^2\Big) \\ &= N\left(1 - \frac{\sin 2\phi_0}{2\phi_0}\right) \end{aligned} \quad (14)$$

Similarly, from (13) and (14) we see, for the Rayleigh fading channel scenario, $\sigma_{a_R}^2$ and $\sigma_{b_R}^2$ are not equal, thus the expression of the PDF of $\|H(t)\|^2$ is difficult to compute.

## 4. MATHEMATICAL ANALYSIS OF ERROR PROBABILITY

The BER of BPSK over a fixed channel in the presence of AWGN is given by [7] in Chapter 5:

$$P_e(\gamma) = \frac{1}{2}\text{erfc}(\sqrt{\gamma}) \quad (15)$$

where $\gamma$ is the received signal-to-noise ratio per bit, and erfc(.) is the complementary error function.

When the channel gain is random, the average BER for BPSK over all values of $\gamma$ is given by [7] in Chapter 14:

$$P_e = \int_0^\infty P_e(\gamma)p(\gamma)d\gamma \quad (16)$$

where $\gamma = \|H(t)\|^2\frac{A^2}{\sigma_n^2}$ in our system model described in Section 2.

In Section 3, we have analyzed the SNR gain $\|H(t)\|^2$ of the distributed beamforming system, and the expression of the PDF of $\|H(t)\|^2$ was not obtained due to the variances of the real part and the imaginary part of the equivalent channel being unequal. Consequently, $p(\gamma)$ is not available in either the static channel scenario or the Rayleigh fading channel scenario. Formula (16) cannot be solved directly to get a closed-form expression of the integration for our model, and can only be evaluated by numerical techniques. Instead, we provide another method to approximate the BER results as follows.

### Method 1:

For both the static channel scenario and the Rayleigh fading channel scenario, we set the variances of the real part and the imaginary part of $H(t)$ to be equal and use the maximum value between them:

$$\sigma_S^2 = \max(\sigma_{a_S}^2, \sigma_{b_S}^2) \quad (17)$$

for the static channels, and

$$\sigma_R^2 = \max(\sigma_{a_R}^2, \sigma_{b_R}^2) \quad (18)$$

for the Rayleigh fading channels.

Because the real part and the imaginary part of the equivalent channel $H(t)$ now have different means but same variances, the magnitude gain of $H(t)$ is approximated as a Rician distribution.

The closed-form of BER for BPSK through Rician fading channel with a coherent receiver is given by [8]:

$$P_E = Q_1(u,w) - \frac{1}{2}\left(1 + \sqrt{\frac{d}{1+d}}\right)\exp\left(-\frac{u^2+w^2}{2}\right)I_0(uw) \quad (19)$$

where

$$d = 2\sigma^2\frac{A^2}{\sigma_n^2} \quad (20)$$

$$u = \sqrt{\frac{\mu_a^2+\mu_b^2}{2\sigma^2}\cdot\frac{1+2d-2\sqrt{d(1+d)}}{2(1+d)}} \quad (21)$$

$$w = \sqrt{\frac{\mu_a^2 + \mu_b^2}{2\sigma^2} \cdot \frac{1 + 2d + 2\sqrt{d(1+d)}}{2(1+d)}} \tag{22}$$

and $I_0(x)$ is the zeroth-order-modified Bessel function of the first kind, defined as:

$$I_0(x) = \sum_{k=0}^{\infty} \frac{(x/2)^{2k}}{k!\Gamma(k+1)}, \qquad x \geq 0 \tag{23}$$

$Q_1(x, y)$ is the Marcum $Q$-function, defined as:

$$Q_1(x, y) = \int_y^{\infty} z \cdot \exp\left(-\frac{z^2 + x^2}{2}\right) I_0(xz) dz \tag{24}$$

Using (19) to (24), we can get the BER for our static channel scenario by substituting (6), (7), (17) for $\mu_a, \mu_b, \sigma^2$ in (20), (21), (22), and get the BER for our Rayleigh fading channel scenario by substituting (11), (12), (18) for $\mu_a, \mu_b, \sigma^2$ in (20), (21), (22).

An approximation of $I_0(x)$ is given by [9] in Chapter 6:

$$I_0(x) \approx \frac{1}{\sqrt{2\pi x}} \exp(x), \qquad x \gg 0 \tag{25}$$

and after manipulation, (19) can be simplified as:

$$P_E = Q_1(u, w) - \frac{1}{2\sqrt{2\pi uw}} \left(1 + \sqrt{\frac{d}{1+d}}\right) \exp\left(-\frac{(u-w)^2}{2}\right) \tag{26}$$

## Method 2:

We are currently investigating the approximation of the BER performance by an additive white Gaussian noise formula. This is the subject of ongoing work.

## 5. DISTRIBUTED BEAMFORMING GAIN WITH CONSTANT TOTAL TRANSMIT POWER

As the received signal-to-noise ratio cannot show the advantages of beamforming gain, and is uncertain due to independent and random phase errors $\phi_i$, our simulation results are plotted as BER vs total transmit power. Before we present our simulation results, we first analyze the beamforming gain with constant total transmit power. We use $P$ to represent the total transmit power of all the sensor nodes. In the static channel scenario,

$$\begin{aligned} P &= \sum_{i=1}^{N} A^2 \\ &= A^2 \cdot N \end{aligned}$$

In the Rayleigh fading channel scenario,

$$\begin{aligned} P &= \sum_{i=1}^{N} (A|p_i(t)|)^2 \\ &= A^2 \sum_{i=1}^{N} |p_i(t)|^2 \end{aligned}$$

With large $N$, by the law of large numbers, it becomes:

$$P \approx A^2 \cdot N$$

Generally, with a constant $P$, we can represent $A$ as:

$$A = \sqrt{\frac{P}{N}} \tag{27}$$

Putting (27) and $s(t) = A \cdot m(t)$ into (4), we obtain:

$$r(t) = H(t) \sqrt{\frac{P}{N}} m(t) + n(t) \tag{28}$$

and

$$\gamma = \frac{1}{N} \|H(t)\|^2 \frac{P}{\sigma_n^2} \tag{29}$$

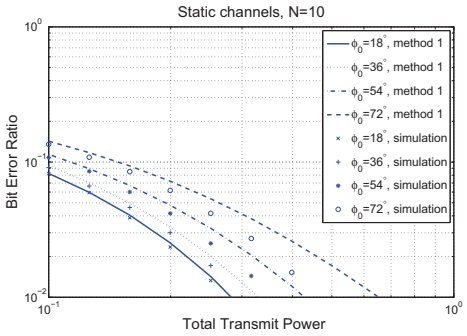
Since the mean of $\|H(t)\|^2$ grows linearly with $N^2$, the mean of the received signal-to-noise ratio per bit $\gamma_{mean} \propto P \cdot N$, and with a constant $P$, $\gamma_{mean}$ is proportional to $N$.

## 6. SIMULATION RESULTS

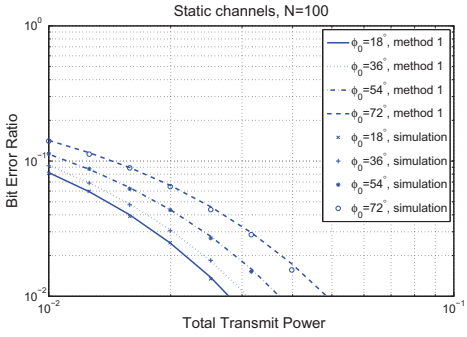
In this section, we present some simulation results in accordance with our previous assumptions, and compare them with our mathematical analysis given in Section 4.

Fig. 2 shows the comparison of the simulation results with the mathematical analysis based on method 1 for BPSK modulation over *static channels* with phase errors. The simulation results are conducted over $10^5$ symbols with different number of nodes $N = 10, 100, 1000$, and different phase error ranges $\phi_0 = 18°, 36°, 54°, 72°$. We set $n(t) \sim CN(0, 1)$. All curves in Fig. 2 are drawn by (19) except the curves for $\phi_0 = 18°$ with $N = 100$ in part (b) and $\phi_0 = 18°, 36°, 54°$ with $N = 1000$ in part (c). These four curves cannot be drawn out by (19) because of the overflow caused by the function $I_0(x)$ in (23) used in MATLAB. Instead of (19), We use (26) to draw these four curves in Fig. 2.

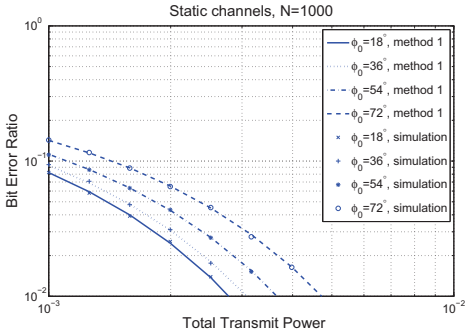
By comparing the simulation results plotted in parts (a), (b), (c) in Fig. 2 and noting the order of magnitude difference of total transmit power in (a), (b), (c), we find that, with similar BER performance in each part, when increasing the number of nodes $N$ by a factor of 10, the total transmit power is reduced by a factor of 10, which means the energy transmitted by each node is reduced by a factor of $10^2$. Thus, we have the conclusion that increasing the number of nodes $N$




(a)

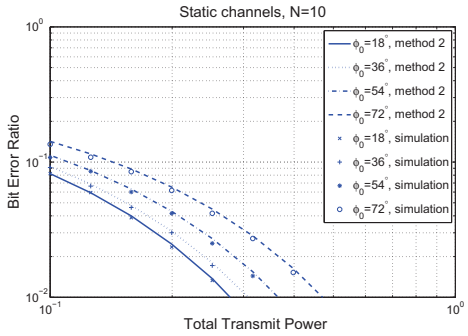



(b)

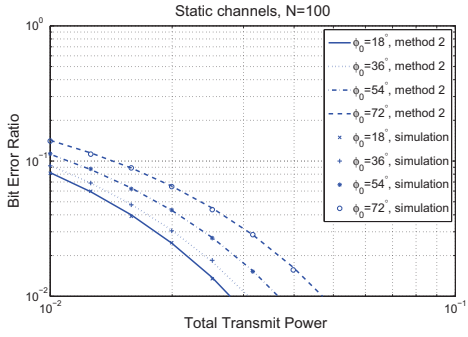



(c)

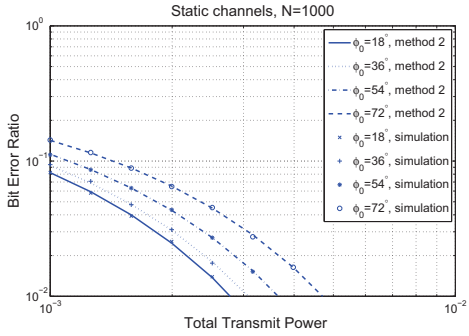
**Fig. 2**. Comparison of mathematical analysis based on method 1 with simulation results of BER versus total transmit power in the static channel scenario with different numbers of nodes $N$ =(a)10, (b)100, and (c)1000, and different phase error ranges $\phi_0 = 18°, 36°, 54°, 72°$.




(a)

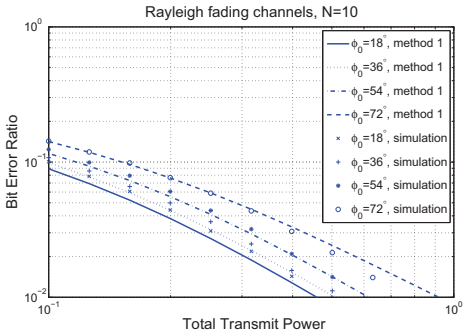



(b)

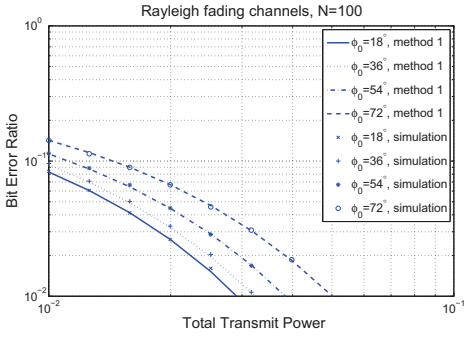



(c)

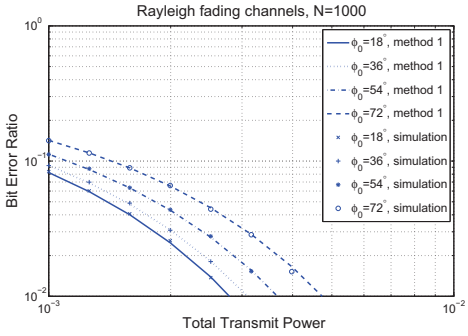
**Fig. 3**. Comparison of mathematical analysis based on method 2 with simulation results of BER versus total transmit power in the static channel scenario with different numbers of nodes $N$ =(a)10, (b)100, and (c)1000, and different phase error ranges $\phi_0 = 18°, 36°, 54°, 72°$.




(a)

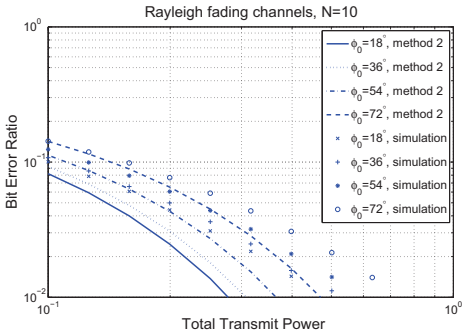



(b)

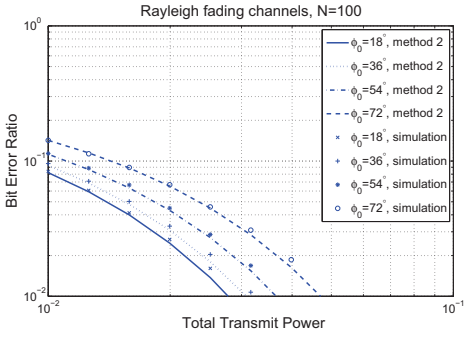



(c)

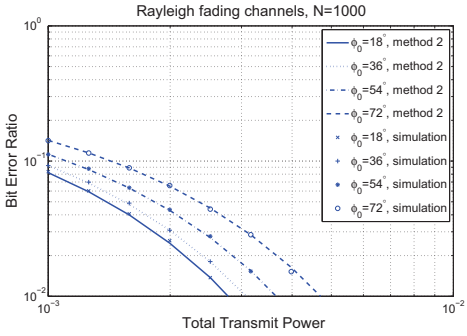
**Fig. 4**. Comparison of mathematical analysis based on method 1 with simulation results of BER versus total transmit power in the Rayleigh fading channel scenario with different numbers of nodes $N$ =(a)10, (b)100, and (c)1000, and different phase error ranges $\phi_0 = 18°, 36°, 54°, 72°$.




(a)

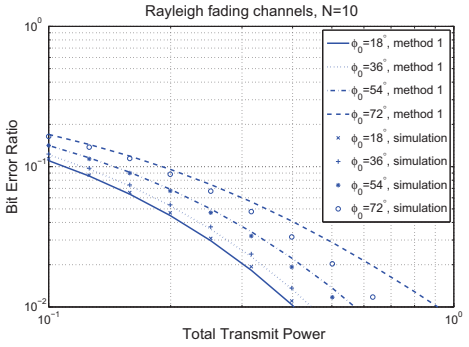



(b)

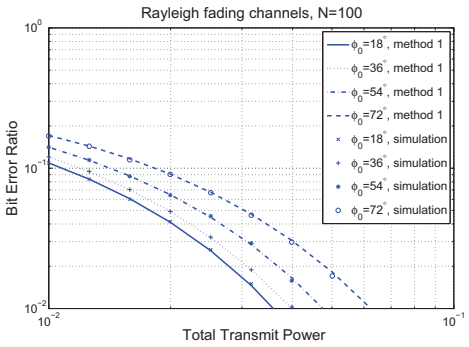



(c)

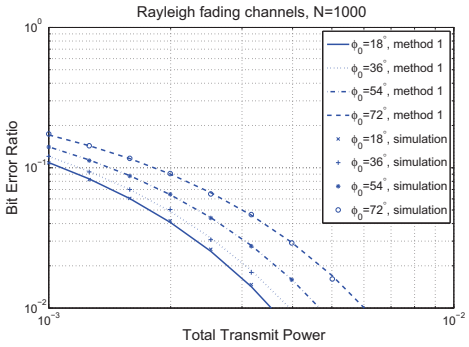
**Fig. 5**. Comparison of mathematical analysis based on method 2 with simulation results of BER versus total transmit power in the Rayleigh fading channel scenario with different numbers of nodes $N$ =(a)10, (b)100, and (c)1000, and different phase error ranges $\phi_0 = 18°, 36°, 54°, 72°$.




(a)

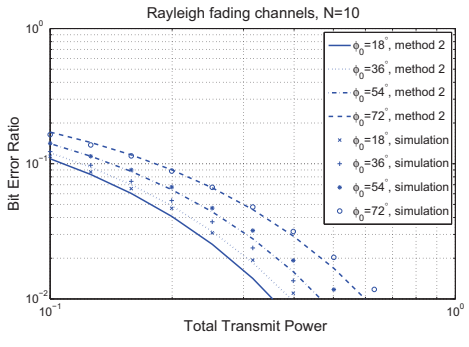



(b)

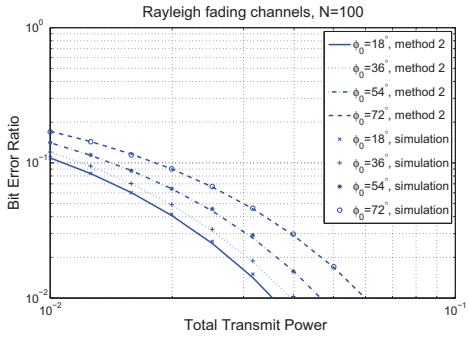



(c)

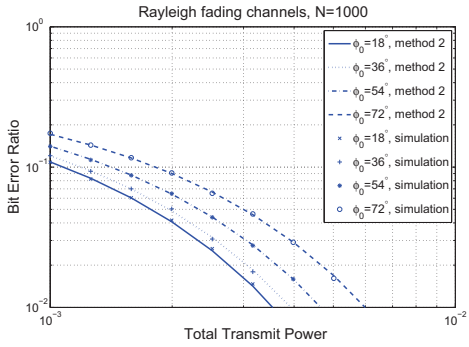
**Fig. 6**. New Comparison of mathematical analysis based on method 1 with simulation results of BER versus total transmit power in the Rayleigh fading channel scenario with different numbers of nodes $N$ =(a)10, (b)100, and (c)1000, and different phase error ranges $\phi_0 = 18°, 36°, 54°, 72°$.




(a)




(b)




(c)

**Fig. 7**. New Comparison of mathematical analysis based on method 2 with simulation results of BER versus total transmit power in the Rayleigh fading channel scenario with different numbers of nodes $N$ =(a)10, (b)100, and (c)1000, and different phase error ranges $\phi_0 = 18°, 36°, 54°, 72°$.

can dramatically reduce the energy requirement for each sensor node subject to the same BER performance, and the number of nodes $N$ has a much larger effect on BER performance than the phase error range $\phi_0$.

From Fig. 2, we see, on the one hand, with a large number of nodes $N = 1000$, the BER analysis based on method 1 matches the simulation results accurately. On the other hand, with a small number of nodes $N = 10$, the BER analysis based on method 1 has a slight difference with the simulation results. This is due to the limitation that central limit theorem does not apply for a small number of nodes.

Fig. 6 shows the comparison of the simulation results with the mathematical analysis based on method 1 for BPSK modulation over *Rayleigh fading channels* with phase errors. The simulation results are also conducted over $10^5$ symbols with different number of nodes $N = 10, 100, 1000$, and different phase error ranges $\phi_0 = 18°, 36°, 54°, 72°$. We also set $n(t) \sim CN(0, 1)$.

Similarly, from Fig. 6 we see, with large $N$, method 1 gives an accurate prediction of the BER, but with small $N$, method 1 gives a better prediction in the Rayleigh fading channel scenario than that in the static channel scenario.

By comparing the simulation results plotted in parts (a), (b), (c) in Fig. 6, we can also have the conclusion that increasing the number of nodes $N$ can dramatically reduce the energy requirement for each sensor node subject to the same BER performance, and the number of nodes $N$ has a much larger effect on BER performance than the phase error range $\phi_0$.

By comparing the simulation results plotted in Fig. 2 with those in Fig. 6, we see when increasing the number of nodes $N$, the BER performance in the Rayleigh fading channel scenario comes close to that in the static channel scenario, which highlights the ability to mitigate fading through path diversity.

## 7. CONCLUSION

We have simulated the BER performance for BPSK modulation in distributed beamforming with phase errors in the static channel scenario and the Rayleigh fading channel scenario, where the results show a good match with our mathematical analysis. The whole beamforming process has been viewed as an equivalent channel and the system performance has been analyzed for different numbers of nodes and different phase error ranges. As the closed-form expression of BER is not easily obtained, we provide a method to approximate the BER results. Generally, method 1 gives a better prediction in the Rayleigh fading channel scenario than the static channel scenario. We are currently working on other approximations of the BER performance, such as method 2 outlined above. The effect of the energy limitation of each sensor node on the BER performance, and BER analysis for other modulation schemes in distributed beamforming with phase errors are also of particular interest for future work.

## 8. REFERENCES


[1] R. Mudumbai, G. Barriac, U. Madhow. *On the Feasibility of Distributed Beamforming in Wireless Networks*. IEEE Transactions on Wireless Communications, Vol. 6, No. 5, pp. 1754-1763, May 2007.

[2] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac. *Scalable Feedback Control for Distributed Beamforming in Sensor Networks*. Proceedings. International Symposium on Information Theory, Adelaide(SA), pp. 137-141, September 2005.

[3] P. Jeevan, S. Pollin, A. Bahai, and P.P. Varaiya. *Pairwise Algorithm for Distributed Transmit Beamforming*. IEEE International Conference on Communications, Beijing, pp. 4245-4249, May 2008.

[4] A.G. Marques, Xin Wang, and G.B. Giannakis. *Minimizing Transmit Power for Coherent Communications in Wireless Sensor Networks With Finite-Rate Feedback*. IEEE Transactions on Signal Processing, vol. 56, No. 9, pp. 4446-4457, September 2008.

[5] M. Tummala, C.C. Wai and P. Vincent. *Distributed Beamforming in Wireless Sensor Networks*. Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, California(USA), pp793-797, October 2005.

[6] H. Ochiai, P. Mitran, H.V. Poor, and V. Tarokh. *Collaborative Beamforming for Distributed Wireless Ad Hoc Sensor Networks*. IEEE Transactions on Signal Processing, Vol. 53, No. 11, pp. 4110-4124, November 2005.

[7] J.G. Proakis. Digital Communications $4^{th}$ ed., McGraw-Hill, Boston, 2001.

[8] W.C. Lindsey. *Error Probabilities for Rician Fading Multichannel Reception of Binary and N-ary Signals*. IEEE Transactions on Information Theory, Vol. 10, No. 4, pp. 339-350, October 1964.

[9] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Numerical Recipes in C $2^{nd}$ ed., Cambridge University Press, 1992.

# References


[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1965.

[2] G. Barriac, R. Mudumbai, and U. Madhow, "Distributed beamforming for information transfer in sensor networks," in *Proc. Third International Symposium on Information Processing in Sensor Networks*, (Berkeley, CA, USA), pp. 81–88, April 2004.

[3] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1754–1763, 2007.

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

[5] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming in ad hoc networks," in *Proc. IEEE Information Theory Workshop*, (San Antonio, TX, USA), pp. 396–401, October 2004.

[6] R. Mudumbai, D. R. Brown, U. Madhow, and H. V. Poor, "Distributed transmit beamforming: challenges and recent progress," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 102–110, 2009.

[7] "10 emerging technologies that will change the world," *MIT's Technology Review Magazine*, pp. 33–49, February 2003.

[8] C. Y. Chong and K. S. P., "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[9] N. Hoult, P. J. Bennett, I. Stoianov, P. Fidler, C. Maksimovic, C. Middleton, N. Graham, and K. Soga, "Wireless sensor networks: creating smart infrastructure," *Proceedings of ICE, Civil Engineering*, vol. 162, no. CE3, pp. 136–143, 2009.

[10] F. Stajano, N. Hoult, I. Wassell, P. Bennett, C. Middleton, and K. Soga, "Smart bridges, smart tunnels: transforming wireless sensor networks from research prototypes into robust engineering infrastructure," *Ad Hoc Networks*, vol. 8, no. 8, pp. 872–888, 2010.

[11] A. Swami, Q. Zhao, Y. W. Hong, and L. Tong, *Wireless Sensor Networks: Signal Processing and Communications Perspectives*. John Wiley & Sons, 2007.

[12] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. John Wiley & Sons, 2007.

[13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[14] C. A. Balanis, *Antenna Theory: Analysis and Design*. John Wiley & Sons, 2003.

[15] T. Shu and M. Krunz, "Coverage-time optimization for clustered wireless sensor networks: a power-balancing approach," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 202–215, 2010.

[16] B. D. Van Veen and K. M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[17] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[18] H. Van Trees, *Optimum Array Processing: Part IV Detection, Estimation and Modulation Theory*. New York: Wiley, 2002.

[19] L. C. Godara, "Application of antenna arrays to mobile communications, part II: beamforming and direction-of-arrival considerations," *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1195–1245, 1997.

[20] J. N. Al-Karaki and A. E. a. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[21] H. Luo, Y. Liu, and S. K. Das, "Routing correlated data in wireless sensor networks: a survey," *IEEE Network*, vol. 21, no. 6, pp. 40–47, 2007.

[22] Z. Ding, W. H. Chin, and K. K. Leung, "Distributed beamforming and power allocation for cooperative networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 5, pp. 1817–1822, 2008.

[23] V. Havary-Nassab, S. Shahbazpanahi, A. Grami, and Z. Q. Luo, "Distributed beamforming for relay networks based on second-order statistics of the channel state information," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4306–4316, 2008.

[24] H. Chen, A. B. Gershman, and S. Shahbazpanahi, "Filter-and-forward distributed beamforming in relay networks with frequency selective fading," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1251–1262, 2010.

[25] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[26] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[27] F. Ren, C. Lin, and F. Liu, "Self-correcting time synchronization using reference broadcast in wireless sensor network," *IEEE Wireless Communications Magazine*, vol. 15, no. 4, pp. 79–85, 2008.

[28] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 81–97, 2008.

[29] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4110–4124, 2005.

[30] M. F. A. Ahmed and S. A. Vorobyov, "Collaborative beamforming for wireless sensor networks with Gaussian distributed sensor nodes," *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 638–643, 2009.

[31] A. Amar, "The effect of local scattering on the gain and beamwidth of a collaborative beampattern for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2730–2736, 2010.

[32] K. Zarifi, S. Affes, and A. Ghrayeb, "Collaborative null-steering beamforming for uniformly distributed wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1889–1903, 2010.

[33] M. F. A. Ahmed and S. A. Vorobyov, "Sidelobe control in collaborative beamforming via node selection," *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6168–6180, 2010.

[34] E. Koyuncu, Y. Jing, and H. Jafarkhani, "Distributed beamforming in wireless relay networks with quantized feedback," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 8, pp. 1429–1439, 2008.

[35] R. Mudumbai, B. Wild, U. Madhow, and K. Ramchandran, "Distributed beamforming using 1 bit feedback: from concept to realization," in *Proc. 44th Allerton Conference on Communication, Control, and Computing*, (Monticello, IL, USA), pp. 1020–1027, September 2006.

[36] C.-W. Chang, A. Kothari, A. Jafri, D. Koutsonikolas, D. Peroulis, and Y. C. Hu, "Radiating sensor selection for distributed beamforming in wireless sensor networks," in *Proc. IEEE Military Communications Conference*, (San Diego, CA, USA), pp. 1–7, November 2008.

[37] D. R. Brown and H. V. Poor, "Time-slotted round-trip carrier synchronization for distributed beamforming," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5630–5643, 2008.

[38] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "On the effects of phase estimation errors on collaborative beamforming in wireless ad hoc networks," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Philadelphia, PA, USA), pp. 657–660, March 2005.

[39] L. Dong, A. P. Petropulu, and H. V. Poor, "Performance analysis of a cross-layer collaborative beamforming approach in the presence of channel and phase errors," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, (Las Vegas, NV, USA), pp. 3233–3236, April 2008.

[40] T. K. Y. Lo, "Maximum ratio transmission," *IEEE Transactions on Communications*, vol. 47, no. 10, pp. 1458–1461, 1999.

[41] D. J. Love and R. W. Heath, "Equal gain transmission in multiple-input multiple-output wireless systems," *IEEE Transactions on Communications*, vol. 51, no. 7, pp. 1102–1110, 2003.

[42] D. G. Brennan, "Linear diversity combining techniques," *Proceedings of the IRE*, vol. 47, no. 6, pp. 1075–1102, 1959.

[43] X. Zhang and N. C. Beaulieu, "A closed-form BER expression for BPSK using MRC in correlated CCI and Rayleigh fading," *IEEE Transactions on Communications*, vol. 55, no. 12, pp. 2249–2252, 2007.

[44] L. Cao and N. C. Beaulieu, "Closed-form BER results for MRC diversity with channel estimation errors in Ricean fading channels," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1440–1447, 2005.

[45] E. K. Al-Hussaini and A. A. M. Al-Bassiouni, "Performance of MRC diversity systems for the detection of signals with Nakagami fading," *IEEE Transactions on Communications*, vol. 33, no. 12, pp. 1315–1319, 1985.

[46] J. Luo, J. Zeidler, and S. McLaughlin, "Bit-error probability analysis of compact antenna arrays with maximal-ratio combining in correlated Nakagami fading," in *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop*, (Cambridge, MA, USA), pp. 52–57, March 2000.

[47] B. D. Rao and Y. M., "Performance of maximal ratio transmission with two receive antennas," *IEEE Transactions on Communications*, vol. 51, no. 6, pp. 894–895, 2003.

[48] Y. Chen and C. Tellambura, "Performance analysis of maximum ratio transmission with imperfect channel estimation," *IEEE Communications Letters*, vol. 9, no. 4, pp. 322–324, 2005.

[49] Q. T. Zhang, "Probability of error for equal-gain combiners over Rayleigh channels: some closed-form solutions," *IEEE Transactions on Communications*, vol. 45, no. 3, pp. 270–273, 1997.

[50] Q. T. Zhang, "A simple approach to probability of error for equal gain combiners over Rayleigh channels," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 4, pp. 1151–1154, 1999.

[51] N. C. Beaulieu and A. A. Abu-Dayya, "Analysis of equal gain diversity on Nakagami fading channels," *IEEE Transactions on Communications*, vol. 39, no. 2, pp. 225–234, 1991.

[52] Y.-S. Tu and G. J. Pottie, "Coherent cooperative transmission from multiple adjacent antennas to a distant stationary antenna through AWGN channels," in *Proc. IEEE 55th Vehicular Technology Conference*, (Birmingham, AL, USA), pp. 130–134, May 2002.

[53] A. G. Marques, X. Wang, and G. B. Giannakis, "Minimizing transmit-power for coherent communications in wireless sensor networks using quantized channel state information," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, (Honolulu, HI, USA), pp. 529–532, April 2007.

[54] A. G. Marques, X. Wang, and G. B. Giannakis, "Minimizing transmit power for coherent communications in wireless sensor networks with finite-rate feedback," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4446–4457, 2008.

[55] J. G. Proakis, *Digital Communications*. 4th ed. New York: McGraw-Hill, 2001.

[56] L. J. Slater, *Confluent Hypergeometric Functions*. Cambridge University Press, 1960.

[57] G. E. P. Box, "Some theorems on quadratic forms applied in the study of analysis of variance problems," *The Annals of Mathematical Statistics*, vol. 25, no. 2, pp. 290–302, 1954.

[58] A. Abdi, C. Tepedelenlioglu, M. Kaveh, and G. Giannakis, "On the estimation of the $K$ parameter for the Rice fading distribution," *IEEE Communications Letters*, vol. 5, no. 3, pp. 92–94, 2001.

[59] W. Lindsey, "Error probabilities for Rician fading multichannel reception of binary and $N$-ary signals," *IEEE Transactions on Information Theory*, vol. 10, no. 4, pp. 339–350, 1964.

[60] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*. Cambridge University Press, 1992.

[61] S. Jonqyin and I. S. Reed, "Performance of MDPSK, MPSK, and noncoherent MFSK in wireless Rician fading channels," *IEEE Transactions on Communications*, vol. 47, no. 6, pp. 813–816, 1999.

[62] J. S. Thompson and A. Smokvarski, "Bit error ratio performance of a receiver diversity scheme with channel estimation," *IET Communications*, vol. 1, no. 1, pp. 92–100, 2007.

[63] M. S. Alouini, X. Tang, and A. J. Goldsmith, "An adaptive modulation scheme for simultaneous voice and data transmission over fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 837–850, 1999.

[64] D. R. Brown, G. B. Prince, and J. A. McNeill, "A method for carrier frequency and phase synchronization of two autonomous cooperative transmitters," in *Proc. IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, (New York, NY, USA), pp. 260–264, June 2005.

[65] I. Ozil and D. R. Brown, "Time-slotted round-trip carrier synchronization," in *Proc. Forty-First Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA, USA), pp. 1781–1785, November 2007.

[66] D. R. Brown, B. Zhang, B. Svirchuk, and M. Ni, "An experimental study of acoustic distributed beamforming using round-trip carrier synchronization," in *Proc. IEEE International Symposium on Phased Array Systems and Technology*, (Waltham, MA, USA), pp. 316–323, October 2010.

[67] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Scalable feedback control for distributed beamforming in sensor networks," in *Proc. International Symposium on Information Theory*, (Adelaide, SA, USA), pp. 137–141, September 2005.

[68] C. Lin, V. V. Veeravalli, and S. P. Meyn, "A random search framework for convergence analysis of distributed beamforming with feedback," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6133–6141, 2010.

[69] C. C. Chen, C. S. Tseng, and C. Lin, "A general proof of convergence for adaptive distributed beamforming schemes," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, (Prague, Czech), pp. 1–4, May 2011.

[70] F. J. Solis and R. J.-B. Wets, "Minimization by random search techniques," *Mathematics of Operations Research*, vol. 6, no. 1, pp. 19–30, 1981.

[71] J. A. Bucklew and W. A. Sethares, "Convergence of a class of decentralized beamforming algorithms," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2280–2288, 2008.

[72] J. A. Bucklew, T. Kurtz, and W. A. Sethares, "Weak convergence and local stability of fixed step size recursive algorithms," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 966–978, 1993.

[73] J. A. Bucklew and W. A. Sethares, "The covering problem and $\mu$-dependent adaptive algorithms," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2616–2627, 1994.

[74] J. Podpora, L. Reznik, and G. V. Pless, "Intelligent real-time adaptation for power efficiency in sensor networks," *IEEE Sensors Journal*, vol. 8, no. 12, pp. 2066–2073, 2008.

[75] M. Seo, M. Rodwell, and U. Madhow, "A feedback-based distributed phased array technique and its application to 60-GHz wireless sensor network," in *Proc. IEEE MTT-S International Microwave Symposium*, (Atlanta, GA, USA), pp. 683–686, June 2008.

[76] J. Thukral and H. Bolcskei, "Distributed spatial multiplexing with 1-bit feedback," in *Proc. 45th Allerton Conference on Communication, Control, and Computing*, (Monticello, IL, USA), pp. 502–509, September 2007.

[77] P. Fertl, A. Hottinen, and G. Matz, "Perturbation-based distributed beamforming for wireless relay networks," in *Proc. IEEE Global Telecommunications Conference*, (New Orleans, LO, USA), pp. 1–5, November 2008.

[78] P. Fertl, A. Hottinen, and G. Matz, "A multiplicative weight perturbation scheme for distributed beamforming in wireless relay networks with 1-bit feedback," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Taipei, Taiwan), pp. 2625–2628, April 2009.

[79] M. Johnson, M. Mitzenmacher, and K. Ramchandran, "Distributed beamforming with binary signaling," in *Proc. IEEE International Symposium on Information Theory*, (Toronto, Canada), pp. 890–894, July 2008.

[80] P. Jeevan, S. Pollin, A. Bahai, and P. P. Varaiya, "Pairwise algorithm for distributed transmit beamforming," in *Proc. IEEE International Conference on Communications*, (Beijing, China), pp. 4245–4249, May 2008.

[81] G. Lim and L. J. Cimini, "Partitioned one-bit feedback for cooperative beamforming," in *Proc. 44th Annual Conference on Information Sciences and Systems*, (Princeton, NJ, USA), pp. 1–5, March 2010.

[82] W. Tushar and D. B. Smith, "Distributed transmit beamforming based on a 3-bit feedback system," in *Proc. IEEE Eleventh International Workshop on Signal Processing Advances in Wireless Communications*, (Marrakech, Morocco), pp. 1–5, June 2010.

[83] S. Sigg and M. Beigl, "Algorithmic approaches to distributed adaptive transmit beamforming," in *Proc. 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, (Melbourne, Australia), pp. 433–438, December 2009.

[84] R. M. EI Masri, S. Sigg, and M. Beigl, "An asymptotically optimal approach to the distributed adaptive transmit beamforming in wireless sensor networks," in *Proc. European Wireless Conference*, (Lucca, Italy), pp. 511–518, April 2010.

[85] H. Wu, Y. Ding, and H. M. Kwon, "Phase synchronization for distributed sensor networks in feedback bit errors," in *Proc. IEEE International Conference on Wireless Communications, Networking and Information Security*, (Beijing, China), pp. 282–286, June 2010.

[86] C. S. Tseng, C. C. Chen, and C. Lin, "A bio-inspired robust adaptive random search algorithm for distributed beamforming," in *Proc. IEEE International Conference on Communications*, (Kyoto, Japan), pp. 1–6, June 2011.

[87] B. Basu and G. K. Mahanti, "Fire fly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna," *Progress In Electromagnetics Research B*, vol. 32, pp. 169–190, 2011.

[88] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 3, pp. 771–779, 2004.

[89] N. Karaboga, K. Guney, and A. Akdagli, "Null steering of linear antenna arrays with use of modified touring ant colony optimization algorithm," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 12, no. 4, pp. 375–383, 2002.

[90] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. J. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151–1162, 1976.

[91] N. Bershad, "On the optimum gain parameter in LMS adaptation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 7, pp. 1065–1068, 1987.

[92] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1633–1642, 1992.

[93] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-type algorithm: analysis and simulations," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 631–639, 1997.

[94] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[95] H. Ali and H. L. Thomas, "A general theory of phase noise in electrical oscillators," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 2, pp. 179–194, 1998.

[96] A. Alexiou and M. Haardt, "Smart antenna technologies for future wireless systems: trends and challenges," *IEEE Communications Magazine*, vol. 42, no. 9, pp. 90–97, 2004.

[97] L. Liu and H. Jafarkhani, "Novel transmit beamforming schemes for time-selective fading multiantenna systems," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4767–4781, 2006.

[98] M. Nakagami, "The $m$-distribution–A general formula of intensity distribution of rapid fading," *Statistical Methods in Radio Wave Propagation*, W. C. Hofman, Ed., New York: Pergamon, 1960.