System Architecture and Signal Processing Techniques for Massive Multi-user Antenna Arrays

Antonio Puglielli
Borivoje Nikolic
Elad Alon

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-149
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-149.html

December 1, 2019

System Architecture and Signal Processing Techniques for Massive Multi-user Antenna Arrays

by

Antonio Puglielli

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering — Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Borivoje Nikolić, Chair
Professor Elad Alon
Professor David Aldous

Fall 2017

System Architecture and Signal Processing Techniques for Massive Multi-user Antenna Arrays

Abstract

System Architecture and Signal Processing Techniques for Massive Multi-user Antenna Arrays

by

Antonio Puglielli

Doctor of Philosophy in Engineering — Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Borivoje Nikolić, Chair

Future advancements in wireless communication standards will rely on two inter-related technologies. First, to address the saturation of traditional cellular spectrum in the < 6 GHz bands, new and much-higher frequency mm-wave spectrum will be utilized. The mm-wave bands at 24, 28, 39, 60, and 72 GHz, among others, have tens of gigahertz of available and unused spectrum. Second, because modern coding and modulation techniques make near-optimal use of time and frequency resources, the spatial dimension of wireless channels must be exploited to send data streams in a spatially selective manner only to the desired users.

From an engineering perspective, both of these technology trends rely on the design of antenna arrays — spatial selectivity is achieved by having a large number of radios while mm-wave operation needs directional transmissions to overcome the propagation loss. Designing large antenna arrays — on the order of 64-256 or larger number of radios — represents a huge departure from traditional radios used in wireless networks which have at most 4 independent radiators and transceiver chains.

This thesis explores system architecture, signal processing, and hardware design techniques which are suitable for massive antenna arrays which form many spatial beams simultaneously. First, beamforming is identified as the preferred spatial processing technique for the large array regime, and a beamforming-aware array architecture is proposed which is modular, scalable, and distributed. The core element of the proposed array architecture is a common module which integrates multiple transceivers, analog and digital signal processing, and interconnect. Additionally, a time-domain beamforming algorithm is proposed for channels with strong multipath components.

Next, synchronization architectures and algorithms are proposed for both carrier and baseband synchronization in massive antenna arrays. It is shown that uncorrelated phase noise between different transceivers causes decoherence between the front ends. Accordingly, a synchronization strategy consisting of co-optimized carrier recovery loops and distributed phase-locked loops (PLLs) is proposed to manage the total phase noise impact in antenna arrays. For baseband synchronization, a hierarchical timing alignment strategy is proposed

which uses a background calibration loop to compensate for front-end and sampling skews while using distributed data timing recovery to compensate for true time delay effects in large arrays.

Finally, hardware implementations of these system- and signal-processing-level ideas are demonstrated. The first prototype consists of a system-on-chip for low-frequency massive MIMO. The proposed SoC integrates multiple full digital radios with on-chip DSP and serial links. The second prototype demonstrates the first massive, multi-user phased array operating at 72GHz. A 128-element phased array is implemented using off-the-shelf components which can form 16 simultaneous beams over 250MHz bandwidth, for an aggregate capacity of 20 Gbps.

Dedication

Per Nonno Tonino

Contents

List of Figures

List of Tables

Acknowledgments

It is hard to communicate the depth of emotion and experience behind the acknowledgement section of a PhD dissertation. I find that as I have progressed in grad school, I keep finding more depth of meaning and perspective in the acknowledgement sections of theses I read. I think the acknowledgement section is the most significant and interesting part of a PhD dissertation, conveying the immensity of the task accomplished and offering a snapshot into the bizarre and unusual world of graduate school.

I would like to start by thanking my advisors, Bora Nikolic and Elad Alon. You guys were definitely the right advisors for me, both individually and together. Bora — I have learned so much from your big-picture thinking and broad grasp of the context of our work and our industry. Many times I have dismissed your predictions or advice and later realized that you were right all along. I also really appreciate your focus on big, ambitious, complex systems. When I started working for you, you asked me to look into "beamforming" — who would have known that that would launch a five year endeavor touching analog and digital circuits, signal processing, wireless communications, and so much more. I could not imagine working on any narrower scope. Elad — your deep understanding of all technical topics and lightning-fast intellectual reflexes have kept me on my toes through many a discussion with you. You always ask the right questions and slowly I have been (still) learning both how to give the right answers and more importantly, how to ask those questions myself. Your guidance has helped me find the gaps in my understanding and explanations, and pushed me to not settle for good enough but rather strive for perfection, even when I did it kicking and screaming.

I also owe a debt of gratitude to Tom Courtade and Ali Niknejad, who despite not being my advisors were always willing to take the time to discuss ideas, give me feedback, and point me in the right direction. I also want to thank Paul Wright for serving on my quals committee (with Tom) and David Aldous for serving on my dissertation committee. Finally, thanks to Rikky Muller, for your advice and discussions on life after grad school and for hiring me as a TA.

A huge part of this dissertation was done in collaboration with Ozzy LaCaille. Ozzy, you have been my friend, mentor, and inspiration throughout grad school, not to mention travel companion, fellow monkey trainer, and (worryingly) doppelganger. I have worked with you for over three years straight but I am still dumbfounded by your unbounded passion for the work you do, your herculean work ethic, and your deep insights. You'll make a tremendous professor someday soon.

The best part of grad school by far is the friends you make. Sameet — I couldn't imagine grad school without you. I have enjoyed all our time together, from late-night juice taste tests to heated arguments to constant startup ideas. Luke — you are an incredibly loyal friend and a truly strange person. Your passion for the things you care about is really inspiring.

Emily and Keertana - who are always somehow mentioned in the same sentence - thank you for your companionship, for throwing crazy parties and fun picnics, for cheering me up when I am depressed, for long pointless debates (Emily) and always being ready to get lunch

(Keertana). John, thank you for being my friend outside of work: skipper, backpacking buddy, DIY guru. You have way too many hobbies for me to hope to emulate. Bonjern and Nathan, thanks for thousands of lunches at up to 10 different places, and ensuing inane discussions. Krishna, thanks for always taking the time to chat when we bump into each other in the kitchen. Lorenzo and Matteo, thank you for strengthening my tie with the motherland.

I want to thank many others at BWRC for your friendship and technical discussions: Marko for commiserating about project management and system design; Ben and Stevo for serving as my digital tapeout consultants; Paul and Chris Yarp for discussing comm systems; Kosta, Amy, Eric, Zhongkai, and Pengpeng for long ewallpaper tapeout hours; as well as Pavan, Katerina, Matthew, Nandish, Sidney, and Andrew. Thanks also to all the other members of the ComIC group — Amanda, Angie, Brian, Charles, Jaehwa, Luis, Matt, Milos, Mira, Nick, Pi-Feng, Rachel, Sharon, and Vladimir — and the EEIS group — Ali, Jaeduk, Kristel, Nick, and Seobin.

The staff at BWRC secretly keep the place from turning into lord of the flies, so a deep thank you to Candy and Yessica for always being cheerful and getting the engineers to have fun, James for always going above and beyond, Fred for keeping the lab running, and the rest of the team: Amber, Brian, Bira, Erin, Leslie, Melissa, Olivia, and Sarah.

I have a deep gratitude to Greg Wright and Ajith Amerasekara for your mentorship and friendship. Greg, you are a bottomless fount of wireless knowledge, business trends, and random and hilarious stories. Thanks for all your guidance and advice, coffees and beers drunk, and for hosting me at Bell Labs many years ago. Ajith, thank you for your mentorship on career paths, industry trends, and always seeing the bigger picture. You spent countless hours helping us learn how to think about business and how to understand the context of our research, and then had the gall to treat us to dinner. I have really enjoyed your friendship.

Thank you to industry partners and internship mentors — Farhana Sheikh, Chris Hull, Chintan Thakkar, and Bryan Casper of Intel and Brucek Khailany, Rangha Venkatesan, and Bill Dally of Nvidia — for technical discussion, feedback, and guidance as well as great intern experiences.

Friends outside BWRC have helped me stay slightly in touch with the world. Arjun, thanks for countless nights of video games, grad school horror stories, junk food and unreasonably large pancakes, and excessive drinking of CENSORED. Abi, we've been brothers for ages so I guess you had to at least half-heartedly pretend to care about my research. Rachel, your enthusiasm and optimism is infectious. Maggie, thanks for the constant reminder that I could have been a software engineer. Susan, thanks for your encouragement and for teaching us to backpack.

Almost finally, thanks to my parents, little brothers, and grandparents. Dad, you've been preparing me for this since I was a tiny kid reading your papers and grants. Mom, your life's passion has been your kids and at least two of them turned out fine. Ale and Lorenzo, you look up to me even when it's not reasonable. Since I've been away you have grown into men and I see a lot of myself in you guys. Nonno Enzo and Nonna Maria Teresa, thanks for always caring about what I work on even though I can't explain it to you, least of all

in Italian. Finally, Nonno Tonino — you are stubborn as a mule and frequently a pain in the ass, but I guess that's where my dad and I get that from. Without you we wouldn't be where we are today, and we are all more similar than you'd think.

Finally finally: I cannot express my gratitude to my girlfriend Jiwon. You dealt with me for three years from across the country and subjected yourself to another year and a half up close and personal. Sometimes I try to imagine what grad school would have been like without you by my side — I simply draw a blank. You believed in me when I didn't believe in myself, kept me going when I didn't want to, made me smile when I needed it, and stood out as a constant bright spot in my life. It hasn't always been smooth sailing but I could not have done it without you. Trust me, I'm a doctor.

Chapter 1

Introduction

1.1 Motivation: Large Arrays are Needed to Extend Network Capacity

Recent years have witnessed a dramatic increase in network traffic [1], as rapidly proliferating mobile devices drive increased consumption of media-rich services such as mobile video. Current projections anticipate that mobile traffic will grow seven-fold in the five years from 2016-2021. By 2021 it is anticipated that almost two-thirds of global IP traffic will originate over a wireless connection, with 20% coming over a cellular network.

Mobile data consumption both drives and results from advances in communications technologies, which have made it possible to deliver up to hundreds of megabits per second over wireless connections using standards such as 802.11 wireless LAN (WLAN) [2] or LTE [3]. On the back of this capability, the smartphone has become the fastest growing consumer electronic device in human history [4]. The spread of smartphones ensures that everyone in the world has a high-bandwidth device in their back pocket and demands to use it on the go for activities such as mobile gaming, video streaming, and live broadcasts. Today video accounts for almost 75% of all consumer IP traffic; this will increase to 82% by 2021. It is further expected that new device classes, such as augmented reality (AR) or virtual reality (VR) glasses, along with new applications such as connected vehicles and robots, will drive even more traffic growth in the next five years [5]. In fact, AR/VR traffic is expected to grow 20-fold in the next five years.

Even as smartphone traffic consumption has been growing with compound annual growth rate (CAGR) in excess of 50% [1], the capabilities of wireless communication techniques have approached their theoretical limits. Since Shannon proposed his theory of information and communication in 1948 [6], innovations such as low-density parity check (LDPC) codes [7] and orthogonal frequency-division multiplexing (OFDM) modulation [8, 9] have brought the spectral efficiency of wireless standards close to the theoretical Shannon bound. 802.11 and LTE today are within a fraction of a dB away from the Shannon limit [10], meaning that these standards make near-optimal use of time and frequency resources.

Though this is a great achievement, it means that there are no simple solutions to address existing and emerging traffic demands. Previous generations of cellular networks increased datarates in one of two ways. First, wider channels were used (from 200kHz in 2G GSM to 5MHz in 3G WCDMA and 20MHz in 4G LTE to 60 or even 100MHz using carrier aggregation). Second, the infrastructure deployments have been dramatically densified [11]. Today both of those techniques are running into roadblocks. Existing spectrum bands below 6GHz are nearly all allocated. This spectrum scarcity is such that the small remaining chunks of spectrum are auctioned for tens of billions of dollars [12], which is almost an order of magnitude higher than the capital cost of the network equipment itself [13]. In parallel, network densification has run into a cost and complexity bottleneck. Ubiquitous infrastructure deployments are not economically feasible given the high cost of infrastructure equipment today as well as the cost and time of acquiring backhaul connections and zoning/siting permits. Additionally, densification results in interference-limited networks where the achievable performance is limited by the large number of base-stations in close proximity [11]. Mitigating this requires coordination and management of radio resources to deal with interference between densely-deployed and irregularly-spaced base-stations.

Based on this discussion, we can identify three key requirements for next-generation cellular networks. First, physical-layer technologies must make more efficient use of existing spectrum. Second, future base stations must be equipped with techniques to manage and mitigate interference in dense deployments. Finally, infrastructure solutions must be as cheap as possible so that they may be ubiquitously deployed.

It will fall on the fifth generation (5G) of mobile networks to begin addressing some of these challenges. The wireless industry expects that 5G standards will introduce new techniques which can increase the capacity of wireless networks [14] and support emerging data-hungry applications and device classes. Addressing today's challenges and tomorrow's needs will require significant innovation at all layers of the network hierarchy. At the PHY layer, the only available solutions are to move to new high-frequency spectrum, where wide channel bandwidths are available, and to introduce complex spatial processing which is able to spatially filter network interference. Both of these capabilities are implemented using large antenna arrays, meaning that cost- and power-efficient array design is the cornerstone technology upon which future network generations will be built.

1.1.1 Spatial Processing

Spatial processing was the focus of extensive research and development in the 1990s [15–19]. This effort led to the invention of space-time codes [20–22] and finally the discovery of the multi-channel capacity formula [23, 24]. In essence, it was discovered that space presents an additional degree of freedom on top of time and frequency which wireless communications systems can exploit. More specifically, if a wireless link is equipped with multiple antennas at both the transmitter and receiver and if the propagation environment is sufficiently rich, then the capacity of this channel scales linearly with the minimum of the number of trans-

mit or receive antennas. This technique is called multiple-input, multiple-output (MIMO) communications.

MIMO techniques present an *engineering* approach to network capacity. Rather than being limited by the frequency allocation, more capacity can be added to a network simply by deploying more antennas and transceivers. Recognizing the potential of this approach, demonstration systems were quickly built [25, 26] and wireless standards adopted MIMO techniques. Single-user (SU) MIMO was the first technique proposed, where multiple spatial streams are sent to a single user which is equipped with multiple transmit/receive antennas [25, 27]. In practice, real propagation environments can only support limited number of streams to an SU-MIMO device. Multi-user (MU) MIMO is an extention of the MIMO technique where multiple spatial streams are sent to a number of spatially distinct users [28]. The MU channel is significantly more decorrelated since the user devices are fundamentally spatially separated. As such, MU-MIMO can reuse spectrum to communicate simultaneously with many users in the same frequency band, by exploiting those users' separation in space.

MIMO techniques employ spatial processing to manage in-network interference. A related concern is external interference sources which degrade the link quality and performance. These interferers may come from adjacent cells in a cellular network, different operators which occupy separate frequency bands, communication systems on different standards, or even non-communication devices such as radars and microwaves. In the absence of any knowledge about these interferers, spatial filtering is the only known technique which can address the problem of general interferers. Spatial filtering is a form of spatial processing which filters out signals using their spatial characteristics or direction of arrival [29]. A spatial filter could identify external interferers by their spatial signature and reject them.

In summary, spatial processing provides two key benefits. First, spatial multiplexing can be used to send multiple data streams simultaneously over the same frequency band, providing a multiplicative capacity gain. Second, spatial filtering can be used to reject arbitrary interference in the environment. These capabilties make aggressive spatial processing an attractive candidate for 5G networks. One could envision deploying infrastructure nodes equipped with a large number of antennas, serving dozens of simultaneous users in the same band while rejecting interference from nearby cell sites.

In fact, this vision has strong theoretical backing. In 2010, Marzetta proposed the concept of "massive MIMO", where a base station with a large number of elements serves a much smaller number of users using MU-MIMO [30]. Marzetta's theoretical results show that as the number of base-station antennas grows large, simple spatial signal processing techniques can eliminate both intra- and inter-cell interference. Intra-cell interference is explicitly managed by the cell itself using its spatial resolution. Inter-cell interference disappears because, in the limit of very large number of antennas, all links use vanishingly thin pencil beams so the probability of crossing beams goes to zero. In other words, the massive MIMO paradigm — deploying large antenna arrays — provides a scalable approach to simultaneously increase network capacity (by a large factor!) *and* mitigate network-level interference.

Massive MIMO will be exploited to enhance the value of traditional $<$ 6GHz cellular bands. Today, high-order MIMO is being introduced into the latest versions of the LTE

standards [31–33] and infrastructure vendors are commercializing systems using 32-96 antennas [34–36], which provide up to 3-5x capacity improvements over existing 4G networks. However a key limitation of < 6GHz massive MIMO is the physical size and weight of the antenna array. Because the antenna size is related to the wavelength, antenna arrays at < 6GHz are very bulky — up to 3m on a side with a weight up to 40kg. The weight is mostly contributed by power supplies and cooling structures, so addressing power consumption via optimized electronics may help mitigate that. However, the physical antenna size is constrainted by the available site size on standard masts. Unless base-station sites are substantially modified to support huge antenna sizes, it will be impossible to go beyond 96 or certainly 128 antennas in < 6GHz systems, which means that we must look elsewhere for further network enhancements.

1.1.2 Moving to higher frequencies

At the same time, to address the congestion in exisiting low-frequency network deployments, regulatory bodies around the world are opening up higher-frequency mm-wave spectrum, including at 24, 28, 39, 60, and 72 GHz [37, 38]. These mm-wave bands provide much higher bandwidths than existing spectrum which can be used to dramatically increase network capacity. Moreover, the much smaller wavelength at mm-wave frequencies provides the opportunity to deploy arrays with thousands of elements in a very small form factor.

Traditionally, the challenge of operating at high frequencies is the higher propagation loss. This has either limited mm-wave links to short distances, or required the use of large mechanically-steered dish antennas with very high directivity. Recent measurement campaigns in both 28 and 72 GHz bands reveal that with modern technologies, communcation over a range around 200-500m is feasible [39–42]. The proposed systems use phased arrays to synthesize a directional transmission/reception which compensates for the propagation loss. Accordingly, a numer of groups have shown 16- or 32-element arrays with a single-beam range of 100-300m [43–46].

These capabilities will underpin the first deployments of mm-wave spectrum for fixed-access to the home, deploying commercial services at high frequency for the first time [47]. Wireless fixed access services exploit the wide mm-wave channel bandwidths to deliver fiber-quality gigabit home or business internet without the costly and time-consuming fiber deployments.

Looking forward, the true promise of mm-wave lies in deploying a ubiquitous fixed + mobile access network which provides gigabit internet on-the-go, everywhere. 1000-element mm-wave arrays can be realized in a form factor comparable to a WiFi router today. These tiny infrastructure nodes can be deployed anywhere — on lamp-posts, buildings, and bus shelters — providing a super-dense wireless coverage. The huge number of antennas provide super-directional pencil beams which can overcome shadowing, glass or concrete obstructions, and outdoor-to-indoor operation. Spatial processing algorithms exploit the 1000 antennas to serve dozens of clients simultaneously, anywhere from homes to smartphones to vehicles,

while cancelling interference arising from the dense deployments. All of this is complemented by network intelligence which coordinates services across a wide geographic range.

In order to achieve this vision, it is necessary to develop techniques for implementing 1000-element arrays in a cost- and power-efficient manner. The technologies underlying today's 16- or 32-element, single-beam arrays will not scale to the envisioned scenario. As such, developing scalable array design techniques will be a critical step toward tomorrow's glorious wireless future.

1.2 Prior work in array design

In light of the above discussion, it is unsurprising that the design and implementation of antenna arrays is a focus of research effort in academia and industry.

Low-order SU- and MU-MIMO have been introduced in wireless standards over the past decade, including 802.11n/ac and LTE [2, 3]. The recent WLAN standard, 802.11ac, can support up to 8 spatial streams to a single device or 4 simultaneous users through downlink-only MU-MIMO. The forthcoming 802.11ax standard will additionally introduce uplink MU-MIMO capabilities. LTE Advanced supports similar levels of spatial multiplexing.

Today's cellular MIMO radios in the handset are implemented using an integrated RF transceiver chip along with per-antenna front-end electronics (filters, switches, and power amplifiers) and a baseband processor [48]. WLAN radios are more integrated — it is common for radio and baseband processing to be integrated on the same die, and the number of external components is smaller [49]. In contrast, due to their more stringent specifications, LTE base-stations use a larger number of front-end components, particularly power amplifiers (PAs) and their associated predistortion capabilities [50]. Additionally, celllular base-stations frequently separate the radio and baseband processing into physically distinct boxes with an electrical or optical interconnect between them. Finally, while baseband processors in handsets (for both LTE and WiFi) are implemented on dedicated chips, base-stations primarly use more generic hardware, realized as an application-specific integrated circuit (ASIC) which integrates digital signal processor (DSP) cores, CPUs, and some custom accelerators.

Several university and industry research groups have built massive MIMO testbeds in the $<$ 6GHz bands, using 64-128 antennas to serve up to 20 simultaneous users [51–59]. These testbeds have shown successful over-the-air measurements and validated many of the theoretical predictions. However, from a hardware design perspective, they reveal the limitations of scaling up existing radio design paradigms from 8 to 128 antennas. Almost all of these testbeds adopt a fully centralized architecture, where all processing, synchronization, and calibration is performed globally on a set of powerful processors. The radios and front-ends are not that challenging from this perspective — cost and power can be managed by tailoring the radio design to the system specifications. Instead the bulk of the cost, power, and complexity of these testbeds arise from the data aggregation and array processing. Commercial systems entering the market today [35, 36] can brute-force this complexity by using

complex baseband ASICs with custom accelerators for array tasks. However, architectural innovations toward more distributed operation could significantly reduce the power and cost of this solution.

At mm-wave frequencies, over the past decade a number of research groups have shown 16- or 32-element arrays monolithically integrated on a single die [43, 44, 46, 60–65]. Because of the small wavelength and small circuit area, it is possible to fit up to 32 elements (or even more!) on a single chip. These arrays use analog beamforming techniques to form a single transmit/receive beam. These demonstrations were enabled by innovations surrounding the beamforming circuits and the LO distribution to all the front-end elements.

There is no clear way to scale today's 32-element mm-wave arrays to hundreds or even 1000 elements. In particular, such large arrays would necessarily be implemented using a number of front-end subarrays which are fused to form the overall array. These subarray chips have to be interconnected with appropriate data aggregation, synchronization, and signal processing techniques. Moreover, the capability to form dozens of simultaneous beams does not exist in today's mm-wave array solutions.

In summary, the main difference between conventional multi-antenna technology and its 5G counterpart is that future antenna arrays will require massive hardware duplication. As such, the radios, signal processing, etc fundamentally cannot all be implemented on the same chip. Existing array design techniques break down in this regime; the cost, complexity, and power of global synchronization, data aggregation, and processing tasks makes the resulting systems impractical to build and deploy.

1.3 Thesis scope and outline

Motivated by this background, this thesis considers system-level techniques to manage the cost and design complexity of very large arrays. The key role of an array can be thought of as taking a large number of antennas/front-ends and "making them play well together". Accordingly, the main objective of this thesis is to develop techniques for efficiently and cheaply interconnecting the radio front ends in order to form an antenna array with very large spatial aperture.

In keeping with the discussion above, the main areas of focus are data aggregation, signal processing, and synchronization. Chapter 2 provides a brief overview of existing spatial signal processing techniques and presents simulation results showing how these techniques perfom in real channel models. Building off of this background, Chapter 3 proposes a distributed array architecture designed around the spatial signal processing task. Additionally, Chapter 3 presents innovative signal processing techniques which may be used in some array implementations.

Chapters 4 and 5 then move on the synchronization tasks. Chapter 4 analyzes the carrier synchronization across the array and presents innovations and design guidelines into the carrier generation architecture as well as the specifications for circuit components within

this subsystem. Chapter 5 studies the baseband synchronization of the array and proposes techniques to accomplish this in a distributed fashion.

Finally, Chapters 6 and 7 describe hardware implementations of these ideas. Chapter 6 presents a highly-integrated system-on-a-chip in 65nm CMOS operating < 6 GHz, while Chapter 7 describes a 72GHz MU-MIMO array implemented using off-the-shelf radios, data converters, and FPGAs.

Chapter 2

Array Processing

Antenna arrays have been used for decades for their ability to resolve the spatial characteristics of the environment. For example, radars can precisely track multiple targets in space while filtering out the impact of ground reflections or other clutter. Similarly, spatially selective communication systems can send multiple data streams in the same frequency channel by exploiting different propagation paths through the environment.

The input (output) of a transmit (receive) antenna array system is one or more signals of interest along with each signal's desired spatial pattern. The key components of the array are the actual antennas themselves (and any front-end electronics as required by the application) along with a signal processing engine that converts between the signals of interest and the transmit or receive signals at each antenna. For example in a phased-array radar, a large network of phase shifters is used to transmit or receive in a specific direction.

There are a large number of spatial signal processing algorithms, ranging from phase shifts in phased arrays to maximum likelihood detection in communication systems. This chapter briefly overviews existing spatial processing techniques and concludes with a performance study that compares the performance of various algorithms in real channels.

2.1 Phased Arrays

Phased arrays were initially used in World War II [66] but received sustained attention starting in the late 1950s and early 1960s [67] as the next-generation of military radars [68]. Radars require the use of directional radiation to precisely localize targets in space. Early radars used directional antennas, such as parabolic dishes, which were mechanically steered to find and track their targets [69]. However, mechanically steered antennas suffer from slow steering, inability to track multiple targets, and mechanical stress and failures. This motivated the search for more flexible and reliable radar systems.

An antenna array is a virtual directional antenna which is electronically steerable. As shown in Figure 2.1(a), phased arrays operate by applying a per-element phase shift to each antenna's signal. The phase shift is a function of the desired look direction, and ensures that

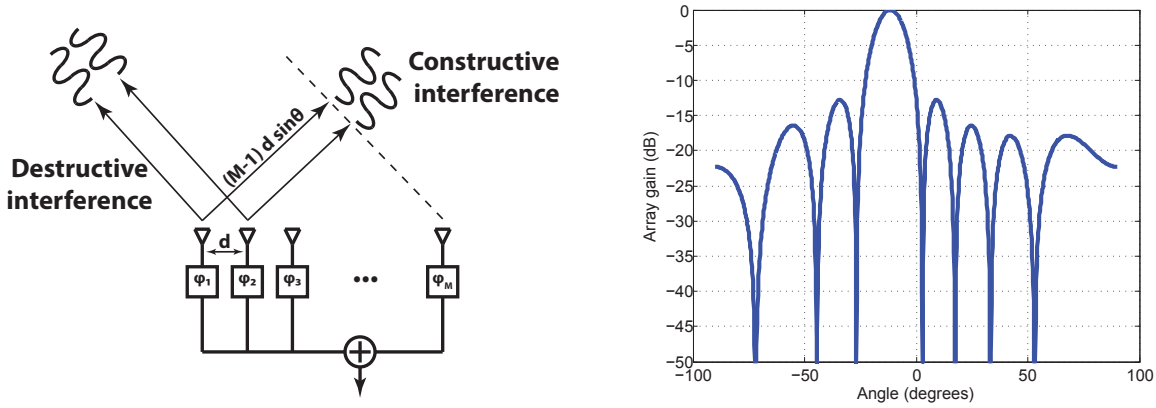


Figure 2.1: Phased array principle of operation, and example beampattern.

signals in that direction add constructively, while signals from all other directions add destructively. In this way, the look direction of the array can be steered simply by changing the phase shifts, which requires no mechanical components and can occur on the time constant of the phase shifter setting. Additionally, multiple beams can be formed simultaneously by implementing parallel phase shifter networks, one for each desired beam. This operation is referred to as "beamforming".

An example of an array's spatial response is shown in Figure 2.1(b), showing the spatial processing gain as a function of direction arrival or departure. This "array pattern" is described in terms of three key parameters. The look direction refers to the direction in which the array is pointed. The main lobe width measures the angular width of the array response in that direction. Finally, the sidelobe level describes the array gain at angles other than the look direction of the array.

The array pattern can be expressed as a function of the beamforming coefficients:

$$G(\theta, \phi) = \sum_{i=0}^{M-1} w_i e^{j\boldsymbol{k}\cdot\boldsymbol{d}_i} \tag{2.1}$$

where $\boldsymbol{k}$ is the wave-vector of the desired look direction and $\boldsymbol{d_i}$ is the position of a given element in the array. For a planar 2D array this simplifies to:

$$G(\theta, \phi) = \sum_{i=0}^{M-1} w_i e^{jk(x_i \sin\theta\cos\phi + y_i \sin\theta\sin\phi)} \tag{2.2}$$

where θ and ϕ are the azimuth and elevation angles of arrival relative to the array's normal vector. To point in direction (θ, ϕ), the phase at position (x_i, y_i) should be $-k(x_i \sin\theta\cos\phi + y_i \sin\theta\sin\phi)$.

This equation describes a discrete Fourier transform (DFT) of the beamforming coefficients, where the original variable is position $\boldsymbol{d}$ and the transform variable is wave-vector $\boldsymbol{k}$.

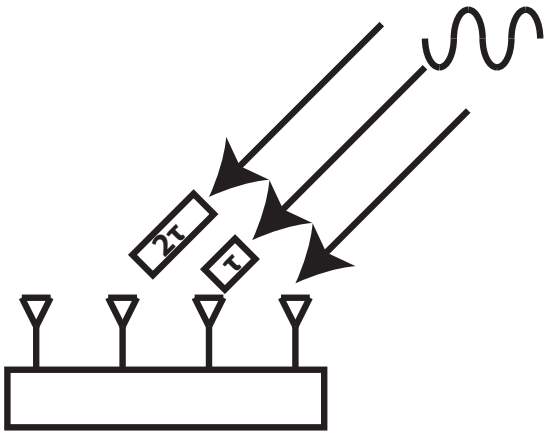


Figure 2.2: True time delay effects in phased arrays.

This gives two key insights. First, if $|w_i| = 1$ (a pure phase shift), then the array pattern is a shifted sinc function (defined as $\sin(x)/x$). Second, the sidelobes and null positions can be controlled by applying non-uniform amplitudes to w_i. For example, standard DFT windows such as Kaiser or Hamming [70] could be applied to achieve a certain array pattern.

Beamforming as described above uses only phase shifts to direct the beam, which is appropriate for narrowband signals or small arrays. A more complete solution uses *time delays*. The spatial selectivity of an antenna array is fundamentally a time-domain phenomenon: waves arriving from or departing to a specific direction experience a direction-dependent propagation delay across the array aperture (Figure 2.2). For narrowband signals, this propagation delay can be approximated as a phase shift only. However if the propagation delay across the array is comparable to the signal bandwidth, this approximation is not valid and instead actual time delays must be applied — so-called "true time delay (TTD) beamforming". The cutoff between phased array and timed array regimes can be expressed in terms of the ratio

$$\frac{Asin(\theta)}{c}B = Nd\sin(\theta)\frac{B}{f_c} \tag{2.3}$$

where A is the array dimension, consisting of N elements with spacing d (in wavelengths), θ is the direction of arrival relative to broadside, c is the speed of light, f_c is the carrier frequency, and B is the signal bandwidth. The cutoff between these regimes depends on the acceptable levels of beam squinting or inter-symbol interference.

In summary, the traditional paradigm of beamforming in phased arrays provides a flexible and powerful set of signal processing techniques for synthesizing desired spatial responses in line of sight environments. Once the desired look angle and sidelobe levels are known, phase, amplitude, and time weights can be quickly calculated to synthesize the desired array response.

2.2 Spatial Processing for Communications

The theory of phased arrays was developed largely for radar systems operating in free space, utilizing a fairly general class of waveforms. Compared to radars, directional communication systems present two differences. First, propagation environments are generally more complex, with rich multipath, fading, and shadowing effects. Second, the signals of interest always consist of a bandpass pseudo-random modulation on top of an RF carrier. As such, much of the intuition and signal processing techniques from phased arrays can be augmented by communication-specific assumptions.

2.2.1 Channel Matrices and Channel Estimation

To handle the range of complicated propagation environments observed in communication systems, it is necessary to develop a thorough description of the spatial environment, or channel. This initial description will focus on narrowband signals (where the multipath delay and the propagation time of a wavefront across the array aperture are small relative to the symbol period). Wideband channels are described in depth below in Section 2.3. Because of the bandpass nature of wireless communications, the channel can be described by a baseband equivalent vector of complex gains, $\boldsymbol{h}$ [71]. For propagation from a single source to an array with M elements, $\boldsymbol{h}$ will have dimension $M \times 1$. For example, for a uniform linear array (ULA) with direction of arrival θ and inter-element spacing d,

$$\boldsymbol{h} = \begin{bmatrix} 1 & e^{jkd\sin\theta} & e^{j2kd\sin\theta} & \dots & e^{j(M-1)kd\sin\theta} \end{bmatrix}^T. \tag{2.4}$$

This line-of-sight (LOS) channel is in fact exactly the environment considered above in the context of radars. However this formulation can also describe environments with more complicated propagation characteristics, such as multipath, diffraction, and shadowing, in terms of the overall complex propagation gain. The proagation environment from the array to the desired signal target can be similarly described by a transmit channel vector. If the transmit and receive frequencies are the same (as in, for example, time-division duplex — TDD — communications), the propagation is reciprocal[1] and the transmit channel vector is simply $\boldsymbol{h}^T$.

With multiple signals of interest, this description can be extended to an $M \times K$ channel matrix $\boldsymbol{H}$. Let $\boldsymbol{y}$ be the $M \times 1$ vector of signals at every antenna and $\boldsymbol{s}$ be the $K \times 1$ vector of desired signals. In receive mode, the array receives

$$\boldsymbol{y}_{rx} = \boldsymbol{H}\boldsymbol{s}_{rx} \tag{2.5}$$

In transmit mode, the K targets receive

$$\hat{\boldsymbol{s}}_{tx} = \boldsymbol{H}^T\boldsymbol{y}_{tx} \tag{2.6}$$

[1]Transmit-receive reciprocity holds true only for the propagation environment. Since the analog front ends are subject to random gain and phase errors, they contribute a non-reciprocal portion to the channel. Generally this is compensated by an appropriate calibration algorithm; see [53] for an algorithm and implementation results.

Modern wireless standard use coherent detection, meaning that the receiver estimates the channel gain and uses that knowledge to recover the amplitude and phase of the transmitted waveform. This estimation is accomplished by transmitting known pilots over the air with sufficient time and frequency resolution to sample the channel response [2]. The receiver then uses the known pilots to estimate the channel's transfer function. As a result, it is generally safe to assume that the receiver has some channel state information (CSI). In contrast, for the transmitter to have any knowledge of the channel it must be fed back from the receiver. To avoid this overhead, generally only very coarse information is fed back — as a result, the TX usually does not have very accurate CSI.

In single-input, single-output (SISO) links, the channel estimate is used mainly to measure the signal-to-noise ratio (SNR) and establish a constellation reference for demodulation. In single-input, multiple-output (SIMO) or multiple-input, multiple-output (MIMO) channels the CSI is also used to configure the spatial signal processing algorithm. In the remainder of this thesis, except where noted, we ignore channel estimation and write expressions in terms of the true channel matrix $\boldsymbol{H}$.

2.2.2 Single-user and Multi-user MIMO

MIMO techniques have been widely used in communication standards for over a decade. The key idea, discovered in the late 1990s, is that in a multi-antenna channel (with sufficiently rich propagation), the Shannon capacity is proportional to the minimum of the number of transmitters or receivers. In this context, "sufficiently rich propagation" means that there are at least as many propagation paths as antennas in order to support these spatial streams. The main attraction of MIMO deployments lies in engineering the channel capacity by adding more antennas.

Single-user (SU) MIMO attempts to send multiple spatial streams to a single user. As long as the environment exhibits rich scattering, it is possible to increase this user's experienced data rate. In indoor channels some SU-MIMO gains are common but many outdoor environments only have one dominant propagation path and therefore do not exhibit very significant SU-MIMO gains. Multi-user (MU) MIMO extends the SU-MIMO scenario by sending multiple data streams simultaneously to different users. Mu-MIMO can overcome propagation challenges of SU-MIMO: the natural spatial diversity of different users means that the channels are almost always sufficiently decorrelated. As a result, MU-MIMO offers a more promising avenue to realizing spatial multiplexing gains.

In SU-MIMO systems, the spatial signal processing may be carried out either at the RX side, TX side, or both, depending where CSI is available. When CSI is available at both sides of the link, the singular value decomposition (SVD) implements the optimal TX/RX processing. More commonly, CSI is available only at the RX side — in this scenario, all the processing has to be done at the receiver (Figure 2.3). A number of techniques have been developed for RX-side MIMO processing, which are described in the subsequent sections.

MU-MIMO differs in one important way from SU-MIMO: it is generally not possible for the users to collaborate in spatial processing. This means that the base-station must

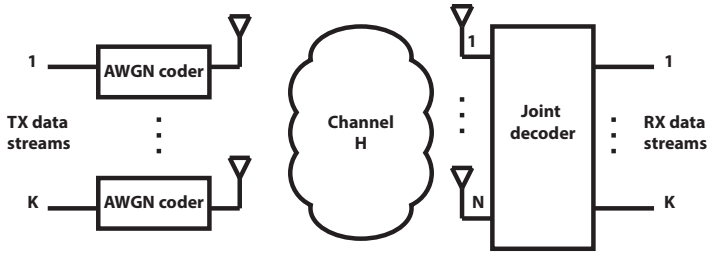


Figure 2.3: SU or MU MIMO scenario with only RX CSI. All spatial processing must be performed at the receiver.

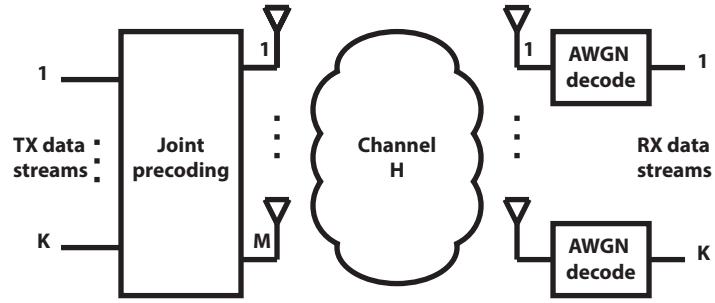


Figure 2.4: Downlink MU-MIMO, where all spatial processing must be performed at the base-station.

perform all spatial signal processing [72, 73]. In the uplink, this is equivalent to the SU-MIMO case with only RX-side processing so the same techniques can be adopted. However, in the MU-MIMO downlink, the base-station *must* acquire CSI and perform *all* the spatial processing (Figure 2.4). This is significantly more challenging for two reasons. First, fewer signal processing techniques have been developed for this scenario. Second, acquiring CSI at the transmitter can be challenging. One option is to perform downlink channel estimation and feed back the RX CSI. Another option is to exploit TDD uplink-downlink reciprocity. If the user transmits uplink pilots *using the downlink frequency channel*, the base-station can estimate the channel transfer function and use that estimate to perform downlink spatial processing.

2.2.3 Linear Beamforming

As the name suggests, linear beamforming is exactly the same as phased array beamforming described above, generalized to handle more complex environments. Instead of computing the beamforming coefficients based on the desired look direction, the channel matrix is used to explicitly compute the required amplitude and phase weights according to some objective

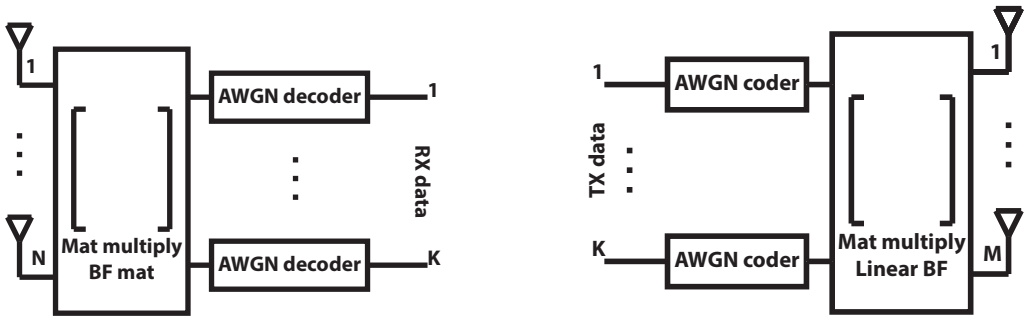


Figure 2.5: Linear beamforming at RX (left) and TX (right).

function. Applying the beamforming weights is accomplished with a matrix multiplication, as illustrated in Figure 2.5. Let $\boldsymbol{y}$ be the $M \times 1$ vector of signals at every antenna and $\boldsymbol{s}$ be the $K \times 1$ vector of desired signals. In the transmit direction, $M \times K$ transmit beamforming matrix $\boldsymbol{G}_{tx}$ is used to compute the output voltage at every antenna:

$$\boldsymbol{y}_{tx} = \boldsymbol{G}_{tx}\boldsymbol{s}_{tx} \tag{2.7}$$

In the receive direction, signals are reconstructed using the $K \times M$ receive beamforming matrix $\boldsymbol{G}_{rx}$:

$$\hat{\boldsymbol{s}}_{rx} = \boldsymbol{G}_{rx}\boldsymbol{y}_{rx} \tag{2.8}$$

As described below, $\boldsymbol{G}_{rx}$ and $\boldsymbol{G}_{tx}$ are constructed based on the estimated channel matrix $\boldsymbol{H}$ according to various objective functions.

Conjugate Beamforming

A natural objective is to maximize the beamformed signal energy. In transmit operation, let the signal received at the targets be the $K \times 1$ vector $\boldsymbol{x}$. Then,

$$\boldsymbol{x} = \boldsymbol{H}^T\boldsymbol{G}_{tx}\boldsymbol{s} + \boldsymbol{n}_K \tag{2.9}$$

where $\boldsymbol{n}_K$ is the $K \times 1$ vector of white Gaussian noise at each target location. Maximizing the received signal energy consists of finding the columns of the beamforming matrix $\boldsymbol{G}_{tx}$ which maximizes the trace of $\boldsymbol{H}^T\boldsymbol{G}_{tx}$.

Because there is no constraint on inter-stream interaction, each column of $\boldsymbol{G}_{tx}$ can be found independently [74, 75]. Since the dot product between two vectors is maximized when they are complex conjugates, the transmit beamforming matrix which maximizes the received signal strength[2] is

$$\boldsymbol{G}_{tx,conj} = \boldsymbol{H}^* \tag{2.10}$$

The same analysis can be performed for receive operation, giving

$$\boldsymbol{G}_{rx,conj} = \boldsymbol{H}^H \tag{2.11}$$

Here the superscript ^{H} refers to Hermitian (conjugate) transpose. This algorithm is called conjugate beamforming or maximum ratio combining (MRC) and the resulting matrix is the spatial matched filter to the channel. Note that if conjugate beamforming is applied to the LOS channel in (2.4), the result is precisely the phased array coefficients from (2.2).

Conjugate beamforming does not consider inter-stream interference at all. As a result, this technique performs very well in low SNR regimes (where there is no appreciable interference), but its performance in interference-dominated, high-SNR links is poor.

Zero-forcing

In interference-limited channels, it is natural to seek a beamformer which eliminates inter-stream interference [76]. In the transmit direction, this constraint can be framed as:

$$\begin{aligned} \boldsymbol{H}^T \boldsymbol{G}_{tx} \boldsymbol{s} &= \boldsymbol{s} \\ \boldsymbol{H}^T \boldsymbol{G}_{tx} &= \boldsymbol{I}_K \end{aligned} \tag{2.12}$$

where $\boldsymbol{I}_K$ is the $K \times K$ identity matrix. This over-determined system of equations can be solved in the least-squares sense using the right pseudo-inverse of the channel matrix:

$$\boldsymbol{G}_{tx,zf} = \boldsymbol{H}^* (\boldsymbol{H}^T \boldsymbol{H}^*)^{-1} \tag{2.13}$$

Similarly, in the receive direction solving the dual problem gives the receive beamforming matrix:

$$\boldsymbol{G}_{rx,zf} = (\boldsymbol{H}^H \boldsymbol{H})^{-1} \boldsymbol{H}^H \tag{2.14}$$

This beamformer is referred to as zero-forcing (ZF) or null-forcing, since it eliminates inter-stream interference.

Zero-forcing can be extended to cancel additional interferers. An extended $M \times (K+V)$ channel matrix $\boldsymbol{H}_e$ incorporates V additional interferers. The spatial signatures of these interferers are estimated and the extended channel matrix is formed by concatenating these to the desired channels H. Then the zero-forcing solution is given by:

$$\begin{aligned} \boldsymbol{G}_{tx} &= \boldsymbol{H}^* (\boldsymbol{H}_e^T \boldsymbol{H}_e^*)^{-1} \\ \boldsymbol{G}_{rx} &= (\boldsymbol{H}_e^H \boldsymbol{H}_e)^{-1} \boldsymbol{H}^H \end{aligned} \tag{2.15}$$

The extended ZF beamformer can null out external interference sources up to a total of $M-1$ interferers.

The ZF beamformer performs well in high SNR channels. However, since there is no constraint on the signal energy, it is possible that the interference is eliminated at the cost of significantly lowering the signal strength. As a result, the ZF beamformer performs poorly in low SNR conditions when the inter-stream interference is overwhelmed by noise.

[2]Transmit beamforming optimization problems must always be accompanied by a power constraint, otherwise the trivial solution is to let output power be infinite. For simplicity and to clearly bring out the key ideas, this constraint is not explicitly included in this section.

MMSE Beamforming

Since conjugate and ZF beamforming are suited to different SNR regimes, is there a linear beamformer which can optimally trade off between the noise and inter-stream interference? Such an algorithm is the minimum mean-squared error (MMSE) beamformer[3].

The MMSE beamformer minimizes the total mean-squared error from both noise and interference sources. In receive mode, this problem can be framed as:

$$\hat{\boldsymbol{G}}_{rx} = \arg\min_{\boldsymbol{G}_{rx}} \mathbb{E}[|\boldsymbol{s} - \boldsymbol{G}_{rx}(\boldsymbol{H}\boldsymbol{s} + \boldsymbol{n}_M)|^2] \tag{2.16}$$

where $\mathbb{E}$ denoted expectation over all transmit sequences $\boldsymbol{s}$ and noise vectors $\boldsymbol{n}$.

This equation can be solved by applying the orthogonality conditions, which state that:

$$\mathbb{E}[(\boldsymbol{s} - \hat{\boldsymbol{s}})\boldsymbol{y}_{rx}^H] = 0 \tag{2.17}$$

Plugging in for these quantities and simplifying, we obtain the result [77]:

$$\begin{aligned} \boldsymbol{G}_{rx,mmse} &= \boldsymbol{H}^H(\boldsymbol{H}\boldsymbol{H}^H + \sigma^2\boldsymbol{I}_M)^{-1} \\ &= (\boldsymbol{H}^H\boldsymbol{H} + \sigma^2\boldsymbol{I}_K)^{-1}\boldsymbol{H}^H \end{aligned} \tag{2.18}$$

where σ^2 is the variance of the white noise at each array element, and the last step relies on the matrix inversion lemma. Similarly, the transmit MMSE beamformer is

$$\boldsymbol{G}_{tx,mmse} = \boldsymbol{H}^*(\boldsymbol{H}^T\boldsymbol{H}^* + \sigma^2\boldsymbol{I}_K)^{-1} \tag{2.19}$$

where here σ^2 is the variance of the white noise at each target.

The MMSE beamformer works well in all SNR conditions because it computes the full correlation matrix, incorporating both noise and interference contributions. In high SNR conditions, the noise term can be neglected and the MMSE result approaches the ZF beamformer; in low SNR conditions, the noise term dominates and the MMSE result converges to the conjugate beamformer.

As in the ZF case, additional interferers can be included. Consider the receive direction corrupted by interference $\boldsymbol{q}$ with known (or estimatable) spatial signature:

$$\boldsymbol{y}_{rx} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{q} + \boldsymbol{n}_M \tag{2.20}$$

Following the same procedure as above and denoting the correlation matrix of the interference as $\boldsymbol{Q} = \mathbb{E}[\boldsymbol{q}\boldsymbol{q}^H]$,

$$\boldsymbol{G}_{rx,mmse} = \boldsymbol{H}^H(\boldsymbol{H}\boldsymbol{H}^H + \boldsymbol{Q} + \sigma^2\boldsymbol{I}_M)^{-1} \tag{2.21}$$

This expression can also be simplified using the matrix inversion lemma into a form similar to (2.18).

Figure 2.6 compares the conjugate, ZF, and MMSE array patterns for user 1 of an 8x2 MIMO LOS channel. The ZF and MMSE techniques cancel user 2's interference at the expense of a wider mainlobe and increased sidelobe levels.

[3]https://www.youtube.com/watch?v=rLDgQg6bq7o

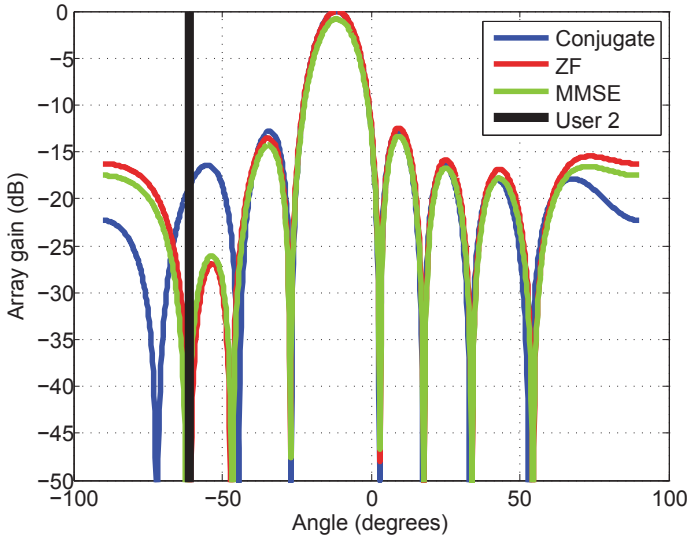


Figure 2.6: Comparison of conjugate, zero-forcing, and MMSE beampatterns for an 8-element array serving 2 users.

2.2.4 Maximum Likelihood Detection, Sphere Decoding, and K-Best

Linear beamformers are agnostic to the statistics of the underlying signals of interest, and as such are applicable in many scenarios such as radars, imagers, and communication systems. In contrast, a set of more robust and higher performance signal processing techniques have been designed for communication-specific scenarios.

In a communication system, optimal performance is obtained by exploiting knowledge of the transmit signal alphabet. The optimal receiver is the maximum a posteriori (MAP) algorithm, which uses the received signal to determine which transmit signal was most likely to have been sent. For equi-probable transmit alphabets, this is equivalent to the maximum likelihood (ML) receiver:

$$\hat{\boldsymbol{s}}_{rx} = \arg\max_{\boldsymbol{s}_{tx}} P(\boldsymbol{y}_{rx}|\boldsymbol{H}) \tag{2.22}$$

ML detection searches for the transmitted symbols which maximizes the probability of observing the received sequences. The performance is improved compared to a linear detection scheme since it exploits information about the the possible transmitted symbol alphabet. The ML receiver is provably optimal for all communication links (with equi-probably transmit symbols), meaning that it can achieve the full sum capacity of the MU channel. However, there are no general solutions to (2.22). As a result, the ML receiver must search through all possible transmitted sequences in order to find the one which best explains the observations. This has computational complexity which is exponential in the number of spatial streams, K.

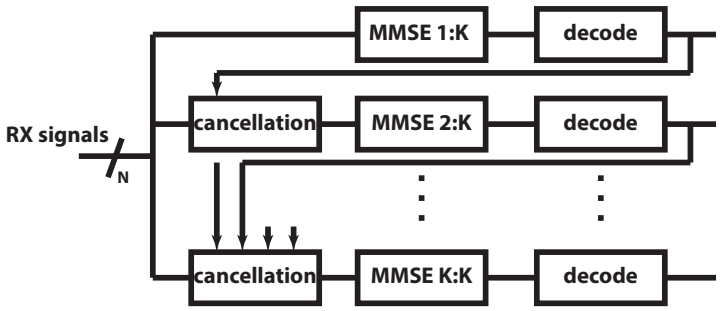


Figure 2.7: MMSE successive interference cancellation (SIC) receiver.

To overcome this bottleneck, (2.22) can be relaxed to a linear problem:

$$\hat{\boldsymbol{s}}_{rx} = \arg\min_{\boldsymbol{s}_{tx}} |\boldsymbol{y}_{rx} - \boldsymbol{H}\boldsymbol{s}_{tx}|^2 \tag{2.23}$$

This relaxation gives rise to a number of simplified approximate ML algorithms which approach the ML performance while reducing the complexity significantly. The two most common such algorithms are sphere-decoding and K-best decoding [78–80]. Both of these achieve near-ML performance with computational complexity which is approximately cubic in the number of signals.

2.2.5 Successive Interference Cancellation

Another popular receiver algorithm for MIMO channels is successive interference cancellation (SIC). The key idea is to iteratively process a single user, compute that user's contribution to the receive signal at each antenna, and subtract it out [25, 81, 82]. In this way, subsequent users experience reduced interference. This technique is essentially a spatial, multi-user decision feedback equalizer (DFE).

In more detail, MMSE-SIC processing proceeds as follows (Figure 2.7). First, the users are ordered by signal strength. The strongest user is spatially processed using the full MMSE beamforming matrix. Then, that user's signal is demodulated and the estimated contribution to the signal at every antenna is subtracted out. Then, the next strongest user is spatially processed using the *reduced* MMSE beamformer for users 2 through K, and so on.

Like the ML receiver, MMSE-SIC is theoretically optimal. As with all other decision-feedback receivers, however, it can suffer from error propagation — if there are any errors in demodulation then feeding back incorrect data will introduce new interference which degrades the overall system accuracy. As such, good performance is only practically achieved in strong channel conditions; for weaker channels, sphere decoding performs better.

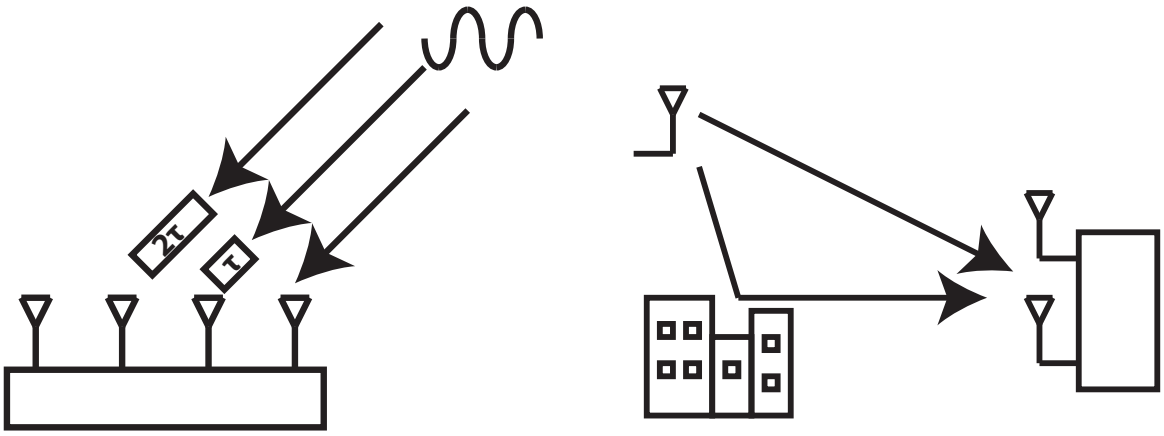


Figure 2.8: True time delay (left) and multipath (right) channels, showing how time and space are inextricably linked.

2.2.6 Dirty Paper Coding

The preceding sections have described a number of receiver algorithms suitable for the SU-MIMO downlink/uplink and MU-MIMO uplink. However downlink precoding for the MU-MIMO channel is much more challenging. Transmit beamforming as in Section 2.2.3 is one option. Are there any other feasible options?

In general transmit precoding must *pre-cancel* the inter-user interference which the transmitter can predict. This scenario is exactly captured by the information theoretical result known as dirty paper coding (DPC), which states that if a channel is corrupted by interference *which is known perfectly at the transmitter*, the transmitter can code around the interference and achieve the capacity of the interference-free channel [83].

DPC itself is a theoretical result — it still requires codes to be invented for a particular interference scenario. One example is Tomlinson-Harashima precoding (THP), which implements a transmit-side DFE to cancel inter-symbol interference in SISO links [84, 85]. In MIMO situations, a suboptimal implementation of DPC for crosstalk cancellation in a DSL link was proposed in [86] using the QR decomposition. This technique was further expanded by [87–89] for the MU-MIMO channel. In practice, these schemes suffer from very high computational complexity due to the complex and noncausal joint processing of all the user's data streams. As a result, DPC techniques have not been used in wireless systems, and so modern standards use linear beamforming for the MU-MIMO downlink.

2.3 Linear Beamformers for Wideband Channels

The discussion so far has focused on MIMO processing in narrowband channels. In practice, real channels exhibit multipath propagation phenomena and other effects which are

not captured by this formulation (Figure 2.8). To capture frequency-selective propagation effects, the propagation environment between a transmit and receive antenna pair can be described by a finite impulse response (FIR) filter. Because of the bandpass nature of the communication link any continuous time impulse response can be sampled at the baseband rate to give the equivalent baseband (discrete-time) impulse response.

A SIMO or MISO link can be described as an $M \times 1$ vector FIR filter:

$$\boldsymbol{h}(t) = \sum_{l=0}^{L} \boldsymbol{h}^{(l)} \delta(t - \tau_l) \tag{2.24}$$

Here, the l'th tap arrives with spatial signature $\boldsymbol{h}^{(l)}$ and delay τ_l. This formulation easily captures true time delay effects or any other per-element fractional delays. Finally, the MIMO case is obtained by letting each channel vector become an $M \times K$ matrix and summing over all paths from all signal sources.

The channel vectors $\{\boldsymbol{h}^{(l)}\}$ describe very complex propagation effects, including the spatial signature of the arriving or departing signal, the reflection coefficients of scatterers, and the time of flight. Real wireless deployments are require extensive measurement campaigns which characterize the propagation effects and result in the development of statistical channel models. These statistical channel models provide a statistical formula for generating realistic and representative channels for various scenarios. For the purpose of this discussion on beamforming techniques, it suffices to work with an abstract channel model of the form (2.24), supplemented with some statistical models as appropriate (and where noted).

Multipath propagation environments give rise to the phenomenon of frequency-selective fading, which simply means that the channel gain depends on the frequency. This can be characterized by a parameter called the coherence bandwidth, which loosely speaking refers to the bandwidth over which the channel gain is largely unchanged. If a communication signal has bandwidth larger than the channel's coherence bandwidth, then this signal experiences frequency-selective fading and requires equalization. This naturally leads to the question of how to design spatial processing algorithms suitable for this frequency-dependent scenario. Reflecting the linked nature of time and space, such algorithms would perform spatio-temporal equalization.

2.3.1 Frequency-Flat Beamforming

The most naive strategy is to simply ignore the frequency-dependent behavior. Consider linear beamforming in a SIMO channel: the strongest channel tap is chosen as the cursor and used for conjugate beamforming. Without loss of generality we can assume the first tap, $\boldsymbol{h}^{(0)}$, to be the strongest. Then the effective single-input, single-output (SISO) channel after beamforming is:

$$h_{siso} = \boldsymbol{w}^H \boldsymbol{h}(t) = |\boldsymbol{h}^{(0)}|^2 \delta(t) + \sum_{l=1}^{L} (\boldsymbol{h}^{(0)} \cdot \boldsymbol{h}^{(l)}) \delta(t - \tau_l) \tag{2.25}$$

Using this technique, the multipath energy is completely neglected and the multipath components are spatially filtered only to the extent that their spatial signature happens to be orthogonal the cursor's. This same technique and intuition could be extended to the MIMO scenario, by collecting each user's cursor and beamforming only to that matrix.

Frequency-flat beamforming only collects the energy in the cursor, and can only separate out inter-user interference arising from the cursor paths. There are two main shortcomings. First, frequency-flat beamforming cannot process inter-user, inter-symbol interference since this shows up as frequency-dependent inter-user interference. Second, frequency-flat beamforming cannot collect all the multipath energy since it cannot matched filter all the spatial components of the signal.

Simple frequency-flat beamforming can be augmented by following it with a $K \times K$ MIMO frequency-domain equalizer (FDE). Frequency-flat beamforming collapses the $M \times K$ FIR channel into a $K \times K$ FIR channel — this can be equalized using a MIMO FDE. The MIMO FDE can solve the first shortcoming mentioned above because it can compensate frequency-dependent inter-user interference. However this technique serves only to equalize, not to collect multipath energy.

One could also propose frequency-flat analogs of the other spatial processing techniques such as ML or SIC. However, these techniques rely on precise channel state information to perform detection. In a frequency-dependent channel it is generally not appropriate to ignore the variation of the channel response with frequency while using these algorithms, since this results in unacceptable channel estimation error.

2.3.2 Frequency-Domain Beamforming

To fully process a channel's spatio-temporal response, *wideband* spatial processing techniques are needed. The standard approach in modern communication systems is to perform spatial processing in the frequency domain. Wideband processing is required because the spatial signature of the signal changes substantially as a function of frequency. If it were possible to divide this large signal bandwidth into many narrowband channels, then each of these sub-channels would experience frequency-flat fading and could be treated as a separate narrowband channel.

Fortunately, this decomposition into many narrowband channels is a key feature of modern modulation schemes such as OFDM and single-carrier OFDM (SC-OFDM) [90]. The exact implementation of these modulation schemes is not relevant, as long as the channel is separated into mutliple narrowband, orthogonal subcarriers. Under those conditions, the MIMO time-domain channel I/O relationship (2.24) is transformed into:

$$\boldsymbol{R}_k = \boldsymbol{H}_k \boldsymbol{X}_k + \boldsymbol{W}_k \quad \forall k \in [-\frac{N_{sc}}{2}, \frac{N_{sc}}{2} - 1] \tag{2.26}$$

for the receive mode (uplink) where $\boldsymbol{R}_k$ is $M \times 1$, $\boldsymbol{X}_k$ is $K \times 1$, and $\boldsymbol{H}_k$ is $M \times K$. Each subcarrier experiences frequency-flat fading, *fully* described by the subcarrier channel matrix $\boldsymbol{H}_k$. As a result, each subcarrier can be treated as a separate and independent channel.

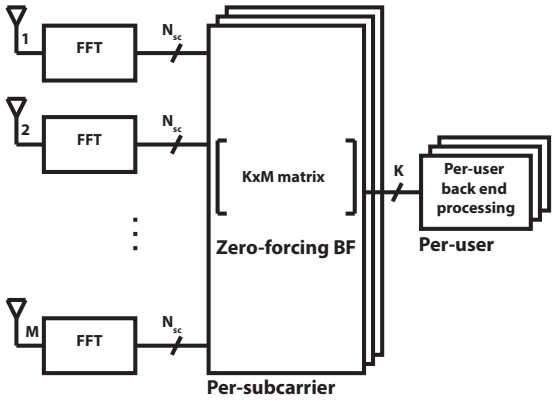


Figure 2.9: Block diagram of a receiver using full frequency-domain beamforming — a different $M \times K$ beamforming matrix is used for each subcarrier.

(Alternatively, a small subset of carriers is jointly processed, but each subset is treated independently.) The channel response is estimated independently for each subcarrier and spatial processing is applied to every subcarrier using any of the techniques in Section 2.2. More specifically, one could apply per-subcarrier ML, SIC, or beamforming spatial processing.

2.4 Case Study on Wideband Beamforming Performance

When should frequency-dependent techniques be used, and when do frequency-flat approximations suffice? These questions can be answered by simulating the performance of these two transceiver architectures in different channel conditions.

A block diagram of a frequency-domain beamformer ("full FDE") is shown in Figure 2.9. Each element is equipped with an FFT unit. Then, each subcarrier is independently processed, including beamforming and other baseband tasks. This receiver architecture is contrasted with the frequency-flat beamformer with MIMO FDE ("ZF-FDE") in Figure 2.10. Here, a single beamformer is applied in the time domain, each stream is transformed into the frequency domain, and a $K \times K$ MIMO FDE is applied. A third receiver option is to take a frequency-flat beamformer followed only by SISO FDEs ("Flat BF") (Figure 2.11). This receiver cannot process frequency-dependent inter-user interference.

Comparing these receivers, the fully frequency-dependent processor requires an FFT at each element, while the other two structures only require an FFT per user stream. Furthermore, the fully flat receiver simplifies the equalizer by using only K SISO FDEs instead of a $K \times K$ MIMO FDE.

It is desired to evaluate the relative performance of these three receiver structures in various channel conditions. This comparison can be carried out by generating a large number

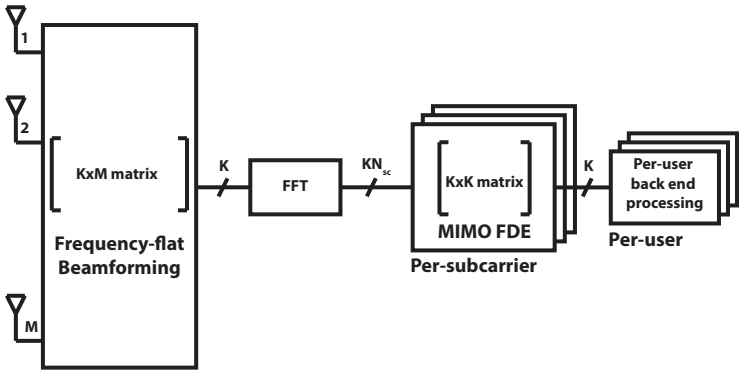


Figure 2.10: Block diagram of a receiver using frequency-flat beamforming with a MIMO FDE — a single $M \times K$ beamforming matrix is applied in the time domain followed by a $K \times K$ MIMO FDE.

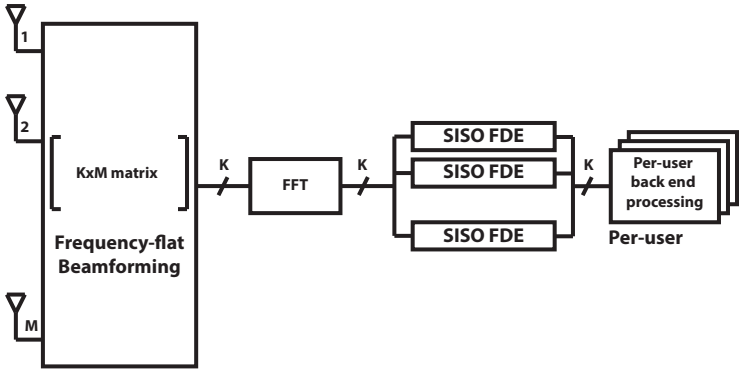


Figure 2.11: Block diagram of a receiver using frequency-flat beamforming with only per-user SISO equalization.

of channel instances drawn from a statistical channel model. For each one of these channel instances, the SNR is swept over a range and the bit error rate (BER) of each receiver is simulated as a function of the SNR. Finally, the SNR required to achieve a target BER of 10^{-3} in that channel instance is extracted and recorded. Therefore, the overall characterization of the receiver is described in terms of a distribution of SNRs required to achieve a certain target BER.

The 802.11 standardization committee defines a family of statistical channel models describing multi-user MIMO operation in typical WLAN operating environments [91]. There are 6 such models, enumerated A-F, which describe progressively more multipath environments. Channels A-C reflect, respectively, line-of-sight, small office, and medium office environments [91].

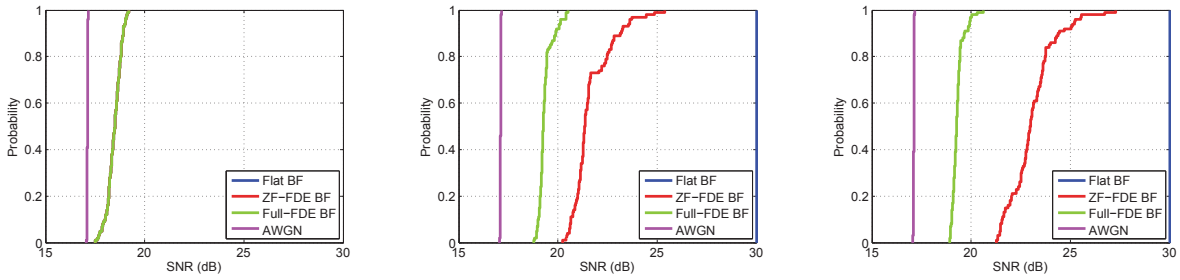


Figure 2.12: CDF of SNR required to achieve 10^{-3} BER in WLAN channel models A-C, with $M = 128$ and $K = 16$, using Full-FDE, ZF-FDE, and Flat-BF receivers. (a) Channel model A (line of sight). (b) Channel model B (small office). (c) Channel model C (medium office).

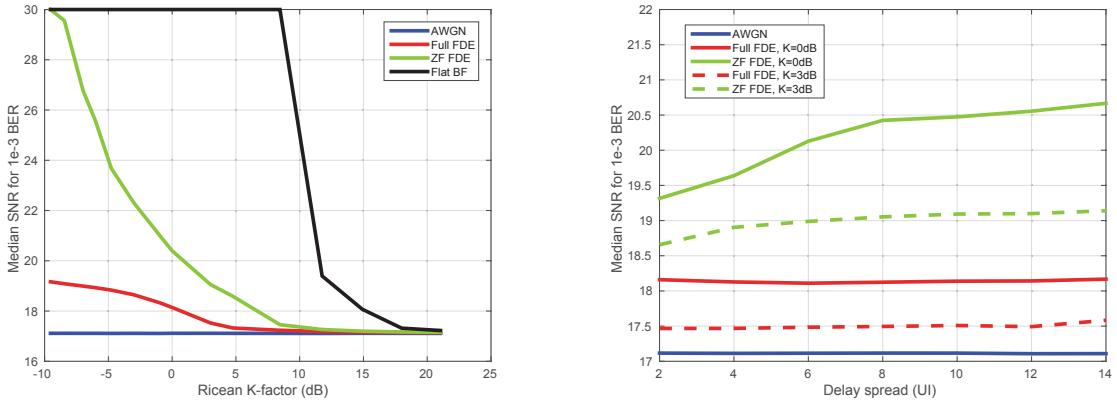


Figure 2.13: Median SNR required to obtain 1e-3 BER with $M = 128$ and $K = 16$. (a) Ricean K-factor is swept while delay spread is constant at 6. (b) Delay spread is swept for K-factor of 0dB or 3dB.

The three receiver structures were characterized according to the methodology described above for WLAN channel models A, B, and C, with 128-element arrays and 16 users. The results are shown in Figure 2.12 In channel A, since there is no multipath propagation, all three beamforming schemes are identical as expected. As more multipath energy is progressively added in channel B and then C, the performance gap between the three signal processing implementations increases. In particular, flat BF is completely unable to handle these environments. Meanwhile, full FDE beamforming provides a performance gain (in both median and 95th percentile cases) compared to only ZF-FDE. This reflects its greater ability to adapt to and compensate for multipath environments.

To understand more precisely the differences between these signal processing techniques, a more detailed and parametrized channel model is needed. Using the techniques in [92–95], we

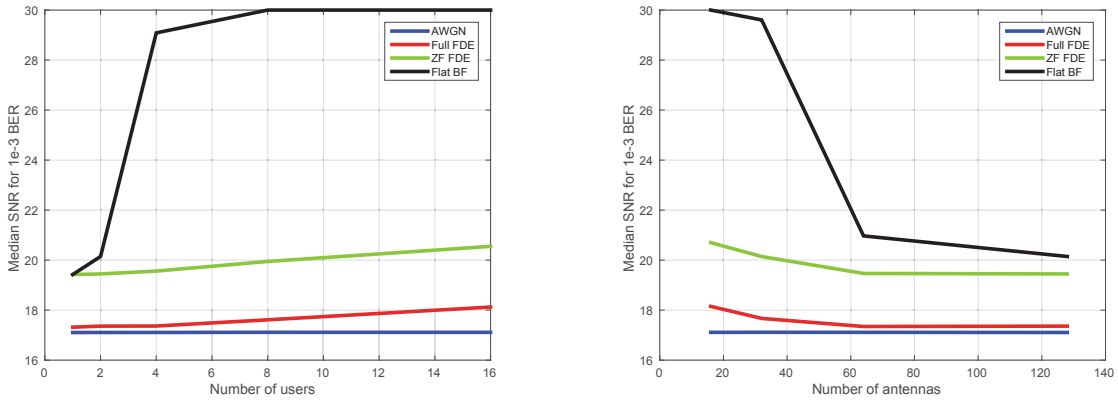


Figure 2.14: Median SNR required to obtain 1e-3 BER with 0dB K-factor and delay spread of 18. (a) Varying number of users, with $M = 128$. (b) Varying number of array antennas, with $K = 2$.

implemented a parametrized statistical channel model following the proposal of [96]. These channels describe a multipath environment where each path has some strength, delay, and angle-of-arrival which follow realistic distributions. The overall channel is parametrized by the Ricean K-factor, which describes the relative strength of the LOS and NLOS components, and the delay spread which measures the length of the channel impulse response. Using this model, it is possible to sweep the channel parameters and observe how the performance of the three signal processing techniques changes accordingly.

Figure 2.13 presents the performance of flat BF, ZF-FDE, and full FDE as a function of the Ricean K-factor and the delay spread. These results reveal a great deal about the behavior of these signal processing techniques. For strongly LOS channels, all three techniques are equivalent, as expected. For moderate LOS channels, with only a small amount of NLOS energy ($< 10\%$ or so of the total), ZF-FDE and full FDE have equivalent performance. In this regime, the main goal is merely to cancel inter-user interference from multipath propagation; there is no need to coherently combine multipath energy from all directions. For strongly NLOS channels (below 5dB K-factor), the performance of ZF-FDE drops off as well. In this regime, it becomes important to collect the energy traveling through the various paths in the environment. Since only the full FDE beamformer is able to do this, it experiences a significant SNR gain over the ZF-FDE case.

Figure 2.14 shows the performance of these techniques with various combinations of M and K. The ZF-FDE and full FDE schemes show relatively low sensitivity to the number of antennas or users, with only a fixed performance gap between the two techniques. In particular, for large M they clearly converge to a constant performance. In contrast, the flat BF scheme shows extreme sensitivity to the communication system parameters. To expand on this, Figure 2.15 sweeps the number of users K, while keeping the ratio of M/K constant at 8. This plot reveals the fundamental difference between the three schemes. ZF-FDE and

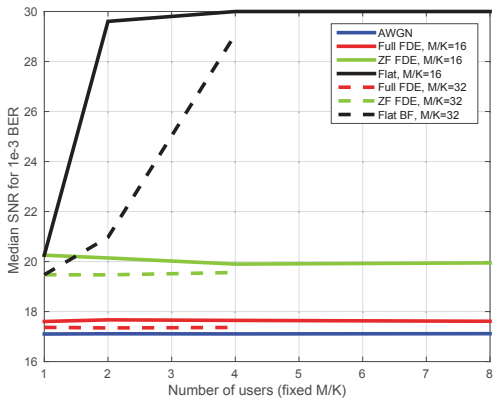


Figure 2.15: Median SNR required to obtain 1e-3 BER with M/K fixed at 8 and varying number of users.

full FDE maintain constant performance independent of K; their performance is just set by the propagation environment and the ratio M/K. In contrast, flat-BF requires not just a large M/K ratio, but an absolutely small *number of users*, regardless of any other parameter. The intuition for this is clear. Each user which is added to the system brings in a new set of propagation paths. As a result, as the number of users is increased, the inter-user interference channel becomes highly frequency dependent, which requires frequency-dependent ZF to manage.

These results can be summarized as follows. First, because flat-BF requires a small number of users, regardless of the ratio M/K, *flat-BF is NOT massive MIMO.* Second, ZF-FDE and full FDE beamforming are distinguished mainly by a performance gap in strongly NLOS environments. This performance gap is mainly a function of the Ricean K-factor; for a given K-factor it is largely independent of delay spread, M, or K. When operating in heavily NLOS regime, it is advantageous to use full FDE processing. When it is known that the channel is more favorable, the processing complexity can be safely relaxed.

2.5 Summary

This chapter has reviewed existing techniques in spatial processing for both communication systems and other applications such as radars. A number of techniques were presented which will be analyzed in subsequent chapters for their suitability to next-generation communication systems. Finally, the performance of spatial processing in wideband MIMO channels was studied. For the massive MIMO limit to be achieved, at the very minimum a frequency-dependent zero-forcing stage is required. In strongly scattering channels, frequency-dependent conjugate beamforming may also confer an SNR advantage by collecting more multipath energy.

Chapter 3

Array Architecture for the Massive Regime

The previous chapter described a number of spatial processing algorithms and techniques, drawing from a broad set of applications. Now focusing specifically on next-generation wireless networks, two important constraints emerge. First, it is desired to maximize the capacity gain from spatial processing, which naturally pushes to increase array sizes in both number of antennas and number of users. Second, the cost and power must be kept as low as possible, since it is desired to deploy wireless infrastructure in a very dense manner.

How do we achieve these objectives? This is fundamentally a question of the system-level design of the array. In the communications arena, arrays at both traditional cellular and mm-wave frequencies have used only a small number of elements and processed a handful of simultaneous spatial streams. These arrays are small enough that often the entire system can be implemented on a single chip. At worst, multiple subarrays are combined in the analog domain on package to form a small number of beams [46]. This dramatically simplifies the tradeoffs in array design and hides many of the challenges that emerge when processing a huge number of antennas or forming a huge number of beams.

Very large arrays are found in the military or aerospace sectors. In fact, many useful insights can be gleaned by studying these systems [97–100]. However, these systems are much less cost-sensitive than commercial arrays (and face other unique design challenges), so the parallels are limited.

In the end, since we are proposing to design complex array systems, the dominant engineering effort and innovations must be at the system-level. If the entire array architecture is carefully designed to be scalable, low-power, and low-cost, then each individual piece may be designed optimally within that framework. This architecture is the organizing principle and strategy which guides the implementation of the entire complex system.

Since spatial processing is the core task of an array, it forms the backbone of the architecture and the implementation. This chapter considers how to develop system-level architectures and spatial processing algorithms specifically for the large-array regime. We first analyze in detail why linear beamforming is the preferred spatial processing algorithm.

Next, we propose a beamforming-aware array architecture which permits a modular and scalable implementation of many-antenna arrays. Finally, we present a novel time-domain beamforming technique.

3.1 Spatial Processing for the Large Array Regime

Any specific array instance will have a fixed number of antennas and serve a maximum number of users, where these parameters are chosen at design time. However it is desired to devise an array architecture which scales well to any number of elements and users. Such an architecture provides a framework for designing and implementing a broad range of arrays, suitable for a variety of deployment scenarios. For example, we could use the same insights and system architecture to implement a very large macro cell array, a medium distance urban micro-cell, and an indoor mm-wave array.

To enable this vision, the scalability of signal processing algorithms and array architectures with both M and K is an important consideration when selecting a spatial processing technique for such systems. In the downlink direction, linear beamforming is the only feasible transmitter architecture since efficient DPC techniques have not been developed [72]. In contrast, there are several receiver structures which could be applied to the MU-MIMO uplink. This section will focus on comparing their suitability for large arrays.

3.1.1 Computational Complexity

An important consideration is the computational complexity of the detection algorithm itself. ML or near-ML detection (such as sphere decoding) is commonly used in today's wireless systems. A significant drawback of these techniques is their high computational complexity. The computational complexity of true ML detection is V^K, where V is the constellation order. This exponential dependence on K makes ML detection infeasible for almost all applications. Near-ML techniques such as sphere decoding can provide reduced computational complexity which scales approximately as K^3. In practice this is feasible for current MIMO orders (2-4 spatial streams), but will prove to be burdensome if K is increased significantly.

In contrast, both linear beamforming and SIC essentially consist of a matrix multiplication, with computational complexity MK. SIC has higher complexity than linear beamforming, but the scaling with M and K is the same. We can conclude that these techniques most easily scalable to large M and K.

3.1.2 Performance Comparison of Linear and Nonlinear schemes

In traditional low order SU-MIMO or MU-MIMO systems, where the number of transmitted streams is similar to the number of receive antennas, near-ML detection is generally preferred due to its much better performance. Marzetta [30] showed that in massive MU-MIMO systems, where $M \gg K$, linear beamforming techniques are asymptotically optimal and can

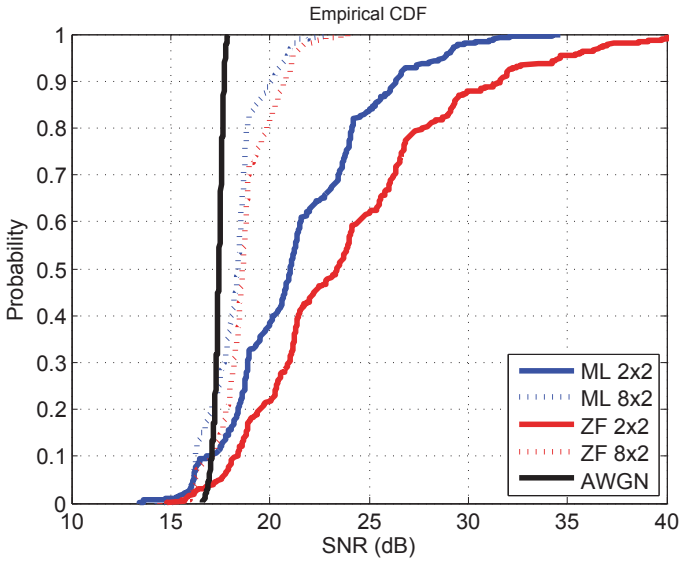


Figure 3.1: Cumulative density function of SNR required to achieve 1e-3 bit error rate (BER) in 2 different MIMO Rayleigh channels with ML and zero-forcing detection.

achieve the capacity of the channel. This is a very promising result because it suggests that it is sufficient to use simple linear beamforming.

Figure 3.1 shows the performance gap between ML processing and the linear ZF receiver for a 2x2 and 8x2 MIMO scenario (2 streams and either 2 or 8 receive antennas). 1000 random instantiations of a Rayleigh channel are simulated by varying the thermal SNR and measuring when the bit error rate (BER) meets the desired threshold of 1e-3. For 2x2 MIMO — a common configuration in LTE and 802.11n/ac — the ML technique yields a large (3-5dB) improvement in performance for these small number of receive antennas. In contrast, simply going to $M/K = 4$ significantly closes the gap between ML and linear detectors. Moreover, the performance approaches that of an additive white Gaussian noise (AWGN) channel, which indicates that inter-user interference becomes less significant.

Why is this important? For two streams this is not a very meaningful result. The computational complexity of sphere decoding and zero-forcing on a 2x2 MIMO channel are approximately the same. But in scaling to a large number of users the difference purely in computational complexity is very meaningful (not to mention related issues such as data movement around the array, discussed below). Suppose it is desired to serve 20 users. Zero-forcing on an 80x20 MIMO channel has about 5x lower computational burden than sphere-decoding on a 20x20 MIMO channel. As long as the extra radios in the 80-element array are not too expensive, it is easier to deploy more hardware and serve the users that way. Put another way, for the same computational cost as a 20x20 sphere decoding operation, it would be possible to implement a 200x50 zero-forcing, serving 2.5 times as many users!

3.1.3 Performance Comparison of Linear Beamformers

A key result in massive MIMO is that, under benign conditions, not only is linear beamforming optimal as M grows large, but *conjugate beamforming* is asymptotically optimal. This result implies that for a "large enough" array, there is no need to use any more complicated technique than conjugate beamforming.

This result holds true as long as the columns of the channel matrix are asymptotically uncorrelated. Essentially, as M grows large, it is desired that

$$\frac{1}{M}\boldsymbol{h}_i \cdot \boldsymbol{h}_j \rightarrow \delta_{ij} \tag{3.1}$$

where δ_{ij} is the Kroenecker delta. Under these conditions, the interference correlation matrix $\boldsymbol{R}_H = \frac{1}{M}\boldsymbol{H}^H\boldsymbol{H}$ approaches the identity matrix and

$$\lim_{M\rightarrow\infty}(\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H = \boldsymbol{H}^H \tag{3.2}$$

In other words, as the number of elements grows large, ZF and MMSE beamforming *converge* to conjugate beamforming. This result can be understood intuitively as follows: as the number of antennas grows larger, the array acquires finer and finer spatial resolution and can more easily distinguish different users.

Is this condition observed in practice? Numerous measurement campaigns have reported that practical environments do appear to exhibit so-called favorable propagation [101–106]. A large part of this comes from the multi-user nature of massive MIMO — just by having physically separate users, it is very likely that the channels are quite uncorrelated. This can be intuitively verified for two important cases.

Case 1: Line of sight propagation: In a line of sight environment, each user's channel vector is a so-called Vandermonde vector:

$$\boldsymbol{h}_i = [1 \quad e^{j\theta_i} \quad e^{j2\theta_i} \quad ... \quad e^{j(M-1)\theta_i}]^T \tag{3.3}$$

where $\theta_i = kdsin(\phi_i)$. As long as all the directions of arrival/departure ϕ_i are distinct, as M grows large the inner product between any two of these channel vectors will go to zero.

Case 2: Rayleigh channel: In a Rayleigh channel, each element of each channel vector is iid $\mathcal{CN}(0,1)$. It then follows that:

$$\lim_{M\rightarrow\infty}\frac{1}{M}\boldsymbol{h}_i \cdot \boldsymbol{h}_j = \lim_{M\rightarrow\infty}\frac{1}{M}\sum_{k=1}^{M} h_{ik}h_{jk}^* = \mathbb{E}[|h_{ij}|^2]\delta_{ij} \tag{3.4}$$

by the Strong Law of Large Numbers. This result shows that for extremely complex scattering environments, channel vectors are asymptotically orthogonal.

Given that favorable propagation is observed in practice, how quickly does conjugate beamforming approach the performance of ZF? In other words, what is a "large enough" value of M such that conjugate beamforming can safely be employed? Fig. 3.2a shows the

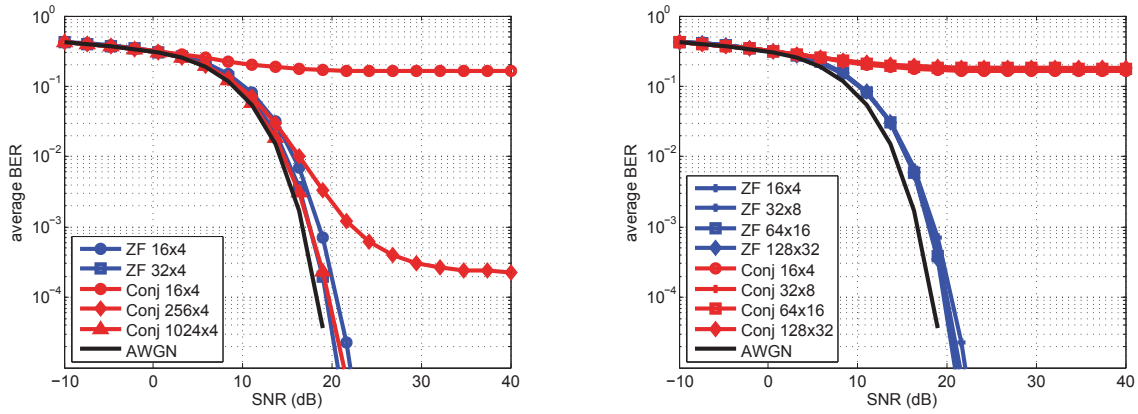


Figure 3.2: Average BER vs SNR for ZF and conjugate beamforming in a Rayleigh channel. (a) $K = 4$, with varying array size. (b). Varying number of users with $M/K = 4$.

average BER vs SNR for both conjugate and ZF beamforming as the number of streams is held constant and the number of array antennas is increased. First, it is clear that ZF always outperforms conjugate beamforming. Second, for M/K ratio of between 4 and 8, ZF beamforming achieves performance very close to an AWGN channel. In contrast, conjugate beamforming requires M/K ratio of 256 to achieve similar performance.

Fig. 3.2b shows a similar setup, where now the ratio of M/K is fixed at 4 and the number of elements in the array is increased. The BER performance is essentially unchanged for each scenario, indicating that the achievable capacity is related to the ratio of M/K rather than their absolute values. Note that since the BER is unchanged but the number of users is increasing, the total cell capacity is being increased constantly with K, as expected from the theory of massive MIMO. The key conclusion is that, for fixed M/K, as K is increased both ZF and conjugate beamforming exhibit capacity growth proportional to K with a fixed performance gap between the two schemes.

3.1.4 Summary

This analysis reveals three key facts. First, with an M/K ratio as low as 4, near-AWGN (interference-free) performance can be achieved using only linear ZF beamforming in Rayleigh and line-of-sight channels. Second, when $M/K = 4$, ML performance is only marginally better than ZF beamforming. Finally, ML or near-ML complexity is far higher than linear beamforming.

These results explain the attractiveness of massive MIMO for next-generation wireless technologies — very simple spatial processing can be used to achieve nearly interference-free multi-user communication, scalable to nearly arbitrary number of users.

3.2 A Scalable Beamforming-Aware Array Architecture

Linear beamforming is the best candidate for supporting aggressive spatial processing in next-generation wireless systems. But in order to practically deploy this capability in real networks, it is not sufficient that the algorithm have minimal complexity. It is equally necessary to ensure that the actual array — consisting of radios, data converters, signal processing, data interconnect, synchronization, etc — be easy to design, low-cost, and low-power. If these criteria are not met then it will not be possible to deploy large arrays in the field. This objective is addressed by devising an array architecture which implements the linear beamforming with low complexity, cost, and power.

In particular, a key goal is to design a modular and scalable array architecture which can be easily extended to support a large number of antenna elements and a large number of simultaneous beams. To this end, there are two fundamental questions that must be answered. First, how should the antennas be connected to the central processor? Second, how should the required hardware and signal processing functions be organized, ordered, and grouped?

Depending on the scenario, including carrier frequency and channel bandwidth, the actual circuits used in the implementation may differ widely. For example, in different scenarios it may be preferred to utilize all-digital beamforming or a hybrid analog/digital approach. Accordingly, another guiding principle in proposing an array architecture is to abstract away as much as possible the specific implementation details of the architecture — then, for example, the choice of how to implement the beamforming is simply an engineering design choice.

These design goals are addressed in this section by proposing an array architecture suitable for a large range of implementation goals and scenarios [107, 108]. The key conclusion is that by exploiting the natural parallelism of linear beamforming, large beamforming arrays can be readily mapped into an efficient, modular, and scalable hardware architecture.

3.2.1 State of the Art

Presently reported massive $<$ 6GHz arrays nearly exclusively use fully centralized array architectures [52–57, 59] to implement arrays in the 32-128 element range. Each antenna is connected to a full digital transceiver, spanning RF front end to data converters. The digital I/Q samples are sent over a high-capacity backplane to a central processor, which is generally implemented as one or more field-programmable gate arrays (FPGAs). This central processor runs the entire baseband processing stack for the full system. Synchronization and calibration loops are generally implemented in an analogous manner, with the central processor updating and configuring all the distributed radios.

In contrast to this fully centralized implementation, some distributed processing was proposed in [53]. That work proposed to use distributed conjugate beamforming (but not

zero-forcing) in some conditions. In order to do this, part of the baseband processing is located close to the radio, on a small FPGA. The interconnect only moves receive samples after beamforming (transmit samples before beamforming), and the data aggregation is embedded in the interconnect.

By situating all processing and synchronization at the central processor, these centralized arrays make the data interconnect and the processor the cost and power bottlenecks in the system. For example, the Lund University testbed's backplane needs over 450 Gbps of aggregate capacity [55]. Similarly, the Samsung prototype has 80% FPGA utilization (on a single high-end Xilinx Virtex 7-690T model) using only 20MHz bandwidth with 12 simultaneous users.

In practice, commercializations of massive MIMO technology today are overcoming these issues by brute force. Network infrastructure OEMs are designing baseband ASICs with sufficient I/O bandwidth and custom accelerators to process massive MIMO workloads. This approach can be made to work up to about 64 antennas. Beyond that, a more clever approach is needed.

Currently reported mm-wave arrays are even simpler. Almost all existing arrays integrate 4 to 16 or 32 elements on a single die, using analog beamforming to form a single beam [43, 60–65]. Some recent publications have proposed to co-package subarrays, with package- or board-level analog combining to form a small number of aggregate beams. IBM and Ericsson [46] report a single chip solution which implements two 16-element phased arrays (one per polarization) at 28GHz. Multiple such chips can be combined on package to form a 64-element array, which can be operated in two modes: as a 64-element 1-beam phased array or as 4 16-element phased arrays, each forming a different beam. Importantly, there is a strong tradeoff between the number of beams and the array size. Similarly, UCSD [45] has shown a 28GHz front-end IC with four elements; eight of these ICs are combined on board to form a 32-antenna single-beam array which operates with 300m range [44]. Finally, some multi-beamforming mm-wave arrays are beginning to emerge. The authors in [109] report a 60GHz front-end with two separate 4-element arrays, each forming a single beam. The two beams are digitally processed for interference nulling. North Carolina State University [110] has shown the ability to form multiple beams from a single subarray, with a 4-element 60GHz array which can form 2 simultaneous beams using analog beamforming. However the two beams cannot be independently steered; instead, the second beam is constrained to be reflected across broadside from the main beam.

To summarize, there has so far been little effort to develop a scalable, power- and cost-efficient, and frequency agnostic architecture for large antenna arrays. In low-frequency bands, massive MIMO has been demonstrated and is being commercialized largely through a brute force approach. This is made possible by the relatively narrow channel bandwidths, which make the I/O rates and processing speeds achievable in high-performance ASICs. In mm-wave bands, there have been not been any attempts to implement massive arrays. Instead, mm-wave arrays today can realize a small number of beams (1-4), either using analog beamforming on a single chip or by analog combining multiple subarrays on-package. The brute force approach taken in < 6 GHz bands cannot be expected to work at mm-wave,

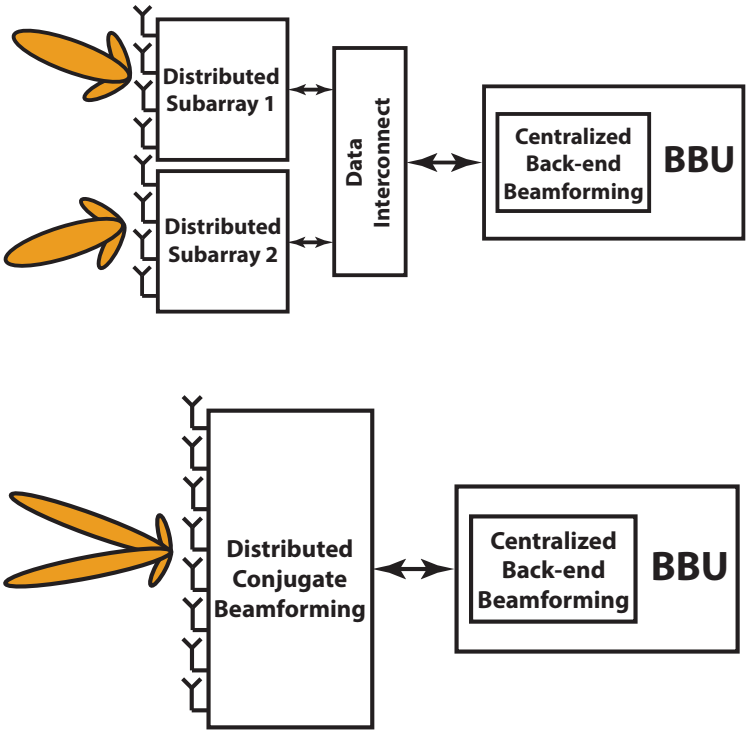


Figure 3.3: Partially (top) and full (bottom) connected array architectures.

where the wide channel bandwidths will make the datarates and signal processing burden too high for the centralized architecture.

Importantly, multi-beam mm-wave arrays are generally limited to a partially-connected architecture (Figure 3.3a). In this architecture, each subarray only forms a single beam. This has two negative consequences. First, for a desired number of beams more hardware is needed since the hardware used is disjoint. Second, each beam's performance depends only on the subarray, not the full array. In contrast, a fully connected array architecture (Figure 3.3b) maps every single user stream to every single antenna. This architecture overcomes the two limitations of a partially connected array but suffers from greater implementation complexity — it is necessary to somehow perform this full mapping function. It is desired to devise system architectures suitable for implementing fully connected arrays at mm-wave.

What is needed is a novel array architecture devised specifically for the massive array regime. This will enable low-frequency arrays to scale beyond 64 elements, and will provide the basis for achieving massive MIMO at mm-wave frequencies. Accordingly, we now seek to devise a general, efficient, and scalable array architecture which can be used to implement very large, multi-beam arrays at both cellular and mm-wave frequencies.

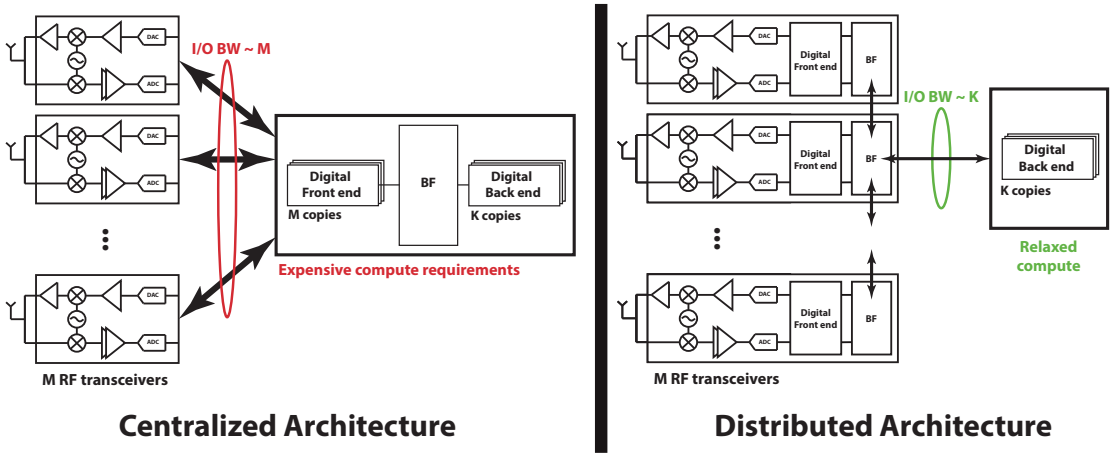


Figure 3.4: Comparison of centralized and distributed processing architectures for massive MIMO arrays.

3.2.2 Large Arrays Must Use Distributed Processing

Because antennas must have a physical size which is on the order of the wavelength, an array with many elements will be physically large relative to the carrier frequency. For instance, arrays operating in the low GHz range will have dimensions on the order of meters; arrays operating at 60GHz will have dimensions on the order of 10s of centimeters. At the same time, the transmit/receive data streams have a single physical interface between the physical layer and higher layers of the network and or application stack. As a result, antenna arrays fundamentally require movement of information between the phyaically dispersed antenna elements and the central processor/network interface. This data movement is the main bottleneck in array design and *how* this information flow occurs and is organized touches many aspects of the array implementation.

As a reference point, consider a fully centralized array architecture (left panel of Figure 3.4). In this architecture, all computation is performed at the central processor — in the transmit direction, this processor computes the signal for every single antenna element, while in the receive direction every receiver forwards its ADC samples for processing. Under this organization, it is clear that the central processor must be equipped with a total I/O bandwidth proportional to M:

$$R_{centr} = M f_s N_b \tag{3.5}$$

where f_s is the data sampling rate and N_b is the number of bits for each sample. This requirement is problematic because of the resulting very large I/O bandwidths. As an example, a 128-element array operating over 100 MHz bandwdith with 20 bits for I/Q samples representation would require at least 256 Gbps I/O bandwidth at the central processor. This would require a complex backplane consisting of aggregation switches solely to merge 128 data flows into a smaller number of high-rate lanes. Even then, the high datarate would consume a large number of I/O lanes on the processor with cutting edge serial link technologies.

Is there anything that can be done about this bottleneck? One important observation is that while each antenna transmits/receives a different signal, these signals are not linearly independent. Rather, the antenna signals lie in a K-dimensional subspace generated by the K distinct users. It should be possible to exploit this redundancy to reduce the dimensionality of the data interconnect and exchange only K rather than M signals with the central processor:

$$R_{distr} = K f_s N_b. \tag{3.6}$$

This analysis focuses on the *maximum* datarate needed in the array, which is the I/O bandwidth of the central processor. This will be the main limitation to array scalability since it presents the first bottleneck to cost or complexity. A related consideration is the aggregate throughput required across all links in the array. This total throughput depends on how many antennas and beams are served by one distributed processing element. From an interconnect perspective, the power consumption of the aggregation network will be the second-order limitation to scalability, and may be significant in extremely large arrays or architectures where the distributed elements interact with more beams than antennas.

This dimensionality reduction can be unlocked through distributed beamforming. Since the antenna signals lie only in a K-dimensional subspace, they can be processed by *any* rank K matrix without loss of information. In the receive direction, the M antenna signals r_{ant} are processed by a $K \times M$ matrix $\boldsymbol{G}_{distr}$ to form K data streams. In the transmit direction, the K data streams (or a linear combination of those) are broadcast to all the antenna elements where they are processed with transpose of that matrix, $\boldsymbol{G}_{distr}^T$.

This results in the distributed processing architecture shown in the right panel of Figure 3.4. Depending on the modulation scheme and where beamforming fits in the signal processing chain, antenna-specific signal processing functions (denoted by digital front end) are also implemented locally at each transceiver. Figure 3.5 shows how the beamforming and data distribution are implemented for uplink and downlink cases, using the distributed array architecture. By its nature, matrix multiplication can be easily performed in a distributed fashion; this fact unlocks the ability to reduce the aggregation bandwidth to order K.

Figure 3.6 shows the ramifications of this architectural choice on the DSP chain for an OFDM based communication system. In order for the beamforming operation to be distributed, any per-antenna signal conditioning must also be implemented in a distributed fashion. Exactly which functions this corresponds to depends on the modulation scheme as well as whether the beamforming is frequency-flat or frequency-dependent. For the depicted OFDM system using frequency-dependent beamforming, this includes transceiver calibration, channel filtering, sampling rate adjustment, timing recovery and FFT/IFFT. At the same time, per-user stream functions such as carrier recovery, coding/decoding, and scheduling must be performed in the central processor. The data interconnect and distributed beamforming connects these two separate signal processing chains to implement the full digital baseband.

In summary, distributed signal processing can significantly relax the I/O bandwidth needed to move data around the array, mitigating this key bottleneck in massive array

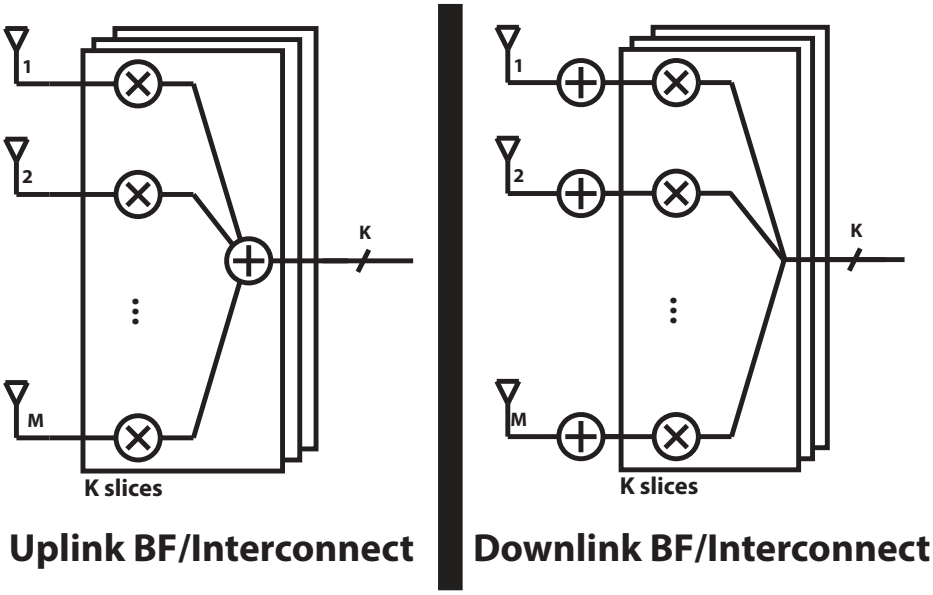


Figure 3.5: Implementation of distributed beamforming and data interconnect for uplink and downlink.

implementation. In exchange, this requires that each RF transceiver be equipped with sufficient signal processing capabilities to implement this distributed computation. As a final note, this section has largely focused on all-digital implementations but the conclusion is implementation-agnostic. Similar design guidelines apply for analog signal processing and beamforming.

3.2.3 The Interconnect Must be Digital

The distributed signal processing described above is amenable to both analog and digital implementations. With a digital interconnect, data is distributed and aggregated using serdes lanes and digital adders. With an analog interconnect, signal distribution and summation is performed using analog splitters and combiners.

When the number of antennas *and* the number of beams is small, all-analog interconnect is quite feasible and possibly preferred [44, 46]. However, analog signal distribution does not scale well to large number of antennas and beams. First, analog routing introduces loss that depends exponentially on distance, requiring power-hungry drivers for long-range routing. Second, wider channel bandwidths result in increased loss and frequency-dependent fading which compound the equalization challenge. Third, analog routes are susceptible to crosstalk and external interference which may limit the performance of beamforming and spatial filtering. Particularly for a large number of beams, crosstalk and EMI management can significantly drive up the complexity and cost of the distribution network.

Based on these issues, we can conclude that *long-distance* routing should be performed

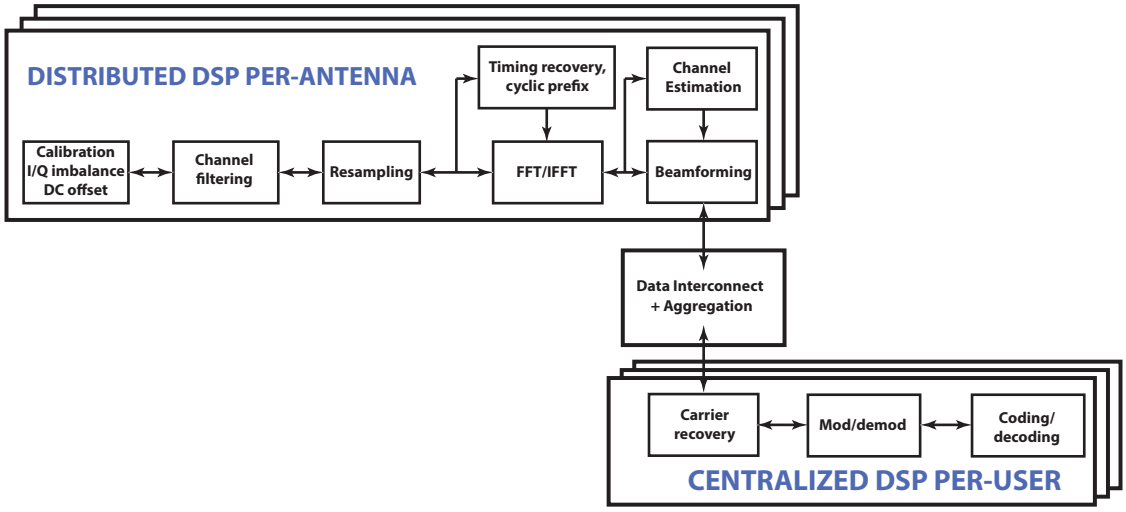


Figure 3.6: Distributed datapath for an OFDM based communication system — per-antenna functions are local to the transceiver while per-user functions are centralized.

using a digital interconnect. A digital interconnect more flexibly and scalably extends to large numbers of elements and users. In practice this means that a cluster of antennas should be co-processed (either in analog or digital fashion, depending on the signal processing requirements), and each cluster should communicate to its neighbors and central processor with a digital interconnect. The boundary between analog and digital interconnect is largely a function of the specifications (datarate, distance, etc) and the available interconnect technology. There is no precise definition of long-distance, but this design guideline is valid for large enough arrays.

Even a digital interconnect can be complex, expensive, and power-hungry when the number of elements and beams is increased. One potential technology would be to use optical signal distribution. Here an extremely wideband analog signal can be modulated onto an optical carrier and easily distributed around the array. If such a technology were developed it would very naturally be applied to data movement in extremely large arrays.

3.2.4 The Array should be Composed of Common Modules

Thus far, we have proposed that signal processing should be implemented in a distributed fashion, with digital interconnect providing long-range communication capabilities between the distributed nodes and the central processor. This naturally suggests grouping a cluster of S nearby antennas together into a subarray, implemented by a common module which contains the RF transceivers, distributed signal processing, and analog/digital conversion (Figure 3.7).

One key benefit of this architecture is amortizing auxiliary functions across multiple

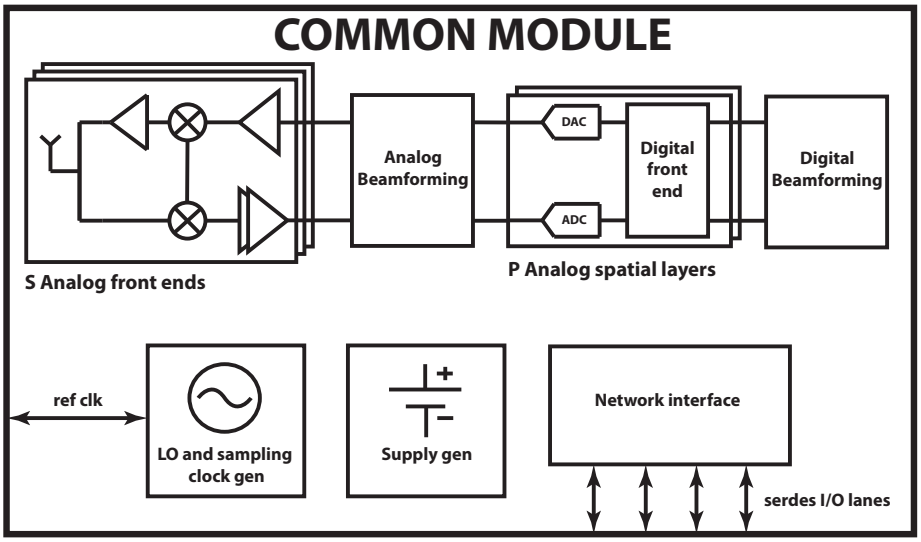


Figure 3.7: Block diagram of a generic common module for a massive MIMO array, including signal path and shared support functions.

elements. For example, each common module can share a network interface, frequency generation, and supply generation, reducing the overhead of those functions. This naturally imposes hierarchy on the array, which is very useful from a design and implementation perspective since it simplifies the task of supply and frequency distribution to all of the transceivers in the array.

How should the number of antennas per module, S, be chosen? This is largely an engineering decision, which trades off the benefits of amortizing shared functions against the implementation challenge of co-packaging a large number of transceivers and antennas. More specifically, the common module provides a natural logical organization for packaging and assembly of the array. Each common module could consist of one or more integrated circuits (IC) along with in-package antennas. Much of the packaging complexity comes from fitting many transceivers on a single die and routing out RF traces from the IC to the antenna. Consequently, the level of module integration is dependent on the carrier frequency and channel bandwidth, as well as the silicon area of the transceivers and other engineering considerations. At mm-wave frequencies, it is common to integrate 32 elements on a single die, and it may be possible to increase this further to 64 depending on silicon area/utilization, the number of I/O pads, and the length of the antenna routing. At lower carrier frequencies generally the number of elements per die is smaller since the silicon area tends to be larger (for the passives) and the antennas are farther apart.

The module abstraction also provides a clean logical partition in the hierarchy. The implementation of the module is an engineering decision which should not impact how the overall system is put together. For example, a module could be implemented as a single mixed-signal system-on-chip, as separate analog and digital chips, or even as multiple front-

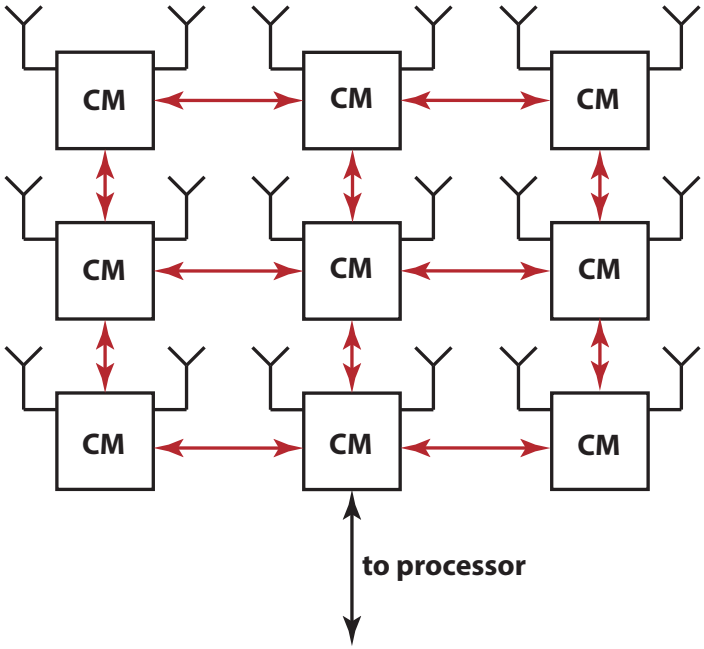


Figure 3.8: Modular and scalable implementation of a massive MIMO array using common modules.

end ICs with analog combining on-package and an FPGA-based digital processor. The module abstraction hides all of these implementation decisions behind a common interface.

As a result, the modular design permits simple scaling to larger number of antenna elements as depicted in Figure 3.8. Since the modules are identical, more can be added onto the interconnect without changing the fundamental way in which the array is organized. For these reasons, this distributed, modular design is the preferred design paradigm for implementing next-generation large antenna arrays.

3.3 Fully Distributed Signal Processing with Two-Stage Beamforming

Section 3.2 proposed that distributed beamforming is critical to managing the data interconnect bandwidth. Any rank K pre-processing matrix $\boldsymbol{G}_{distr}$ is sufficient to perform this dimensionality reduction. In that case, is there any strategy to intelligently choose $\boldsymbol{G}_{distr}$? In particular, the conjugate, zero-forcing, or MMSE beamforming matrices all meet this requirement — is there any reason to prefer one over another? Furthermore, while distributed beamforming eliminates global dependences in *applying* the beamforming weights, it does not necessarily remove global relationships in *computing* the beamformer itself. Is it possible to make the computation of the beamformer fully decentralized as well? This section

proposes a novel two-stage beamforming algorithm consisting of a fully distributed conjugate beamformer, with no inter-element estimation or computation dependencies, followed by a centralized stage which performs user separation. A simpler form of this structure was proposed in [111] for a hybrid beamforming scenario with zero-forcing only. The main novelty here is to extend to more general processing and algorithms and to introduce a fully distributed channel estimation structure to complement the beamformer.

There is a key *practical* difference between conjugate, zero-forcing, and MMSE beamforming algorithms. The conjugate beamforming matrix is obtained solely through a complex conjugate operation on each element of the channel matrix. As a result, no global channel state information is needed. Each transceiver could autonomously compute its conjugate beamforming coefficients for each user stream using purely local channel state information. In contrast, ZF and MMSE beamformers must compute the inverse of a correlation matrix, which is only possible after pooling channel state information from all elements and users. As a result, a naive implementation of ZF or MMSE would require channel estimates to be forwarded the central processor, which computes the global beamforming matrices and sends the relevant information back to each element.

Based on this discussion, when evaluating how to choose $\boldsymbol{G}_{distr}$ it is clear that the conjugate and ZF/MMSE algorithms are not equivalent. Any of these techniques can appropriately reduce the dimensionality of the *data payload*. However using the ZF or MMSE beamformer for $\boldsymbol{G}_{distr}$ requires an initial step of pooling channel estimates, computing matrix inverses, and then broadcasting the beamforming weights. This is a relatively modest burden from the perspective of interconnect bandwidth but imposes significant latency. This processing latency would require each transceiver to buffer incoming data until the global beamforming weights are computed and broadcasted.

This bottleneck can be overcome using a two-stage beamforming architecture which implements ZF or MMSE in a fully distributed fashion. The key identification is that the conjugate, ZF, and MMSE beamformers — in (2.11), (2.14), and (2.18), respectively — share a common structure. Each of these algorithms can be expressed as a conjugate beamformer followed by a decorrelation operation (Figure 3.9). The conjugate beamforming step is responsible for physically forming the beam lobe which points towards the incoming energy. In a sense this creates a virtual directional antenna which tracks the user's physical location. The decorrelator then separates out the multiple spatial streams according to the desired objective function. This deconstruction applies for both receive and transmit directions.

As such, all three of these algorithms could be mapped onto the common structure shown in Figure 3.10. First, conjugate beamforming is applied in a fully distributed manner as described in Section 3.2.2, with both channel estimation and beamforming taking place autonomously at each element. With this operation, the data at the central processor experiences effective $K \times K$ channel $\boldsymbol{H}_{eff} = \boldsymbol{G}_{conj}\boldsymbol{H}$. Therefore, the central processor estimates this effective channel and computes the appropriate decorrelation matrices.

How should the central processor estimate the effective channel? If the channel estimation pilots could be processed with conjugate matrix $\boldsymbol{G}_{conj}$ then they would naturally carry information about $\boldsymbol{H}_{eff}$. This can be achieved using the structure shown in Figure 3.11.

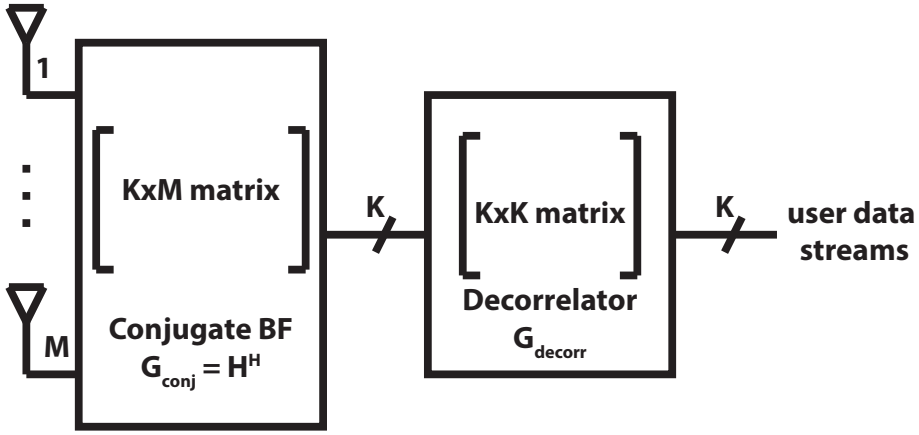


Figure 3.9: Conceptual mapping of any beamformer (including conjugate, ZF, or MMSE) into a two-stage structure — beamforming + decorrelation — for both transmit and receive directions.

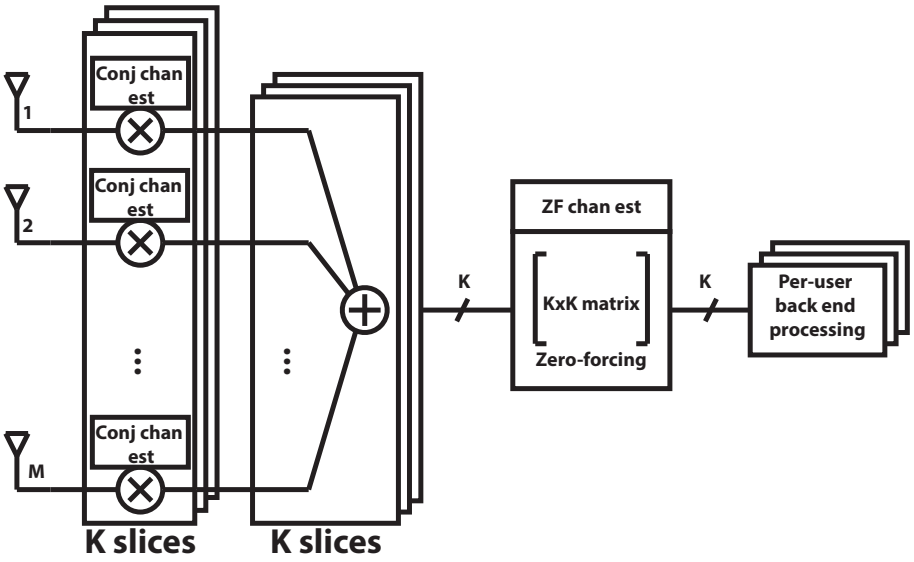


Figure 3.10: Implementation of two-stage receive-side beamforming, for a frequency-domain beamformer, with distributed conjugate estimation/beamforming and centralized zero-forcing.

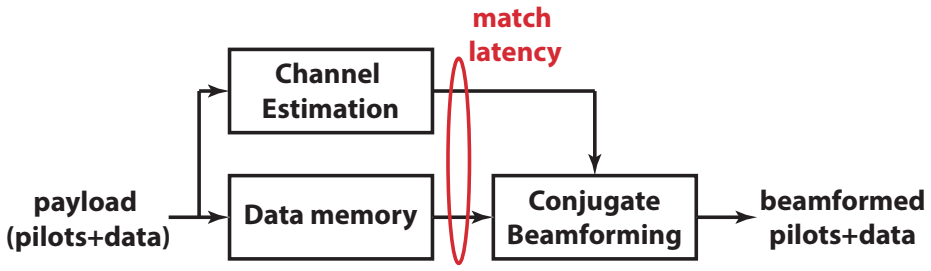


Figure 3.11: Pilot beamforming algorithm. Incoming data payload is delayed to match the latency of the channel estimation block so that both pilots and data may be conjugate beamformed.

The exact implementation depends on the relevant standard; however the key idea is to delay the incoming data while the channel estimate is computed and then apply the conjugate beamforming weights to both pilots and data. This delay only has to match the latency of the channel estimation block which is quite low due to the relative simplicity of this task. Generally a delay of a single OFDM symbol would be sufficient and the overhead is quite low. For example, this could be implemented as an in-memory processing unit (where memory and logic are co-located and tightly coupled) combined with several other functions on the common module. Overall, the two-stage approach provides a methodology for implementing any form of central processing algorithm with no additional communication overhead or processing latency.

The proposed two-stage beamforming also provides a level of abstraction between energy collection (conjugate beamforming) and user stream separation (ZF/MMSE/etc) — two tasks which are conceptually separate. This abstraction cleanly decouples the implementation of the conjugate beamformer from the decorrelator. For instance, the conjugate beamforming could be applied in the analog domain while back-end zero-forcing is performed in digital[1]. Or, a frequency-flat conjugate stage could be coupled with a frequency-dependent decorrelator. This complements and builds on the approach and design philosophy in Section 3.2 for devising a scalable and modular array architecture.

3.4 FIR Filter-Bank Beamforming

So far, this chapter has proposed a modular and scalable array architecture encompassing signal processing, data interconnect, and hardware functions. An important goal of this effort was to devise an architectural framework that encompasses both analog and digital beamforming implementations. In the communications field, “analog beamforming” is almost

[1]If the conjugate beamforming is done in analog, then an analog memory or delay network would be needed to store the pilots while the beamforming ccoefficients are computed. One simple way around this (with throughput overhead) would be to send duplicate pilots for conjugate and ZF stages.

always taken to mean phased arrays implementing RF, LO, or baseband phase shifting (in a frequency-flat manner). This section expands this way of thinking by presenting a more sophisticated time-domain beamforming technique. The proposed technique uses a bank of FIR filters to implement spatio-temporal processing which can form multi-user, frequency-dependent beamforming patterns. Though this is not inherently an analog technique, a key motivation behind this study is to devise an analog-friendly implementation.

Wideband beamforming has been an important consideration in array engineering, particularly in radar applications. Because the notion of "wideband" in arrays is defined relative to the array size, very large arrays experience progressively more significant wideband effects even for relatively narrow instantaneous signal bandwidths. It was recognized early on that time-domain processing was needed to implement frequency-dependent beamformers [29, 112, 113]. A common way to implement this generalizes a phased array by introducing an FIR filter of phase shifts. The added time dimension enables simultaneous spatial *and* temporal processing of the incoming signal, forming wideband beams.

In the communications field, time-domain processing has been studied under the label of "time-reversal processing" [114–118]. Generally these studies consider a single-user or multi-user scenario where the base-station is equipped with only a single antenna. It then uses the temporal resolution of the channel to identify users and separate out their signal. This work extends those results by considering explicitly multi-antenna base-stations which can use spatial and temporal processing.

This section seeks a time-domain implementation of the frequency-domain algorithm in section 2.3.2. This extends previous work in wideband arrays in three ways. First, we apply the wideband beamforming principles from other array applications to the field of communications. Second, we consider the case of multi-user spatial + temporal zero-forcing — temporal equalization combined with spatial separation of streams — using the least-squares algorithm based on channel estimates. Third, we consider this structure in the massive MIMO limit and some implementation details.

Why bother looking for a time-domain alternate to the FFT beamformer? Despite its simplicity, frequency-domain beamforming has some limitations. First, frequency-domain beamforming is not easily amenable to analog implementations since it requires an FFT. Second, in some conditions the computational complexity may be dominated by the requirement to perform one FFT per element. Finally, frequency-domain equalization requires cyclically-prefixed modulation formats, which lose some spectral efficiency from the cyclic prefix. A fully time-domain implementation could overcome one or more of these drawbacks.

3.4.1 Review of FIR Equalizers for SISO Channels

In order to develop the full MIMO time-domain beamformer, it is useful to review how FIR equalizers (also known as feedforward equalizers — FFEs) are constructed. This well-known problem consists in solving a least-squares equation to compute the coefficients for a SISO FIR equalizer. It is commonly used in wireline trasceivers for FFEs.

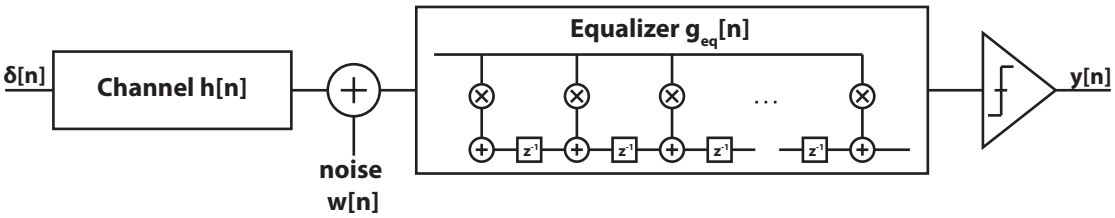


Figure 3.12: SISO FIR equalizer design scenario.

Consider a SISO channel with impulse response

$$h[n] = \sum_{l=0}^{L-1} h_l \delta[n-l] \tag{3.7}$$

where L is the impulse response length. Suppose it is desired to equalize this channel using a finite impulse response (FIR) filter g_{eq} at the receiver (Fig. 3.12). The objective is:

$$(g_{eq} * h)[n] = \sum_{k=-\infty}^{\infty} g_{eq}[k]h[n-k] = \delta[n-n_0] \tag{3.8}$$

where the equalized latency n_0 and the equalizer length N are chosen by the designer. This can be reframed as a matrix equation. The equalized impulse response has length $L+N-1$. The $(L+N-1)\times N$ Toeplitz convolution matrix of the channel h is:

$$\tilde{\boldsymbol{H}} = \begin{bmatrix} h_0 & 0 & 0 & \dots & 0 \\ h_1 & h_0 & 0 & \dots & 0 \\ h_2 & h_1 & h_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_{N-2} & h_{N-3} & \dots & h_0 \\ 0 & h_{N-1} & h_{N-2} & \dots & 0 \\ 0 & 0 & h_{N-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & h_{L-1} \end{bmatrix} \tag{3.9}$$

It can be easily verified that when multiplying by an $N \times 1$ vector of coefficients v, this matrix implements the convolution of h with v.

Now (3.8) can be cast into matrix form as follows:

$$\boldsymbol{e}_{n_0} = \tilde{\boldsymbol{H}}\boldsymbol{g} \tag{3.10}$$

where $\boldsymbol{e}_{n_0}$ is an $(L+N-1)\times 1$ vector of all zeros except for a one in the n_0'th position. This is an over-determined system since there are $N+L-1$ equations but only N free variables

(the equalizer coefficients). Therefore, it must be solved in a least-squares sense:

$$\hat{\boldsymbol{g}}_{zf}^H = (\tilde{\boldsymbol{H}}^H \tilde{\boldsymbol{H}})^{-1} \tilde{\boldsymbol{H}}^H \boldsymbol{e}_{n_0} \tag{3.11}$$

The resulting N-coefficient equalizer is the least squares optimal FIR equalizer for the given channel[2]. This is known as "zero-forcing" equalization since it forces the inter-symbol interference to zero.

It is possible to construct an MMSE equalizer for a SISO channel as well. Here we have that

$$\begin{aligned} r[n] &= \sum_{l=0}^{L-1} h[l]d[n-l] + w[n] \\ \hat{d}[n] &= \sum_{k=0}^{N-1} g[k]r[n-n_0-k] \end{aligned} \tag{3.12}$$

where, as before, the latency n_0 is arbitrary and can be chosen by the designer. Applying the orthogonality conditions:

$$\mathbb{E}[(d[n] - \hat{d}[n])r[k-n_0]] = 0 \quad \forall k \in [0, N-1] \tag{3.13}$$

results in the MMSE equalizer:

$$\boldsymbol{g}_{mmse}^H = (\tilde{\boldsymbol{H}}^H \tilde{\boldsymbol{H}} + \sigma^2 \boldsymbol{I}_N)^{-1} \tilde{\boldsymbol{H}}^H \boldsymbol{e}_{n_0} \tag{3.14}$$

where σ^2 is the thermal noise variance. It is interesting and satisfying that the zero-forcing and MMSE equalizer give results very similar to those obtained for beamforming above. This is very natural since beamforming could be thought of as spatial equalization.

3.4.2 Design of Time-Domain Beamformers

The objective is then to generalize the SISO technique to cover SIMO and MIMO channels. We begin with the SIMO or MISO case. Consider a SIMO channel where the channel from the transmitter to antenna i is denoted by h_i. The objective is to design a receive filter at each antenna, g_i, such that

$$\sum_{i=1}^{M} h_i[n] * g_i[n] = \delta[n - n_0] \tag{3.15}$$

Notice that this is a statement of both beamforming (sum the signals arriving at each antenna) as well as zero-forcing equalization (eliminate inter-symbol interference). Following the same process as in the SISO case, construct the channel convolution matrix as the concatenation of the the individual channel convolution matrices:

$$\tilde{\boldsymbol{H}} = [\tilde{\boldsymbol{H}}_1 \quad \tilde{\boldsymbol{H}}_2 \quad \dots \quad \tilde{\boldsymbol{H}}_M] \tag{3.16}$$

[2]Note this is how feed-forward equalizers (FFEs) are designed for wireline links.

Note that this matrix has dimensions $(N+L-1) \times MN$. Similarly, construct the $MN \times 1$ vector of equalizer coefficients by concatenating the coefficients of the equalizer at every element:

$$\boldsymbol{g}_{simo} = [\boldsymbol{g}_1^T \quad \boldsymbol{g}_2^T \quad \dots \quad \boldsymbol{g}_M^T]^T \tag{3.17}$$

Finally, the constraint (3.15) can be written as a matrix equation

$$\boldsymbol{e}_{n_0} = \tilde{\boldsymbol{H}}\boldsymbol{g}_{simo} \tag{3.18}$$

Because this system of equations is under-constrained, the least-squares solution is:

$$\boldsymbol{g}_{simo,zf} = \tilde{\boldsymbol{H}}^H(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^H)^{-1}\boldsymbol{e}_{n_0} \tag{3.19}$$

Using a similar procedure as above, it is possible to derive the MMSE FIR beamformer.

$$\boldsymbol{g}_{simo,mmse} = \tilde{\boldsymbol{H}}^H(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^H + \sigma^2\boldsymbol{I}_{N+L-1})^{-1}\boldsymbol{e}_{n_0} \tag{3.20}$$

This result can be straightforwardly extended to the MIMO case. Now, each element in the array is equipped with K FIR equalizers (Fig. 3.13). Both the channels and equalizers are labeled with indices i and j, where i is the antenna index and j the stream index. Now the beamforming and equalization constraint is

$$\sum_{i=1}^{M} g_{ij}[n] * h_{jk}[n] = \delta_{jk}\delta[n-n_0] \tag{3.21}$$

where δ_{jk} is the Kroenecker delta. This objective extends the SIMO/MISO case by introducing an additional *zero-forcing beamforming* constraint which states that user j's filter bank should completely reject user k's signal if $j \neq k$.

Proceeding as before, construct the $K(N+L-1) \times MN$ channel Toeplitz matrix as

$$\tilde{\boldsymbol{H}} = \begin{bmatrix} \tilde{\boldsymbol{H}}_{11} & \tilde{\boldsymbol{H}}_{21} & \dots & \tilde{\boldsymbol{H}}_{M1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{H}}_{1K} & \tilde{\boldsymbol{H}}_{2K} & \dots & \tilde{\boldsymbol{H}}_{MK} \end{bmatrix} \tag{3.22}$$

Similarly, form the $MN \times K$ matrix of receive filters as

$$\boldsymbol{g}_{mimo} = \begin{bmatrix} \boldsymbol{g}_{11} & \cdots & \boldsymbol{g}_{1K} \\ \vdots & \ddots & \vdots \\ \boldsymbol{g}_{M1} & \cdots & \boldsymbol{g}_{MK} \end{bmatrix} \tag{3.23}$$

Finally, form the $K(N+L-1) \times K$ matrix of desired responses as

$$\boldsymbol{D}_{mimo} = diag(\boldsymbol{e}_{n_0}) \tag{3.24}$$

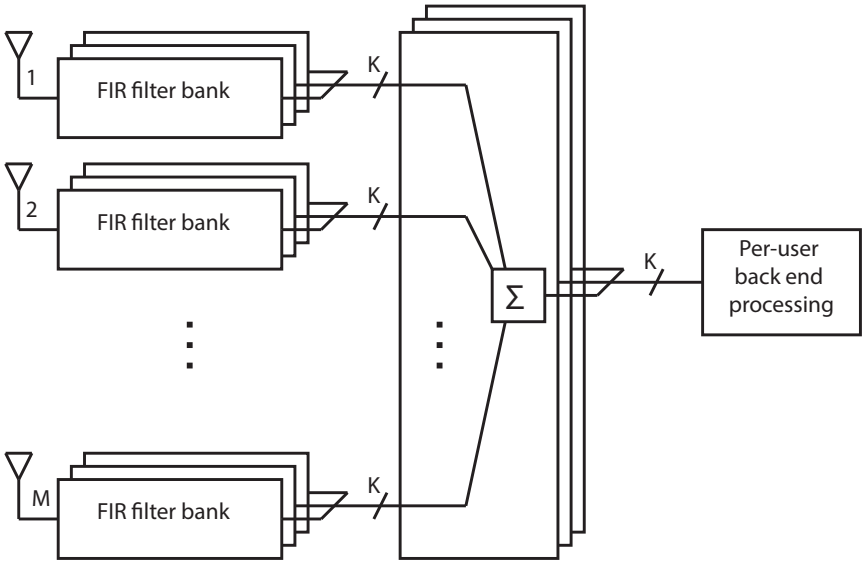


Figure 3.13: Block diagram of a receiver using the proposed time-domain beamforming and equalization — each receiver is equipped with a bank of K FIR filters.

which is a block diagonal matrix with $\boldsymbol{e}_{n_0}$ on the diagonal and zeros elsewhere. Note in general the latency n_0 could be chosen independently for each user, but without loss of generality we omit that case. $\boldsymbol{D}_{mimo}$ establishes the beamforming and equalization constraints by enforcing equalization on the diagonal and zero-forcing beamforming off-diagonal. As before, since the system is under-constrainted the zero-forcing solution is:

$$\boldsymbol{g}_{mimo,zf}^{H} = \tilde{\boldsymbol{H}}^{H}(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^{H})^{-1}\boldsymbol{D}_{mimo} \tag{3.25}$$

The MMSE solution is:

$$\boldsymbol{g}_{mimo,mmse}^{H} = \tilde{\boldsymbol{H}}^{H}(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^{H} + \sigma^2\boldsymbol{I}_{K(N+L-1)})^{-1}\boldsymbol{D}_{mimo} \tag{3.26}$$

This result gives a recipe for designing a K-parallel filter bank at each element which separates user streams in the space-time domain. It is helpful to consider the special case of $N = 1$, $L = 1$: If the channel has no postcursors ($L = 1$), then no equalization is required ($N = 1$). Under these conditions, $\tilde{\boldsymbol{H}}$ becomes the narrowband flat-fading channel matrix $\boldsymbol{H}^T$ and $\boldsymbol{D}_{mimo} = 1$. Consequently, (3.26) and (3.25) become exactly (2.19) and (2.13), respectively. It is clear that the FIR formulation contains as a special case narrowband beamformer design.

3.4.3 MIMO FIR Beamformer in the Large Array Regime

The asymptotic behavior of the FIR beamformer as M grows large reveals useful intuition about how this algorithm operates. Additionally, it uncovers striking parlalels with the conventional frequency-domain implementation.

Let us consider first the SIMO/MISO case (3.19) and analyze the correlation matrix R:

$$\begin{aligned}\boldsymbol{R} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^H &= [\tilde{\boldsymbol{H}}_1 \quad \tilde{\boldsymbol{H}}_2 \quad \dots \quad \tilde{\boldsymbol{H}}_M][\tilde{\boldsymbol{H}}_1 \quad \tilde{\boldsymbol{H}}_2 \quad \dots \quad \tilde{\boldsymbol{H}}_M]^H \\ &= \sum_{i=1}^{M} \tilde{\boldsymbol{H}}_i \tilde{\boldsymbol{H}}_i^H = \sum_{i=1}^{M} \boldsymbol{R}_i\end{aligned} \tag{3.27}$$

Using the fact that $\tilde{\boldsymbol{H}}_i$ is Toeplitz, each such $\boldsymbol{R}_i$ has the following structure:

$$\begin{aligned}[\boldsymbol{R}_i]_{jk} &= \boldsymbol{h}_i[-(N-1-j):j] \cdot \boldsymbol{h}_i[-(N-1-k):k]^H \\ [\boldsymbol{R}_i]_{jk} &= [\boldsymbol{R}_i]_{kj}^* \quad \forall j,k \in [0, N+L-2]\end{aligned} \tag{3.28}$$

Here the notation $\boldsymbol{h}[a:b]$ indicates the part of vector $\boldsymbol{h}$ from index a to b, with zero-padding for all negative indices. Therefore we can simplify:

$$\begin{aligned}[\boldsymbol{R}_i]_{jj} &= \sum_{l=-(N-1-j)}^{j} |\boldsymbol{h}_i[l]|^2 \\ [\boldsymbol{R}_i]_{jk} &= \sum_{l=-(N-1-j)}^{j} \boldsymbol{h}_i[l]\boldsymbol{h}_i[l-j+k] \quad j \neq k\end{aligned} \tag{3.29}$$

Intuitively, each column of matrix $\boldsymbol{R}_i$ is formed by taking an N-sample window of zero-padded impulse response $\boldsymbol{h}_i$ and convolving it with the conjugate, time-reversed impulse response $\boldsymbol{h}_i^{(tr)}$.

Plugging this result into (3.27), we obtain:

$$\begin{aligned}\frac{1}{M}[\boldsymbol{R}]_{jj} &= \frac{1}{M}\sum_{i=1}^{M}\sum_{l=-(N-1-j)}^{j} |\boldsymbol{h}_i[l]|^2 \\ \frac{1}{M}[\boldsymbol{R}]_{jk} &= \frac{1}{M}\sum_{i=1}^{M}\sum_{l=-(N-1-j)}^{j} \boldsymbol{h}_i[l]\boldsymbol{h}_i[l-j+k] \quad j \neq k\end{aligned} \tag{3.30}$$

Taking the limit as $M \to \infty$, the diagonal elements of $\boldsymbol{R}$ converge to the average power in a given channel impulse response window. The off-diagonal elements depend on the correlation between $\boldsymbol{h}_i[l]$ and $\boldsymbol{h}_i[l-j+k]$ for $j \neq k$ — that is, the correlation between different taps of the channel. As long as the taps of the channel impulse responses are uncorrelated, the off-diagonal elements converge to 0. This condition is satisfied both for Rayleigh fading as well as for environments where postcursors have different direction of arrival than the cursor. Under these conditions of channel impulse response orthogonality, as M grows large,

$$\boldsymbol{g}_{simo,zf} \to \tilde{\boldsymbol{H}}^H \boldsymbol{\Sigma} \boldsymbol{e}_{n_0} \tag{3.31}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix. Since $\boldsymbol{e}_{n_0}$ is a vector of all zeros except for the n_0'th, $\boldsymbol{g}_{simo,zf}$ is just a scaled copy of the n_0'th column of $\tilde{\boldsymbol{H}}^H$. Therefore, the equalizer $\boldsymbol{g}_i$ at element i has coefficients:

$$\boldsymbol{g}_i = \boldsymbol{h}_i^*[n_0 - N - 1 : n_0]^{(tr)} \tag{3.32}$$

That is to say, in the large array limit, the optimal SIMO receive filter is given by conjugate time-reversing an N-sample window of the channel impulse response $\boldsymbol{h}_i$. This is nothing but a windowed matched filter!

This result can easily extend to the MIMO case:

$$\begin{aligned} \frac{1}{M}[\boldsymbol{R}]_{(L+N-1)p+j,(L+N-1)p+j} &= \frac{1}{M}\sum_{i=1}^{M}\sum_{l=-(N-1-j)}^{j}|\boldsymbol{h}_{ip}[l]|^2 \\ \frac{1}{M}[\boldsymbol{R}]_{(L+N-1)p+j,(L+N-1)q+k} &= \frac{1}{M}\sum_{i=1}^{M}\sum_{l=-(N-1-j)}^{j}\boldsymbol{h}_{ip}[l]\boldsymbol{h}_{iq}[l-j+k] \quad j \neq k \end{aligned} \tag{3.33}$$

If the channel impulse responses of different users are uncorrelated, then this result converges in the same way as the SIMO case above. Consequently,

$$\boldsymbol{g}_{mimo,zf} \rightarrow \tilde{\boldsymbol{H}}_{mimo}^H \boldsymbol{\Sigma}\boldsymbol{D}_{mimo} \tag{3.34}$$

where $\boldsymbol{\Sigma}$ is some diagonal matrix, and therefore

$$\boldsymbol{g}_{ip} = \boldsymbol{h}_{ip}^*[n_p - N - 1 : n_p]^{(tr)} \tag{3.35}$$

where n_p is the chosen latency of user p's receive filters. This result indicates that in the large array limit, the optimal MIMO receive filter is a windowed matched filter for each user, with no need for inter-user zero-forcing.

This is a significant result for several reasons. First, it was shown in Section 3.1.3 that in the "massive" array limit, conjugate beamforming was optimal. This result is simply the generalization of that to multi-tap channels. Second, from basic Fourier theory it is known that if $h[n]$ has DFT H_k, then the DFT of the conjugate, time-reversed sequence $h[-n]^*$ is H_k^*, which gives exactly the conjugate beamforming coefficients expected from a frequency-domain beamformer. Consequently, at least in the limit of large M, the FDE and FIR beamformers are identical and the tradeoff of which to use depends on the channel characteristics.

Why should this optimal beamformer converge to a matched filter? As long as different taps of the channel impulse response are uncorrelated, the channel will be self-equalized just by virtue of being orthogonal to the receive filter. Consquently, the receive filter design should merely focus on collecting as much energy as possible, which is accomplished by a matched filter. This also suggests a design intuition for selecting N and n_0. These parameters should be chosen to maximize the energy collected from the channel impulse response.

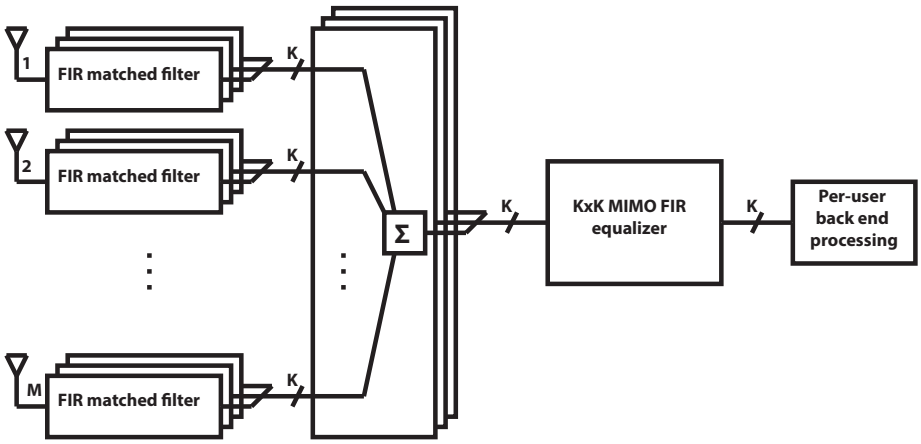


Figure 3.14: Two-stage FIR beamforming: bank of matched filters at each antenna followed by summation network and a $K \times K$ matrix FIR zero-forcing beamformer/equalizer.

The asymptotic analysis reveals a deep insight into how the FIR beamforming algorithm operates. Much as the frequency-domain zero-forcing could be split into a conjugate beamformer followed by a zero-forcing operation, a similar factoring applies to the FIR beamforming algorithm. A two-stage implementation of FIR beamforming consists of a bank of matched filters followed by a MIMO matrix equalizer which compensates for inter-symbol and inter-user interference (Figure 3.14).

3.4.4 Equivalence of FIR and FDE Beamformers

The analysis in the previous section showed that FIR and FDE beamformers are identical in the limit as M grows large. Is this true in general or only in this limit?

We can motivate the equivalence of FIR and FDE techniques by comparing the computed equalizers for individual channel instantiations as well as the aggregate performance across an ensemble of channels. Figure 3.15 plots the frequency response and impulse response of one branch of the FIR or FDE beamformers in a MIMO system. As shown, these results reveal exact correspondence between the beamformer computed with the frequency-domain or time-domain algorithm.

Figure 3.16 compares the performance of FIR and FDE beamforming with $M = 16$ and $K = 2$ across a collection of channels fitting WLAN model C. It is clear that the performance of FIR and FDE is identical, indicating that FIR filter bank beamforming can fully exploit the frequency-dependent spatial signature of the channel.

Combining both the individual channel instantiations and the statistical results, it is fair to conclude that the FIR and FDE beamformers implement the equivalent beamforming algorithm.

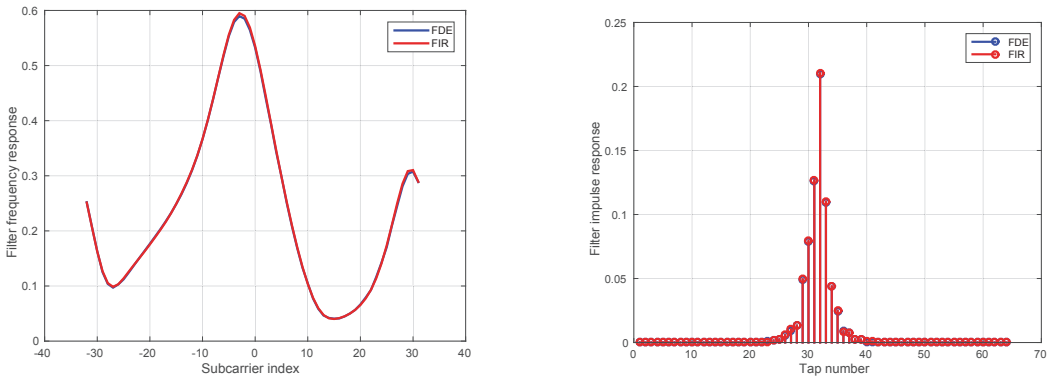


Figure 3.15: (a) Frequency response of FIR and FDE beamformers in a WLAN channel model C. (b) Impulse response of FIR and FDE beamformers in a WLAN channel model C.

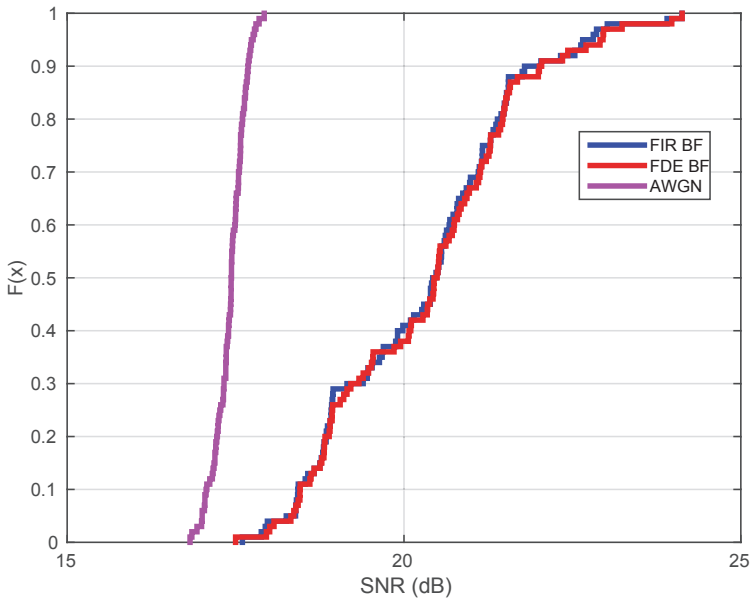


Figure 3.16: CDF of SNR required to achieve a 10^{-3} BER in a WLAN channel model C environment, with $M = 16$ and $K = 2$, comparing FIR and FDE schemes.

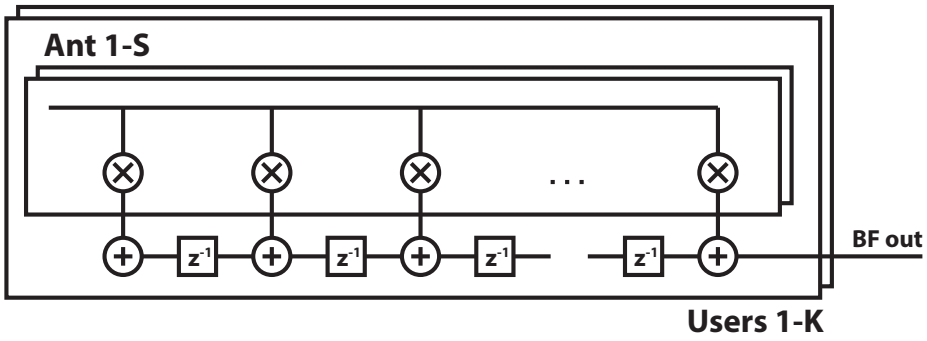


Figure 3.17: Optimal FIR-beamforming structure for implementation on a subarray basis when $K < S$.

3.4.5 Computational Complexity of FIR and FDE Beamformers

The preceding sections have described how to design and implement a time-domain MIMO beamformer which is equivalent to an FDE ZF beamformer. These two algorithms are equivalent and implement the identical solution. What is the difference in computational complexity between the time- and frequency-domain implementations?

This comparison should take into the account the arithmetic operations required, the number of storage elements and memory operations, and the complexity of computing the beamformer. For the FDE structure, each antenna element is equipped with an N_{sc}-point FFT, each of which requires approximately $\frac{1}{2}N_{sc}\log_2(N_{sc})$ arithmetic operations and approximately N_{sc} storage elements. Additionally, to process a full OFDM symbol through the beamformer, MKN_{sc} complex multiplies are required.

For the FIR structure, each antenna element is equipped with K FIR filters, each of length L where $L < N_{sc}$ [3]. Each filter bank requires KLN_{sc} complex-multiplies to process an OFDM symbol. The storage elements can be shared across S front ends, as shown in Figure 3.17 (using a transpose FIR filter; if $K > S$, a transverse structure is preferred). Consequently, each filter bank only needs KL/S storage elements.

Finally, to compute the FDE zero-forcing matrices for each subcarrier, N_{sc} $K \times K$ matrix inversions are needed. In contrast, to compute the MIMO equalizer for the FIR structure, a single $KL \times KL$ matrix inversion is required. The computational complexity of an $N \times N$ matrix inversion is roughly N^3.

The overall computational complexity of the FDE and FIR algorithms is summarized in Figure 3.18. Three points stand out. First, unless L is very small (on the order of 2-4), FIR beamforming requires more arithmetic operations. This is quite intuitive — after all, a key advantage of OFDM is that frequency-domain equalization is much simpler than time-domain equalization. Second, delay cell sharing across a subarray cuts down on the number of storage elements needed by FIR beamforming. Depending on the exact implementation

[3] A reasonable ballpark for L is around $N_{sc}/4$

	FDE-BF	FIR-BF
Integer mul-adds	$MN(0.5\log_2(N)+K)$	MNKL
Registers	2MN	MKL/S
Matrix inversion cost	NK^3	$(KL)^3$

Figure 3.18: Comparison of the computational complexity of FDE and FIR beamformers to process one OFDM symbol ($N = N_{sc}$).

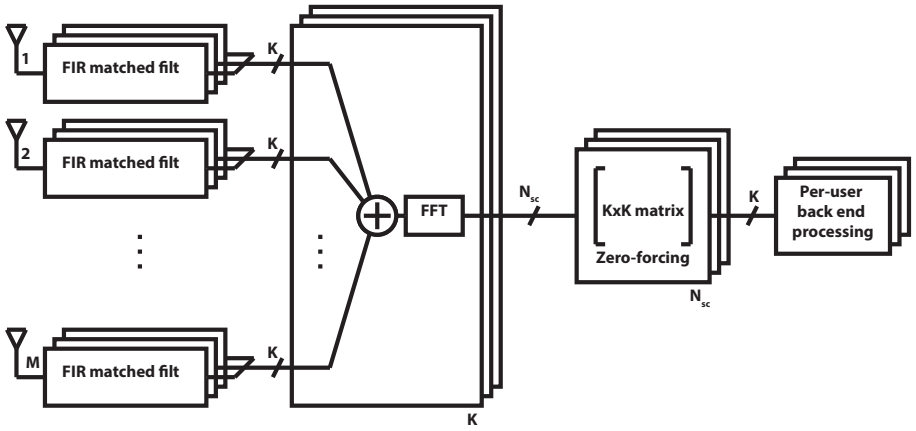


Figure 3.19: Block diagram of a receiver using combined time- and frequency-domain processing: a matched filter bank per receiver followed by a ZF or MMSE FDE on a per-subcarrier basis.

details, FIR beamforming may have somewhere between 1 and 1/4 as many storage elements as the FDE.

Finally, and most importantly, there is an enormous disparity in the matrix inversion cost. This again reflects the vastly simplified complexity of frequency-domain equalization. The requirement to invert enormous matrices to evaluate the FIR beamformer makes this algorithm impractical to use in this form.

3.4.6 Mixed Time- and Frequency-Domain Algorithm

The number one drawback of the FIR beamformer is that the computation of the matrix equalizer portion is extremely complex. In contrast, the front-end matched filter is no more complex to compute than the equivalent frequency-domain conjugate beamformer.

Since the FIR and FDE beamformer implementations are equivalent, one could imagine implementing the beamforming-equalization algorithm partially in the time domain and partially in the frequency domain. As noted above in Section 3.3 and Section 3.4.3, both FDE and FIR algorithms can be factored into a conjugate step followed by a decorrelation or zero-forcing step. This suggests an implementation consisting of a time-domain conjugate

beamformer followed by frequency-domain post-processing (Fig. 3.19). Note that this is simply a generalization of the frequency-flat beamformer by allowing each conjugate beamformer to have frequency-varying response. According to the equivalence demonstrated above, this will be identical to a completely time- or frequency-domain implementation.

This structure offers a few advantages compared to a fully time-domain implementation. First of all, as described in Section 3.3 the matched filter can be easily estimated locally at each element. Second, the equalization and post-processing is simpler to compute and implement in the frequency-domain, avoiding the huge computational complexity of solving (3.26).

The main disadvantage of this structure is that it requires CP-equipped modulation formats to enable the frequency-domain processing in the back-end. Additionally, the length of the receive matched filter must fit within the CP requirements. Since each receiver utilizes a matched filter with N taps to beamform, the effective channel impulse response length is $N+L-1$. In order to ensure that the circular convolution property of OFDM and SC-OFDM still holds, it is necessary that the length of the guard interval be greater than $N+L-1$. Alternatively, it is possible to "under-budget" the CP and instead apply the beamforming matched filter in a block manner rather than as a streaming FIR filter. This would require each receiver to recover each CP-precoded symbol, strip the CP, cyclically extend it again, and apply the receive filter.

3.4.7 Summary

In this section, a time-domain equalizer-beamformer has been proposed for MIMO channels. By applying the simultaneous constraints of equalization and beamforming (in either a ZF or MMSE sense), this algorithm constructs a bank of K FIR filters at each element which both equalize and beamform a multi-stream signal. This algorithm can be considered a generalization of narrowband beamformers. Furthermore, it is completely equivalent to the FDE beamforming problem and represents a time-domain implementation which achieve the identical solution. Finally, it exhibits the expected massive MIMO convergence behavior to a matched filter.

Chapter 4

Carrier Generation and Synchronization

Chapter 3 considered the task of data aggregation and processing in large arrays. A modular and distributed system architecture was proposed in order to manage the complexity of this task. In a similar vein, synchronization of the many array elements has a significant impact on both the design/implementation complexity and the achievable performance. It is critical to ensure that all the elements are carrier (RF) synchronized to each other for proper array operation. However, if not done carefully, this may require global synchronization loops which impose a bottleneck to the array power consumption, design complexity, and scalability.

This chapter considers the carrier synchronization of massive MIMO arrays, which refers to the generation and tracking of the carrier frequency and phase. Up- and down-conversion mixers use a local oscillator (LO) to translate in frequency between the baseband signal and the modulated RF carrier. This LO must be phase and frequency synchronized, both across the entire array as well as "over the air" with the other side of the wireless link.

In traditional SISO links carrier synchronization is accomplished with a variety of hardware and signal processing blocks. This chapter considers how to design carrier synchronization subsystems for the massive MIMO regime, proposing a combination of architectural, hardware, and signal processing tools to ensure low-power and high-accuracy carrier synchronization. In short, we motivate that global carrier recovery is ideal from a performance standpoint but excessively power-hungry to implement, and instead propose more sophisticated techniques to reduce the power of the LO subsystem and mitigate performance impacts that arise. First we will analyze how the choice of distribution architecture impacts the total power consumption and scalability of the array. Next we will show that the LO generation architecture also impacts the array performance and describe the mechanisms by which this occurs. Finally, we will develop design guidelines for hardware and signal processing blocks in the LO subsystem which can mitigate the performance loss mechanisms in large arrays.

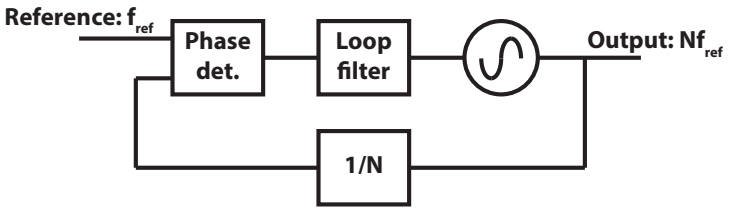


Figure 4.1: Block diagram of standard PLL

4.1 LO Distribution Architectures

The LO is usually generated using a phase-locked loop (PLL) which generates a high-frequency oscillation locked to a stable low-frequency reference. Due to their tunability, voltage-controlled oscillators (VCOs) have poor frequency and phase stability. On the other hand, crystal oscillators (XOs) generate very precise and stable oscillations but at a fixed, and typically low, frequency. Common XO frequencies range from single to few hundred MHz and are determined by the mechanical properties of the crystal. By placing a high-frequency VCO in a feedback loop referenced to an XO (Fig. 4.1), a PLL can transfer the phase and frequency stability of the crystal to the VCO. PLL-based carrier generation is used so widely because it combines the very high precision of the XO with the variable multiplication ratio and wide tuning range of the VCO.

In SISO links, the TX and RX are each equipped with a PLL which generates their LOs; carrier synchronization is only needed to synchronize the receiver to the transmitter. In arrays, there is an additional degree of complexity in ensuring that all the elements in the array are synchronized to each other[1]. In this case, the conceptually simplest LO distribution architecture is a fully centralized one': generate a single centralized LO and distribute that to every single transceiver in the array. This attempts to replicate the SISO architecture in the massive MIMO regime and ensures that all the transceivers are perfectly synchronized. In fact this is a nearly universal design choice for single-chip arrays with 16-32 elements on a single die. In this regime the total LO subsystem power consumption is relatively insensitive to the architecture and the complexity reduction of a fully centralized LO generation justifies this design choice.

When moving to larger arrays, particularly where the transceivers are divided among multiple chips, the brute-force solution is much more challenging. Much as data aggregation across the large physical size of the array proved challenging, the power cost of routing a high-frequency LO across large distances is significant. This motivates a desire to consider distributed LO generation architectures more suitable for the massive array regime which could avoid this problem. The danger is that distributed LO generation may introduce some de-synchronization of the LOs. As such, both the power consumption and the performance

[1]This applies to baseband analog or digital beamforming architectures, where each transceiver needs an LO, but also to architectures where multiple RF-beamforming subarrays are fused in the baseband domain.

impacts of these distributed LO generation architectures must be considered. This section introduces three main LO generation architectures and presents a model which captures the LO subsystem power consumption. In subsequent sections we will analyze the performance of large MU-MIMO arrays with distributed LO generation and derive design intuitions which ensure proper array synchronization.

4.1.1 LO Subsystem Components

Regardless of how the LO architecture is designed, at the very minimum the entire array must share a single low-frequency reference. If this were not the case, it would be impossible to reliably synchronize the entire array. This case has been studied in the literature, and it is shown that with asynchronous operation the array has a "coherence time" beyond which a calibration procedure must be re-initiated [119–122]. However, as long as the entire array shares a single XO, all other architectural decisions are fair game.

With this assumption, the entire LO subsystem consists of distribution network for the low-frequency (XO) reference, one or more VCO/PLLs, a distribution network for the LO itself, and finally the load, which almost always consists of the up-/down-conversion mixers. We can express the power consumption of the entire LO chain as:

$$P_{LO} = P_{load} + P_{distr} + P_{VCO} + P_{PLL} + P_{ref}. \tag{4.1}$$

Here P_{load} represents the power that must be delivered to the load (e.g. the mixers), P_{distr} is the power required to route the high-frequency LO, P_{VCO} is the power burned in the VCO, P_{PLL} is the power burned in the rest of the PLL, and P_{ref} is the power for reference distribution.

The load power depends on the mixer design, which is determined by its signal-path specifications such as noise figure and conversion gain or loss. As such, the mixer design does not depend on the LO chain design, but rather on the signal chain design, and therefore does not change with LO architecture.

In contrast, the LO and reference distribution networks are largely determined by the LO generation hierarchy and specifically the physical distance over which these signals must be routed. Furthermore, since the reference is much lower frequency than all the other signals in the array, its distribution network consumes very little power compared to other LO components. Therefore, it is generally safe to neglect the power consumed in the reference distribution.

The VCO power is largely set by its performance requirements. Various noise mechanisms in the VCO lead to random fluctuations in its phase compared to that of an ideal oscillator, referred to as phase noise. Noisy LOs can lead to undesirable degradations in the radio performance, such as constellation error, out-of-band emissions, and sensitivity to jammers. A VCO can be characterized by its figure of merit (FoM), which is a function of the quality factor and circuit architecture. In general, the amount of VCO phase noise is inversely

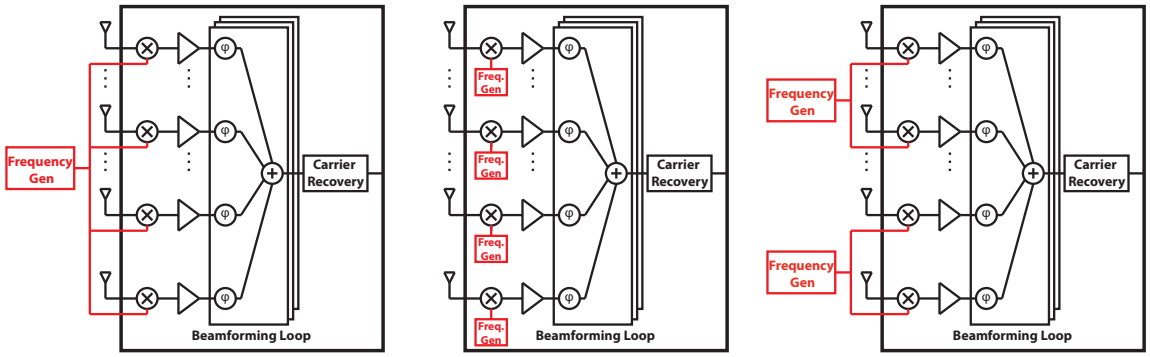


Figure 4.2: Three main LO distribution architectures. (a) Central carrier generation (CCG), (b) Local carrier generation (LCG), (c) Generalized carrier generation (GCG).

proportional to its DC power consumption [123–125].

$$\mathcal{L}(f_\Delta) = \frac{(\frac{f_{LO}}{f_\Delta})^2 FoM_{VCO}}{P_{VCO}} \tag{4.2}$$

Given a phase noise specification, the VCO power consumption is increased until that spec is met.

It has been shown that in arrays, the uncorrelated phase noise of individual VCOs is averaged to yield an improvement proportional to the number of oscillators [119, 126, 127]. This means that, compared with a single VCO, an array of S VCOs can relax the performance of each individual VCO by S; consequently, each VCO's power can be reduced by the same factor. As a result, to first order the power-performance point of an array of VCOs is independent of the number of VCOs.

In contrast, the other elements of a PLL such as phase detector and divider contribute overhead power which does not depend on the required performance level. Therefore, even if the VCO specifications can be relaxed to save power, the power consumption from the remaining PLL components cannot be reduced.

In summary, the LO subsystem consists of a variety of components and stages. Of these, the load, VCO, and reference distribution consume approximately the same power regardless of LO architecture. Instead, it is mainly the PLL overhead and the carrier distribution network which can be changed as a function of the LO architecture. The goal is then to find the optimal architecture which will manage these two sources of LO power.

4.1.2 Comparison of LO Distribution Architectures

Figure 4.2 presents several different LO distribution architectures. The conceptually simplest option is a central carrier generation (CCG) scheme, where a single central PLL generates the LO with the desired phase noise profile and that LO is distributed at the carrier frequency to each element (Figure 4.2a). In this scenario, the LO distribution network needs enough gain

to overcome the large loss associated the long-distance routing and splitting; the distribution buffers would therefore burn a large amount of power.

The opposite option is a local carrier generation (LCG) scheme, where a local PLL is used at each element and all the PLLs are locked to a common low-frequency reference (Figure 4.2b). In this scenario the power consumption is dominated by the hundreds of PLLs since the mm-wave distribution is very short and the reference distribution consumes a trivial amount of power. As described above, by exploiting the averaging of uncorrelated phase noise, to first order the combined power of the VCOs in LCG should equal the power of the VCO in the CCG scheme [127]. As a result, in the LCG scheme it is the *PLL overhead* rather than the VCO which dominates the power consumption due to the huge number of PLLs.

The trade-off between the CCG and LCG schemes comes from balancing distribution power against PLL overhead power. In practice we can consider a generalized carrier generation (GCG) scheme, consisting of several PLLs that each serve multiple elements (Figure 4.2c).

The power of different LO distribution architectures can be compared quantitatively as a function of N, the number of elements per PLL. $N = 1$ corresponds to the LCG scheme and $N = M$ to the CCG scheme, where M is the number of array elements. As discussed above, P_{distr} and P_{PLL} are most closely tied to the overall system architecture. P_{distr} accounts for the power needed to overcome loss in the distribution network, which primarily consists of routing loss and the loss in power splitters (*excess* loss above the desired power splitting). It is important to note that these losses increase with frequency. P_{PLL} depends on both the design of the PLL as well as the architectural choice of how many PLLs to use.

To quantitatively compare these architectures, a model for the routing loss is needed. Consider a scenario with an M-element array, implemented with a unit IC which contains P elements. Therefore, there are M/P such ICs. For simplicity, we model the on-chip distribution and splitting as "free" since the chip is small relative to the wavelength and the metal stackup is high quality. Instead, only the off-chip (board or package) routing loss is considered. As a function of the number of elements per PLL, N, the routing loss is

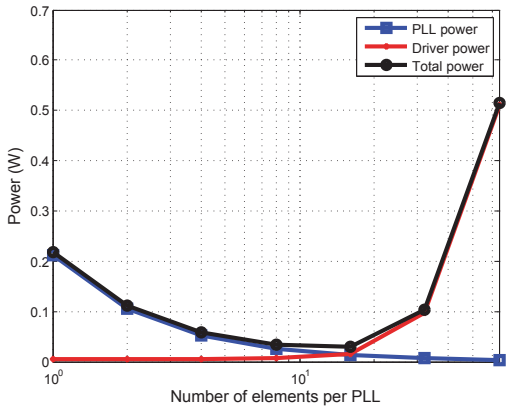
$$L_{route} = \frac{1}{2}\sum_{s=\log_2 P_X}^{\log_2 N_X - 1} \frac{2^s}{\sqrt{M}} D_X L_{mm} + \frac{1}{2}\sum_{s=\log_2 P_Y}^{\log_2 N_Y - 1} \frac{2^s}{\sqrt{M}} D_Y L_{mm} \tag{4.3}$$

This models the use of an H-tree to distribute the LO on the board or package, with loss L_{mm} per unit millimeter and total array x/y dimensions of D_X by D_Y. Similarly, the splitting loss from using a S-way splitter is

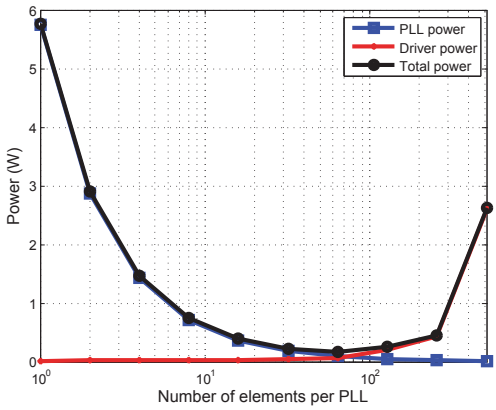
$$L_{sep} = \max\left(0, \log_S N/P\right) L_{split} \tag{4.4}$$

Using (4.2), it is possible to find the VCO's DC power consumption for a given phase noise spec, P_{VCO}. In dBm, the total distribution loss is then

$$P_{distr} = P_{VCO} + 10\log_{10}\eta_{osc} + L_{route} + L_{sep} - 10\log_{10}\eta_{driver} \tag{4.5}$$



(a)



(b)

Model Parameter	5 GHz	75 GHz
Number of elements	64	512
Transceivers/chip	8	32
4-way splitter loss (dB)	0.7	1.5
Routing loss (dB/mm)	0.06	0.2
VCO FoM (dBc/Hz)	177	177
VCO PN @ 1MHz offset (dBc/Hz)	-110	-90
VCO efficiency	20%	15%
PLL overhead mW)	3.5	10
LO Driver efficiency	20%	10%

(c)

Figure 4.3: LO subsystem power consumption versus PLL hierarchy. (a) 5GHz array. b) 75GHz array. (c) Model parameters.

where the driver amplifiers have efficiency η_{driver}. Note that all the N-dependence in this equation arises from the L_{route} and L_{sep} terms.

To illustrate the significance of these design choices, this model is used to analyze the power consumption of the LO subsystem in a cellular band (5GHz) and E-band (75 GHz) massive MIMO array. Figure 4.3 presents the results of this architectural analysis, plotting the power consumption as a function of N in both scenarios. In both low- and high-frequency scenarios, the choice of architecture is very significant to the overall power and complexity of the LO chain. By choosing the optimum architecture, the power can be reduced by 5-10x compared to a CCG or LCG implementation.

This architectural comparison makes it clear that massive MIMO arrays require careful design of the LO subsystem to manage power and complexity. In particular, the optimum architecture involves distributed generation of the LO which potentially runs the risk of de-

synchronizing the elements. In light of these architectural optimizations, it is important to analyze how the choice of LO architecture influences the performance of the overall system. How does distributed frequency generation influence inter-element and inter-user synchronization? The remainder of this chapter will study how the LO generation architecture affects the achievable performance of large array systems. We will use the CCG and LCG schemes to illustrate extremes of behavior and finally show that with multi-user operation, the performance of a GCG scheme strongly depends on the choice of N.

4.2 Phase Noise Filtering Loops

In a wireless link, the instantaneous phase of the received signal is a function of the transmitted data sequence, the transmitter's phase and frequency drift, the phase of the channel propagation, and the receiver's phase and frequency drift. As such, the receiver is equipped with a number of signal processing loops which attempt to estimate and cancel these various phase contributions, with the goal of isolating the phase of the transmitted data sequence from the other contributions. In general, channel estimation is used to measure the propagation environment, PLLs are used to syntesize phase- and frequency-stable LOs, and carrier recovery is used to track residual phase and frequency offsets in the baseband data prior to demodulation.

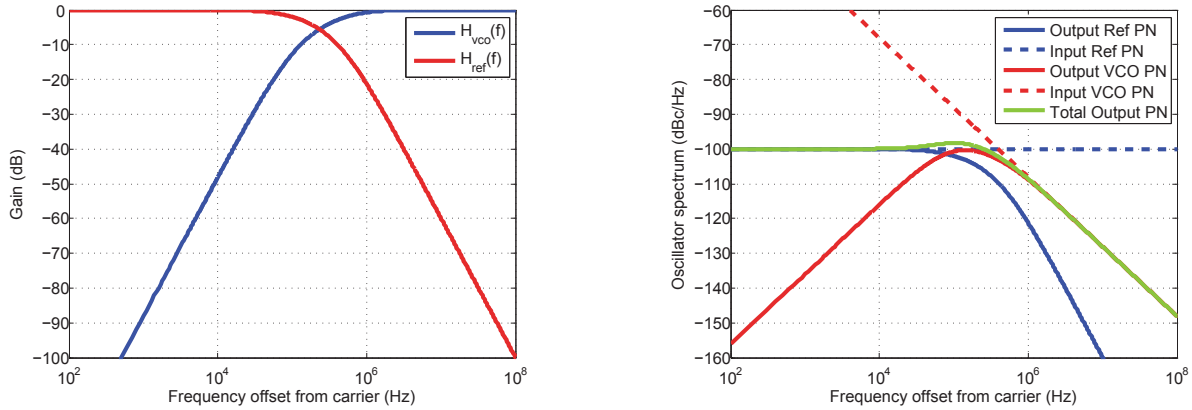
4.2.1 Phase-locked Loops

As described above, PLLs are used to generate a stable carrier from a low-frequency reference. As part of this operation, PLLs filter the phase noise of both the XO and the VCO, making it possible to achieve very good phase noise characteristics at the carrier.

The phase noise of an open-loop oscillator follows a Wiener process, which describes a random walk in phase. The phase variance in this case is well-known: $\sigma^2[n] = n\sigma_0^2$ [119, 120, 122, 128–130]. In the frequency domain this creates a $1/f^2$ characteristic. The time-domain variance and the frequency-domain linewidth both characterize the quality of the oscillator: a very high quality oscillator will have low variance/linewidth while a lower quality one has large variance and linewidth.

The reference comes from a high quality XO which achieves excellent phase noise performance due to its very high Q. In fact, the XO phase noise is good that typically its $1/f^2$ noise can be neglected, and instead the phase noise is dominated by the white thermal jitter introduced by its distribution buffers. In contrast, the VCO's phase noise consists primarily of its $1/f^2$ PSD.

The PLL's loop filter is designed to filter the XO and VCO phase noise (see [131] for a discussion of how the loop filter is designed in a common charge-pump PLL). The loop filter acts to low-pass filter the reference phase noise while high-pass filtering the VCO phase



(a) (b)

Figure 4.4: (a) PLL reference and VCO noise transfer functions. (b) Phase noise PSDs at the input and output of the PLL.

noise.

$$\begin{aligned} H_{ref}(s) =& \frac{f_{LO}}{f_{ref}} \frac{H_{LF}(s)}{1 + H_{LF}(s)} \\ H_{vco}(s) =& \frac{1}{1 + H_{LF}(s)} \end{aligned} \tag{4.6}$$

Within the PLL bandwidth, B_{PLL}, the output phase tracks the reference while any VCO noise is suppressed. Outside of the PLL bandwidth, the VCO operates in open loop and all of its noise therefore appears at the output. Figure 4.4 shows the PSD of a PLL's noise and its different contributions. These transfer functions are typically second order, but significant design intuition can be obtained by assuming that all phase noise below the PLL bandwidth comes from the reference and all noise above the PLL bandwidth comes from the VCO.

The LO's phase noise variance can be written in terms of the reference and VCO contributions. The reference phase noise consists of a $1/f^2$ part and a much stronger white component from the distribution buffers. In contrast, the VCO phase noise consists primarily of a $1/f^2$ contribution which is filtered by the PLL transfer function and is therefore wide-sense stationary [130, 132, 133]. The total phase variance is then:

$$\sigma_\phi^2 = n\sigma_{ref}^2 + \sigma_{dist}^2 + \sigma_{VCO}^2 \approx \sigma_{dist}^2 + \sigma_{VCO}^2 \tag{4.7}$$

Since the reference and VCO noise have different PSDs, the total phase noise variance is a convex function of the PLL bandwidth. To first order, the optimum PLL bandwidth is the frequency at which the reference and VCO noise contribute equal amounts of noise to the output, denoted by B_{jitt}. Graphically, this can be identified as the point where the open-loop reference and VCO phase noises intersect.

The phase noise $\phi[n]$ impacts the transceiver via the oscillator voltage $v[n] = e^{j\phi[n]}$. The statistics of this process have been studied in [130, 132, 133], where the power spectral density (PSD), $S_v(\omega)$, is derived. We use the prototype PSD in Fig. 4.4, which is representative of a wide range of PLL architectures, to analyze the effect of phase noise on communication systems. This allows us to make design decisions in terms of spectra which are well-characterized and understood.

4.2.2 Channel Estimation

As discussed in Chapter 2, channel estimation is used to track variations in the wireless channel. In array receivers and MIMO systems, channel estimation is performed on a per-element basis to estimate the full $M \times K$ channel matrix. Since variations in LO phase are indistinguishable from variations in the phase of the wireless channel, this per-element channel estimation naturally also tracks the slow fluctuations in the LO phase at every element, essentially acting as a per-element carrier recovery loop. As such, the channel estimation loop sets an absolute lower bound on the frequency at which phase noise will impact the receiver. Depending on mobility, common channel estimation periods can range from 10 μs to 1 ms, filtering phase noise below 10 kHz or so. This is another reason why the $1/f^2$ noise of the reference can generally be neglected.

4.2.3 Carrier Recovery

The PLL is designed for optimal phase and frequency stability but is ultimately limited by the quality of the reference and circuits which are a function of technology limits. Therefore, carrier recovery (CR) loops are used to compensate for residual carrier phase and frequency offsets. The key idea behind CR algorithms is to use information in the data sequence itself to identify the ideal (true) phase trajectory of the modulated data and the deviations from it, and thereby compute correction signals which can minimize these deviations.

Many implementations of CR with varying complexity and performance trade-offs exist [134, 135]. These can be divided into two main classes. In data-aided (DA) CR loops, known pilots are time or frequency multiplexed with the data, which the receiver uses to compute an instantaneous estimate of the phase and frequency error. In non-data aided (NDA) CR loops, the phase/frequency error is estimated directly from the data. An example of an NDA algorithm is the decision-directed estimator, which compares the phase of the measured symbol with the true constellation point to estimate the instantaneous phase error.

Regardless of whether DA or NDA estimation is used, the estimate of the phase error is filtered and then used to compute a correction which cancels out the time-varying phase and frequency offsets. Filtering and correction can be applied in either a feedforward or feedback scheme. Regardless of the actual implementation, the CR loop acts as a phase filter which tracks frequency drift and cancels out phase noise within the CR bandwidth.

The CR implementation depends on the modulation scheme and other details of the physical layer such as pilot locations. In subsequent analysis we will describe CR algorithms suitable for both OFDM and SC modulations.

4.3 System Model for Carrier Generation in Large Arrays

We propose a carrier generation and synchronization subsystem built around the following principles.

- **Common XO Reference:** Every element in the array shares a single low-frequency reference, ensuring basic frequency synchronization. LCG, CCG, or GCG LO generation architectures are used to multiply this reference up to the carrier frequency. These architectures are parametrized by the number of transceivers per PLL, N.
- **VCO Phase Noise Scaling:** In an array with M/N VCOs, each VCO's phase noise spec is *relaxed* by a factor of M/N compared to the baseline CCG (which only has a single VCO in the whole array). This design choice is necessary to manage the aggregate VCO power consumption. To first order, this means that the total VCO power in the entire array is independent of M.
- **Centralized Carrier Recovery:** There are no per-element carrier recovery loops; instead, all CR is performed in a centralized fashion. This is necessary because CR loops require high SINR to operate well, so must follow beamforming, zero-forcing, and equalization.

In this section we briefly introduce system models for OFDM and single-carrier modulations, then proceed in subsequent sections to analyze how phase noise impacts SIMO and MIMO operation.

4.3.1 OFDM System Model

Consider an OFDM communication system using N_{sc} subcarriers, each with bandwidth B_{sc}, for a total channel bandwidth of $B = N_{sc}B_{sc}$. All symbols have energy E_s and are statistically independent. The transmitter modulates a data and pilot sequence $\{d_k\}$ onto the subcarriers and transmits time-domain sequence $s[n]$ through a channel with impulse response $h[n]$ and additive white Gaussian noise $w[n]$ with variance σ^2. The receive signal is corrupted by the unit energy oscillator phase noise voltage drawn from the process $v[n]$. Since the phase noise does not alter the statistical properties of the additive white noise, the received signal is

$$y[n] = (s[n] \odot h[n])v[n] + w[n] \tag{4.8}$$

The multiplication by $v[n]$ creates a circular convolution in the frequency domain between the data and oscillator spectra. Denoting the DFT of the phase noise sequence, channel, and white noise as $\{Q_k\}, \{H_k\}$, and $\{W_k\}$, respectively, the received frequency domain samples $\{Y_k\}$ are

$$Y_k = d_k H_k Q_0 + \sum_{j \neq k} d_j H_j Q_{i-j} + W_k \tag{4.9}$$

Phase noise has two main effects. First, each symbol is affected by a common phase error (CPE) Q_0, which appears as the multiplication of the complex channel gain equally across all subcarriers. Second, the symbols are corrupted by inter-carrier interference (ICI): mixing of data streams caused by loss of orthogonality between subcarriers.

Impact of Subcarrier and PLL Bandwidths

To gain a deeper understanding of the SINR, the CPE and ICI can be characterized in terms of the phase noise PSD. Define $\mathbf{1}_{N_{sc}}$ as a rectangular window of N_{sc} ones. Then the moving average filter of length N_{sc} has impulse response $h_{N_{sc}} = \frac{1}{N_{sc}} \mathbf{1}_{N_{sc}}$ and continuous-time frequency response

$$H_{N_{sc}}(\omega) = \frac{sin(N_{sc} \frac{\omega}{B})}{N_{sc}^2 \frac{\omega}{B}} \tag{4.10}$$

The width of this filter's main lobe is B_{sc}. Since the CPE is the DC component of the windowed phase noise, its energy is simply the energy of the random process generated by filtering $v[n]$ with $h_{N_{sc}}$. This can be expressed as:

$$\mathbb{E}[|Q_0|^2] = \int_{-\infty}^{\infty} S_v(\omega) |H_{N_{sc}}(\omega)|^2 d\omega \tag{4.11}$$

Since $v[n]$ has unit energy, the ICI energy is:

$$\mathbb{E}[|ICI|^2] = 1 - \mathbb{E}[|Q_0|^2] = \int_{-\infty}^{\infty} S_v(\omega)(1 - |H_{N_{sc}}(\omega)|^2) d\omega \tag{4.12}$$

This reveals a key insight: the CPE is contributed primarily by oscillator energy below the subcarrier bandwidth while ICI comes mainly from energy above the subcarrier bandwidth. This provides an intuitive explanation for how the subcarrier bandwidth affects the SINR. If B is kept fixed as N_{sc} is increased, the bandwidth of $h_{N_{sc}}$ is reduced. This will reduce the CPE energy and increase ICI, degrading the SINR.

Fig. 4.5 shows the simulated CPE and ICI power as a function of the VCO phase noise compared with the predictions in (4.11) and (4.12). These match well, including the signal energy degradation at very high phase noise levels. Also, over a wide range of phase noise levels, the ICI power increases linearly with the VCO's phase noise (1 dB/dB).

The CPE and ICI transfer functions can be cascaded with the PLL transfer functions to isolate the contributions of both reference and VCO to the CPE and ICI. Since the reference transfer function is lowpass, its phase noise will mostly contribute CPE and little ICI.

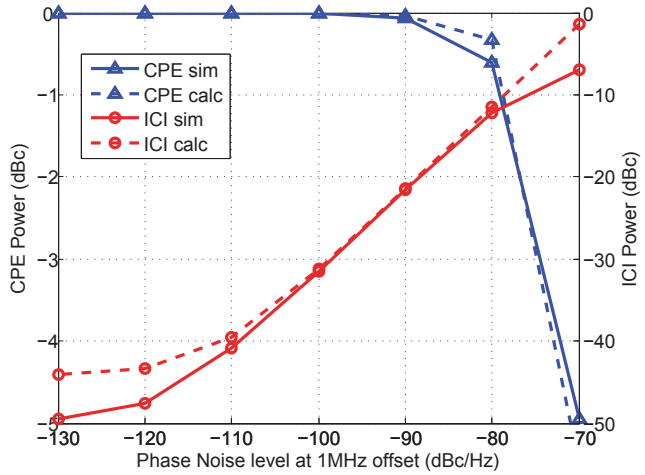


Figure 4.5: Simulated and predicted results of ICI and CPE power based on the model in (4.11) and (4.12), for a PLL with 200kHz bandwidth and B_{sc} of 624kHz.

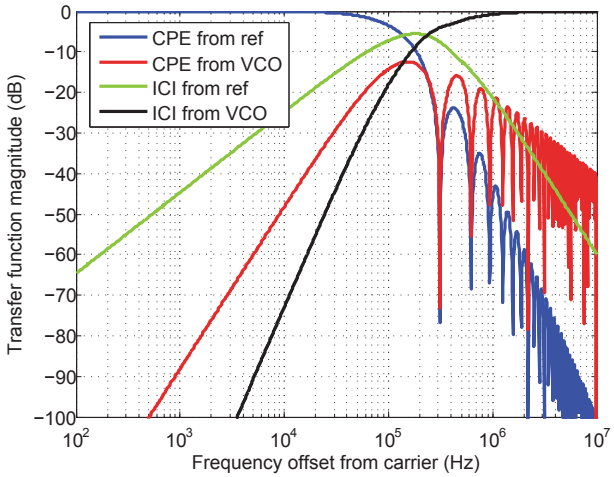


Figure 4.6: Cascaded transfer function of reference and VCO phase noise to the CPE and ICI, with $B_{PLL} = 200kHz$ and $B_{sc} = 312kHz$. These results validate the conclusion that reference noise dominates the CPE while VCO noise dominates the ICI generation.

Similarly, the VCO's phase noise will primarily add ICI rather than CPE. These conclusions are illustrated in Fig. 4.6 which captures the cascade of the PLL transfer functions with $H_{N_{sc}}(\omega)$.

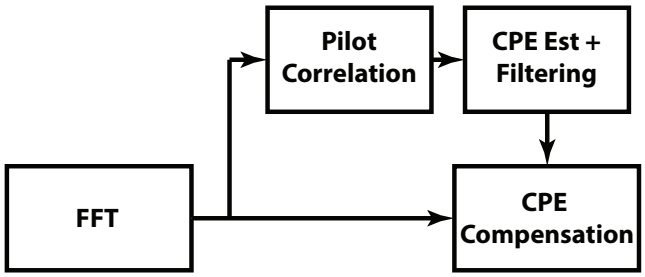


Figure 4.7: Feedforward pilot-based carrier recovery for OFDM modulation

Carrier Recovery for OFDM Signals

It is common in OFDM systems to reserve a small number of subcarriers for pilots [128, 136] which are used at the receiver to estimate various imperfections. Using the N_p pilots, found at the set of indices $\mathcal{P}$, the CPE can be estimated as:

$$\hat{Q}_0 = \frac{1}{N_p} \sum_{k \in \mathcal{P}} \frac{Y_k}{d_k} \tag{4.13}$$

The CPE can then be compensated in a feedforward manner by dividing all received symbols by this estimate (Figure 4.7). Ideally, this cancels the sample mean of the phase noise process on a symbol-by-symbol basis, giving an average SINR of

$$SINR = \frac{\mathbb{E}[|Q_0|^2]}{\mathbb{E}[|ICI|^2] + \frac{\sigma^2}{E_s}}. \tag{4.14}$$

Building on the discussion above, CPE estimation and correction is equivalent to a CR loop with bandwidth B_{sc}. More complex carrier recovery schemes have been proposed for OFDM signals. As described in (4.9), on top of the CPE, phase noise also generates ICI. Through joint processing of all the subcarriers, it is possible to estimate these higher-order terms (Q_1, Q_2, etc). Once this estimate is obtained, CPE and some ICI terms can be mitigated by deconvolving the received frequency-domain signal with the estimated DFT of the phase noise. In practice this technique is not used due to its high complexity, both in ICI estimation and deconvolution. As a result, in OFDM systems the CR bandwidth is almost always restricted to the subcarrier bandwidth.

4.3.2 Single-carrier System Model

In a single-carrier communication link using channel bandwidth B, the transmitter sends data symbols $s[n]$ at interval $1/B$. All symbols have energy E_s and are statistically independent. The data sequence propagates through a channel with impulse response $h[n]$ and additive white Gaussian noise $w[n]$ with variance σ^2. The receive signal is corrupted by a unit energy

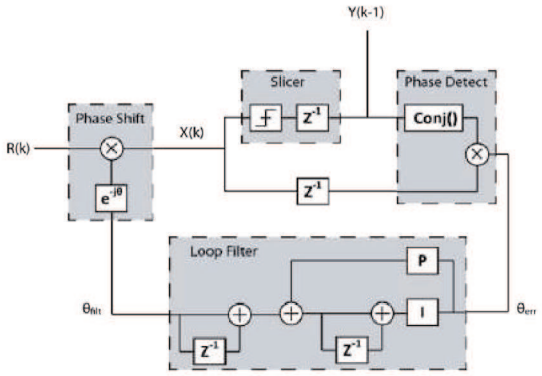


Figure 4.8: Block diagram of decision-directed carrier recovery loop.

oscillator phase noise voltage drawn from the process $v[n] = e^{j\phi[n]}$. Since the phase noise does not alter the statistical properties of the additive white noise, the received signal is

$$y[n] = (s[n] \star h[n])v[n] + w[n] = (s[n] \star h[n])e^{j\phi[n]} + w[n] \tag{4.15}$$

The phase noise in a single-carrier link creates phase errors in the received data symbols. The variance of this phase error is the variance of the phase noise process $\phi[n]$, given by σ_ϕ^2, which comes from the integrated energy in the phase noise process up to the channel bandwidth B. For this reason, this error is referred to integrated phase error. If the integrated phase error is large enough, it can lead to demodulation errors.

Carrier Recovery for SC Modulation

Carrier recovery loops can be used in single-carrier modulation to suppress phase and frequency errors, such as the phase noise sequence. As a time-domain modulation scheme, it is natural to implement single-carrier CR loops using a time-domain adaptive filter.

One example of a CR loop is the decision-directed PLL shown in Figure 4.8. Once the loop is locked, the phase of the measured symbol is compared to the true constellation point to compute the instantaneous phase error. This error signal is fed back through a loop filter to apply a correction to the signal path. The loop filter typically contains two poles at DC to eliminate static phase and static frequency errors.

This operation is essentially identical to the PLL of Figure 4.1 where the "reference" is the sequence of ideal constellation points. As a result, the CR loop applies a high-pass transfer function to the input phase noise. If the CR bandwidth is set very low, the loop will only remove static phase and frequency offsets. However, if the bandwidth of the CR is allowed to be large it can also track and filter out the instantaneous phase error arising from high-frequency components of the phase noise.

CR loops are important in any communication links to manage the impact of phase and frequency errors. In mm-wave systems, due to the high carrier frequency, CR loop design is

particularly important and has strong impacts on the achievable performance. This topic is discussed in detail in [137].

4.4 Phase Noise in Single-User Arrays

We will first analyze the case of a single-user beamforming array (SIMO configuration) to discover how phase noise and LO generation impacts the performance in this scenario. In a SIMO array, it is important to consider how the phase noise at each element interacts with the beamforming operation. Because we are analyzing SIMO operation, it is sufficient to only consider conjugate beamforming.

In general, the phase noise contribution at each element can be expressed as the combination of a spatially correlated and spatially uncorrelated term. This formulation captures the behavior of LCG, CCG, and GCG LO architectures. This decomposition applies on a per-element basis in the LCG architecture and on a per-subarray basis in the GCG architecture. Finally, for the CCG architecture, there is no uncorrelated phase noise contribution.

For a single-carrier modulation, the beamformed signal is given by:

$$x[n] = \boldsymbol{h}^H \mathbf{diag}\{e^{j\phi_c[n]+j\phi_{u,i}[n]}\}(\boldsymbol{h}s[n] + w[n]) \tag{4.16}$$

Here $\mathbf{diag}\{\mathbf{a}\}$ forms a diagonal matrix from vector $\mathbf{a}$. The phase noise at each element can in general be expressed as a component ϕ_c which is correlated across all elements and a component $\phi_{u,i}$ which is uncorrelated and depends on element index i.

In OFDM, the beamformed signal is:

$$X_k = \boldsymbol{H}_k^H \mathbf{diag}\{Q_0^{(c)} + Q_{0,i}^{(u)}\}\boldsymbol{H}_k d_k + \boldsymbol{H}_k^H(\mathbf{ICI} + W_k) \tag{4.17}$$

The CPE at each element is split into correlated and uncorrelated components, $Q_{0,i} = Q_0^{(c)} + Q_{0,i}^{(u)}$.

Consider only the signal component for each of the above expressions. For the single-carrier modulation, we have:

$$x[n] = e^{j\phi_c[n]} \sum_{i=1}^{M} |h_i|^2 e^{j\phi_{u,i}[n]} s[n] \tag{4.18}$$

For the OFDM modulation, the result is:

$$\begin{aligned} X_k =& Q_0^{(c)} d_k + \sum_{i=1}^{M} |H_{k,i}|^2 Q_{0,i}^{(u)} d_k \\ =& \sum_{n=1}^{N_{sc}} e^{j\phi_c[n]} d_k + \sum_{i=1}^{M} (|H_{k,i}|^2 \sum_{n=1}^{N_{sc}} e^{j\phi_{u,i}[n]}) d_k \end{aligned} \tag{4.19}$$

These results are very similar across the two modulation schemes. First, the correlated phase noise component looks indistinguishable from phase noise in a SISO link. Second, the uncorrelated phase noise component is spatially averaged across the array elements to form an effective signal gain term. The aggregate impact of the uncorrelated phase noise consists of a signal energy filtering along with fluctuations that cause random errors. We will analyze each in turn.

4.4.1 Signal Energy Loss from Uncorrelated Phase Noise

Because the channel fading statistics are independent of the phase noise statistics, we can separate out the two effects when computing the signal gain. For the single-carrier modulation, the gain is given by:

$$G_{sc} = \mathbb{E}[\frac{x[n]}{s[n]}] = \mathbb{E}[\sum_{i=1}^{M} e^{j\phi_{u,i}[n]}] \tag{4.20}$$

For OFDM modulation, the gain is:

$$G_{ofdm} = \mathbb{E}[\frac{X_k}{d_k}] = \mathbb{E}[\sum_{i=1}^{M} (\sum_{n=1}^{N_{sc}} e^{j\phi_{u,i}[n]})] \tag{4.21}$$

In these expressions, $\mathbf{E}[]$ denotes expectation. Importantly, the phase noise at the output of a PLL is wide-sense stationary. Therefore, the process is ergodic and the spatial and temporal averages are one and the same. Putting these observations together:

$$G_{sc} = G_{ofdm} = \mathbf{E}[e^{j\phi_i}] = e^{-\sigma_\phi^2} \tag{4.22}$$

using the characteristic function of the phase noise process [130], where σ_ϕ^2 is the variance of the phase noise.

What does this mean? The uncorrelated part of the phase noise imparts a random phase to the signal at each element. When summing these signals together during the beamforming step, they do not sum perfectly in phase. Instead, there is a beamforming error caused by the uncorrelated phase noise, which leads to signal energy loss. Naturally, the magnitude of this energy loss is proportional to the variance of the phase noise.

4.4.2 Gain and Phase Fluctuations from Uncorrelated Phase Noise

In addition to the static signal energy loss, the uncorrelated phase noise leads to gain and phase errors in the received signals.

In a single-carrier modulation, we can represent the total signal gain as:

$$e^{j\phi_c[n]} \sum_{i=1}^{M} |h_i|^2 e^{j\phi_{u,i}[n]} = e^{j\phi_c[n]} \mathbf{E}[e^{j\phi_u}] g[n] e^{j\theta[n]} \boldsymbol{H}^H \boldsymbol{H} \tag{4.23}$$

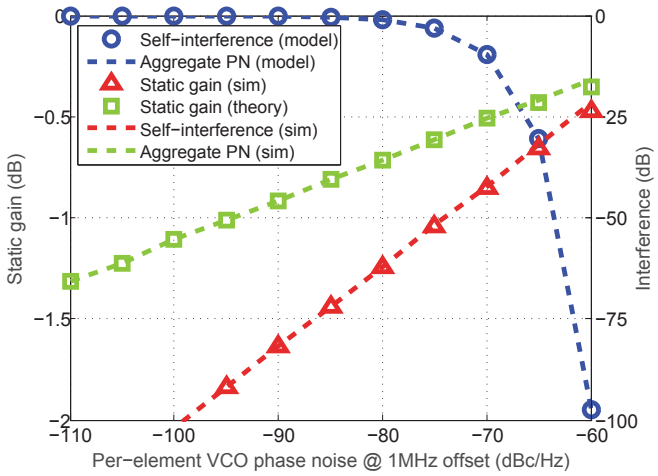


Figure 4.9: Characterization of error mechanisms from uncorrelated VCO phase noise in a 16-element array.

where n is the time index. The sum in (4.16) is represented as the product of its mean and time-varying residual gain and phase $g[n]$ and $\theta[n]$.

The residual gain errors come from rapid fluctuations of the instantaneous gain about its average value[2]. The residual phase errors form an equivalent phase noise process $\theta[n]$ at the array level. The total phase noise at the output of the beamformer is $\phi_c + \theta[n]$; this phase noise can be filtered by the CR loop exactly as in the single-element case. As a result, following the CR loop the leftover noise consists of high-frequency phase noise and the gain self-interference. While the residual high-frequency phase noise is also present in a single-element receiver, the presence of gain self-interference is unique to array based systems and arises specifically from uncorrelated phase noise.

The gain and phase errors can be analyzed by considering the Taylor expansion

$$\sum_{i=0}^{M-1} e^{j\phi_i} \approx \sum_{i=0}^{M-1} (1 - \frac{1}{2}\phi_i^2) + j \sum_{i=0}^{M-1} \phi_i \tag{4.24}$$

This approximation is valid for small levels of phase noise. In this regime it is clear that the real part is very close to 1 while the imaginary part comes from the linear average of the phase noise at each element. Consequently for small levels of phase noise

$$\begin{aligned} \theta[t] &\approx \sum_{i=0}^{M-1} \phi_i \\ g[t] &\approx \sum_{i=0}^{M-1} (1 - \frac{1}{2}\phi_i^2) \end{aligned} \tag{4.25}$$

[2]These gain variations could be tracked by a *fast* AGC loop. However as discussed below this is only applicable to a single-user scenario.

For large phase noise variance it is difficult to obtain analytical expressions. Instead, we conduct Monte Carlo simulations to empirically obtain the distribution of the gain error. Figure 4.9 shows the static gain, gain variations, and residual phase noise for a 75GHz LO as a function of the VCO's phase noise level, for $M = 16$ and 5MHz PLL bandwidth. The theoretical analysis for the static gain error and the Taylor approximations for gain and phase error match the simulations very well.

For the single-carrier modulation, we can draw three interesting conclusions. First, gain self-interference arises uniquely in array systems due to uncorrelated phase noise. Second, for small levels of phase noise, phase variations dominate over gain variations. Third, the gain and phase errors are averaged by a factor of M as seen in (4.25).

In an OFDM modulation, we could similarly break the CPE sum in (4.17) into its mean and gain/phase variations. However, the CPE consists of the phase noise average over a sequence of N_{sc} phase noise samples. Therefore, the variance of these CPE gain/phase errors is reduced by a factor of N_{sc} compared to the equivalent single-carrier case. These can be neglected. Instead, the fast-varying phase noise impact in OFDM systems comes from the ICI term in (4.17). As long as the ICI is totally uncorrelated at each element (which is a reasonable assumption as discussed below), summation across the array will provide a gain of M in the signal-to-ICI ratio.

4.4.3 SINR for SIMO Arrays with Phase Noise

Both single-carrier and OFDM SIMO arrays experience similar impacts from uncorrelated phase noise. First, the signal energy is reduced due to beamforming errors. Second, residual array-level phase noise errors (whether gain/phase errors in single-carrier or ICI in OFDM) are averaged by a factor of M due to the array gain.

The latter observation justifies exploiting VCO scaling to save system power, as stated in the system design assumptions in Section 4.3. The design procedure is to select acceptable phase noise error budgets, design a reference SISO oscillator which achieves the required performance, and then scale that oscillator down by M/N for each physical VCO used in the system.

Interestingly, this system-level choice has significant repercussions on the achievable performance. Consider the reference SISO VCO, which has a certain phase noise level corresponding to phase variance of σ^2_{VCO}. Now, each physical VCO used in the array has higher level of phase noise characterized by variance $\frac{M}{N}\sigma^2_{VCO}$.

Recall that the signal energy loss from phase noise-mediated beamforming error depends exponentially on the phase variance (Section 4.4.1). This means that VCO scaling makes this effect worse. Quantitatively, the achievable SINR in both single-carrier and OFDM modulations is:

$$SINR = \frac{ME_s e^{-\frac{M}{N}\sigma^2_{\phi,uncorr}}}{\beta\sigma^2_{PN} + \sigma^2} \tag{4.26}$$

Here σ^2_{PN} is the phase noise variance, corresponding to either ICI in OFDM or gain/phase errors in single-carrier, *after* beamforming and carrier recovery; β is a constant of proportionality; and σ^2 is the white noise variance. The signal loss from beamforming error only depends on the uncorrelated portion of the phase noise.

The phase noise impacts two terms in (4.26). First, the signal energy is inversely proportional to the uncorrelated phase noise variance. Second, the total phase noise error depends on the final phase noise variance, after accounting for the filtering in the PLL, uncorrelated averaging across the array, and the filtering in the CR loop. This second contribution is minimized when the PLL bandwidth is chosen as $\max(B_{CR}, B_{jitt})$.

The relative power of correlated and uncorrelated phase noise components, as well as their PSDs, is determined by the LO generation architecture of the array. For the CCG scheme, all the reference and VCO phase noise is correlated at every element. For the LCG scheme, the reference phase noise is correlated at every element while the VCO phase noise is fully uncorrelated. Furthermore, due to the phase noise filtering property of PLLs, in the LCG scheme the correlated phase noise will appear below the PLL bandwidth while the uncorrelated phase noise lies above it. Therefore, the uncorrelated phase noise variance can also be controlled by the PLL bandwidth.

These observations suggest that there is an optimum PLL bandwidth. If the PLL bandwidth is chosen too low, it will let in excessive uncorrelated phase noise from the VCO, which will degrade the signal energy. If the PLL bandwidth is chosen too high, the total integrated error (from ICI or gain/phase errors) will be too large, degrading the SINR. What is the optimum bandwidth? It is hard to derive this analytically, but a good rule of thumb is to pick $B_{opt} = \max(B_{CR}, B_{jitt})$. This also presents an engineering knob to control the SINR loss. By increasing the CR bandwidth, it is possible to increase the PLL bandwidth and therefore suppress beamforming error.

Figures 4.10 and 4.11 plot the SINR versus PLL bandwidth for an OFDM and single-carrier array[3], respectively. These simulations clearly reveal the optimum PLL bandwidth as described above. For low PLL bandwidth, the performance is limited by the beamforming error; for high PLL bandwidth, the performance is limited by the integrated phase error or ICI.

Finallly, Figure 4.12 shows the performance of an OFDM array as a function of array size. Phase noise scaling is exploited to relax the oscillator performance proportionally to the array size. PLL bandwidth optimization is used to recover the performance loss from uncorrelated phase noise. This system uses 20MHz bandwidth at 2.5GHz carrier, with 64 subcarriers corresponding to subcarrier bandwidth of 312kHz. Phase noise scaling works as expected to maintain excellent performance across a wide variety of array sizes, for constant total VCO power consumption.

[3]The single-user array uses an IF-PLL at 5GHz, as described in [137].

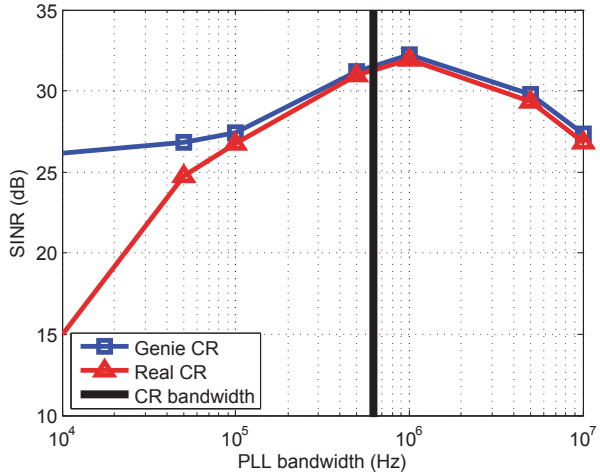


Figure 4.10: SINR versus PLL bandwidth for a 2.5GHz 64-element MIMO-OFDM array with B_{sc} of 624kHz and effective phase noise of -100dBc/Hz at 1MHz offset.

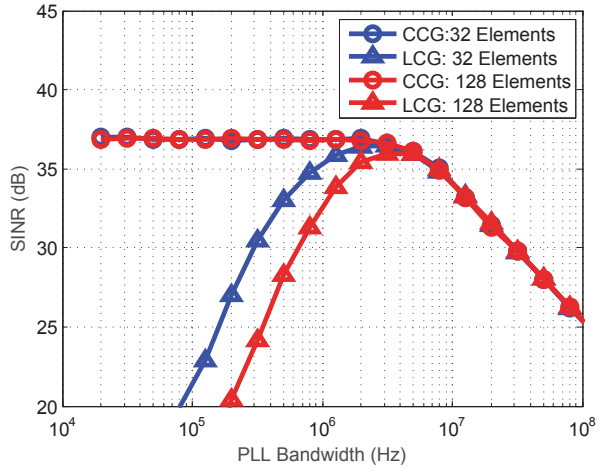


Figure 4.11: SINR versus PLL bandwidth for a 75GHz single-carrier massive MIMO array with CR bandwidth of 10MHz, reference noise at -140dBc/Hz and VCO effective phase noise of -90dBc/Hz at 1MHz offset.

4.5 Phase Noise in Multi-User Arrays

Compared to the single-user scenario, multi-user operation introduces spatial processing to separate out the user streams. For this analysis, we will focus on the zero-forcing beamformer. It is useful to think of the ZF beamformer as a conjugate beamformer H^H, followed by a zero-forcing matrix $(H^H H)^{-1}$. Thus the effective $K \times K$ channel before ZF is given by $A = H^H H$.

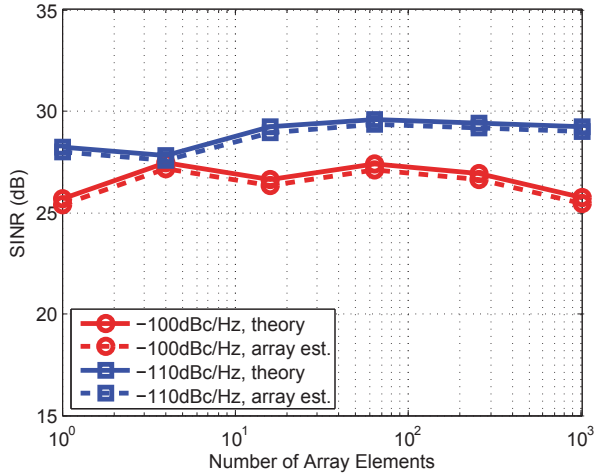


Figure 4.12: Simulated and predicted SINR for synchronous array with various levels of phase noise. $B_{PLL} = 500kHz$, $B_{sc} = 312kHz$, and $N_{sc} = 64$.

With a single-carrier modulation, the received signal in the presence of correlated and uncorrelated phase noise is

$$\begin{aligned}\hat{\boldsymbol{x}}[n] =& (\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H\mathbf{diag}\{e^{j\phi_c[n]+j\phi_{u,i}[n]}\}(\boldsymbol{H}\boldsymbol{s}[n]+\boldsymbol{w}[n]) \\ =& e^{j\phi_c[n]}(H^H\boldsymbol{H})^{-1}\boldsymbol{H}^H\mathbf{diag}\{e^{j\phi_{u,i}[n]}\}\boldsymbol{H}\boldsymbol{s}[n]+(\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H\boldsymbol{w}[n]\end{aligned} \tag{4.27}$$

Analogously, with OFDM modulation, the received signal is:

$$\begin{aligned}\boldsymbol{X}_k =& (\boldsymbol{H}_k^H\boldsymbol{H}_k)^{-1}\boldsymbol{H}_k^H\mathbf{diag}\{Q_0^{(c)}+Q_{0,i}^{(u)}\}\boldsymbol{H}_k\boldsymbol{d}_k+(\boldsymbol{H}_k^H\boldsymbol{H}_k)^{-1}\boldsymbol{H}_k^H(\mathbf{ICI}+\boldsymbol{W}_k) \\ =& Q_0^{(c)}\boldsymbol{d}_k+(\boldsymbol{H}_k^H\boldsymbol{H}_k)^{-1}\boldsymbol{H}_k^H\mathbf{diag}\{Q_{0,i}^{(u)}\}\boldsymbol{H}_k\boldsymbol{d}_k+(\boldsymbol{H}_k^H\boldsymbol{H}_k)^{-1}\boldsymbol{H}_k^H(\mathbf{ICI}+\boldsymbol{W}_k)\end{aligned} \tag{4.28}$$

For single-carrier modulation, the effective channel in the presence of phase noise is

$$\begin{gathered}\boldsymbol{A} = \hat{\boldsymbol{H}}^H\mathbf{diag}\{e^{j\phi_i}\}\boldsymbol{H} \\ A_{ii} = \sum_{i=0}^{M-1}|H_{ii}|^2e^{\phi_i} \qquad A_{lk} = \sum_{i=0}^{M-1}H_{il}H_{ik}^*e^{\phi_i}\end{gathered} \tag{4.29}$$

For OFDM modulation, the effective channel is

$$\begin{gathered}\boldsymbol{A} = \boldsymbol{H}_k^H\mathbf{diag}\{Q_{0,i}^{(u)}\}\boldsymbol{H}_k \\ A_{ii} = \sum_{i=0}^{M-1}|H_{ik}|^2Q_{0,i}^{(u)} \qquad A_{lk} = \sum_{i=0}^{M-1}H_{il}H_{ik}^*Q_{0,i}^{(u)}\end{gathered} \tag{4.30}$$

Regardless of modulation scheme, the diagonal elements A_{ii} are identical to the single user case; as such, the same self-interference effects are present. In addition, the multi-user

interaction must be considered. As can be seen in the off-diagonal terms, the uncorrelated phase noise causes a time-varying channel drift. Because this phase noise-corrupted channel is mismatched to the static ZF beamforming matrix, the ZF is unable to fully cancel inter-user interference, leading to a residual phase noise-induced zero-forcing error[4].

In the OFDM case, it is also important to study how the ICI behaves under beamforming. Consider just the ICI inflicted by subcarrier $k+1$ onto subcarrier k:

$$ICI_{k+1\rightarrow k} = \mathbf{G_{zf}^{(k)}} diag\{Q_{1,i}\}\mathbf{H^{(k+1)}}\mathbf{y_{k+1}} \tag{4.31}$$

The ICI experiences an effective phase noise channel composed of the true channel $\mathbf{H^{(k+1)}}$ scrambled by the uncorrelated ICI coefficients $\{Q_{1,i}\}$. This has two important effects. First, it decorrelates the ICI across the array. Second, it causes the ICI channels to lose orthogonality, creating *inter-user ICI* in addition to the self-ICI.

4.5.1 SINR for Multi-User Arrays with Phase Noise

The multi-user SINR depends on the channel structure, particularly the underlying correlation of the different users' channels. We capture these effects by modeling the SINR as:

$$SINR = \frac{S_u}{N_t + S_u N_p + \alpha\gamma \sum_{j=1}^{K-1} S_j N_p} \tag{4.32}$$

where S_u is the signal power, N_t is the thermal noise power, N_p is the phase noise power, and S_j the power of the j'th user. If uplink power control is applied this becomes simply

$$SINR = \frac{S_u}{N_t + S_u N_p + \alpha\gamma(K-1) S_u N_p} \tag{4.33}$$

The model parameters α and γ capture important effects. The parameter α describes the ratio of self- to inter-user interference power and depends on several factors including modulation scheme, phase noise levels, channel structure and correlation, and the distributions of self-interference and zero-forcing errors. γ is a purely architecture-dependent parameter which describes how the level of uncorrelated VCO phase noise depends on the array architecture. For LCG schemes, $\gamma = 1$ since all VCO phase noise is uncorrelated. For CCG schemes, $\gamma = 0$ because all VCO phase noise is correlated. For GCG, γ assumes an intermediate value that is a function of N.

Much as before, we can exploit VCO scaling by the ratio M/K to relax the performance of distributed oscillators. Fig. 4.13 shows the SINR for a multi-user synchronous array at 2.5GHz. The channel experiences Rayleigh fading, and the array utilizes zero-forcing beamforming to recover the user signals. The ratio M/K is fixed at 1/8 for $M \geq 16$. Each

[4]While the self-interference could be tracked by a fast AGC loop, tracking the inter-user interference would require a fast beamforming loop that tracks the effective channel A and adapts the ZF matrix accordingly. This is computationally quite intensive.

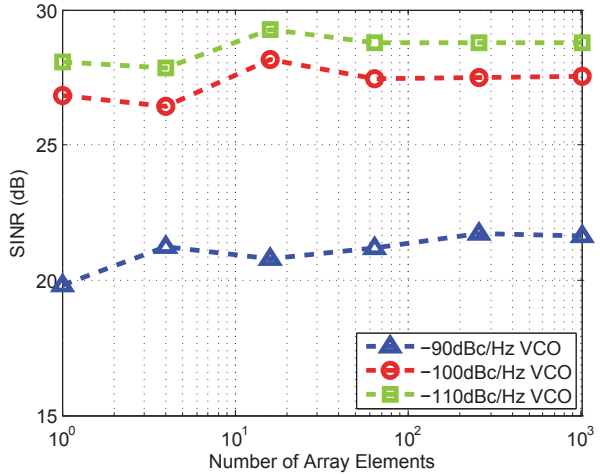


Figure 4.13: Simulated SINR with multi-user, synchronous array. $B_{PLL} = 500kHz$, $B_{sc} = 624kHz$, and $N_{sc} = 64$

.

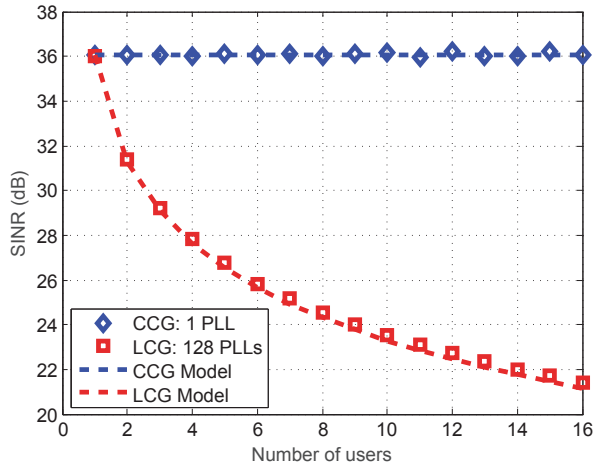


Figure 4.14: Average SINR vs number of users for 128-element array with CCG or LCG scheme. Carrier recovery bandwidth is 10 MHz and PLL bandwidth is 5 MHz.

element's phase noise is scaled with the ratio M/K. As expected, the multi-user SINR is the same as the single-user case. Furthermore, the phase noise scaling is able to maintain constant performance across a wide range of array sizes.

To illustrate the architecture dependence of the multi-user SINR, Figure 4.14 compares the performance of CCG and LCG for a 75GHz single-carrier array. The mm-wave PLL bandwidth is 5 MHz in both scenarios. In this example the users are placed in line-of-sight channels with 10 degrees of angular separation between them. As expected, the CCG

schemes performance is independent of the number of users, while the LCG performance degrades as the number of users is increased. These results match exactly with the model (4.33), using $\alpha = 2$ and $\gamma = 1$.

4.5.2 PLL Sharing in Multi-User Arrays

As stated earlier, a CCG architecture is unappealing for large scale arrays due to difficulty in distributing the LO over such a large area. However, the CCG scheme fundamentally outperforms an LCG scheme in multi-user operation. In order to compromise between these two objectives, it is desired to keep some amount of correlation in the VCO noise in order to suppress inter-user interference. This can be achieved by using a GCG scheme. In the GCG scheme N presents a design parameter that is available to influence γ and therefore tune the dependence of the SINR on the number of users. Conceptually, the GCG architecture applies some spatial filtering prior to adding uncorrelated phase noise, thereby reducing the impact of inter-user phase noise errors. This topic is discussed in detail in [137].

4.5.3 Full-System Simulations

The previous sections have ignored all additional impairments to the receiver and only investigated the impact of phase noise. Figure 4.15 shows full system simulations of a mm-wave array with 128 elements serving 16 users over 2GHz bandwidth in the presence of AWGN. This scenario utilizes a GCG architecture with 4 VCOs, each with -84 dBc/Hz of phase noise at 1 MHz offset and serving 32 elements.

By incorporating the proposed system architecture and design optimizations, the LO chain achieves excellent performance for QAM constellations of 4, 16, and 64. The phase noise-limited SINR is 36 dB, which is also sufficient to reach a BER floor of 5×10^{-4} using 256-QAM. Note that the performance of 256-QAM is limited by burst errors in the DD-PLL. This could be fixed by using alternative CR schemes in this regime.

4.6 Summary: LO Generation Approach for Large Arrays

In summary, this chapter has developed design guidelines for LO subsystem architecture, circuits, and signal processing blocks which enable high-performance, power-efficient, scalable LO subsystems for massive arrays at both RF and mm-wave frequencies.

First, it was shown that the choice of architecture has a strong impact on the total LO subsystem power consumption. Centralized LO generation schemes, commonly used today, break down in the massive array regime due to excessive LO routing power. Meanwhile, fully distributed LO generation incurs a very high duplication overhead. The optimum architecture consists of subarray-based LO generation which balances LO routing versus PLL overhead.

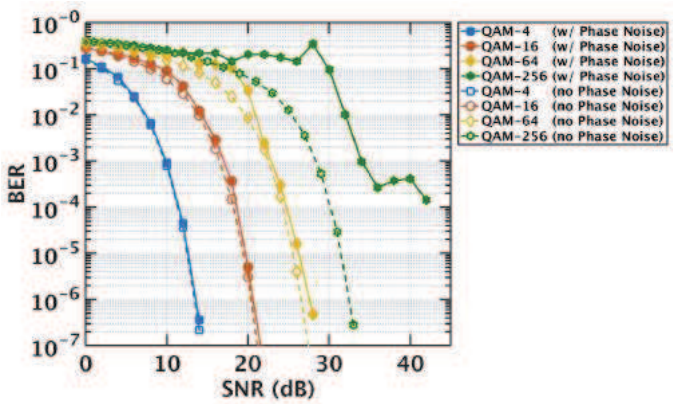


Figure 4.15: Sum BER for 16 users versus thermal SNR with and without phase noise, for various constellation orders. The carrier recovery bandwidth is optimized for each thermal SNR level. The phase noise limited SINR is 36dB.

In order to reduce the total power burned in the VCOs in the array, phase noise averaging can be used to scale the performance of each VCO proportionally to the total number of VCOs used. In this way, architectures with a greater number of VCOs will burn less power in each circuit, and vice versa. This achieves roughly constant power-performance point independent of array size.

However, this deliberate degradation of the VCO performance in distributed arrays induces a performance loss due to beamforming error arising from uncorrelated phase noise. This performance loss can only be mitigated by controlling the correlation level of each PLL's phase noise. This is accomplished by a co-optimization of the CR and PLL bandwidths. High CR bandwidth is used to filter most of the phase noise, while the PLL bandwidth is increased accordingly to maximize correlation. This dual loop strategy serves to correlate the phase noise across the array while nevertheless filtering it.

Finally, when multiple simultaneous users are served over spatial multiplexing, their phase noise errors interact. This effect can only be suppressed by partially spatially filtering the users prior to introducing uncorrelated phase noise. This can be achieved by subarray-based LO generation. Within a subarray, the phase noise is fully correlated such that users may be spatially filtered without any inter-user phase noise effect. Subsequently, summation across subarrays introduces negligible inter-user loss.

Interestingly, from both a power optimization perspective and a multi-user performance perspective, subarray-based LO generation is preferred. Moreover, the analysis in this chapter gives design guidelines on selecting the subarray size and on optimizing the CR and PLL bandwidths for this architecture.

Chapter 5

Hierarchical Baseband Synchronization

Chapter 4 considered carrier generation and synchronization in massive MIMO arrays. In that context, static or very slowly-varying phase offsets could be estimated and tracked by the channel estimation loop. Consequently, any static phase offsets between oscillators at each transceiver do not negatively impact performance. Instead, the main concern for carrier synchronization is the rapid fluctations of the instantaneous phase and the associated techniques for filtering those.

We now turn to baseband synchronization of massive MIMO systems, ensuring that the timing references for all digital samples across the array are appropriately synchronized. Unlike the in the case of carrier synchronization, there is no signal processing loop that naturally corrects for static timing offsets. Therefore the key objective of this study is to design signal processing algorithms to lock all digital data streams to a common absolute time reference.

Consider the modular and distributed architecture proposed in Chapter 3. Baseband synchronization in arrays involves three main tasks. First, when using digital beamforming within a single module, ADC and DAC sampling clocks should be synchronized to ensure proper summation of data streams. Second, the inter-module data aggregation must share a common timebase. Finally, all the users' sampling clocks are not frequency synchronized and therefore may suffer from sampling frequency offsets (SFO). The per-user SFO should be corrected at the base-station.

5.1 Background: Synchronization Requirements and Architecture

During an uplink transmission, the user's transmit signal travels through multiple different propagation paths and hardware blocks before being recombined in the beamformer. The beamforming unit simply combines these multiple copies of the same underlying signal with

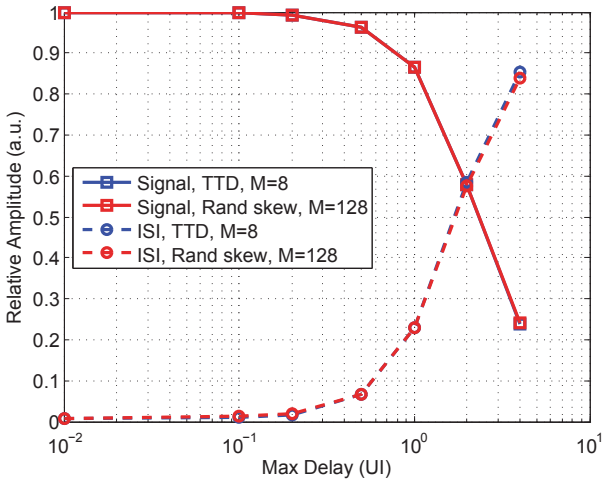


Figure 5.1: Total signal energy loss and ISI energy as a function of worst-case timing offset.

appropriate weights. If the user signal experiences delay which varies according to antenna element, the inputs to the beamformer will not be perfectly time-synchronized. During the summation, adding together these time-shifted copies of the user's data stream will result in both loss of signal energy as well as creation of inter-symbol interference.

Where could these element-dependent delays come from? There are many possible sources. First, true time delay effects in the channel will result in different propagation delay to different elements depending on the direction of arrival. Second, bandwidth mismatch anywhere in the transceivers will result in different group delays between elements. Finally, skew between data converters will cause different ADCs or DACs to sample the continuous-time waveforms at different phases, resulting in digital data sequences that are not aligned relative to contunous time.

Consider first a SIMO link. Between propagation, transceiver front end, and data converter, the total delay variation between elements must be made small enough not to cause degradation in the received SINR. After beamforming, the received signal is:

$$s(t) = \sum_{i=0}^{M-1} |h_i|^2 y_i(t - \tau_i) \tag{5.1}$$

where τ_i is the delay at element i. These τ_i consist of a uniformly randomly distributed component, arising from random bandwidth mismatch and sampling skew, along with a deterministic per-element component $i\Delta t_p$ from TTD effects.

It is necessary to characterize how the SINR of $s(t)$ depends on the distribution of the τ_i in order to determine the level of synchronization accuracy that is required. There are two main impacts arising from per-element skew. First, the low-pass filtering effect will reduce the signal energy. Second, the frequency-selective aggregate transfer function will introduce inter-symbol interference (ISI).

Figure 5.1 shows how both the signal energy and ISI energy vary as a function of the worst-case delay between array elements. The results are shown for two array sizes and two delay models. TTD skews models the skew at each element as $\tau_i = i\Delta t_p$, while random skew models τ_i as being uniformly distributed on the appropriate interval. The latter effect captures the impact of random variations between elements such as sampling clock skew and bandwidth mismatch. As shown in this figure, the SINR depends solely on the worst-case skew and is independent of both array size and the underlying physical origin.

The aggregate SINR impact can be modeled by an equivalent de-synchronization filter which is just a rectangular pulse in time:

$$d(t) = rect(0, \tau_{max}) \tag{5.2}$$

The baseband synchronization subsystem must therefore achieve accuracy within τ_{max} as determined by the error budget shown in Figure 5.1.

5.1.1 Synchronization Architecture

With knowledge of the required baseband synchronizatation accuracy, we must now design hardware and signal processing blocks to achieve the needed level of time alignment. This can be accomplished with a variety of tools, including analog delay circuits (such as analog TTD beamformers), low-skew ADC/DAC clock generation circuits and distribution schemes, calibration methods, and digital timing recovery and adjustment loops.

The previous analysis considered a SIMO case, where the timing offset at each element needs to be corrected up to some accuracy. When extending to a MIMO scenario, each user stream at each element experiences independent delay. The full-complexity solution would require MK timing recovery and adjustment blocks to fully recover the synchronization of the MIMO array with M antennas serving K users.

If full frequency-domain beamforming is used, then the per-user and per-antenna delays are naturally estimated and compensated by the beamforming algorithm, embedded within the $M \times K$ channel matrix and beamformer on each subcarrier. On the other hand, if the conjugate beamforming stage uses frequency-flat beamforming, then it has no delay resolution and could be sensitive to baseband synchronization issues. This is the context considered in this chapter. Moreover, it would be desirable to reduce the number of synchronization blocks needed in the system.

Of the three skew sources described above, group delay mismatch is likely to be the least significant contribution. Group delay is inversely proportional to the filter bandwidth in both first- and second-order filters. If the analog bandwidth variation is problematic, then bandwidth trimming will need to be used. This calibration will inherently fix frequency response and delay issues caused by bandwidth mismatch.

Sampling skew between data converters will introduce worst-case delay inversely proportional to the sampling period, $1/T_S$. If the system is 2x oversampled, the worst case delay is only half of a UI which introduces only relatively small energy loss as shown above.

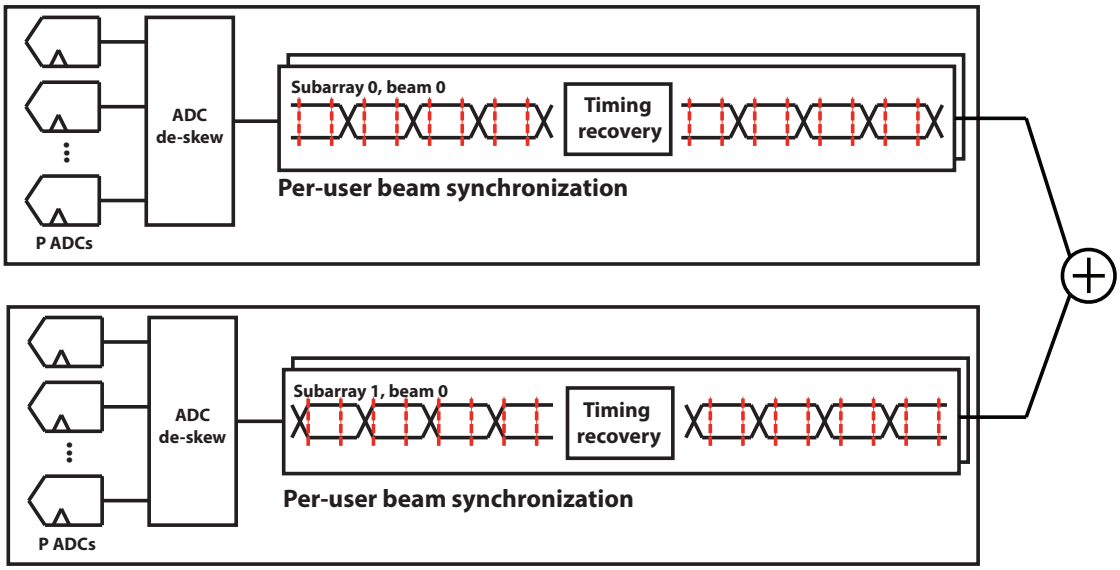


Figure 5.2: Hierarchical baseband synchronization strategy.

Data converter skew can generally be addressed by designing a relatively low-skew clock distribution network on chip. This may be more challenging when data converters are being synchronized on package or board; in this case, digital algorithms which estimate and compensate the sampling skew may be needed.

True time delay (TTD) is the most severe de-synchronization mechanism. Unlike front-end and sampling skews, which are bounded by system parameters, the worst-case wavefront propagation delay can be arbitrarily large depending on the size and fractional bandwidth of the array. There are a number of circuit techniques that have been proposed to realize analog time-delay beamformers, such as tapped transmission lines (whether real or synthetic) and all-pass filters. Generally these structures tend to be bulky, lossy, and power-hungry, although they may be easier to realize at higher carrier frequencies.

To address the TTD problem, we look for a system-level solution. True time delay is inherently a "large array" problem, resulting from a very large array aperture. Small subarrays would not experience significant true time delay effects even for very aggressive fractional bandwidths. For example, a 16-element linear array at 72GHz with field of view of 120° with half-wavelength spaced antennas experiences 100ps of delay between the first and last element. This is only about 20% of a UI at 2GHz of channel bandwidth. For 2-D arrays this is even less significant. For example, a 64-element subarray arranged as a 4x16 2-D array experiences the exact same level of true-time delay loss. An 8x8 subarray would only have 50ps of maximum delay, supporting channel bandwidths up to 10GHz with only half-UI loss.

At carrier frequencies below 100GHz, achievable subarray sizes and fractional bandwidths are such that TTD effects can be neglected within a single subarray. This significantly sim-

plifies the synchronization problem, since we can correct TTD effects between subarray modules instead of between elements. With this realization, we can eliminate the need for K synchronization loops at every receive element. Instead, we can have one synchronization loop per-element to adjust for ADC sampling and front end skew, and an additional synchronization loop per beam per subarray to compensate for subarray-level TTD effects.

In summary, the proposed synchronization approach is designed to be hierarchical and modular. This is illustrated in Figure 5.2. Each common module (subarray) is internally synchronized on its own. Then, the modules are synchronized to each other to compensate for large array TTD effects between modules. The remainder of this chapter will discuss digital timing recovery techniques for baseband synchronization.

5.2 Distributed ADC de-skew in a Modular Digital Array

We first consider synchronization within a subarray. As mentioned above, if the subarray uses exclusively analog beamforming, there is no need to synchronize the samplers. The user data streams have no enforced timing relationship between them, so there is no need to control the skew between their samplers at the base-staiton. In contrast, subarrays which use any amount of digital beamforming require ADC/DAC synchronization. In this scenario, multiple digital data streams within the subarray contain information about the same user data, so they must be properly aligned before digital combining. In the remainder of the section we will focus on receiver processing. Transmit synchronization proceeds broadly in the same way, with the added need to perform some type of loopback to measure the inter-DAC skew.

How should data converters within a subarray be synchronized? The simplest option is to design an accurate, low-skew clock distribution network such as an H-tree. If all the converters are implemented on a single chip, this may be a very reasonable objective. There may be some drawbacks, particularly if the physical floorplan for the clock distribution network interferes with other physical design objectives. Moreover, low-skew clock distribution networks consume more power compared to a skew-insensitive clock tree due to the need to maintain balance to each leaf node. Overall, in single-chip implementations there are not strong reasons to avoid balanced clock trees.

In other scenarios, where converters are integrated on package or on board, it may be more challenging to realize a well-balanced clock distribution network. In these cases, synchronization between ADCs should be achieved in some other way. Explicit calibration could be used to measure and compensate for the skew. A calibration source is used to couple an edge into each ADC, and digital processing attempts to align the zero-crossings of all the edges to remove inter-ADC skew. Interestingly, this requires the calibration signal to be distributed with low skew. This may be a bit more feasible than doing so with the clock, since the calibration signal can be made slower than the clock, and does not need to be ac-

tive all the time. Nevertheless, this is a significant drawback of explicit calibration methods. Additionally, this would be a foreground calibration technique, which would require taking an ADC offline to perform calibration.

In this section, we design background calibration techniques which use information within the incoming signal itself to correct for inter-ADC skew. The proposed scheme exploits user-transmitted pilots to estimate channel state information and delay. The delay estimates at each ADC are jointly processed to compute the ADC de-skew coefficients.

5.2.1 Joint Golay Channel and Timing Estimator

Almost all wireless standards use synchronization sequences and pilots to provide or look for beacons, exchange basic information, and estimate the channel. This setup enables very sophisticated modulation schemes to exchange data payloads with much higher spectral efficiency. We propose to re-use the pilot sequences to estimate both channel state information and delay information simultaneously from multiple users. This discussion considers Golay coded pilots, which are used in 802.11ad [38] due to their very simple generation/detection and excellent auto-correlation properties. The same conclusions would apply to most other pilot sequences, since the general principle involves correlating against the known pilot and extracting information from the peak of the correlator output.

A detailed discussion of Golay pilots can be found in [138]. The key ideas are as follows. Golay codes consist of complementary sequences (called A and B sequences) of a desired length which is a power of two. The two sequences are transmitted through the channel sequentially and each received by their respective correlator. The sum of these auto-correlations gives an error-free estimate of the channel impulse response:

$$\hat{h}[n] = g_A[n] \star (h[n] * g_A[n]) + g_B[n] \star (h[n] * g_B[n]) \tag{5.3}$$

If the users successively transmit Golay pilots, correlators at each base-station element can be used to estimate the full channel matrix and compute the desired beamforming coefficients. These correlators can be reused to estimate the timing offset relative to the peak correlator output, as follows. The ADCs are oversampled by a factor of N_{os}. An N_{os} branch polyphase Golay correlator computes the correlation with each phase of the oversampled data, which are re-interleaved together to estimate the channel impulse response in the oversampled time base. Finally, a polynomial interpolator can be used to find the offset between real samples and the maximum of the correlator output.

This structure is shown in Figure 5.3. The polyphase correlator is normalized by the average input power at the ADC (estimated by a very low-bandwidth IIR filter) to provide an accurate measure of the correlation gain. A threshold parameter is used to separate out correlation peaks from spurious peaks in this correlator output. The threshold may be tuned according to the desired behavior. A lower threshold increases the probability of detection (particularly in low SNR conditions), but leads to more frequent false positives. A higher threshold reduces the number of false positives in exchange for more frequent detection failures at low SNRs.

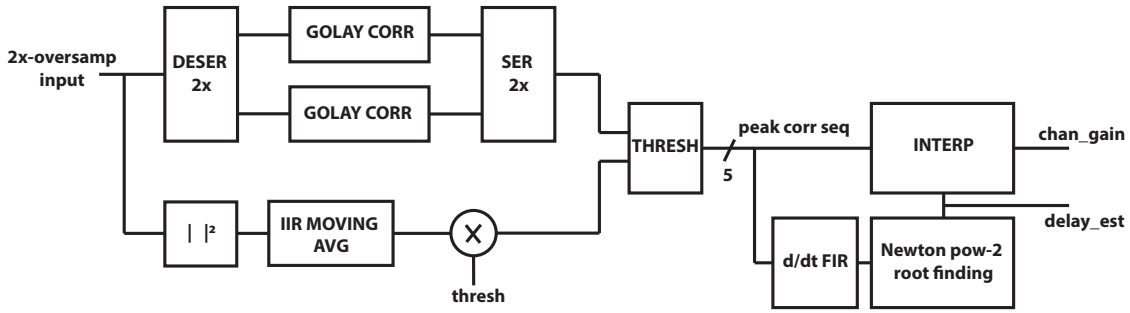


Figure 5.3: Joint Golay channel and delay estimator for 2x oversampled input.

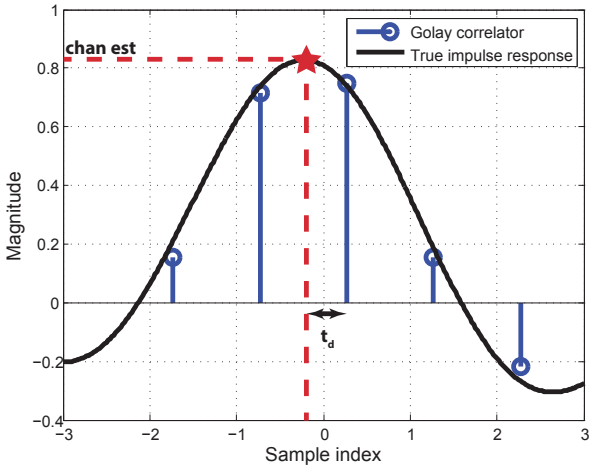


Figure 5.4: Estimated channel impulse response from polyphase Golay correlator.

The output of the correlator consists of the oversampled channel impulse response as shown in Figure 5.4. This contains full information about the channel impulse response but most likely does not have any sample taken precisely at or near the correlation peak. This peak value, and the time at which it occurs, as illustrated in Figure 5.4 is estimated using a polynomial interpolator. Take the N samples centered around the peak of the correlator output. The value of the CIR at any time in this interval can be obtained by Lagrange polynomial interpolation:

$$c(t) = \sum_{i=0}^{N-1} \prod_{j \neq i} \frac{(t - t_j)}{(t_i - t_j)} c(t_i) \tag{5.4}$$

We are interested in the value of t for which $c(t)$ takes its maximum, denoted by t_d:

$$t_d = \operatorname{argmax} c(t) \tag{5.5}$$

This can be found by simple differentiation. If N is small enough, given that we know the interval $[0, N-1]$ is centered on the correlation peak, $c(t)$ will only have a single extremum in this interval which will be precisely its global maximum. For example, if the correlator operates with 2x oversampling, $N = 5$ is a good choice to obtain good performance without sampling any additional extrema.

t_d may be found according to:

$$\frac{dc(t)}{dt}(t_d) = 0 \tag{5.6}$$

This gives a polynomial of order $N-1$, whose coefficients are functions of the sampling instants t_i and the correlator output at those sample instances, $c(t_i)$. The former is known at design time, based on the sequence length N, since we always center this interval about the correlator maximum. As a result, the polynomial coefficients of $dc(t)/dt$ are simply FIR filters of $c(t_i)$. For example, for $N = 5$, we would need four five-tap FIR filters to compute the coefficients for the derivative polynomial.

The root of this polynomial can be found using the Newton-Raphson method. With floating point precision, this algorithm finds the root of a function $f(x)$ by proceeding iteratively from an initial guess x_0:

$$x_i = x_{i-1} - \frac{f(x_{i-1})}{f'(x_{i-1})} \tag{5.7}$$

This can be easily applied to our scenario, since we can compute both $c(t)$ and $c'(t)$ with known polynomial interpolators. The only problem is the need to perform division, which is very expensive in hardware. This can be sidestepped by rounding the value of the derivative at each iteration to the next highest power of two. In this way, the division can be replaced simply with a left or right shift or expansion. This approximation always over-estimates the derivative value, so takes a smaller step than it would have in floating point precision. In this way, the hardware-friendly algorithm is guaranteed to converge if the floating point algorithm would have, but simply in a couple of extra iterations. Because we choose N to be small enough, there is only one extremum in the interval and the algorithm is almost certain to find it.

In summary, the correlator output is processed by interpolation filters and a Newton-Raphson iteration to find the delay between the actual ADC samples and the peak of the continuous time channel impulse response. The correlator impulse response can be interpolated to find the channel gain at this timing offset; this channel gain is used as the frequency-flat channel estimate for this antenna.

Figure 5.5 shows the performance achieved by the proposed Golay-based estimator. As shown here, this algorithm achieves excellent timing and channel detection accuracy above a certain SNR which depends on how the detector threshold parameter is chosen.

5.2.2 Joint ADC de-skew

The Golay-aided estimator presented above will estimate the channel gain and delay experienced by every user at every element in the receive array. The channel estimates are used to

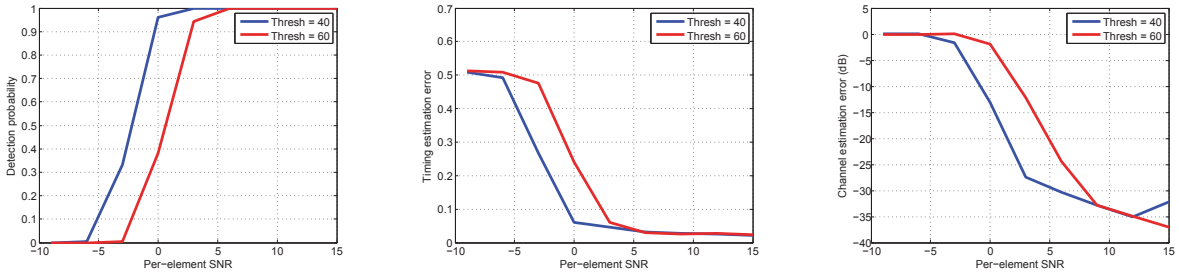


Figure 5.5: Detection, delay estimation, and channel estimation accuracy achieved by the proposed Golay channel/delay estimator.

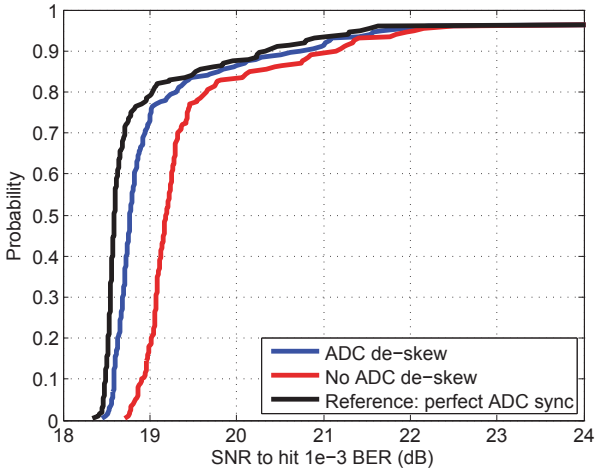


Figure 5.6: ADC de-skew performance in a 16-element subarray serving 4 users.

compute the beamforming matrix. The delay estimates must be fused across users to obtain the deskew coefficients for each ADC.

The delay experienced at element i by user j relative to a clock edge can be split into two components:

$$\tau_{ij} = t_i + \Delta_j \mod (T_S) \tag{5.8}$$

where T_S is the sampling period of the ADC. Because the user-specific portion of the delay τ_{ij} is uncorrelated with the sampling phase and the other user's delays, the Δ_j component is uniformly distributed in this modulo arithmetic. Consequently, it is sufficient to average the delays τ_{ij} across j to obtain the estimate of t_i. This deskew coefficient can then be used to align all the ADC sampling instants to each other. This is implemented using a polynomial interplation filter (5-tap Farrow filter) at each element [139].

Figure 5.6 compares the performance of the ADC de-skew for a 16-element subarray serving 4 users. As shown in this figure, without ADC de-skew there is up to 0.6dB performance

loss due to signal filtering. The ADC de-skew algorithm recovers most of this performance gap, up to about 0.4dB. The residual 0.2dB error is due to the finite timing resolution of this algorithm and the self-noise of the interpolator.

5.3 Distributed Subarray Synchronization and True Time Delay Compensation

The processing described above serves to synchronize all the ADCs in a single subarray to each other. This ensures that subarray digital beamforming can proceed without signal loss due to non-ideal summation. However, the various subarrays in the overall array need to be time aligned to each other in order for the global data aggregation to proceed correctly. As discussed above, the dominant cause of misalignment between subarrays is true time delay propagation effects for large aperture arrays. Accordingly, we need to compensate for the TTD effects before summing together data from different subarrays.

If different subarrays are processing different user streams (referred to as a partially connected array), then there are no inter-subarray synchronization requirements. In contrast, if different subarrays jointly receive the same user streams, then those data sequences need to be aligned before aggregation. Essentially, the sampling instants for a given user sequence should be the same across all subarrays. This can be accomplished using a timing recovery loop operating on the user data itself. By its nature, a timing recovery loop will use timing information implicit in the underlying data sequence to lock on to a particular sampling phase. This could be run autonomously on each subarray to align the samples for each user data stream. This operation would naturally correct for TTD offsets between different subarrays.

This section will compare different timing recovery schemes for subarray synchronization, and then show how this technique can compensate for TTD effects.

5.3.1 Comparison of Timing Recovery Schemes

Per-subarray timing recovery operates in a challenging environment. The SINR at the output of the subarray conjugate beamformer is

$$\mathbb{E}[SINR_{subarray}] = \frac{PE_s}{KE_s + \sigma^2} = \frac{P}{K + \frac{1}{\gamma}} \tag{5.9}$$

where P is the subarray size, E_s is the signal energy, σ^2 is the thermal noise variance, and γ is the thermal SNR. Commonly the subarray size and the number of users will be comparable, so this SINR could be close to 0dB. The exact value and distribution of the SINR will depend on the exact arrangement of the users in space and the resulting inter-user interference.

Moreover, per-subarray timing recovery would occur before critical signal processing tasks such as equalization, so it must handle fairly strong channel dispersion and frequency-selective fading. As a result, the subarray synchronization algorithm must be robust to very

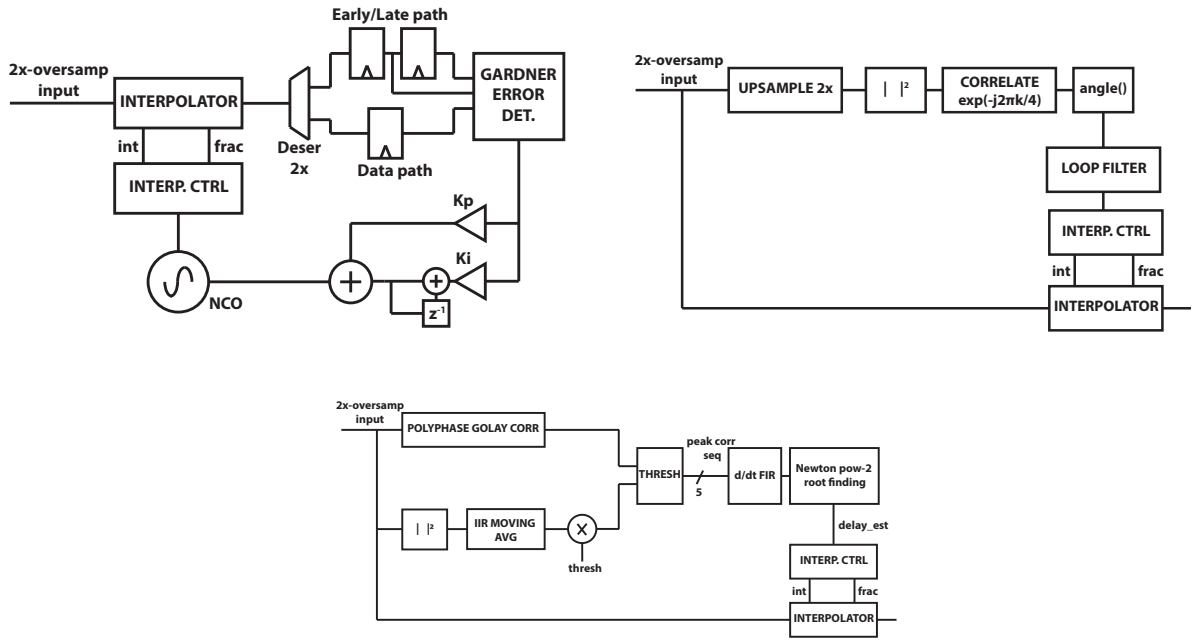


Figure 5.7: Three timing recovery algorithms: NDA feedback loop with Garder detector, O&M feedforward spectral estimator, and Golay pilot-aided estimator.

low SINRs. Additionally, only non-data-aided (NDA) algorithms could be used since there are no symbol decisions available at the subarray level (decisions are only available at the full array level).

There are three main candidates for NDA timing recovery algorithms [134]. A feedback-based timing recovery loop uses a timing error detector to measure the timing offset between the actual and ideal sampling instants. This error signal is fed back through a loop filter to a numerically-controlled oscillator (NCO) which adjusts the sampling instants. NDA feedforward timing recovery schemes use a timing detector to obtain a direct estimate of the ideal sampling instant, and then re-process the data to interpolate it to that sampling instant. Finally, pilot-based (or data-aided — DA) timing recovery systems use some known pilot sequence to estimate and correct for the delay in a feedforward manner.

We compared the performance of three different timing recovery schemes for inter-subarray synchronization. The first was a feedback loop operating at 2x oversampling using a Gardner error detector (Figure 5.7a) [140]. The error signal is fed back through a proportional-integral (PI) loop to control an NCO. The second scheme was an Oerder and Meyr (O&M) feedforward loop (Figure 5.7b) [141]. Using 4x oversampled data, the auto-correlation function of the data is estimated and fit to a Fourier series; the coefficient of the fundamental Fourier tone is extracted to estimate the phase of the data. The third scheme is the Golay pilot-based timing recovery discussed in Section 5.2.1. All three schemes estimate integer and fractional delays and use an interpolation filter (implemented using the Farrow

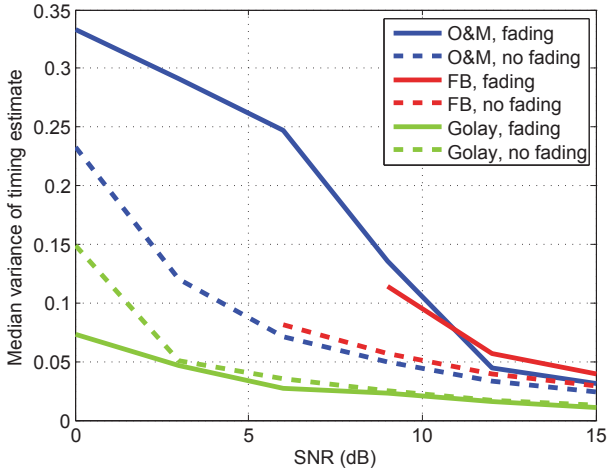


Figure 5.8: Comparison of feedback, O&M feedforward, and Golay timing recovery performance versus thermal SNR.

structure) to adjust the sampling instants of the data sequence [134, 139, 140, 142].

Recall that inter-subarray synchronization proceeds in a distributed fashion. Independent TR algorithms run on each subarray's estimate of each beam, with the goal that for each beam, all subarrays lock onto the same sampling phase. Therefore, to assess the suitability of the above three TR algorithms to this task, we must characterize their reliability and robustness as measured by this criterion. This was simulated as follows. For a given channel response, 30 independent data/noise sequences were generated with the same timing characteristics. These 30 trials were processed by each of the above three TR schemes and the variance of the timing estimate across the 30 trials was computed. This procedure was repeated across 100 random channel instantiations and aggregate statistics about the timing variance were computed.

Figure 5.8 shows the results of this simulation, measured versus thermal SNR at the antenna. Fading channels have a Ricean K-factor of -10dB; non-fading channels only have a single tap. As shown by these results, the feedback TR loop cannot work below a certain SNR. Meanwhile, the Golay timing recovery performs best for all SNR values and fading scenarios. Moreover, because the user Golay pilots are time-multiplexed, Golay pilots do not experience inter-user interference. As a result, the SINR for the Golay pilots is K times stronger than that of the rest of the data sequence. Finally, the Golay scheme converges immediately while the other two algorithms require a long acquisition time. For all these reasons, Golay timing recovery was identified as the best candidate for inter-subarray synchronization.

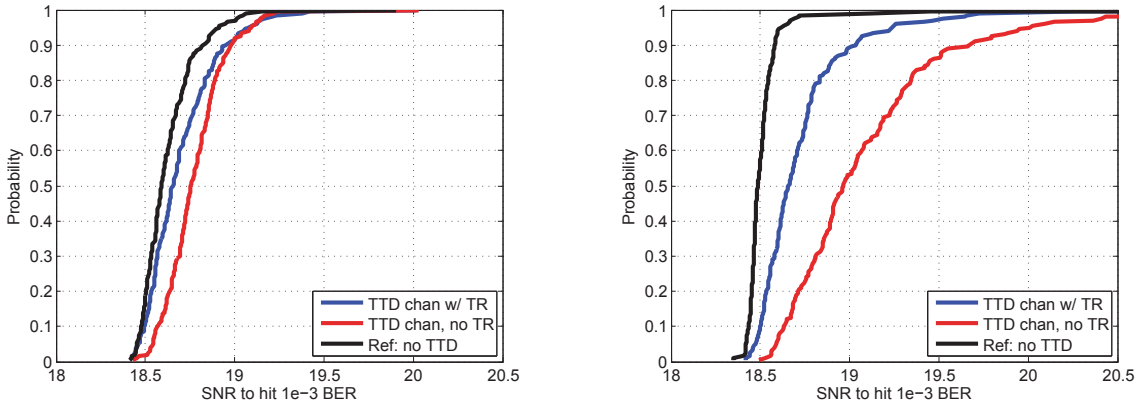


Figure 5.9: True-time delay compensation using distributed Golay-aided timing recovery at each subarray for 200MHz and 2GHz channel bandwidths at 75GHz carrier.

5.3.2 True Time Delay Compensation

Figure 5.9 shows the performance of the distributed subarray synchronization in a TTD channel. The array has 32 elements and serves four users; each subarray is 16-elements so there are two total subarrays. The left panel shows the results for a 200MHz channel at 75GHz. The fractional bandwidth in this scenario is very small so the total TTD effects are quite insignificant. Without TTD compensation there is about a 0.2dB SNR loss, of which 0.15dB is recovered by the distributed synchronization. The right panel shows the same scenario with a 2GHz channel bandwidth. Here TTD effects are very significant — without distributed TR there is over 0.5dB of signal loss. The proposed synchronization scheme is able to recover 0.35dB of that loss.

5.4 Multi-User Sampling Frequency Offset Compensation

The techniques described so far in this chapter are used to achieve baseband synchronization within the base-station array. The final remaining task is to synchronize to the sampling frequency and phase of the user data streams themselves.

Consider the signal processing architecture shown in Figure 5.10. Initial phase synchronization is partially achieved by the distributed Golay timing recovery outlined in Section 5.3. Any residual phase offset can be compensated by the central MIMO equalizer. However, sampling phase drift due to sampling frequency offset relative to the user's data stream is not addressed anywhere.

The sampling frequencies of the users and the base-station are set by their reference oscillators. These frequencies will match up to the tolerance of the oscillator, usually on the

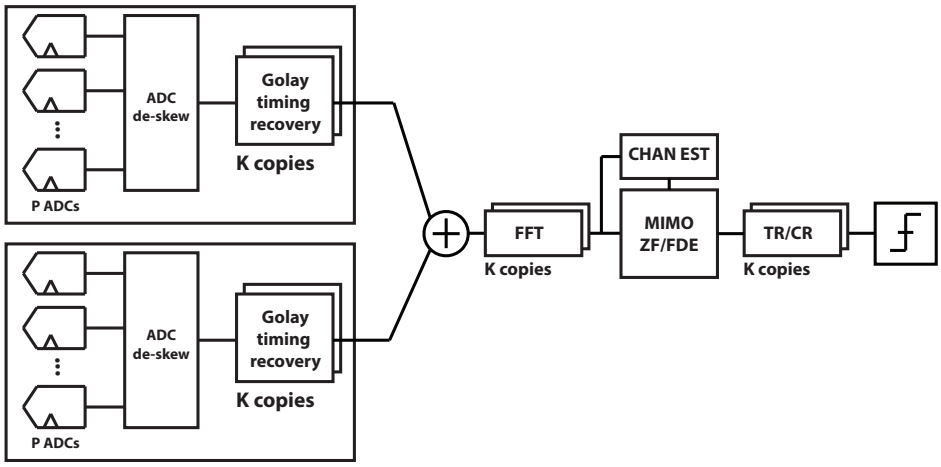


Figure 5.10: Overall array DSP chain including subarrays and central processor.

order of 10s of ppm. This means that the user sampling frequencies will differ from each other, and from the base-station's sampling frequency, by a factor on the order of 20 ppm.

In a modulation scheme using frequency-domain equalization, such as single-carrier frequency-division multiplexing (SC-FDM, used in the LTE uplink), sampling frequency offsets (SFO) have two impacts. First, over the course of a frame it causes drift in the FFT window. Second, the delay manifests as a phase rotation in the frequency domain proportional to subcarrier index [143, 144]. As long as the SFO is small relative to the subcarrier spacing, there is negligible loss of subcarrier orthogonality.

Because the SFO is different for each user, correcting it can be tricky. Before the ZF-FDE, each data stream consists of the desired user's data (with its SFO) corrupted by inter-user interference. Each inter-user interference carries the SFO signature of its respective user stream. This means that to correct SFO *before* the ZF-FDE, we would need to separate out each inter-user interference contribution and adjust it by that user's SFO. Of course, this is impossible — it is the function of the ZF-FDE to separate out those components! This means that SFO compensation must occur on a per-user basis after the zero-forcer and equalizer.

The challenge with post-FDE SFO correction is that the FDE applies a *cyclic shift* in timing since it appears as a circular convolution. That means that drift in the FFT window causes the FDE output to be cyclically rotated such that samples may appear out of order at the output. This implies that either (1) SFO must be compensated in the frequency-domain to undo the cyclic shift, or (2) SFO may be compensated in the time-domain after FDE but must include re-ordering of samples. We choose to go with the first approach, and design a hybrid time-frequency feedback timing recovery loop.

Timing drift of t_d in a FDE symbol creates a per-subcarrier phase rotation proportional to subcarrier index k:

$$\phi_k = \frac{2\pi t_d k}{N_{sc}} \tag{5.10}$$

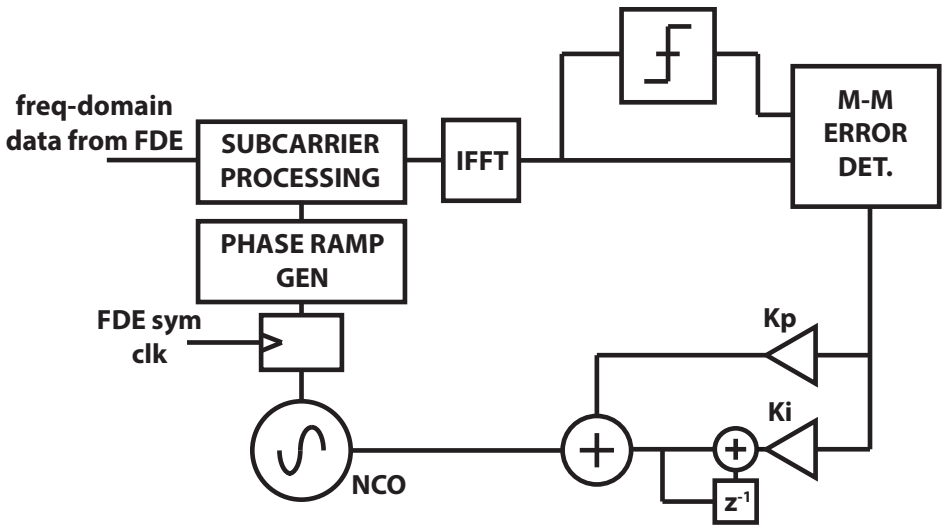


Figure 5.11: Time-frequency timing recovery loop using Mueller-Muller detector.

This means that a hybrid time-frequency TR loop needs only two changes compared to a traditional fully time-domain loop. First, the timing correction is applied by applying appropriate compensating phase shift at each subcarrier. Practically, the delay estimate must be converted to a phase ramp in the frequency domain. Second, the delay is compensated on a FDE symbol-by-symbol basis instead of a sample-by-sample basis. In effect, the continuous timing drift is approximated by a stairstep. As long as the drift is slow enough relative to the FDE symbol period, this error is negligible. Furthermore, since the IFFT has significant delay, this loop should operate with relatively low bandwidth.

The proposed time-frequency TR loop is shown in Figure 5.11. It uses a Mueller-Muller timing error detector because it is the only baud-rate TED (and is decision-directed) [134, 145]. A different TED could be used (such as Gardner) but would need 2x upsampling in the time-domain before the TED. Because this TR loop comes after the equalization and zero-forcing, all inter-user dependencies have been eliminated. As a result, each user stream is independently processed by one of these loops. Figure 5.12 shows that with 20ppm SFO, the proposed timing recovery algorithm can recover 2dB of SNR performance when targeting a 1e-3 bit error rate (BER).

5.5 Summary

In summary, this chapter has proposed baseband synchronization architecture and algorithms suitable for large massive MIMO arrays. The synchronization is accoimplished in a hierarchical manner. First, subarrays are independently synchronized by aligning the ADC sampling instants through a blind background calibration. Next, subarrays are aligned to each other to compensate for true-time delay. This can be done by running distributed timing

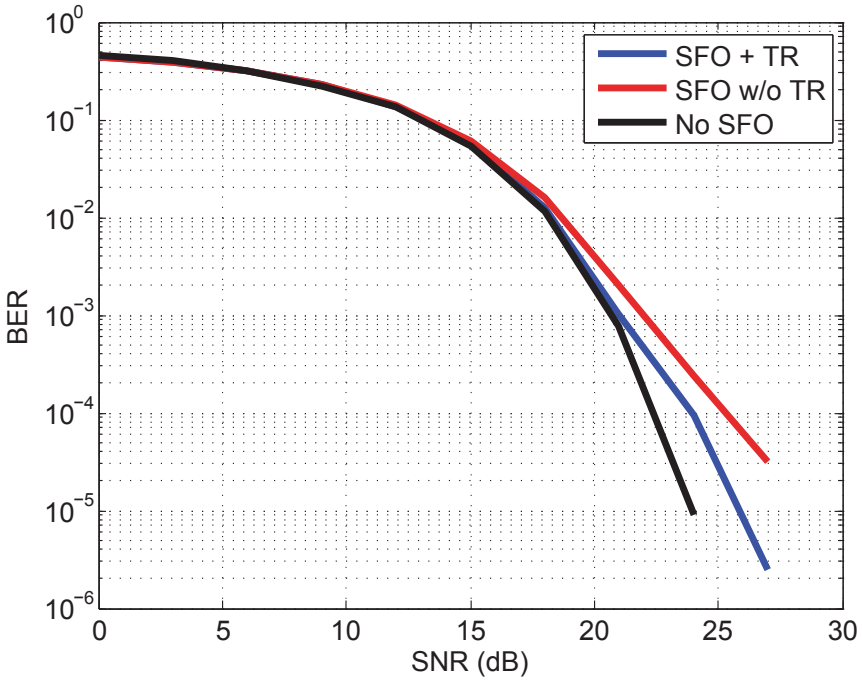


Figure 5.12: Performance of time-frequency timing recovery algorithm with 20ppm offset SFO.

recovery on each beam at each subarray. If a Golay pilot-aided TR is used at each subarray, favorable and repeatable timing lock is achieved which properly corrects for TTD-induced skews between subarrays. Finally, a decision-directed frequency-domain timing recovery loop compensates for per-user SFO following the FDE. Overall, this architecture achieves both synchronization within the array as well as timing recovery of the user data streams.

Chapter 6

System-on-Chip for < 6GHz Massive MIMO

So far, Chapters 2-5 have discussed architecture and algorithm design choices suitable for very large array systems. We now discuss the design of specific arrays using those principles. This chapter considers a massive MIMO system operating in traditional cellular frequency bands below 6GHz. As discussed in the introduction, massive MIMO systems are already rapidly approaching commercial deployments in cellular frequency bands. These commercial deployments use a "brute force" approach of adapting existing base-station designs to the regime of 64 antennas. Though inefficient and not optimized, the cost and power of these solutions are sufficient for commercial requirements. Accordingly, the main challenge in < 6 GHz massive MIMO is not the impossibility of the design, but rather the efficiency. Consequently, this chapter is mostly focused on the design of a very cost- and power-efficient system-on-chip (SoC) for these applications.

6.1 Overview

Massive MIMO systems approaching commercialization in cellular bands generally consist of all-digital beamforming arrays with up to 64 elements in both transmit and receive directions (known as a 64T64R array). The design of these systems suffers from two main inefficiencies. First, fully centralized processing is used, where the digital I/Q samples for each RX and TX stream are communicated to the central processor. This central processor performs all the computation in the system. Research publications (even from industry groups) such as [59] use FPGAs for this processor; practical commercial systems implement this using co-processors or accelerators on a large baseband ASIC. Second, the radios used in massive MIMO systems today consume a large amount of power for relatively mediocre specifications. As an example, the AD9375 consumes up to 5W for two transceivers with only 7dBm output power and 12dB noise figure (NF).

In light of this, the main goals of this project are two-fold. First, distributed signal

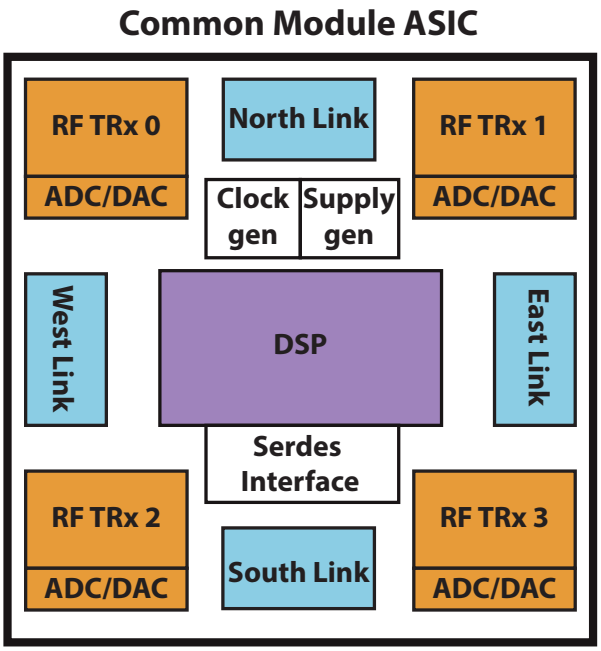


Figure 6.1: Block diagram of < 6 GHz massive MIMO SoC.

processing will be used (as elaborated in Chapter 3) to reduce the interconnect bandwidth and computation requirements on the central processor. Second, the radios will be designed for very high efficiency.

Toward this second end, it is well known that array operation provides array gains in both RX NF and TX power. The receiver NF improves by a factor of M for an M-element array; the TX power improves proportionally to M^2/K thanks to in-phase summation. Consequently, very impressive array-level performance can be achieved with quite modest per-element specifications. As an example, to achieve 50dBm EIRP and 1dB NF with 32 elements, it is sufficient to use a $<$ 30 dBm transmitter and 16dB NF receiver. This performance is close to what is achievable in CMOS processes, so it begins to be reasonable to implement this system with a CMOS chip and an external PA with moderate specs.

Therefore, we propose to build a highly-integrated digital array transceiver in CMOS. If necesssary, this transceiver can be augmented with handset-class external components (such as SiGe PAs) to improve the transmit peformance by a few dB. The proposed CMOS SoC would incorporate a number of full transceivers (probably in the range of 4-8, depending on area and I/O constraints), clock and supply generation, digital signal processing, and inter-chip serdes lanes for interconnect. A block diagram for this chip is shown in Figure 6.1.

This chapter will overview the design of some of the digital components of this chip. The array is designed to operate using the 802.11n wireless standard [2].

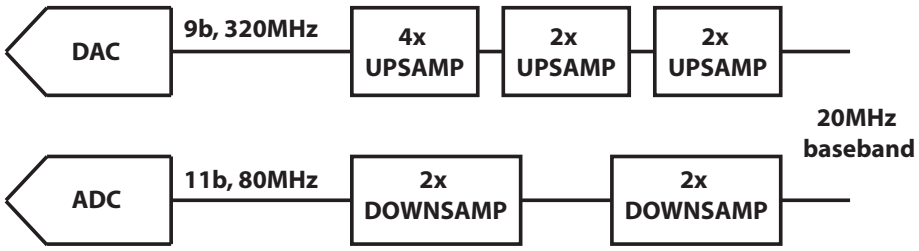


Figure 6.2: Analog interface including ADC/DAC and resampling filter hierarchy.

6.2 RX and TX filters

The channel bandwidth in this system is chosen to be 20MHz. Accordingly, the ADC sampling rate was chosen to be 80MHz based on the blocker and anti-aliasing margins needed in the analog front-end design. Similarly, the DAC sampling rate was chosen as 320MHz (16x oversampling) in order to achieve very aggressive out-of-band emissions targets.

Resampling filters are needed to convert the sampling rate from the baseband rate of 20MHz to and from the ADC and DAC sampling frequencies. These filters must meet the RX and TX mask requirements for the 802.11 standard [2]. For both filters, a multi-stage implementation is used to improve the overall hardware efficiency (Figure 6.2) [146–150].

The RX downsampling filter is implemented as two stages of half-band filters, each downsampling by 2. The first filter uses 10 taps with 7 bits of precision and a passband edge at 9MHz; the second uses 20 taps with 10 bits of precision with passband edge at 8.12MHz. The second filter has more stringent specs since it has a narrower transition band and sets the close-in interferer rejection. It is designed to work with the zero-padded subcarriers at the edge of the band in 802.11n modulation. The number of taps and bits in the weights were determined empirically based on the required specifications. The overall transfer function is shown in Figure 6.3.

The TX upsampling filter must achieve very high oversampling ratio of 16. This is split into two stages of half-band filters implementing 2x oversampling each, and a Nyquist filter achieving 4x oversampling. It is important that all these filters be Nyquist so as to not introduce pulse distortion or inter-symbol interference when oversampling the TX signal. The first stage of filter is a 14-tap half-band filter with 9MHz passband edge and 7-bit coefficients. The second stage uses only 6 taps with 8.1MHz passband edge and 10 bits of resolution. Finally, the Nyquist quarter-band filter uses 13 taps with passband edge at 16MHz and $>$ 40dB stop-band rejection. The aggregate transfer function is shown in Figure 6.4. Transmit masks are discussed below.

All filter stages were implemented in hardware using polyphase structures with fixed coefficients (Figure 6.5). The polyphase structure relaxes the clock rate requirements (thereby simplifying the physical design) and can make use of redundancies such as all the zero coef-

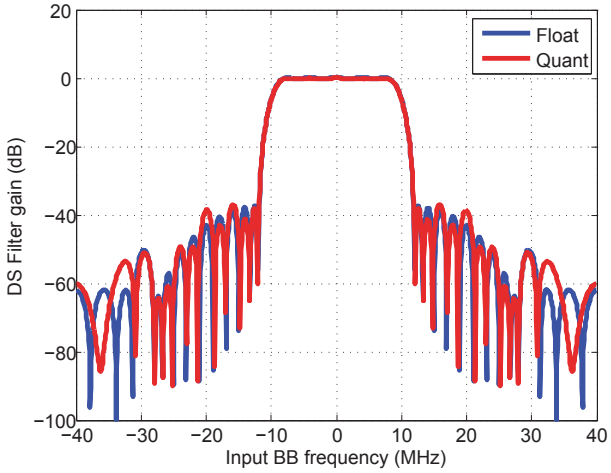


Figure 6.3: Receiver downsampling filter transfer function.

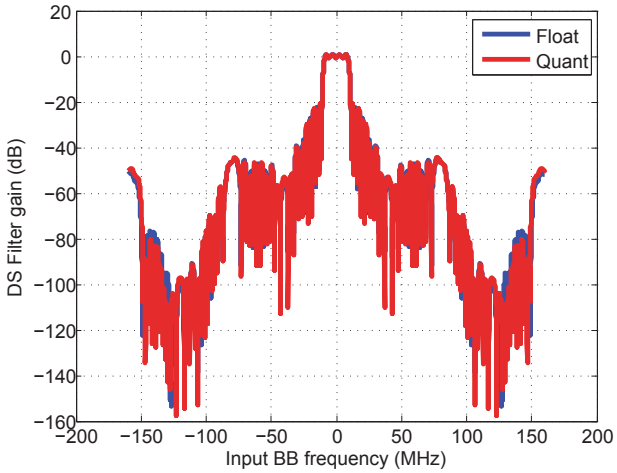


Figure 6.4: Transmitter upsampling filter transfer function.

ficients. The use of fixed coefficients reduced the silicon area by 9x.

6.3 TX Quantization Noise Averaging

A key transmitter spec is the power emitted outside of the assigned channel bandwidth. The out-of-band emissions spec measures how much a radio interferes with other radios operating at different frequencies. Each standard assigns transmit masks which specify the allowable emissions in various parts of the spectrum. Most of the spectral emissions correspond to quantization noise in the TX chain.

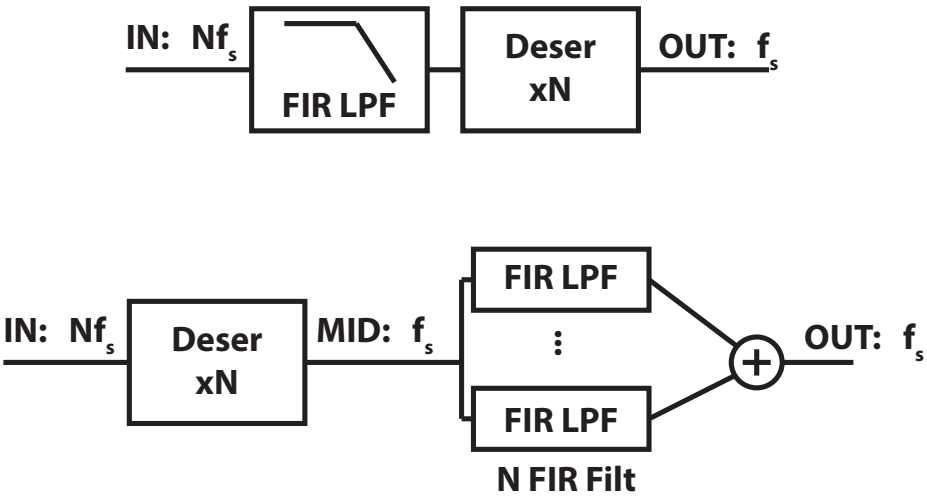


Figure 6.5: Standard downsampling filter (top) and polyphase implementation (bottom), which reduces the clock rate for the entire design.

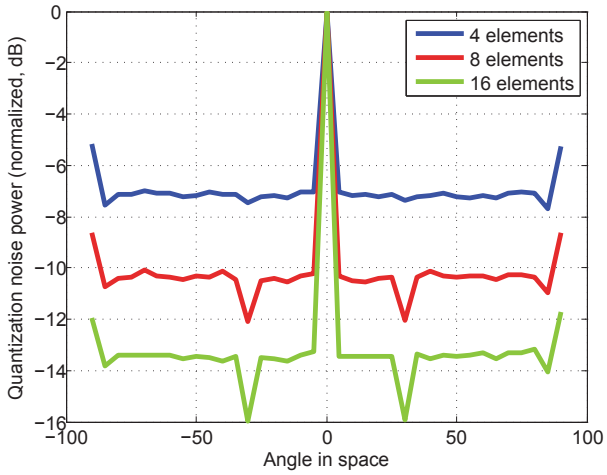


Figure 6.6: Quantization noise versus line of sight angle for various array sizes.

In array transmitters, especially all digital ones, quantization noise at each DAC tends to be uncorrelated. This arises because both the beamforming coefficients as well as the multi-user data sufficiently change the digital samples at the input to each DAC that the DAC quantization noise is decorrelated. As a result it can be expected that the out-of-band quantization noise will be averaged and reduced in an array transmitter.

Figure 6.6 shows the out-of-band quantization noise versus line of sight direction for a 4-, 8-, and 16-element array. As expected, for all directions except broadside, the quantization

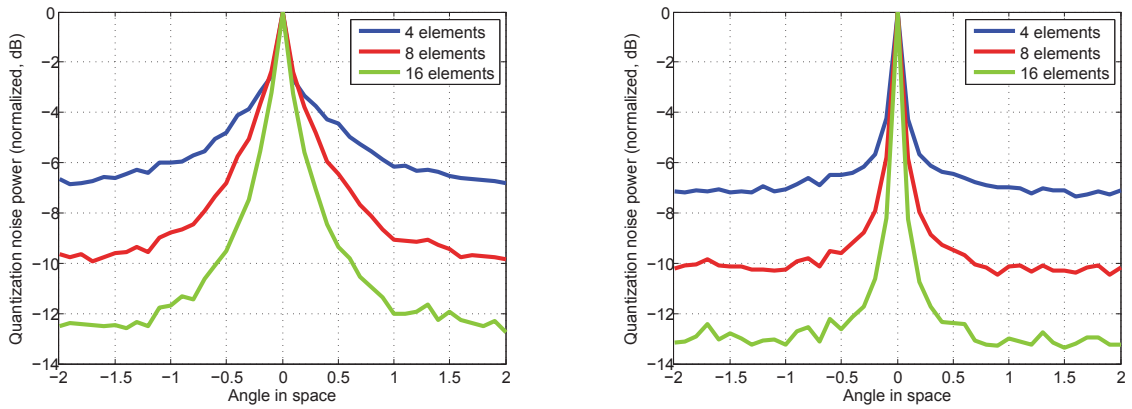


Figure 6.7: Quantization noise in a near-broadside line-of-sight array, for a 5-bit and 7-bit DAC.

noise gain is proportional to the number of transmitters in the array. At broadside, all the digital beamforming coefficients are equal to one, so there is no decorrelation effect. A close-up of the region around 0° is shown in Figure 6.7, for a 5-bit and 7-bit DAC. The "non-averaging" broadside region is shown to be inversely proportional to the DAC's dynamic range. Essentially, the imaginary component of the beamforming coefficient is responsible for mixing enough I into Q and Q into I to decorrelate the quantization noise. Therefore, we can say that the beamforming coefficient should satisfy:

$$\pi \sin \theta_{min} > LSB \tag{6.1}$$

This relationship shows that as the LSB size shrinks (when more bits are added to the DAC), this criterion is met at smaller and smaller angles. Moreover, in a multi-user scenario the presence of the second user would be sufficient to decorrelate the quantization noise.

These results indicate that we can take advantage of quantization noise averaging to achieve much better out-of-band emissions performance in an array compared to a single-element transmitter. Figure 6.8 shows the output spectrum of a 16 element array (with two users), using the upsampling filters designed above and quantization noise averaging. As expected, the in-band power improves by 6dB/element-octave while the quantization noise power increases by 3dB/element-octave, giving 3dB improvement in OOB emissions for each factor of two in array elements. The quantization noise is fully decorrelated such that both users receive the same total emissions power.

6.4 DSP Path

802.11n uses an OFDM-based modulation scheme. Additionally, due to the low-frequency operation it is expected that channel propagation would result in rich scattering and multi-

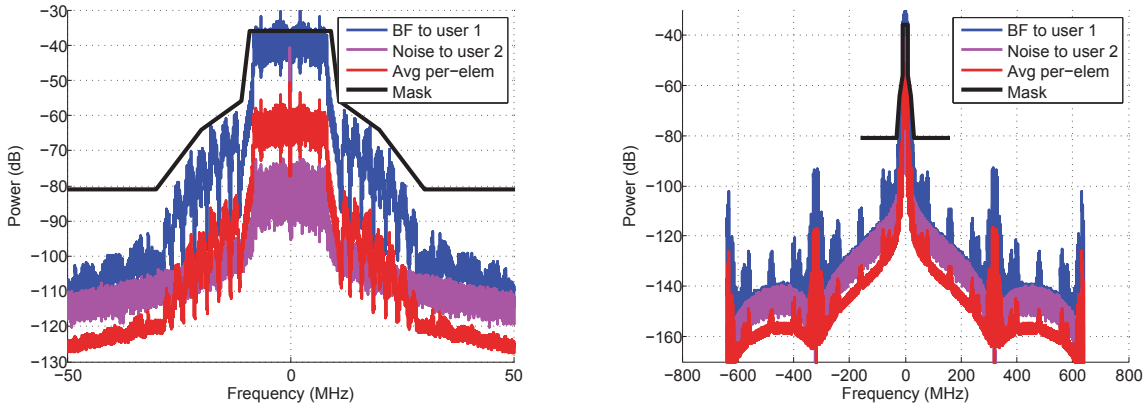


Figure 6.8: Output spectrum of a 16-element array with two users in a 802.11n mask.

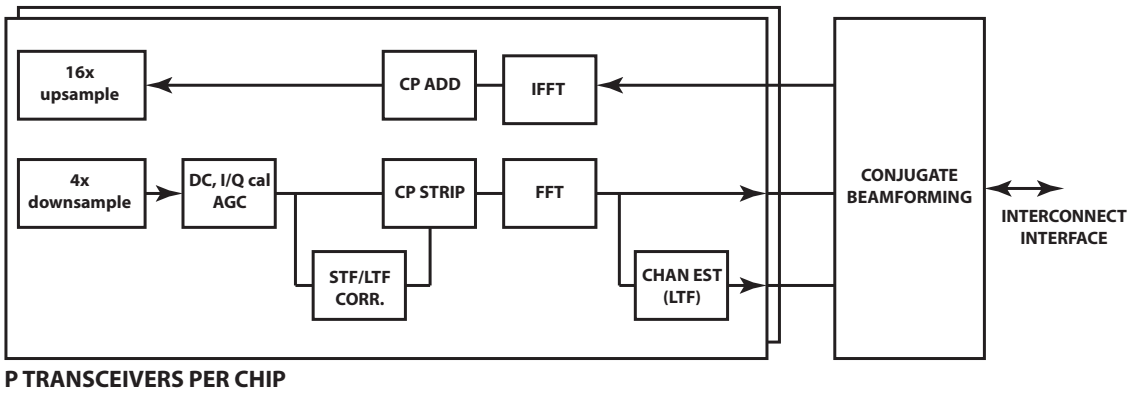


Figure 6.9: DSP path for massive MIMO SoC using 802.11n.

path. Therefore, this system uses full frequency-dependent beamforming.

There are no 802.11 variants that yet support MU-MIMO natively in the uplink. Therefore, we assume some differences in the frame structure to accomodate this functionality. First, we assume that each user transmits a packet header, including short training field (STF) and long-training field (LTF), in a time-multiplexed manner. Second, we assume that carrier frequency offset (CFO) and sampling frequency offset (SFO) are compensated at the user side, using a scheme such as that suggested in [151].

The DSP chain is shown in Figure 6.9. In the receive path, the signal is downsampled to baseband using the resampling filters discussed above. 802.11n correlators are used to detect the STF and establish symbol timing and an FFT window [152]. The LTF is used to estimate the channel in the frequency domain between each user and that antenna element. As discussed in Section 3.3, those channel estimates are used to set local conjugate beamforming coefficients on every subcarrier. They are also themselves processed by the beamformer and

forwarded to the central processor to compute the zero-forcing coefficients. In the signal path, the incoming data is split into FFT symbols, beamformed with the other elements locally on the same SoC, and then sent to the interconnect router.

In the transmit path, the link router receives user data streams from the interconnect. These user data streams are beamformed in the frequency domain and then converted to the time domain through IFFT. After insertion of the cyclic prefix, the resulting TX data is upsampled and sent to the DAC.

The link router is responsible for interfacing with the data interconnect. In the receive direction, its core function is to receive data forwarded from other chips, perform distributed combining, and send the result of that computation to the next step in the chain. It must be able to apply the appropriate delays on both incoming serdes lanes and the data generated on chip to align them properly. In the transmit direction, it applies similar operations but does not need to execute any combining step.

6.4.1 ADC and DAC Timing

The ADC and DAC require a sampling clock which is clean and low-jitter; any jitter in this clock would degrade the effective number of bits (ENOB) and resolution of these converters. At the same time, the supply and clocking domain of the DSP is noisy due to the large amounts of digital switching. Accordingly, it is chosen to generate the sensitive analog clocks independently from the digital clock. These two domains will be mesochronous to each other.

The clock crossing from analog to digital domain is handled using an asynchronous FIFO. These two domains will have a static but unknown phase offset between each other. Because there is no drift between the two clocks, the FIFO can be quite shallow and there is little concern about the edges crossing each other.

The latency of each ADC FIFO (measured in real time units) would be naturally estimated by the channel estimator and included in the relevant beamforming coefficients. Even if the FIFO stalls due to phase drift between analog and digital clocks, this latency is preserved. As discussed previously, this is very unlikely, so can be ignored.

The latency of the DAC FIFO is not known and does not necessarily match that of the ADC FIFO. Accordingly, for a given antenna element the TX delay may differ from the RX delay. This delay mismatch should be less than one ADC sampling period, so its impact is at most 0.25 baseband sample. Nevertheless, it may be desired to correct this delay; this could be done using a loopback calibration.

6.4.2 Reciprocity Calibration

TDD massive MIMO systems rely on measuring uplink channel state information to form the downlink beamforming coefficients. This is accurate only for environmental effects. Reciprocity of uplink and downlink does not extend to the transceiver front-ends themselves. Any mismatch between the RX complex gain and TX complex gain at an antenna element

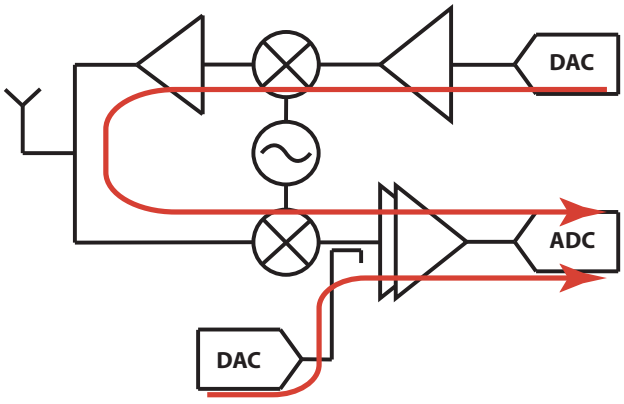


Figure 6.10: Per-element reciprocity calibration scheme consisting of measurement TX and loopback.

will introduce downlink beamforming error. It is necessary to measure and correct for this reciprocity error in order to obtain good downlink performance.

A number of reciprocity schemes have been proposed in the literature, such as [53]. Most of these algorithms use over-the-air measurements to a reference antenna, either in the environment or within the same array. This is undesirable since it is not generally possible to have a calibration antenna with good propagation characteristics to all the array elements.

We propose a reciprocity calibration algorithm consisting only of per-element measurements (Figure 6.10). Each transceiver should be equipped with a reference DAC, to measure the RX path, and a loopback capability to measure the aggregate TX and RX transfer function. The reference DAC is used to measure the receive path up to the resolution of this measurement TX. If this calibration DAC has a stable and accurate voltage reference (such as a bandgap), then it can be assumed that all calibration DACs at every element are implicitly calibrated to each other, up to the resolution of this DAC.

Having measured the RX transfer function precisely at each element, the loopback measurement obtains the cascaded TX-RX gain. This makes it possible to extract the TX transfer function by itself and therefore devise the appropriate reciprocity calibration coefficients. The results of this simulation are shown in Figure 6.11. Note that in practice it may be necessary to make both calibration measurements in a frequency-dependent fashion.

6.4.3 Link Router

The link router handles the interface to the data interconnect. The multiple chips in the array are interconnected in a mesh network. In the receive direction, each chip forwards its data to the next hop in the interconnect chain. The task of the link router is to accept incoming packets from other chips, align them to each other and to the data generated locally, perform the distributed summation, and forward the result to the next hop in the chain (Figure 6.12).

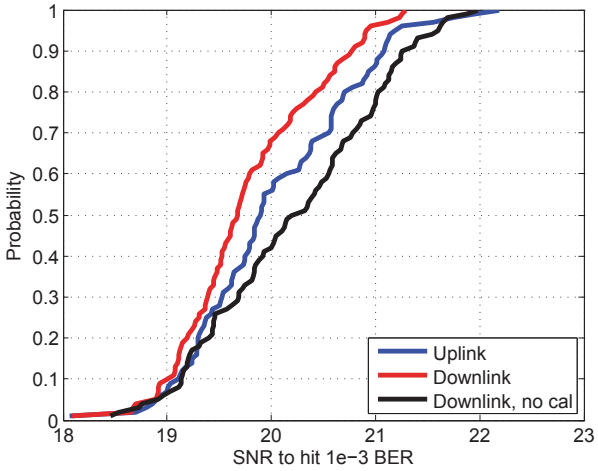


Figure 6.11: Downlink performance with and without reciprocity calibration.

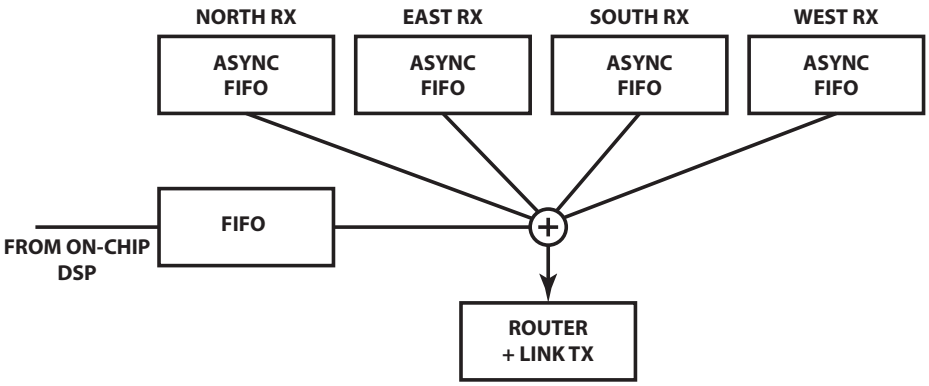


Figure 6.12: Link router for receive (uplink) direction — the incoming data is synchronized, summed with the delayed output from the local DSP, and forwarded.

In the receive direction, this alignment can be accomplished by referencing all timing to the frame start. Each deserializer is equipped with a small FIFO while the DSP is equipped with a larger FIFO. Each of these FIFOs can be augmented with a tag array which stores which FIFO index corresponds to a certain packet ID. When a new frame is detected, all routers reset their packet ID counters. Then as the router receives incoming packets, it extracts the incoming packet IDs, finds the appropriate packet in the DSP FIFO, and sums them together. This only needs to be performed once at the beginning of each frame. Subsequent packets in the frame can remember the measured latencies of each path and avoid searching through the tag array.

A similar operation should be performed in the transmit direction. Even though there is no summation in the transmit direction, it is important to match the latencies to all

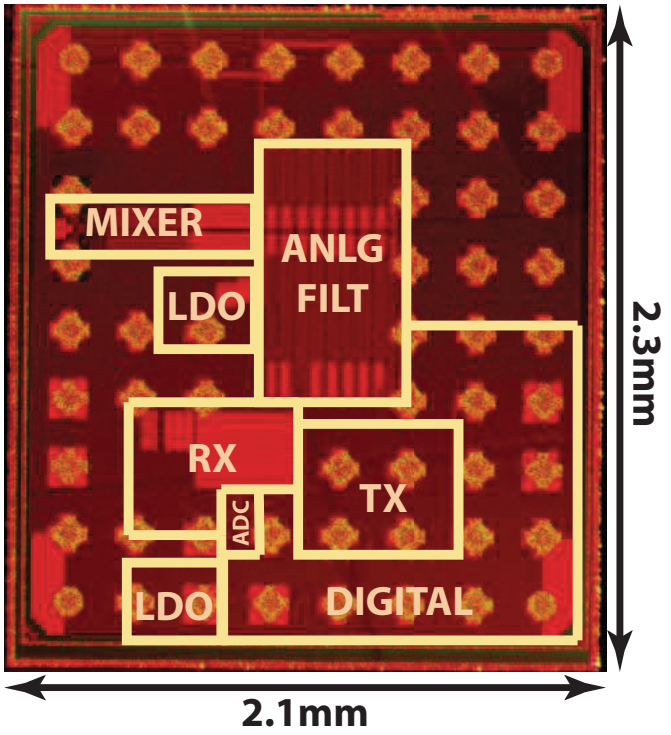


Figure 6.13: Die photo for test chip.

TX chains to ensure that the data at the DAC is aligned everywhere. Unfortunately the interconnect latencies in the TX direction may differ from the RX direction since separate clock crossing FIFOs and serdes IPs are used. Automated transmit interconnect alignment is still an open item for this SoC implementation.

On top of alignment, the link router should check validity of incoming packets. This can be accomplished by using checksums and header fields to detect transmission errors. If an error is detected, the router can throw out that packet and insert all zeros instead.

6.5 Test Chip

A test chip was taped out in 65nm CMOS as an initial design point towards the realization of this CMOS SoC. The test chip incorporated a full RX and TX path, on-chip supply generation, resampling filters, and digital snapshot memories for test stimulus and recording. The die micrograph is shown in Figure 6.13.

This test chip was used to validate the architecture and performance of the radio transceiver. Future chip tapeouts will aim to integrate multiple radios, more DSP, and serdes lanes on a single CMOS SoC.

Chapter 7

Design of a Massive MIMO Array at E-Band

This chapter describes the design and implementation of a massive MIMO uplink array using the principles developed in the preceding chapters. Compared to existing mm-wave arrays, the dual objectives of this project are to increase the number of array elements from 32 to 128, and to increase the number of simultaneous users from 1 to 16. These requirements vividly illustrate the need for the scalable and modular architecture described in Chapter 3. On top of that, the synchronization techniques developed in Chapter 4 and Chapter 5 are used to achieve scalable and low-power system synchronization.

This chapter describes how the techniques in this thesis were practically used to design and implement this prototype. The focus is on the hardware engineering and signal processing stack used to establish wireless links.

7.1 System Overview

The state of the art in mm-wave arrays is surveyed in Section 3.2.1. To recapitulate the key points, mm-wave systems today are characterized by

- Limited array size: Array sizes are generally limited to 32-element phased arrays integrated either on chip or in package/board. Some 64-element arrays are beginning to appear.
- Limited multi-beam support: Only a small number of multi-beam arrays have been reported. These designs all use a partially-connected array architecture, which partitions different beams to disjoint subarrays. This introduces a strong tradeoff between number of beams and effective array size.
- No on-chip multi-beamforming: There are no phased array chips with the ability to form multiple beams using the same transceivers and antennas.

	Nominal	Wide BW	Range	Downlink[a]
Bandwidth (MHz)	250	2000	2000	2000
BS NF (single element) (dB)	10	10	10	10
BS input referred noise (dBm)	-80	-71	-71	-71
BS average link SNR (dB) **(16 QAM)**	14	14	14	14
Number BS RX	128	128	128	8
BS average SNR per element (dB)	-7	-7	-7	5
BS input power requirement (dBm)	-87	-78	-78	-67
Range (m)	**100**	**30**	**300**	**300**
Loss at range (28GHz carrier)	-102	-93	-111	-111
Average TX EIRP requirement (dBm)	15	15	33	44
Number UE TX	1	1	8	128 (16)[b]
UE avg TX power per element (dBm)	**15**	**15**	**15**	**14**

[a] BS and UE quantities are swapped for this column.
[b] 128 BS antennas serving 16 users

Figure 7.1: Link budgets for actual system uplink, along with hypothetical extensions to wider channels, longer range, and downlink scenarios.

This research project aims to design a fully-connected mm-wave array with 128 elements forming 16 simultaneous beams. In keeping with the fully-connected design approach, each beam is mapped to every single antenna. This ensures maximal use of the hardware and consequently maximal spatial processing gain. The initial prototype is implemented using commercial off-the-shelf (COTS) components. Future versions of this prototype will integrate a custom front-end phased array on chip which has the ability to form 16 spatial streams from 16 antenna interfaces [137]. As such, the initial prototype addresses and overcomes the first two limitations outlined above, while future research efforts will address the third and final point.

7.1.1 Link Budget

This mm-wave array is designed to operate with 128 base-station antennas communicating with 16 spatially multiplexed users. Each user is equipped only with a single antenna, for simplicity of design and implementation. This wireless system could be easily extended to support multi-antenna users, which would enable the range of the link to be extended. Only the uplink direction is implemented in this prototype generation; these ideas can be extended in a similar vein to implement the multi-user mm-wave downlink.

This system operates at 72GHz, which is an emerging frequency band of interest to the industry and the FCC. Due to limitations in the COTS components and FPGA platform used, the bandwidth is limited to a 250MHz RF channel.

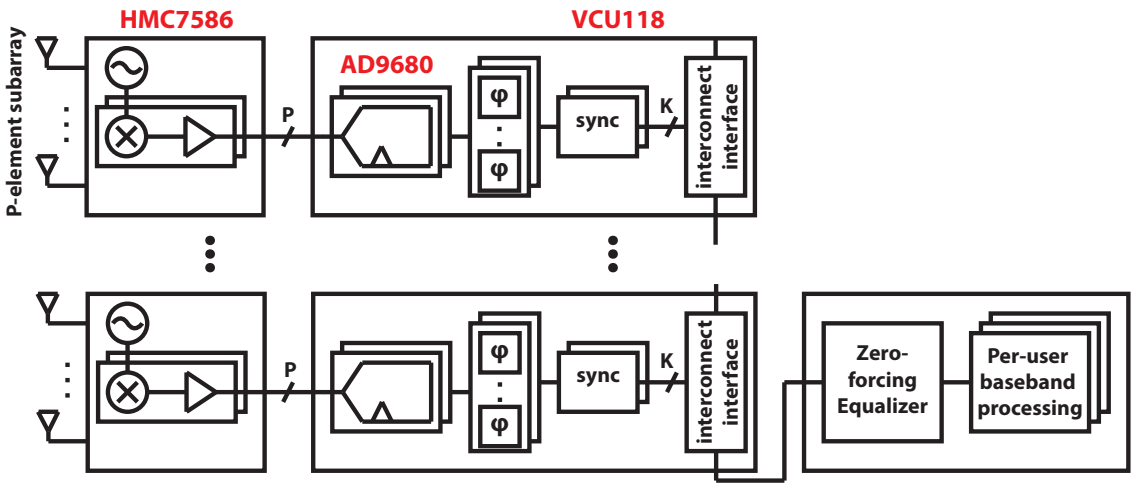


Figure 7.2: Modular array architecture for a massive mm-wave MIMO array.

An uplink link budget is shown in Figure 7.1. This table describes the nominal scenario as well as two hypothetical scenarios with wider channel bandiwdth and multi-antenna user equipments (UEs). Finally, an example downlink link budget is shown to illustrate how a downlink with similar range and datarate could be implemented.

7.2 Hardware Implementation

7.2.1 Modular Architecture

In keeping with the principles outlined in Section 3.2, our massive MIMO mm-wave array is designed around a modular and scalable architecture. As shown in Figure 7.2, the core unit is a 16-element subarray. In this testbed, each radio channel is served by a full digital receiver spanning antenna to ADC and digital samples. The subarray is accompanied by digital signal processing which performs ADC de-skew, channel estimation, subarray conjugate beamforming, and inter-subarray timing alignment (Section 5.3). Other, more analog-intensive implementations, fit cleanly within this array architecture.

A key difference between the present array and the state of the art is the fully-connected architecture employed here. Each subarray receives every single one of the 16 user streams. The per-subarray signal processing essentially results in each subarray's independent estimate of the user streams; these estimates are then fused across subarrays to form the aggregate array-level user beams. Because the distributed synchronization and TTD compensation can cleanly separate and abstract the modular boundaries, the data interconnect consists of a very simple forward-and-sum operation. Finally, user separation, equalization, and back-end processing is carried out following the data aggregation. This operation is largely equivalent to 16 parallel user basebands.

An important highlight of the proposed array architecture is the highly modular and scalable design. Crucially, each subarray operates fully autonomously in terms of both beamforming and synchronization. While the intra-subarray complexity is fairly high, the only signals which cross the subarray hierarchy boundary are the serdes lanes comprising the data interconnect. As a result, it becomes quite simple to implement very large fully-connected mm-wave arrays in this manner.

7.2.2 Testbed design

A fully-functioning prototype was implemented using COTS chips and custom printed circuit boards (PCBs). One 16-element subarray is implemented from a number of boards. The signal path consists of a 72GHz direct-conversion receiver (HMC7586) with on-board aperture-coupled patch antenna (Figure 7.4), followed by variable-gain amplifiers (VGAs) and the AD9680 dual-channel 500MSps ADC. The receive channels are grouped into quads which are implemented on a pair of boards — one mm-wave board and one baseband one. A subarray therefore consists of four such quads. The mm-wave boards are also paired with a power generation and LO distribution board which generate the supply voltages and clocking reference for the radios. In contrast, the baseband board contains power and clock generation for the baseband path. The baseband assembly is shown in Figure 7.3.

Each ADC has a 12.5Gbps high-speed serdes lane (over JESD204B protocol) which transmits its digital samples to an FPGA. The FPGA is chosen as a VCU118 evaluation kit from Xilinx, which comprises a full board centered around a 16nm Ultrascale+ FPGA. The FPGA is responsible for the following main tasks: (1) JESD interace with all 16 data converters in a subarray, (2) per-subarray signal processing, and (3) inter-FPGA data interconnect. Finally, each FPGA is also equipped with an Ethernet link by which it may occasionally send partially processed frames to a PC for further signal processing and visualization.

The UE consists of similar but simpler hardware (Figure 7.5). Each UE is equipped with a TE0714 evaluation module centered around an Artix-7 Xilinx FPGA. The FPGA drives an AD9512 DAC which in turn drives a mm-wave transmit chain consisting of the HMC8118 upconverter and HMC7543 power amplifier. Each UE is equipped with power generation for all internal supplies, a 125MHz oscillator which serves as the reference for all baseband and mm-wave frequency references, and clock generation. As such, each UE is fully autonomous relative to all other users.

7.3 Signal Processing Chain

7.3.1 DSP Chain Overview

The hardware described above is supported by a fully PHY layer baseband chain running in combination of FPGA RTL and Python on a PC. The signal processing chain is illustrated in Figure 7.6.

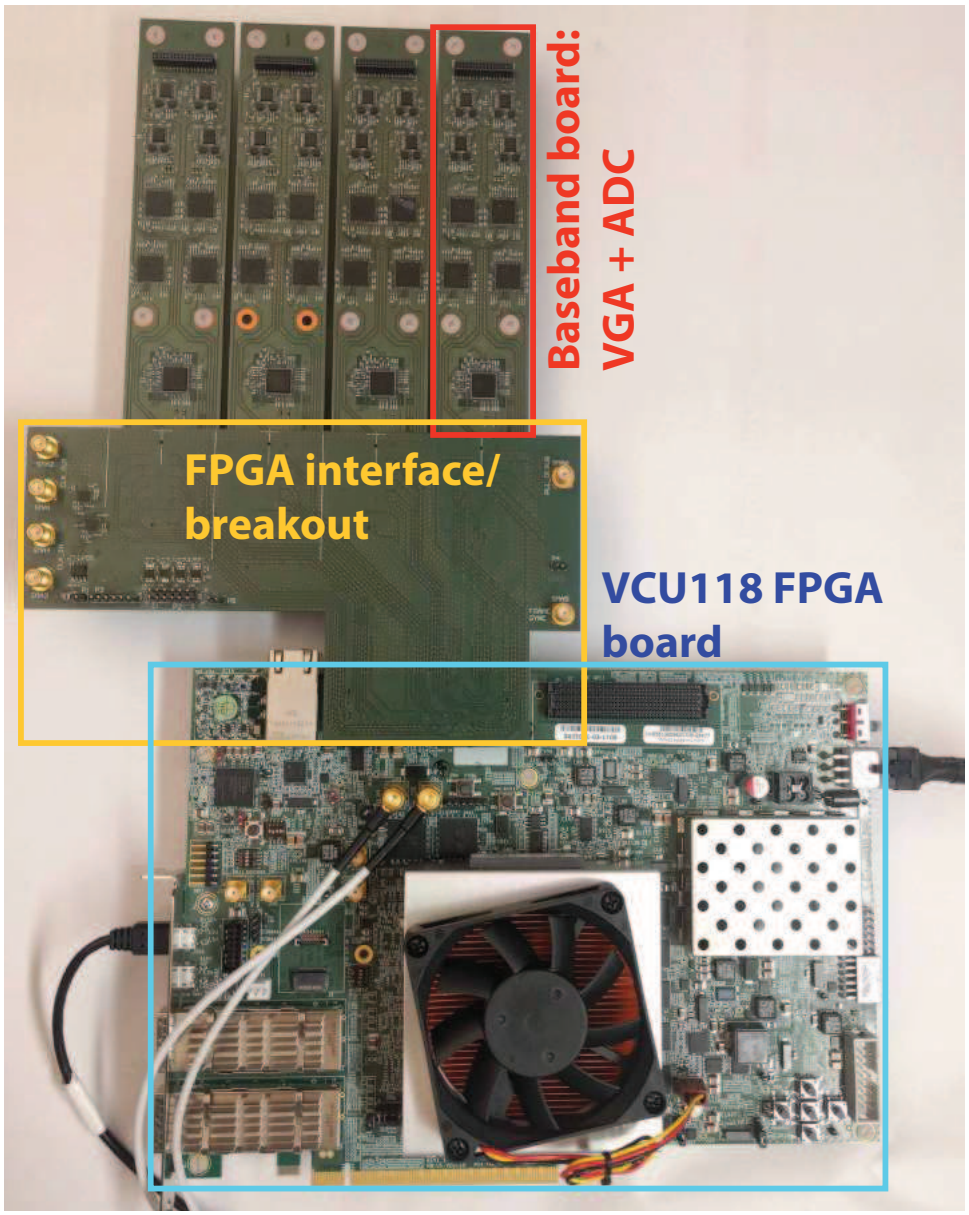


Figure 7.3: Baseband path for a single 16-element subarray.

Each antenna is first processed by calibration loops including DC offset and I/Q imbalance correction as well as noise power estimation. Subsequently, each antenna is filtered by a channel filter which is implemented as a root-raised cosine filter with rolloff factor 0.25. This filter acts to reject blockers in adjacent channels. It also matches to the transmit pulse-shaping filter (an identical root-raised cosine) to give an overall Nyquist filter response ensuring no ISI from resampling operations.

The filtered signal is kept in the oversampled regime. A Golay estimator at each antenna

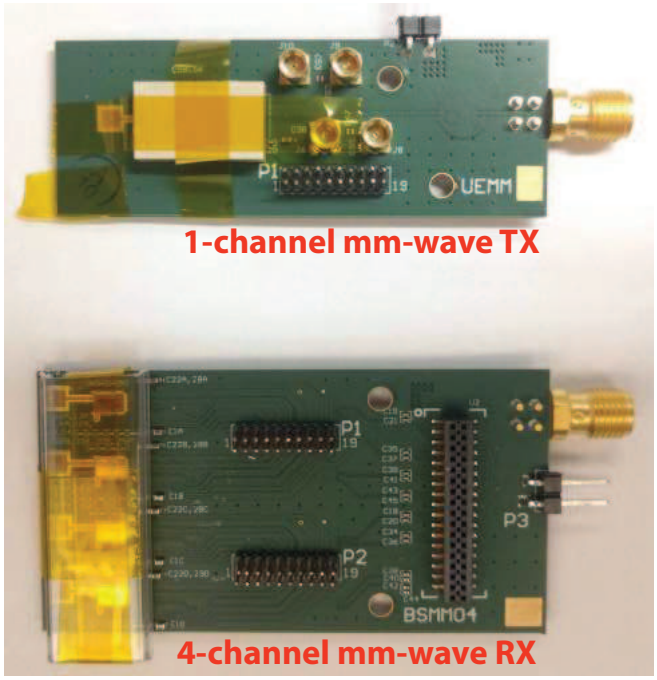


Figure 7.4: RX and TX mm-wave front-end boards including on-board aperture-coupled patch antenna.

exploits the Golay pilots transmitted by each user (in a time-multiplexed manner) and estimates the channel gain and timing delay for each user. The channel gain estimates are fed to the conjugate beamforming network while the delay estimates are fused across users at each antenna element and used to compute the ADC de-skew coefficients. The ADC de-skew is implemented as a 5-th order Farrow interpolation filter. This structure is followed by a conjugate beamforming matrix which estimates the user beams at each subarray autonomously. Because mm-wave channels are expected to be mostly line-of-sight, it is considered sufficient to employ frequency-independent conjugate beamforming in this step. This dramatically simplifies the required hardware complexity since no FFT is needed at each antenna.

Subarray conjugate beamforming converts the processed signals from the antenna domain to the beam domain. Subsequently, each beam on each subarray is processed by a Golay timing estimator which recovers the optimal sampling instant of each beam. This pilot-aided timing recovery loop is used to compensate for TTD effects by aligning the sampling instants of each beam across subarrays. The timing adjustment is again implemented using a 5th order Farrow interpolation filter. This interpolation can be followed by downsampling to a baseband sampling rate, reducing the datarate that must be transmitted over the inter-FPGA serdes links. After beam synchronization, the data interconnect may cleanly sum together the beam samples generated by each subarray. This process is carried out by the data aggregation network.

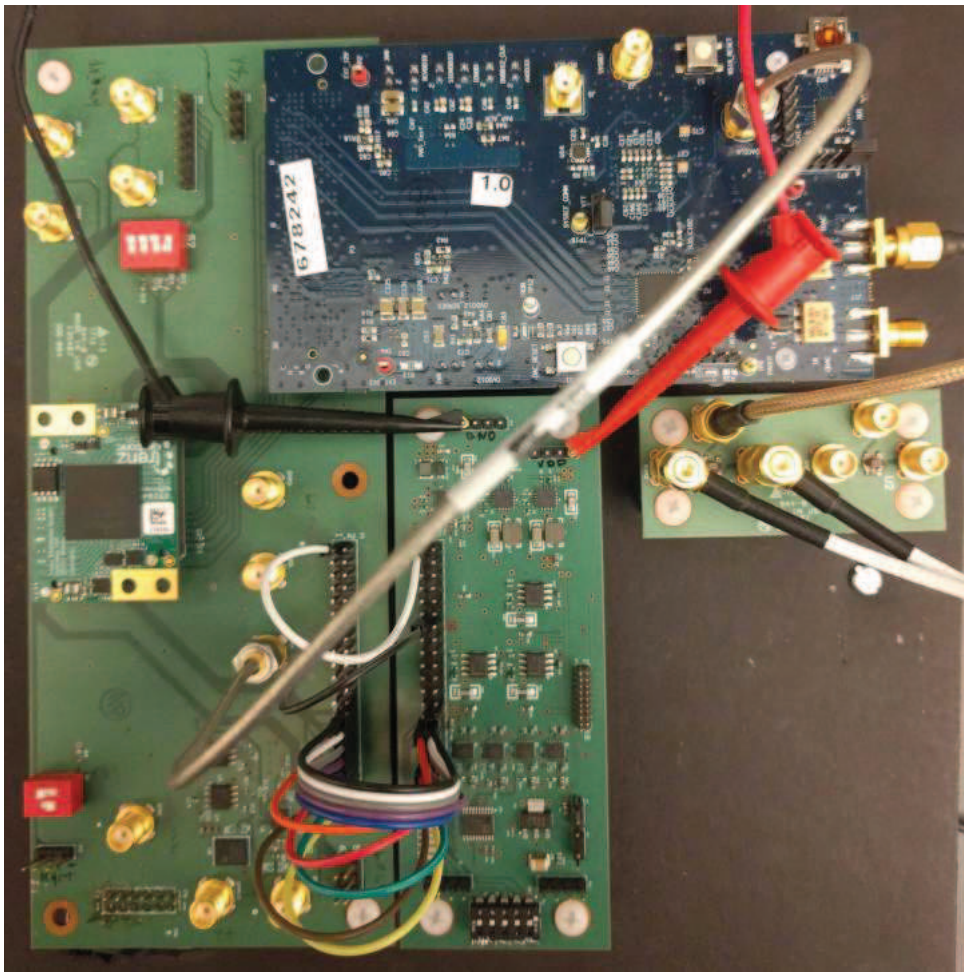
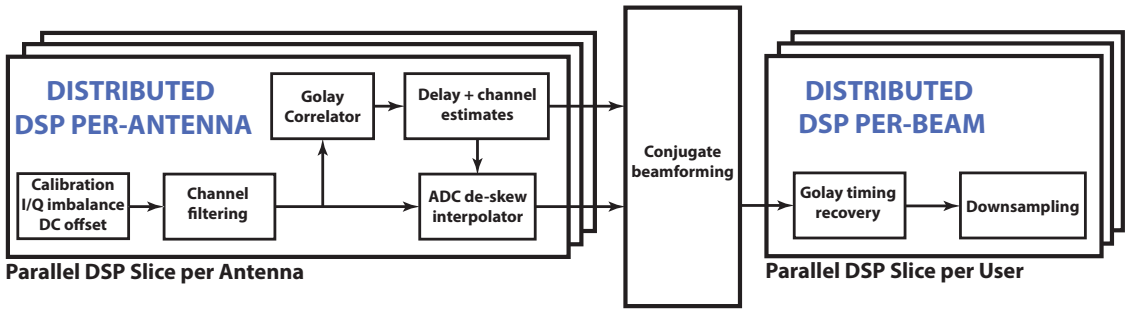


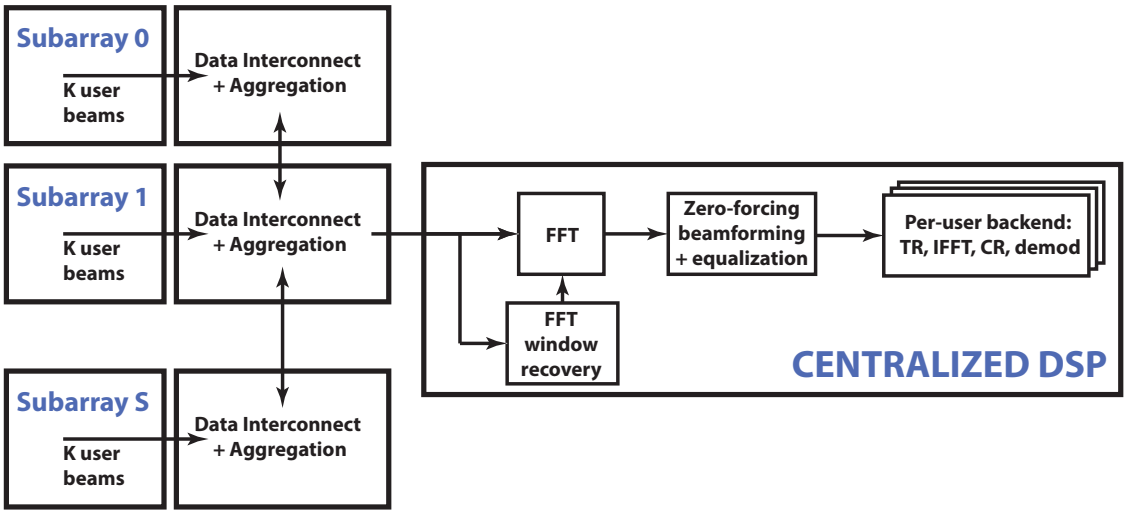
Figure 7.5: User equipment for testing, including FPGA, DAC, clocking, power generation, and LO path.

Following the full subarray fusion, a back-end processor separates out the user data streams, equalizes them, and performs back-end processing. The user separation and equalization is performed by a $K \times K$ MIMO frequency-domain equalizer (FDE). By using frequency-dependent processing, the zero-forcer is able to compensate for multipath propagation which gives rise to frequency-dependent inter-user interference. For this system, a 256-point FFT with 16-element cyclic prefix is utilized. The FDE coefficients are estimated by transmitting FDE pilots consisting of a known pilot tone on each subcarrier. Multiple such FDE estimation pilots can be transmitted to improve the channel estimate. The number of FDE training pilots is a system parameter which is adjusted according to channel conditions.

Following the FDE, the frequency-domain user data streams are fully spatially separated. Finally, each user's back-end consists of inter-twined timing- and carrier-recovery loops. The



(a) Per-subarray distributed signal processing chain.



(b) Data interconnect and centralized user separation and back-end.

Figure 7.6: Signal processing chain for mm-wave massive MIMO link.

timing recovery loop is intended to compensate for each user's SFO using the structure described in Section 5.4. The carrier recovery loop operates in the time-domain to compensate for phase noise and residual carrier frequency offset in each user's data stream. Finally, a slicer demodulates the received constellation and makes hard decisions.

7.3.2 Frame Structure

This signal processing chain is accompanied by the appropriate frame structure used for synchronization and channel estimation (Figure 7.7). First, each user successively transmits (in time-multiplexed fashion) a Golay pilot consisting of two complementary sequences of a specified length. The length of the Golay sequence may be parametrized and chosen based on the channel conditions. In this work the nominal Golay length was 32 samples for each sequence. An inter-user guard interval of 10 samples provides a buffer to prevent inter-user

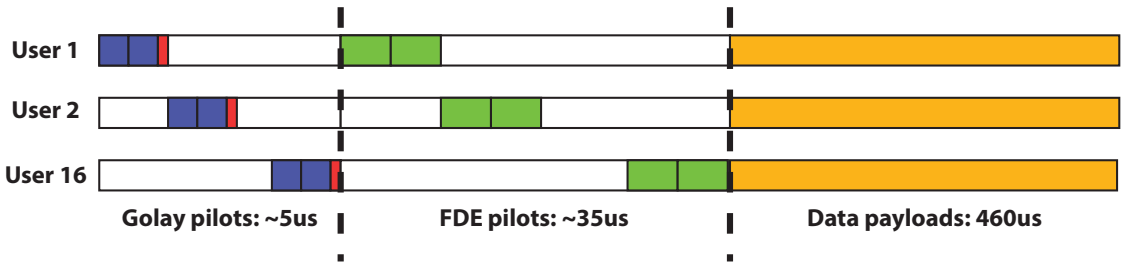


Figure 7.7: Frame structure for mm-wave multi-user MIMO link.

interference.

After each user has transmitted its Golay pilots, each user successively sends a number of FDE training blocks. The structure of the training blocks is described in detail in [137]. The key idea is that one pilot symbol is sent every N subcarriers, with N nominally equal to four. The pilot symbols are used to compute both a coarse CFO estimate as well as a frequency-smoothed channel estimate. As long as the coherence bandwidth is larger than $256/N$ (in units of subcarrier bandwidths), there is no loss from sending subsampled FDE tones.

Finally, after all pilot sequences have been sent all users simultaneously send their data payloads. Each paylod consists of a small number of seeds for the CR loop followed by the actual data packet. For nominal system parameters, each user sends 74 samples worth of Golay pilots and 544 samples of FDE pilots. This results in a total pilot overhead of approximately 10,000 samples, which at a 250MHz sampling rate comes out to about $40\mu s$. If the frame length is at least $500\mu s$, the total pilot overhead is less than 10% of a frame, which is a very acceptable quantity. If needed, this overhead could be reduced by code-multiplexing the user pilots so that they may be sent simultaneously.

7.3.3 Two-stage Beamforming: ZF at Low SNR

It is interesting to note how the two-stage beamformer allows high-performance zero-forcing to operate even with very low per-element SNRs. As shown in the link budget in Figure 7.1, the thermal SNR at each element is 0dB and the SINR is -12dB due to the inter-user interference. By using time-multiplexed Golay pilots, the conjugate channel estimation proceeds with SNR gain proportional to the code length. For the nominal length-32 Golay codes, this comes out to 15dB SNR for conjugate estimation. This SNR is sufficient for moderate but not extremely high resolution. In particular, it is worth pointing out that if a frequency-dependent ZF estimation was performed at each element, the estimation SNR would only be 0dB. This SNR is far too low to accurately compute the ZF matrix, so as a result it would be impossible to cleanly separate out the user streams.

Instead, the ZF FDE pilots are beamformed by the previously estimated conjugate beamformer. In the full 128-element array, this provides a 21dB gain in SNR, resulting in 21dB

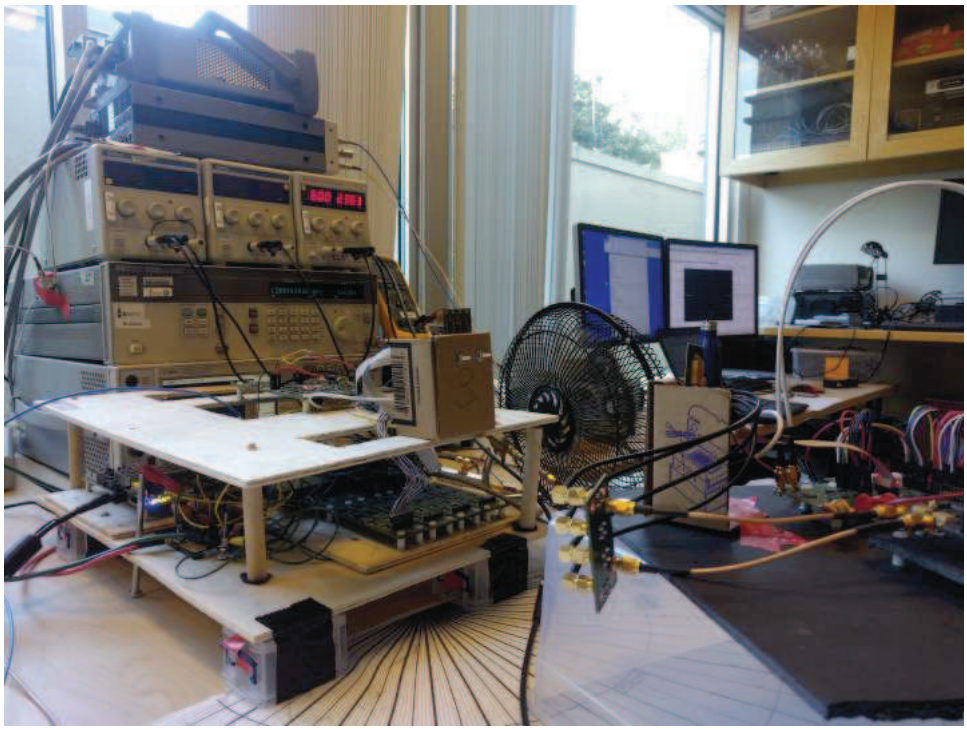


Figure 7.8: Lab setup for 4x1 SIMO testing.

SNR at the input of the ZF channel estimator. Especially when averaging across two or more FDE pilot blocks, this SNR is appropriate to obtain a good estimate of the channel state information. As a result, very high-performance ZF beamforming is achievable even with extremely low per-element thermal SNR. This is made possible through the use of the two-stage beamformer.

7.4 Results and Measurements

The hardware described above was assembled into a full system prototype. The signal processing stack was implemented using a combination of FPGA RTL and Python software running on the lab PC. The PC communicates over Ethernet with one or more FPGAs, collecting the digital samples corresponding to the output of its on-board DSP. The rest of the processing is performed by the Python software.

As of the time of writing, results were only available for a 4x2 MIMO link. This testing shows that the key elements of the hardware are operable and that multi-user beamforming is achievable using this hardware/software setup. A picture of the test setup is shown in Figure 7.8. The left side of the figure contains the base-station panel including test equipment. The foreground shows the UE transmitter and part of its baseband path. In the background is shown the PC running the Python software which visualizes the link status.

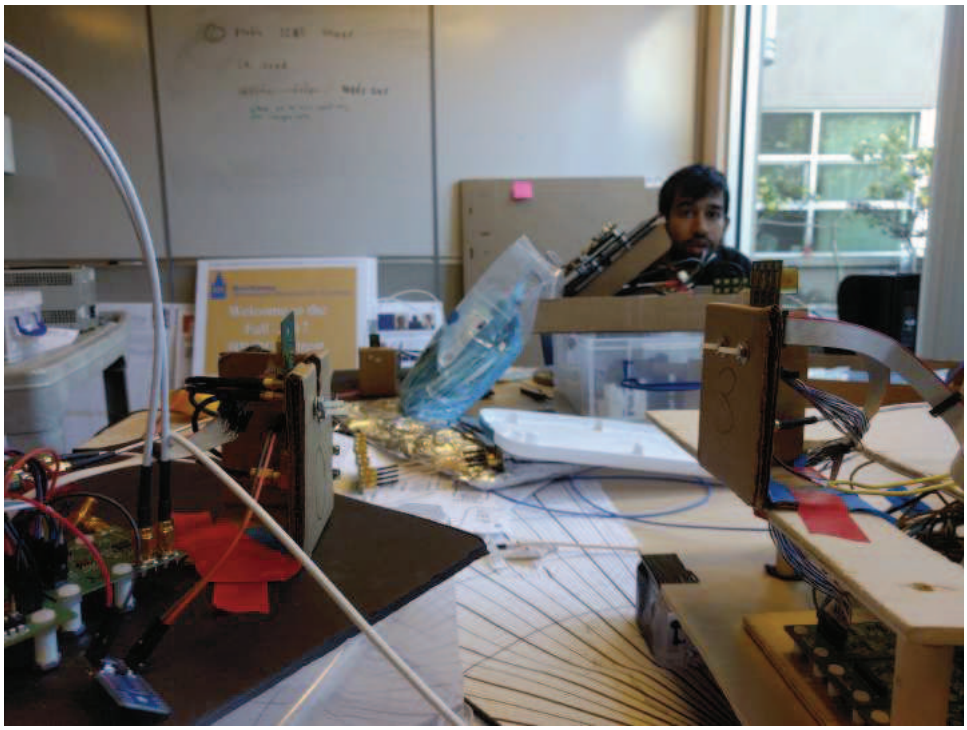


Figure 7.9: Zoomed-in view of 4x1 link with antennas and mm-wave boards.

Figure 7.9 shows a close-up of the TX and RX antennas for a 4x1 link configuration. A postdoc is also shown in the background. Figure 7.10 shows measured constellations for various testing configurations. In a 4x1 MIMO link with LO reference coming from a signal generator, the measured SINR was 23dB. The corresponding 64-QAM constellation is shown. When a on-board mm-wave PLL was used (locked to a 125MHz crystal), the SINR degrades to 18dB, sufficient for a 16-QAM constellation but not 64-QAM. Finally, a 4x2 MIMO scenario was measured with a noisy environment (low gain in the RX front end). SINR was measured at 9 and 10dB for each user, and the dual QPSK constellations are shown in Figure 7.10.

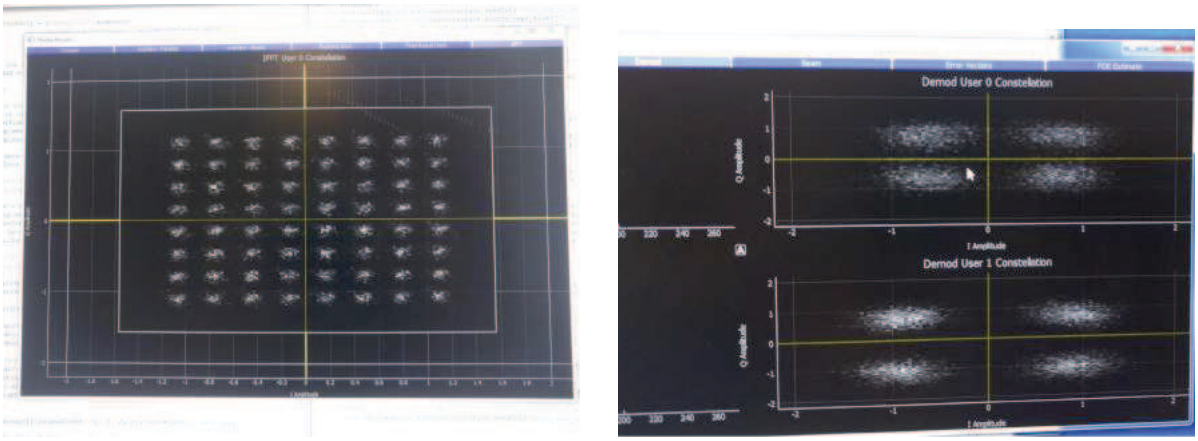


Figure 7.10: Constellations for (a) 4x1 SIMO link with high-quality LO reference and (b) 4x2 MIMO link.

Chapter 8

Conclusion

This work has proposed system architectures, signal processing techniques, and hardware prototypes which can address the needs of future wireless networks. This is accomplished in two ways: first by making use of very high carrier frequencies which have a large amount of unused spectrum, and second by exploiting the spatial dimension of wireless channels to separate users in space on top of traditional time and frequency multiplexing. These two principles underlie the fifth generation of mobile networks and as such, this thesis presents an avenue toward realizing the next ten years of mobile connectivity.

This thesis has explored theoretical and practical questions surrounding the design of massive multi-user beamforming antenna arrays. The capstone is a 128-element phased array operating in the E-band, showing how array design principles can be applied to emerging high-frequency spectrum. This testbed achieves 4x increase in total array size and 16x increase in number of simultaneous spatial beams (from 1 to 16) compared to other mm-wave arrays reported at 28, 39, or 60 GHz. The principles illustrated in this array prototype can be applied to improve the capabilities and capacity of mm-wave systems operating in many frequency bands. In a decade systems such as this, in unobtrusive form factors, may be deployed ubiquitously in the network to provid very-high-capacity wireless networking.

8.1 Thesis Contributions

This work advances the state of the art around algorithm, system, and hardware design for massive multi-user arrays. In particular, the novel contributions include:

- Proposes a scalable and distributed architecture for implementing massive arrays out of unit common modules. The proposed architecture abstracts away specific implementation-specific details but rather focuses on general principles common to all such arrays.
- Develops a time-domain implementation of the per-subcarrier MIMO zero-forcer/equalizer, which achieves the same performance as the frequency-dependent algorithm but permits a time-domain implementation.

- Develops the analysis of phase noise averaging and inter-user phase noise interaction for various LO distribution schemes.
- Identifies the mechanism of phase noise averaging for uncorrelated phase noise in SIMO and MIMO arrays, and proposing scaling of distributed VCOs in order to save power in arrays.
- Identifies interactions between the bandwidths of PLLs and carrier recovery loops and the existence of an optimal PLL bandwidth which depends on the CR bandwidth.
- Proposes a hierarchical baseband synchronization strategy which consists of intra- and inter-subarray synchronization steps.
- Develops a background ADC de-skew algorithm utilizing user Golay pilots.
- Develops a true-time delay compensation scheme consisting of per-subarray timing recovery on each user data stream.
- Proposes a time-frequency multi-user SFO compensation loop using a second-order loop.
- Architects and designing the DSP chain for a massive MIMO system-on-chip operating at $<$ 6GHz bands.
- Proposes a signal processing and frame structure suitable for mm-wave massive MIMO.
- Builds a prototype massive MIMO system operating at E-band, with 128 antennas serving 16 simultaneous users.

8.2 Future Directions

This work can be extended in several ways. First, the demonstrated prototype massive MIMO at E-band operates only in the uplink direction. Generally speaking, the downlink design should be similar, with two important differences. First, reciprocity calibration is a critical component of TDD massive MIMO arrays, so a suitable calibration scheme (both algorithm and hardware) should be proposed. Chapter 6 proposes a first step towards thsi goal by presenting an autonomous loopback-based calibration scheme for each transceiver independently. Second, the signal statistics of the downlink multi-user beamformed signal should be studied both to derive specs for the transmit chain as well as to identify potential array gains which can be exploited to reduce complexity. In particular, both dynamic range and peak-to-average power ratio are key transmit specs. It is expected that multi-user beamformed signals will add in power, increasing dynamic range by a factor of K. Moreover, it is expected that by adding incoherently, multi-user signals will converge to a Gaussian

random variable and therefore that the PAPR of the downlink data will be Gaussian. These hypotheses should be studied.

A second research direction would be to make the signal processing architecture less dependent on the frame structure. In particular, both the conjugate channel estimation and ADC de-skew rely on the Golay pilots. It would be desirable to expand this to more general modulation schemes. One potential avenue would be to exploit the cross-correlation between different antenna signals to compute both direction and delay information. This could operate as a background loop obtaining both channel state information and ADC skew coefficients. A related goal involves devising a more efficient frame structure. In particular, if the user pilots could be code-multiplexed, then the overhead devoted to pilot sequences can be reduced. This would allow more efficient use of short frame lengths and consequently better performance in high-mobility scenarios.

Array design and spatial wireless communications may also intersect with other emerging technologies in the wireless space such as full-duplex. It may be possible to utilize the spatial selectivity of an antenna array to simplify a full duplex operation. For example, if the TX and RX arrays do not share antennas, they could benefit from the spatial isolation conferred by the beamforming network. This may require the use of analog beamforming to realize this separation. Another challenge with massive MIMO is extending it to frequency-division duplex (FDD) scenarios where the TX and RX use different carrier frequencies. FDD operation destroys the channel reciprocity, meaning that uplink CSI cannot be reused to create the downlink beamformer. FDD massive MIMO requires the design of efficient downlink channel estimation codes in order to efficiently obtain this CSI.

Finally, the design of higher network layers natively for multi-user directional systems should be considered. In particular, common MAC layer issues include initial access, synchronization, scheduling, handovers, and so on. These functions may interact significantly with the hardware and PHY layer design. As an example, if an omnidirectional transmit/receive mode is required for initial cell access this would impose significant requirements on the hardware design. As another example, carrier and sampling frequency synchronization can be handled in either or both of L1 and L2. This decision can open up or break design tradeoffs in the signal processing stack.

These related research areas will complete the ecosystem around the technologies developed in this thesis. Based on future progress in these areas, mm-wave massive MIMO technology shows promise for use in mobile network deployments by the mid 2020s.

Bibliography

[1] Cisco. *Cisco Visual Networking Index: Forecast and Methodology, 2016-2021*. Tech. rep. June 2017.

[2] *Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE.

[3] *Evolved Universal Terrestrial Radio Access (E-UTRA); LTE physical layer; General description*. 3GPP, 2015.

[4] Benedict Evans. *Mobile is Eating the World*. 2013. URL: http://ben-evans.com/benedictevans/2013/5/17/mobile-is-eating-the-world (visited on 09/24/2017).

[5] Rachid El Hattachi and Javan Erfanian. *5G White Paper*. Tech. rep.

[6] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423. ISSN: 1538-7305. DOI: 10.1002/j.1538-7305.1948.tb01338.x. URL: http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[7] Robert Gallager. "Low-density parity-check codes". In: *IRE Transactions on Information Theory* 8.1 (1962), pp. 21–28.

[8] J. Salz and S. B. Weinstein. "Fourier Transform Communication System". In: *Proceedings of the First ACM Symposium on Problems in the Optimization of Data Communications Systems*. Pine Mountain, Georgia, USA: ACM, 1969, pp. 99–128.

[9] M. Zimmerman and A. Kirsch. "The AN/GSC-10 (KATHRYN) Variable Rate Data Modem for HF Radio". In: *IEEE Transactions on Communication Technology* 15.2 (1967), pp. 197–204. ISSN: 0018-9332.

[10] Andrea Goldsmith. "5G and Beyond: What Lies Ahead for Wireless System Design". In: *Proceedings of PIMRC 2014*. 2014.

[11] J. G. Andrews et al. "Are we approaching the fundamental limits of wireless network densification?" In: *IEEE Communications Magazine* 54.10 (2016), pp. 184–190.

[12] Marguerite Reardon. *T-Mobile just won the keys to supercharging its network*. 2017. URL: `https://www.cnet.com/news/t-mobile-comcast-directv-wireless-spends-8-billion-as-big-winner-of-fcc-auction-spectrum/` (visited on 09/24/2017).

[13] CORD: Central Office Rearchitected as a Datacenter and ONOS Partnership. *M-CORD: Mobile CORD: Enable 5G on CORD*. 2016. URL: `http://opencord.org/wp-content/uploads/2016/03/M-CORD-March-2016.pdf` (visited on 10/07/2017).

[14] J G Andrews et al. "What Will 5G Be?" In: *IEEE J. Sel. Areas Commun.* 32.6 (June 2014), pp. 1065–1082.

[15] J Winters. "On the Capacity of Radio Communication Systems with Diversity in a Rayleigh Fading Environment". In: *IEEE J. Sel. Areas Commun.* 5.5 (June 1987), pp. 871–878.

[16] J H Winters, J Salz, and R D Gitlin. "The impact of antenna diversity on the capacity of wireless communication systems". In: *IEEE Trans. Commun.* 42.234 (Feb. 1994), pp. 1740–1751.

[17] S Talwar, M Viberg, and A Paulraj. "Blind estimation of multiple co-channel digital signals using an antenna array". In: *IEEE Signal Process. Lett.* 1.2 (Feb. 1994), pp. 29–31.

[18] Arogyaswami J Paulraj and Thomas Kailath. "Increasing capacity in wireless broadcast systems using distributed transmission/directional reception (DTDR)". Pat. 5345599. 1994.

[19] Gerard J Foschini. "Cross-polarization canceler/equalizer". In: *US Patent* (1986).

[20] V Tarokh, N Seshadri, and A R Calderbank. "Space-time codes for high data rate wireless communication: performance criterion and code construction". In: *IEEE Trans. Inf. Theory* 44.2 (Mar. 1998), pp. 744–765.

[21] V Tarokh, H Jafarkhani, and A R Calderbank. "Space-time block codes from orthogonal designs". In: *IEEE Trans. Inf. Theory* 45.5 (July 1999), pp. 1456–1467.

[22] S M Alamouti. "A simple transmit diversity technique for wireless communications". In: *IEEE J. Sel. Areas Commun.* 16.8 (Oct. 1998), pp. 1451–1458.

[23] G J Foschini and M J Gans. "On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas". In: *Wirel. Pers. Commun.* 6.3 (Mar. 1998), pp. 311–335.

[24] I E Telatar. "Capacity of Multi-antenna Gaussian Channels". In: *European Transactions on Telecommunications*. 1999.

[25] P W Wolniansky et al. "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel". In: *Signals, Systems, and Electronics, 1998. ISSSE 98. 1998 URSI International Symposium on*. Sept. 1998, pp. 295–300.

[26] Daniel Avidor et al. "TDM-Based fixed wireless loop system". Pat. 6,961,325. 2005. URL: http://www.google.com/patents/US6961325.

[27] G. J. Foschini. "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas". In: *Bell Labs Technical Journal* 1.2 (1996), pp. 41–59.

[28] A. Goldsmith et al. "Capacity limits of MIMO channels". In: *IEEE Journal on Selected Areas in Communications* 21.5 (2003), pp. 684–702.

[29] B D Van Veen and K M Buckley. "Beamforming: a versatile approach to spatial filtering". In: *IEEE ASSP Magazine* 5.2 (Apr. 1988), pp. 4–24.

[30] T L Marzetta. "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas". In: *IEEE Trans. Wireless Commun.* 9.11 (Nov. 2010), pp. 3590–3600.

[31] Y. H. Nam et al. "Full-dimension MIMO (FD-MIMO) for next generation cellular technology". In: *IEEE Communications Magazine* 51.6 (2013), pp. 172–179.

[32] Y. Kim et al. "Full dimension mimo (FD-MIMO): the next evolution of MIMO in LTE systems". In: *IEEE Wireless Communications* 21.2 (2014), pp. 26–33.

[33] H. Ji et al. "Overview of Full-Dimension MIMO in LTE-Advanced Pro". In: *IEEE Communications Magazine* 55.2 (2017), pp. 176–184.

[34] Blue Danube. *BeamCraft — Blue Danube*. URL: http://www.bluedanube.com/products/ (visited on 10/07/2017).

[35] Nokia. *Nokia 5G Journey*. 2017. URL: https://www.nttdocomo.co.jp/binary/pdf/corporate/technology/rd/tech/5g/5GTBS2017_TECH_WORKSHOP_NOKIA.pdf (visited on 10/07/2017).

[36] Alok Shah. *Samsung and Sprint Conduct Real-World Massive MIMO Testing at Mobile World Congress Fall 2017*. 2017. URL: https://insights.samsung.com/2017/09/11/samsung-and-sprint-conduct-real-world-massive-mimo-testing-at-mobile-world-congress-fall-2017/ (visited on 10/07/2017).

[37] Monica Alleven. *FCC OKs sweeping Spectrum Frontiers rules to open up nearly 11 GHz of spectrum*. 2016. URL: http://www.fiercewireless.com/tech/fcc-oks-sweeping-spectrum-frontiers-rules-to-open-up-nearly-11-ghz-spectrum (visited on 09/24/2017).

[38] *Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE.

[39] T S Rappaport et al. "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!" In: *IEEE Access* 1 (2013), pp. 335–349.

[40] T S Rappaport et al. "Wideband Millimeter-Wave Propagation Measurements and Channel Models for Future Wireless Communication System Design". In: *IEEE Trans. Commun.* 63.9 (2015), pp. 3029–3056.

[41] M R Akdeniz et al. "Millimeter Wave Channel Modeling and Cellular Capacity Evaluation". In: *IEEE J. Sel. Areas Commun.* 32.6 (June 2014), pp. 1164–1179.

[42] S Deng, M K Samimi, and T S Rappaport. "28 GHz and 73 GHz millimeter-wave indoor propagation measurements and path loss models". In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. June 2015, pp. 1244–1250.

[43] H. T. Kim et al. "A 28GHz CMOS direct conversion transceiver with packaged antenna arrays for 5G cellular system". In: *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. 2017, pp. 69–72.

[44] K. Kibaroglu, M. Sayginer, and G. M. Rebeiz. "An ultra low-cost 32-element 28 GHz phased-array transceiver with 41 dBm EIRP and 1.0-1.6 Gbps 16-QAM link at 300 meters". In: *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. 2017, pp. 73–76.

[45] U. Kodak and G. M. Rebeiz. "Bi-directional flip-chip 28 GHz phased-array core-chip in 45nm CMOS SOI for high-efficiency high-linearity 5G systems". In: *2017 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*. 2017, pp. 61–64.

[46] B. Sadhu et al. "7.2 A 28GHz 32-element phased-array transceiver IC with concurrent dual polarized beams and 1.4 degree beam-steering resolution for 5G communication". In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 2017, pp. 128–129.

[47] Mike Dano. *Editor's CornerThe economics of fixed wireless, from LTE to 5G, and what it means for Verizon*. 2017. URL: `http://www.fiercewireless.com/5g/economics-fixed-wireless-from-lte-to-5g-and-what-it-means-for-verizon` (visited on 09/24/2017).

[48] Qualcomm. *RF Products*. 2017. URL: `https://www.qualcomm.com/products/rf` (visited on 09/24/2017).

[49] Broadcom. *BCM4358: 5G WiFi 802.11ac Client*. 2017. URL: `https://www.broadcom.com/products/wireless/wireless-lan-bluetooth/bcm4358#overview` (visited on 09/24/2017).

[50] Mads Barnkob. *Ericsson Radio Base Station RBS6000 teardown*. 2017. URL: `http://kaizerpowerelectronics.dk/teardown/ericsson-radio-base-station-rbs6000-teardown/` (visited on 09/24/2017).

[51] GreenTouch Consortium. 2011. URL: `http://www.greentouch.org/index.php?page=members-collaborate-on-dramatic-new-antenna-system`.

[52] Hajime Suzuki et al. "Highly spectrally efficient Ngara Rural Wireless Broadband Access Demonstrator". In: *2012 International Symposium on Communications and Information Technologies (ISCIT)*. 2012.

[53] Clayton Shepard et al. "Argos: Practical Many-antenna Base Stations". In: *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*. Mobicom '12. New York, NY, USA: ACM, 2012, pp. 53–64.

[54] Clayton Shepard, Hang Yu, and Lin Zhong. "ArgosV2: A Flexible Many-antenna Research Platform". In: *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*. MobiCom '13. New York, NY, USA: ACM, 2013, pp. 163–166.

[55] J Vieira et al. "A flexible 100-antenna testbed for Massive MIMO". In: *2014 IEEE Globecom Workshops (GC Wkshps)*. Dec. 2014, pp. 287–293.

[56] *5G Massive MIMO Testbed: From Theory to Reality - National Instruments*. `http://www.ni.com/white-paper/52382/en/`. Accessed: 2016-4-28.

[57] *TitanMIMO-6: Sub 6 GHz Massive MIMO Testbed*. Nutaq Corp.

[58] *Introducing Facebook's new terrestrial connectivity systems — Terragraph and Project ARIES*. `https://code.facebook.com/posts/1072680049445290/introducing-facebook-s-new-terrestrial-connectivity-systems-terragraph-and-project-aries/`. Accessed: 2016-4-28.

[59] G. Xu et al. "Full Dimension MIMO (FD-MIMO): Demonstrating Commercial Feasibility". In: *IEEE Journal on Selected Areas in Communications* 35.8 (2017), pp. 1876–1886.

[60] A Valdes-Garcia et al. "A Fully Integrated 16-Element Phased-Array Transmitter in SiGe BiCMOS for 60-GHz Communications". In: *IEEE J. Solid-State Circuits* 45.12 (Dec. 2010), pp. 2757–2773.

[61] S Emami et al. "A 60GHz CMOS phased-array transceiver pair for multi-Gb/s wireless communications". In: *2011 IEEE International Solid-State Circuits Conference*. Feb. 2011, pp. 164–166.

[62] A Natarajan et al. "A Fully-Integrated 16-Element Phased-Array Receiver in SiGe BiCMOS for 60-GHz Communications". In: *IEEE J. Solid-State Circuits* 46.5 (May 2011), pp. 1059–1075.

[63] K Okada et al. "A 60-GHz 16QAM/8PSK/QPSK/BPSK Direct-Conversion Transceiver for IEEE802.15.3c". In: *IEEE J. Solid-State Circuits* 46.12 (Dec. 2011), pp. 2988–3004.

[64] N Saito et al. "A Fully Integrated 60-GHz CMOS Transceiver Chipset Based on WiGig/IEEE 802.11ad With Built-In Self Calibration for Mobile Usage". In: *IEEE J. Solid-State Circuits* 48.12 (Dec. 2013), pp. 3146–3159.

[65] M Boers et al. "A 16TX/16RX 60 GHz 802.11ad Chipset With Single Coaxial Interface and Polarization Diversity". In: *IEEE J. Solid-State Circuits* 49.12 (Dec. 2014), pp. 3031–3045.

[66] The Pacific War Online Encyclopedia. *Mark 8 Fire Control Radar*. URL: `http://www.pwencycl.kgbudge.com/M/a/Mark_8_fire_control_radar.htm` (visited on 10/07/2017).

[67] Alan J Fenn et al. "The development of phased-array radar technology". In: *Lincoln Laboratory Journal* 12.2 (2000), pp. 321–340.

[68] White Sands Missile Range. *Multi-function Array Radar I (MAR-I)*. URL: `http://www.wsmr-history.org/MAR-I.htm` (visited on 10/07/2017).

[69] Glenn W Meurer Jr. "The TRADEX Multitarget Tracker". In: *Lincoln Laboratory Journal* 5 (1992), pp. 317–350.

[70] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals &Amp; Systems (2Nd Ed.)* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN: 0-13-814757-4.

[71] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[72] Q. H. Spencer et al. "An introduction to the multi-user MIMO downlink". In: *IEEE Communications Magazine* 42.10 (2004), pp. 60–67.

[73] D. Gesbert et al. "Shifting the MIMO Paradigm". In: *IEEE Signal Processing Magazine* 24.5 (2007), pp. 36–46.

[74] William C Jakes. *Microwave mobile communications*. Wiley-IEEE Press, 1974.

[75] T. K. Y. Lo. "Maximum ratio transmission". In: *IEEE Transactions on Communications* 47.10 (1999), pp. 1458–1461.

[76] T. Haustein et al. "Performance of MIMO systems with channel inversion". In: *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367)*. Vol. 1. 2002, 35–39 vol.1.

[77] M. Joham, W. Utschick, and J. A. Nossek. "Linear transmit processing in MIMO communications systems". In: *IEEE Transactions on Signal Processing* 53.8 (2005), pp. 2700–2712.

[78] U. Fincke and M. Pohst. "Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis". In: *Mathematics of Computation* 44.170 (1985), pp. 463–471.

[79] Zhan Guo and P. Nilsson. "Algorithm and implementation of the K-best sphere decoding for MIMO detection". In: *IEEE Journal on Selected Areas in Communications* 24.3 (2006), pp. 491–503.

[80] B. Hassibi and H. Vikalo. "On the sphere-decoding algorithm I. Expected complexity". In: *IEEE Transactions on Signal Processing* 53.8 (2005), pp. 2806–2818.

[81] M. K. Varanasi and T. Guess. "Optimum decision feedback multiuser equalization with successive decoding achieves the total capacity of the Gaussian multiple-access channel". In: *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No.97CB36136)*. Vol. 2. 1997, 1405–1409 vol.2.

[82] G. Ginis and J. M. Cioffi. "On the relation between V-BLAST and the GDFE". In: *IEEE Communications Letters* 5.9 (2001), pp. 364–366.

[83] M. Costa. "Writing on dirty paper (Corresp.)" In: *IEEE Transactions on Information Theory* 29.3 (1983), pp. 439–441.

[84] M. Tomlinson. "New automatic equaliser employing modulo arithmetic". In: *Electronics Letters* 7.5 (1971), pp. 138–139.

[85] H. Harashima and H. Miyakawa. "Matched-Transmission Technique for Channels With Intersymbol Interference". In: *IEEE Transactions on Communications* 20.4 (1972), pp. 774–780.

[86] G. Ginis and J. M. Cioffi. "A multi-user precoding scheme achieving crosstalk cancellation with application to DSL systems". In: *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat. No.00CH37154)*. Vol. 2. 2000, 1627–1631 vol.2.

[87] G. Caire and S. Shamai. "On the achievable throughput of a multiantenna Gaussian broadcast channel". In: *IEEE Transactions on Information Theory* 49.7 (2003), pp. 1691–1706.

[88] R. Zamir, S. Shamai, and U. Erez. "Nested linear/lattice codes for structured multiterminal binning". In: *IEEE Transactions on Information Theory* 48.6 (2002), pp. 1250–1276.

[89] B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst. "A vector-perturbation technique for near-capacity multiantenna multiuser communication-part II: perturbation". In: *IEEE Transactions on Communications* 53.3 (2005), pp. 537–544.

[90] G. L. Stuber et al. "Broadband MIMO-OFDM wireless communications". In: *Proceedings of the IEEE* 92.2 (2004), pp. 271–294.

[91] IEEE Std. 802.11 03/940r4. *TGn Channel Models*.

[92] Hao Xu et al. "A generalized space-time multiple-input multiple-output (MIMO) channel model". In: *IEEE Transactions on Wireless Communications* 3.3 (2004), pp. 966–975.

[93] Ernst Bonek. "MIMO Propagation and Channel Modeling". In: *MIMO: From Theory to Implementation*. Ed. by A. Sibille, C. Oestges, and A. Zanella. Elsevier Science, 2010, pp. 27–54.

[94] A. F. Molisch et al. "The COST259 Directional Channel Model-Part I: Overview and Methodology". In: *IEEE Transactions on Wireless Communications* 5.12 (2006), pp. 3421–3433.

[95] L. Schumacher, K. I. Pedersen, and P. E. Mogensen. "From antenna spacings to theoretical capacities - guidelines for simulating MIMO systems". In: *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications.* Vol. 2. 2002, 587–592 vol.2.

[96] S. Hur et al. "Proposal on Millimeter-Wave Channel Modeling for 5G Cellular System". In: *IEEE Journal of Selected Topics in Signal Processing* 10.3 (2016), pp. 454–469.

[97] P. K. Bailleul. "A New Era in Elemental Digital Beamforming for Spaceborne Communications Phased Arrays". In: *Proceedings of the IEEE* 104.3 (2016), pp. 623–632.

[98] C. Fulton et al. "Digital Phased Arrays: Challenges and Opportunities". In: *Proceedings of the IEEE* 104.3 (2016), pp. 487–503.

[99] S. H. Talisa et al. "Benefits of Digital Phased Array Radars". In: *Proceedings of the IEEE* 104.3 (2016), pp. 530–543.

[100] J. S. Herd and M. D. Conway. "The Evolution to Modern Phased Array Architectures". In: *Proceedings of the IEEE* 104.3 (2016), pp. 519–529.

[101] Xiang Gao et al. "Measured propagation characteristics for very-large MIMO at 2.6 GHz". In: *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR).* 2012.

[102] Jakob Hoydis et al. "Channel measurements for large antenna arrays". In: *2012 International Symposium on Wireless Communication Systems (ISWCS).* 2012.

[103] Sohail Payami and Fredrik Tufvesson. "Channel Measurements and Analysis for Very Large Array Systems At 2.6 GHz". In: *Antennas and Propagation (EUCAP), 2012 6th European Conference on.* 2012.

[104] Alex Oliveras Martinez, Elisabeth De Carvalho, and Jesper Odum Nielsen. "Towards very large aperture massive MIMO: A measurement based study". In: *2014 IEEE Globecom Workshops (GC Wkshps).* 2014.

[105] X Gao et al. "Massive MIMO Performance Evaluation Based on Measured Propagation Data". In: *IEEE Trans. Wireless Commun.* 14.7 (July 2015), pp. 3899–3911.

[106] P. Harris et al. "Performance Characterization of a Real-Time Massive MIMO System With LOS Mobile Channels". In: *IEEE Journal on Selected Areas in Communications* 35.6 (2017), pp. 1244–1253.

[107] A. Puglielli et al. "A scalable massive MIMO array architecture based on common modules". In: *2015 IEEE International Conference on Communication Workshop (ICCW).* 2015, pp. 1310–1315.

[108] A. Puglielli et al. "Design of Energy- and Cost-Efficient Massive MIMO Arrays". In: *Proceedings of the IEEE* 104.3 (2016), pp. 586–606.

[109] K. Takinami et al. "A 60GHz wireless transceiver employing hybrid analog/digital beamforming with interference suppression for multiuser gigabit/s radio access". In: *2015 Symposium on VLSI Circuits (VLSI Circuits)*. 2015, pp. C306–C307.

[110] Y. S. Yeh et al. "A 28-GHz Phased-Array Receiver Front End With Dual-Vector Distributed Beamforming". In: *IEEE Journal of Solid-State Circuits* 52.5 (2017), pp. 1230–1244.

[111] L. Liang, W. Xu, and X. Dong. "Low-Complexity Hybrid Precoding in Massive Multiuser MIMO Systems". In: *IEEE Wireless Communications Letters* 3.6 (2014), pp. 653–656.

[112] R.A. Monzingo, R.L. Haupt, and T.W. Miller. *Introduction to Adaptive Arrays*. Electromagnetics and Radar. Institution of Engineering and Technology, 2011. ISBN: 9781891121579.

[113] W. Liu and S. Weiss. *Wideband Beamforming: Concepts and Techniques*. Wireless Communications and Mobile Computing. Wiley, 2010. ISBN: 9780470661185.

[114] M. Emami et al. "Matched filtering with rate back-off for low complexity communications in very large delay spread channels". In: *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.* Vol. 1. 2004, 218–222 Vol.1.

[115] R Daniels and R Heath. "Improving on time reversal with MISO precoding". In: *Proceedings of the Eighth International Symposium on Wireless Personal Communications Conference*. 2005, pp. 18–22.

[116] B. Wang et al. "Green Wireless Communications: A Time-Reversal Paradigm". In: *IEEE Journal on Selected Areas in Communications* 29.8 (2011), pp. 1698–1710.

[117] H Yang and T L Marzetta. "Performance of Conjugate and Zero-Forcing Beamforming in Large-Scale Antenna Systems". In: *IEEE J. Sel. Areas Commun.* 31.2 (Feb. 2013), pp. 172–179.

[118] Y. Han et al. "Time-Reversal Massive Multipath Effect: A Single-Antenna Massive MIMO Solution". In: *IEEE Transactions on Communications* 64.8 (2016), pp. 3382–3394.

[119] A Pitarokoilis, S K Mohammed, and E G Larsson. "Uplink Performance of Time-Reversal MRC in Massive MIMO Systems Subject to Phase Noise". In: *IEEE Transactions on Wireless Communications* 14.2 (Feb. 2015), pp. 711–723.

[120] R Krishnan et al. "Algorithms for Joint Phase Estimation and Decoding for MIMO Systems in the Presence of Phase Noise and Quasi-Static Fading Channels". In: *IEEE Transactions on Signal Processing* 63.13 (July 2015), pp. 3360–3375.

[121] M R Khanzadi, G Durisi, and T Eriksson. "Capacity of SIMO and MISO Phase-Noise Channels With Common/Separate Oscillators". In: *IEEE Transactions on Communications* 63.9 (Sept. 2015), pp. 3218–3231.

[122] Rajet Krishnan et al. "Linear massive MIMO precoders in the presence of phase noise—A large-scale analysis". In: *IEEE Transactions on Vehicular Technology* 65.5 (May 2016), pp. 3057–3071.

[123] D Murphy, J J Rael, and A A Abidi. "Phase Noise in LC Oscillators: A Phasor-Based Analysis of a General Result and of Loaded Q". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 57.6 (June 2010), pp. 1187–1203.

[124] A Hajimiri and T H Lee. "Design issues in CMOS differential LC oscillators". In: *IEEE Journal of Solid-State Circuits* 34.5 (May 1999), pp. 717–724.

[125] Jesper Bank. "A Harmonic-Oscillator Design Methodology Based on Describing Functions". PhD thesis. Chalmers University of Technology, 2006.

[126] T Höhne and V Ranki. "Phase Noise in Beamforming". In: *IEEE Transactions on Wireless Communications* 9.12 (Dec. 2010), pp. 3682–3689.

[127] A. Puglielli et al. "Phase noise scaling and tracking in OFDM multi-user beamforming arrays". In: *2016 IEEE International Conference on Communications (ICC)*. May 2016, pp. 1–6.

[128] S. Wu and Y. Bar-Ness. "A Phase Noise Suppression Algorithm for OFDM-Based WLANs". In: *IEEE Communications Letters* 6.12 (Dec. 2002), pp. 535–537.

[129] S. Wu and Y. Bar-Ness. "OFDM Systems in the Presence of Phase Noise: Consequences and Solutions". In: *IEEE Transactions on Communications* 52.11 (Nov. 2004), pp. 1988–1996.

[130] D. Petrovic, W. Rave, and G. Fettweis. "Effects of Phase Noise on OFDM Systems With and Without PLL: Characterization and Compensation". In: *IEEE Transactions on Communications* 55.8 (Aug. 2007), pp. 1607–1616.

[131] P. K. Hanumolu et al. "Analysis of Charge-Pump Phase-Locked Loops". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 51.9 (Sept. 2004), pp. 1665–1674.

[132] A. Demir, A. Mehrotra, and J. Roychowdhury. "Phase Noise in Oscillators: A Unifying Theory and Numerical Methods for Characterization". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 47 (May 2000), pp. 655–674.

[133] A. Mehrotra. "Noise Analysis of Phase-Locked Loops". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 49.9 (Sept. 2002), pp. 1309–1316.

[134] H. Meyr, M. Moeneclaey, and S. Fechtel. *Digital Communication Receivers, Volume 2: Synchronization, Channel Estimation, and Signal Processing*. Wiley, 1998.

[135] U. Mengali and A. D'Andrea. *Synchronization Techniques for Digital Receivers*. Springer, 1997.

[136] P. Robertson and S. Kaiser. "Analysis of the Effects of Phase Noise in Orthogonal Frequency Division Multiplex (OFDM) Systems". In: *Proc. 1995 IEEE Int. Conf. Communications (ICC)*. (Seattle, WA, USA). June 1995.

[137] Greg LaCaille. "TBD". PhD thesis. UC Berkeley, 2018.

[138] R Kimura et al. "Golay sequence aided channel estimation for millimeter-wave WPAN systems". In: *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*. 2008, pp. 1–5.

[139] C. W. Farrow. "A continuously variable digital delay element". In: *1988., IEEE International Symposium on Circuits and Systems*. 1988, 2641–2645 vol.3.

[140] L. Erup, F. M. Gardner, and R. A. Harris. "Interpolation in digital modems. II. Implementation and performance". In: *IEEE Transactions on Communications* 41.6 (1993), pp. 998–1008.

[141] M. Oerder and H. Meyr. "Digital filter and square timing recovery". In: *IEEE Transactions on Communications* 36.5 (1988), pp. 605–612.

[142] F. M. Gardner. "Interpolation in digital modems. I. Fundamentals". In: *IEEE Transactions on Communications* 41.3 (1993), pp. 501–507.

[143] M. Speth et al. "Optimum receiver design for wireless broad-band systems using OFDM. I". In: *IEEE Transactions on Communications* 47.11 (1999), pp. 1668–1677.

[144] M. Speth et al. "Optimum receiver design for OFDM-based broadband transmission .II. A case study". In: *IEEE Transactions on Communications* 49.4 (2001), pp. 571–578.

[145] K. Mueller and M. Muller. "Timing Recovery in Digital Synchronous Data Receivers". In: *IEEE Transactions on Communications* 24.5 (1976), pp. 516–531.

[146] M. Bellanger, J. Daguet, and G. Lepagnol. "Interpolation, extrapolation, and reduction of computation speed in digital filters". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 22.4 (1974), pp. 231–235.

[147] R. Crochiere and L. Rabiner. "Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.5 (1975), pp. 444–456.

[148] R. Crochiere and L. Rabiner. "Further considerations in the design of decimators and interpolators". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.4 (1976), pp. 296–311.

[149] R. Shively. "On multistage finite impulse response (FIR)filters with decimation". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.4 (1975), pp. 353–357.

[150] L. Rabiner and R. Crochiere. "A novel implementation for narrow-band FIR digital filters". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.5 (1975), pp. 457–464.

[151] Clayton Shepard, Abeer Javed, and Lin Zhong. "Control Channel Design for Many-Antenna MU-MIMO". In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 578–591.

[152] Zhipeng Zhao. "IEEE 802.11n Implementation". In: *MIMO: From Theory to Implementation*. Ed. by A. Sibille, C. Oestges, and A. Zanella. Elsevier Science, 2010, pp. 27–54.