REINDEER

D3.1

Analytical Performance Metrics and Physical-Layer Solutions

Project number:	101013425
Project acronym:	**REINDEER**
Project title:	REsilient INteractive applications through hyper Diversity in Energy Efficient RadioWeaves technology
Project Start Date:	1st January, 2021
Duration:	42 months
Programme:	H2020-ICT-52-2020
Deliverable Type:	Report, Prototype, Demonstrator, Other
Reference Number:	ICT-52-2020 / D3.1 / 1.08
Reference Number:	ICT-52-2020 / D3.1 / 1.08
Workpackage:	WP 03
Due Date:	31st December, 2021 (M12)
Actual Submission Date:	26th January, 2022
Responsible Organisation:	Linköping university
Editor:	LiU
Dissemination Level:	PU
Revision:	1.08
Abstract:	This deliverable develops a framework for physical-layer performance evaluation of RadioWeaves. Specifically, an effective-SINR abstraction model is developed and shown to be accurate and highly efficient in terms of computational complexity. Furthermore, this document evaluates complexity of existing algorithms in terms of computational load and front-haul signaling for different cell-free topologies considered in the literature. It also proposes efficient algorithms for distributed MIMO processing of RadioWeaves with centralized and decentralized approaches.
Keywords:	Physical-layer, performance evaluation, effective SINR, cell-free operation, distributed processing.

The REINDEER project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101013425.

Editor

LiU

Contributors

Baktash Behmanesh (ULUND), Benjamin Deutschmann (TU GRAZ), Ove Edfors (ULUND), Pål Frenger (EAB), Unnikrishnan Kunnath Ganesan (LiU), Ke Wang Helmersson (EAB), Erik G. Larsson (LiU), Liang Liu (ULUND), Sofie Pollin (KUL), Vida Ranjbar (KUL), Sarvendranath Rimalapudi (LiU), Juan Francisco Esteban Rivas (TEC), Zakir Hussain Shaik (LiU), Joao Viera (EAB), Thomas Wilding (TU GRAZ), Klaus Witrisal (TU GRAZ).[1]

Disclaimer

The information in this document is provided as is, and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author's view – the European Commission is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.
This document has gone through the consortium's internal review process and is still subject to the review of the European Commission. Updates to the content may be made at a later stage.

[1] The contributors names are ordered alphabetically based on last name.

Executive Summary

This is deliverable 3.1 “Analytical performance metrics and physical-layer solutions”. This document provides a framework for physical-layer performance evaluation of RadioWeaves. It evaluates complexity of existing centralized vs decentralized algorithms and proposes new algorithms.

Chapter 1 describes the main objective and structure of this deliverable. It introduces the RadioWeaves technology and explains the advantages of cell-free architectures. It also describes different cell-free architectures available.

Chapter 2 develops the effective-SINR based physical layer abstraction model that serves as performance evaluation framework for the RadioWeaves. It describes the system model, channel models, mobility models, and antenna array configurations supported by the evaluation framework. Furthermore, it presents the channel estimation techniques considered and the design of precoders and combiners based on the channel estimates obtained. It then explains the SINR generation process and develops a mapping between effective SINRs and link performance, based on the exponential effective SINR mapping, a well-established tool for physical-layer abstraction of wireless link performance.

Chapter 3 summarizes different fronthaul topologies in a cell-free network. It studies different uplink data detection/downlink precoding algorithms for the topologies considered. Given the fronthaul topology, fronthaul signaling and computational complexity of centralized and distributed algorithms are compared. The trade-off between the number of connections to the central processing unit and computational complexity for better spectral efficiency is studied for centralized algorithms. Furthermore, the trade-off between the delay and the number of connections to the central processing unit is studied for distributed algorithms.

Chapter 4 provides an initial discussion of the joint scheduling of communication resources to users and the corresponding computational resources required in the RadioWeaves infrastructure to execute the distributed algorithms delivering services. This discussion is to be seen as a background to take into account when further refining development and selection of algorithms suitable for energy efficient execution, delivering reliable services.

Chapter 5 discuss efficient algorithm candidates for allocation of compute and fronthaul resources. A Kalman filter implementation for uplink processing is described as well as methods to reduce the complexity by means of selecting a sub-set of access points for processing the signals related to each of the users. Furthermore, pros and cons of centralized versus distributed processing are discussed with special focus on calculation of antenna precoding and combining weights and other implementation requirements.

Chapter 6 summarizes the main conclusions of this deliverable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objective of the document

The main aim of this document is to develop a physical-layer performance evaluation framework for RadioWeaves, evaluate existing centralized and decentralized algorithms (decoding and precoding), and propose new algorithms. It first briefly introduces the RadioWeaves technology and discusses the benefits of cell-free architectures. It also compares RadioWeaves to existing cell-free architectures, such as Radio Stripes. It then develops the performance evaluation framework, which serves as an abstraction model for physical layer. Specifically, an effective-SINR based abstraction model is adapted, which is shown to be accurate and computationally efficient. Owing to its simplicity and accuracy, the effective-SINR abstraction is a standard tool to evaluate wireless standards, and is used for example for LTE [5, 6], WiMAX [7] and 5G-NR [8]. The document also describes different options supported by the current version of the framework. The framework described is an initial version and will be updated at later stages of the project. This document also evaluates the computational and front-haul signaling complexity of several existing algorithms for different cell-free topologies. Finally, it proposes new algorithms for efficient processing of RadioWeaves with centralized and decentralized approaches.

1.2 Structure of the document

The main structure of the document is as follows:

- Chapter 1 introduces the RadioWeaves and discuss different cell-free architectures.
- Chapter 2 develops the physical layer performance evaluation framework.
- Chapter 3 evaluates computational and front-haul signaling complexity of the existing algorithms.
- Chapter 4 provides and overview of energy measures and scheduling of users and computational resources.
- Chapter 5 proposes new algorithms for efficient distributed processing of RadioWeaves.
- Chapter 6 summarizes the main conclusions of the deliverable.

1.3 RadioWeaves

RadioWeaves is an emerging wireless access infrastructure, in which a fabric of distributed radio devices and computing resources serve as a massive distributed antenna array. The technology builds upon the foundations of massive MIMO and combines the advantages of distributed cell-free architectures and large intelligent surfaces to achieve superior coverage and low power consumption.

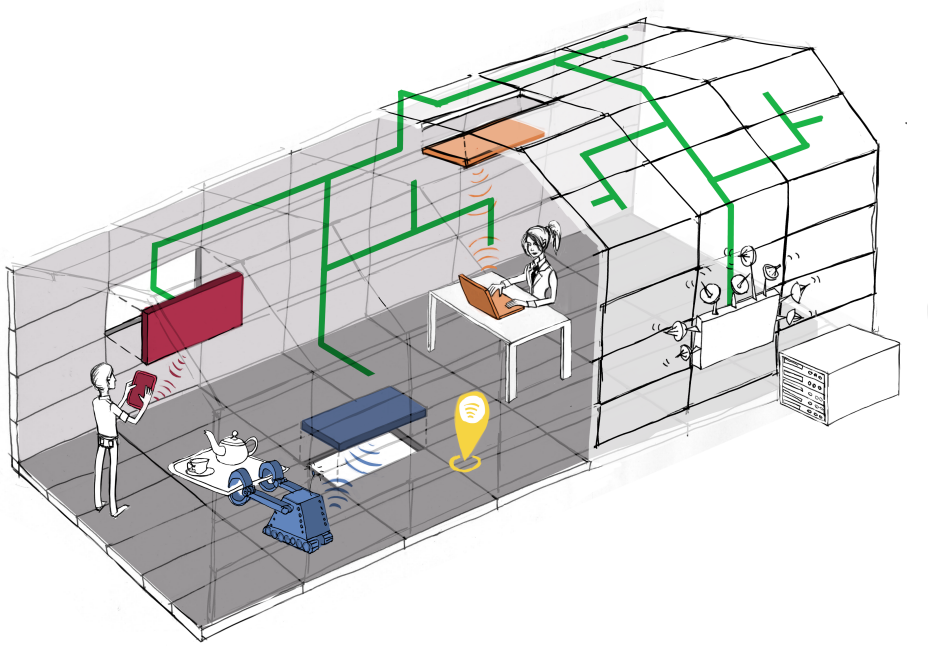


Figure 1.1: A pictorial representation of RadioWeaves deployment.

RadioWeaves technology is the key to enabling new, innovative use cases ranging from robotized factories, warehouses, and logistics to immersive entertainment for crowds of people to care environments, hospitals, and assisted living to smart homes. It is foreseen to be a fundamental enabler of future 6G networks and beyond, which will offer consistent service and scalable network capacity at unprecedented energy efficiency. RadioWeaves also provides a connectivity platform for future applications that require interaction between real and virtual worlds.

1.3.1 Many benefits with RadioWeaves

The RadioWeaves distributed infrastructure is envisioned to achieve many orders of magnitude improvements in quality-of-service and energy efficiency, compared to a conventional collocated MIMO system. There are many benefits with deploying RadioWeaves technology. In terms of spatial processing, RadioWeaves inherits the fundamental advantages of cellular massive MIMO: operation in TDD, reliance on uplink pilots for all channel estimation tasks, and fully digital RF processing per antenna. Beyond this, there are many other benefits as detailed below.

Increased Spatial Diversity

The RadioWeaves topology is designed to maximize the probability of a terminal being close to at least a handful of antennas and in a non/shadowed and favorite direction with respect to the

arrays. This yields a superior degree of macro-diversity against signal blockage, and provides favorable propagation conditions. Moreover, the angular directions from a terminal to the service antennas span a broad range. The phase-aligned transmission towards the terminal results in signal power strongly concentrated in a small region around the user, creating favorable propagation conditions for the transmission of multiple data streams. Thus, RadioWeaves can improve coverage, reduce power consumption and increase system capacity.

Ultra Robust Operation

URLLC is an envisioned use case for future wireless networks with real-time systems requiring imperceptible latency at the application level with high reliability. Implementing a connect-compute fabric that can run applications near the terminals can meet the latency requirements. Also, the dispersed wireless access structure and the directed antenna arrays can ensure more robust coverage to time-critical applications.

Energy and Bandwidth Efficiency

The proximity of access points and extreme multiplexing capabilities can reduce the power consumption of user terminals at the expense of local compute engines. Also, computing locally in RadioWeaves improves the bandwidth efficiency whereas the central cloud computing networks run into bandwidth bottlenecks. The distributed architecture in RadioWeaves with phase-aligned transmission helps in generating a strong energy field in the favorable field of view of certain devices thereby exciting them to generate energy. This helps in wireless power transfer to devices and can bring down global energy consumption. In combination, RadioWeaves can reduce power consumption by larger orders of magnitude.

1.4 Advantages of The Cell-Free Approach

In a cellular network, where a service point (SP) serves multiple user equipments (UEs) the received signal power decays rapidly with the propagation distance. The UEs closer to the BS, i.e., cell-center, experience a higher signal-to-noise ratio (SNR) than those present at the cell edge. Moreover, neighboring cell base stations affect cell-edge users, making the signal-to-interference-plus-noise ratio (SINR) low at the cell-edge locations. The data rate is an increasing function of the SINR; thus, there can be significant variations in data rates in each cell. This problem persists even with massive MIMO. Towards this, cell-free massive MIMO is projected to be a potential solution to tackle this issue and provide uniform data rates everywhere. A cell-free network comprises multiple single/multi-antenna SPs distributed over a geographical area and serving multiple single-antenna UEs simultaneously [9, 10]. The SPs may be interconnected in different topologies, see below.

1.5 Comparison of Cell-Free Architectures

There are two different aspects of the network topology: first, the topology of the SP interconnections, and second, the way the signal processing is organized.

Starting with the interconnect topology, the options are sequential (daisy-chain network), centralized (star-like network), tree, and fully connected (mesh network) as shown in Fig.3.1. The choice

of a particular topology depends on the application of interest. One possible topology for cell-free networks is radio stripes, which utilize the sequential topology [3, 11]. This architecture is suitable for deployments in dense areas such as sports arenas and railway stations with many SPs and UEs per km^2, and large construction elements that the stripes can be attached to. This network comprises sequentially connected SPs in a daisy-chain topology and shares the same cable for fronthaul and power supply [12]. One of the benefits of a sequential network are: (i) ease of deployment and cable routing in practical applications such as railway stations, museums, factories, etc., and (ii) Another possible topology is a tree or a mesh that results if at least one SP or the edge processing unit (EPU) has more than one transmit or receive link [13]. This can add robustness and provide for lower delays but also adds complexity in the implementation and in the timing synchronization. Also the design of fault-tolerant routing mechanisms may be challenging depending on the topology. Moreover, the sequential implementation of cell-free massive MIMO has the potential to deliver the benefits of massive MIMO with much lower fronthaul requirements when compared to centralized.

In terms of the signal processing, the methods found in the literature can be broadly categorized into two types: fully centralized and fully distributed implementation [14–20]. In the former category, all the required processing is done at the EPU [1, 9, 15], while in the latter category, all the processing is done locally at the SPs [3, 4, 13, 14], except for the final fusion at the EPU, using statistical channel state information (CSI). A centralized implementation has superior performance because of its access to complete information but, the downside is the requirement of a very large fronthaul compared to other topologies such as sequential processing. There are a few other methods that do not strictly fall into either of the above categories. For example, the method proposed in [14] is partially decentralizing the process, i.e., the SPs partially process the received signals and forward the Gramian of the channel matrix to the EPU, where most of the signal processing and estimation of the signal is done. The most of the works cited above focused on developing algorithms to decentralize linear methods such as maximum ratio combining (MRC), zero forcing (ZF), and linear minimum mean square error (LMMSE). The optimal centralized MMSE solution for decoding can be computed in a distributed manner, with a fronthaul load that is independent of the number of access points. The resulting distributed algorithm, first presented in [12], can be interpreted as a Kalman filter (sequential MMSE estimator) and is further refined in Chapter 5.

Chapter 2

Framework for Physical-Layer Performance Evaluation of RadioWeaves

In this chapter, we describe the physical-layer abstraction model adopted to evaluate the performance of the RadioWeaves system. This model is based on an effective signal-to-interference-plus-noise ratio (SINR) computation, which will be described later in this chapter. This effective-SINR based physical-layer abstraction model helps to avoid computationally expensive physical-layer link-level simulations. The framework accurately predicts the link performance in a computationally efficient way, and is routinely used for system-level simulation of many standards such as LTE [5, 6], WiMAX [7] and 5G-NR [8].

Outline: A high-level description of the evaluation framework is introduced in Section 2.1. Section 2.2 introduces the physical layer model used in developing the framework. It describes different aspects such as system model, channel models, and antenna array deployment. Section 2.3 describes the SINR generation block, the first block of the framework. The performance mapping block, which is the second block of the framework, is described in Section 2.4. Section 2.5 analyzes the downlink power requirements of RadioWeaves and shows how uniform-quality-of-service is possible with very lower power. Section 2.6 summarizes the chapter.

2.1 Effective SINR Based Evaluation Framework

We now provide a high level description of the performance evaluation framework that is made up of two blocks, namely SINR generation block and performance mapping block.

1. *SINR Generation Block:* The first block calculates a vector of SINRs per user per narrowband channel use. Elements of the SINR vector corresponds to the SINR of symbols that represent the coefficients of the basis functions of the time-frequency space of the transmitted signal [21]. For example, orthogonal-frequency-division-multiplexing (OFDM) symbol, in which case the symbols correspond to the subcarriers.
2. *Performance Mapping Block:* The second block takes the vector of SINRs and predicts the instantaneous packet error rate (PER) based on the modulation and coding scheme (MCS) used for the transmission.

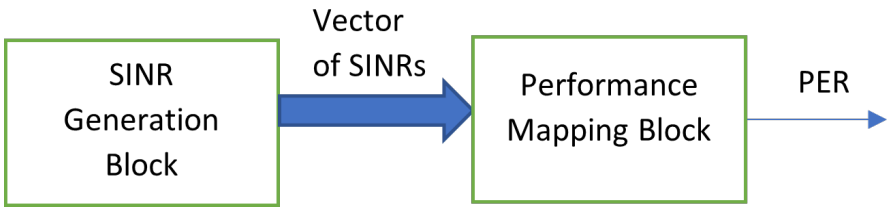


Figure 2.1: Performance Evaluation Framework of RadioWeaves

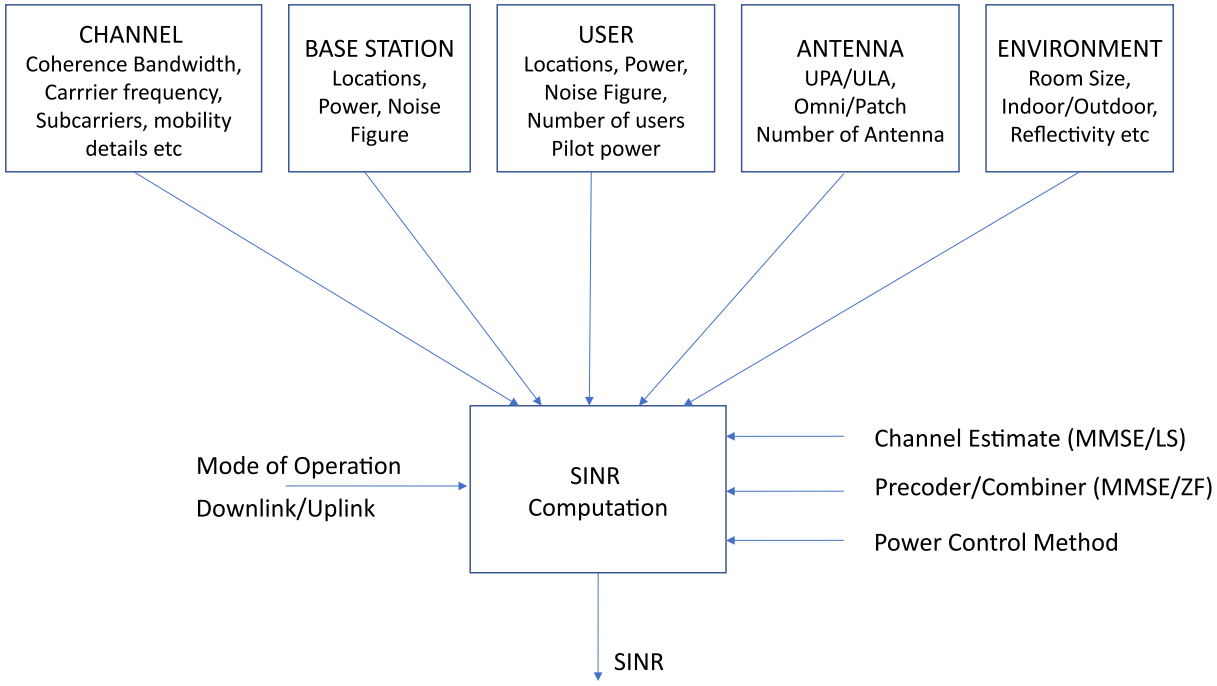


Figure 2.2: SINR Generation Block

2.1.1 SINR Generation Block

This block generates one SINR value for each narrowband channel based on a specification of the scenario, which includes a channel model (coherence time, coherence bandwidth), a mobility model, and other parameters (carrier frequency and bandwidth). The block also takes as input the number of users, user locations, number of antennas, antenna gain patterns, antenna configurations, channel estimates, precoder and combiner, pilot and noise power. The resulting vector of SINRs depends on the locations of transmitter and receiver, the precoder employed at the transmitter or combiner employed at the receiver, interfering transmissions, and the impact of large and small scale fading. This vector of SINRs is generated per user. This SINR computation is explained later in Section 2.3.

2.1.2 Performance Mapping Block

This block contains two stages. The first stage takes the vector of SINRs of each user generated by the SINR generation block and maps them onto a scalar called the effective SINR. This process is referred to as SINR compression. The second stage, which is known as the quality mapping,

maps the effective SINR to the PER. This block is explained in detail later in Section 2.4 and calibration process of mapping block is explained in Section 2.4.2.

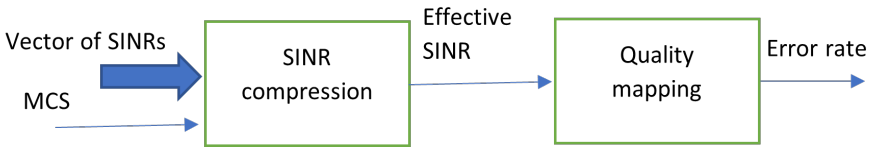


Figure 2.3: Performance Mapping Block

The vector of SINRs contains SINRs for the different coherence blocks over which data is encoded. The mapping of the vector of SINRs onto the effective SINR combines all the different SINRs seen by the block of data into a single scalar number. Specifically, this mapping models the link as an equivalent AWGN channel and the effective SINR can be interpreted the equivalent SNR on an AWGN channel. This mapping in general depends on the MCS employed, and predicts the performance in a computationally efficient manner. Link-level simulations of the RadioWeaves are performed initially to calibrate this quality mapping block. Once calibration has been performed, the computationally intensive link-level simulations need not be performed again.

2.2 Physical Layer

This section provides a detailed overview of physical layer parameters, modes of operation, different deployment settings of RadioWeaves, and how they are used in the evaluation framework model. These parameters are user configurable in the code framework. Various configurable parameters supported by the code framework are tabulated in Table 2.1.

The following notation is used in the sequel. Boldface lowercase letters denote column vectors and boldface uppercase letters denote matrices. The superscripts $(\cdot)^*$, $(\cdot)^{\mathsf{T}}$, and $(\cdot)^{\mathsf{H}}$ denote the conjugate, transpose, and conjugate transpose, respectively. The notation $\mathbf{I}_N$ represents the $N \times N$ identity matrix. The (m,n)th element of a matrix $\mathbf{A}$ is denoted by $[\mathbf{A}]_{mn}$. A block-diagonal matrix is represented by $\mathrm{diag}(\mathbf{A}_1, \cdots, \mathbf{A}_N)$ for matrices $\mathbf{A}_1, \cdots, \mathbf{A}_N$. The notation $\mathbf{D}_\mathbf{a}$ represents a diagonal matrix with entries from vector $\mathbf{a}$. The absolute value of a scalar and l_2-norm of a vector are denoted by $|\cdot|$ and $\|\cdot\|$, respectively. The real value of a scalar is denoted by $\Re\{\cdot\}$. We denote the expectation and variance by $\mathbb{E}\{\cdot\}$ and $\mathrm{Var}\{\cdot\}$, respectively. $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, \mathbf{C})$ denotes that z is a multi-variate circularly symmetric complex Gaussian random vector with zero mean and covariance matrix $\mathbf{C}$.

Spherical Coordinate System

This chapter majorly uses the spherical coordinate system to specify the position of a point. Consider a point (x, y, z) in the Cartesian coordinate system. The spherical coordinates (r, θ, φ), where r is the distance to the point from the origin, $\theta \in [0, \pi]$ is the elevation angle and $\varphi \in [-\pi, \pi)$ is the azimuth angle, can be obtained from (x, y, z) as

$$r = \sqrt{x^2 + y^2 + z^2} \tag{2.1}$$

$$\cos\theta = \left(\frac{z}{r}\right) \tag{2.2}$$

$$\tan\varphi = \left(\frac{y}{x}\right). \tag{2.3}$$

Table 2.1: Configurable parameters for the performance evaluation framework

Parameter	Supported Values/Units
Channel Type	LoS, Rayleigh, Rician, Custom
Mobility Model	Static, Random Walk, Custom
Antenna Configuration	ULA, UPA, RadioStripes, Custom
Antenna Gain Pattern	Omni, Patch
Channel Estimator	MMSE, LS, PCSI
Signal bandwidth	Hz
Carrier Frequency	Hz
Subcarrier bandwidth	Hz
Base station power	W
Base station noise figure	dB
Number of BS antennas	
Antenna Spacing	(in multiples of λ)
Number of users	
User power	W
Pilot power	W
User noise figure	dB
Room Size (Length, Width, height)	m
Precoding/Combining	ZF, MR
Operation Mode	Downlink, Uplink
Temperature of Operation	K

2.2.1 System Model

We consider a RadioWeaves system, where we have M antennas jointly serving K single antenna users. Let $\mathbf{G} \in \mathbb{C}^{M \times K}$ be the channel matrix between the M transmit antennas and the K users defined by

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1K} \\ g_{21} & g_{22} & \cdots & g_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ g_{M1} & g_{M2} & \cdots & g_{MK} \end{bmatrix}. \tag{2.4}$$

For the antenna modeling (see below), we assume that the coupling is insignificant among the service antennas.

Noise Power

Thermal noise is present in all electrical circuits and the noise power N_0 is calculated as

$$N_0 = k_B \cdot T \cdot BW \cdot 10^{-\mathsf{NF}/10} \tag{2.5}$$

where $k_B = 1.23 \times 10^{-23}$ is the Boltzmann constant, T is the temperature in Kelvin, BW is the bandwidth under consideration, and NF is the noise figure in dB, which is a hardware dependent parameter.

Uplink

In the uplink, all the users coherently transmit data to the base station. The collective signal received at the service antennas, $\mathbf{y} \in \mathbb{C}^{M\times 1}$, is

$$\mathbf{y} = \sqrt{\rho_{\mathsf{ul}}}\mathbf{G}\mathbf{x} + \mathbf{w}, \tag{2.6}$$

where ρ_{ul} is the uplink transmit power and $\mathbf{x} = \mathbf{D}_{\boldsymbol{\eta}}^{\frac{1}{2}}\mathbf{q} \in \mathbb{C}^{K\times 1}$. $\mathbf{D}_{\boldsymbol{\eta}} = \mathrm{diag}\{\boldsymbol{\eta}\}$, where $\boldsymbol{\eta} = [\eta_1\ \eta_2 \cdots \eta_K]^{\mathsf{T}}$ is the power control vector satisfying $0 \leq \eta_k \leq 1,\ \forall k$. $\mathbf{q} \in \mathbb{C}^{K\times 1}$ is the data from all users and $\mathbf{w} \sim \mathcal{CN}(0, N_0\mathbf{I}_M)$ is the noise.

Downlink

In the downlink, the service antennas jointly transmit the data $\mathbf{q} = [q_1\ q_2 \cdots q_K]^{\mathsf{T}}$, where q_k is the data intended for user k. The collective signal received at the users is denoted by the vector $\mathbf{y} \in \mathbb{C}^{K\times 1}$ and is expressed as

$$\mathbf{y} = \sqrt{\rho_{\mathsf{dl}}}\mathbf{G}^{\mathsf{H}}\mathbf{x} + \mathbf{w} \tag{2.7}$$

where ρ_{dl} is the downlink transmit power, $\mathbf{x}$ is the downlink transmitted signal given by

$$\mathbf{x} = \mathbf{A}\mathbf{D}_{\boldsymbol{\eta}}^{\frac{1}{2}}\mathbf{q}. \tag{2.8}$$

$\mathbf{A} \in \mathbb{C}^{M\times K}$ is the precoding matrix to spread the K signals into M antennas. $\mathbf{D}_{\boldsymbol{\eta}} = \mathrm{diag}\{\boldsymbol{\eta}\}$, where $\boldsymbol{\eta} = [\eta_1\ \eta_2 \cdots \eta_K]^{\mathsf{T}}$ is the power control vector such that $\sum_{k=1}^{K}\eta_k \leq 1$ and $\mathbf{w} \sim \mathcal{CN}(0, N_0\mathbf{I}_K)$ is the noise. Also we assume that the data $\mathbf{q}$ have maximum transmit power of 1, i.e., $\mathbb{E}\{\|\mathbf{q}\|^2\} \leq 1$.

2.2.2 Channel Models

The code framework supports some simplistic canonical textbook channel models, and the use of custom channel models devised for RadioWeaves and are briefly outlined below.

Line-of-Sight Channel Model

A line-of-sight (LoS) channel is a simplistic channel model when the transmitter and receiver are assumed to be in free space. We assume that there is only one direct propagation path from transmitter to receiver and no scatterers in the environment. Let the UE be relatively at (r, θ, φ) from the transmitter. Let P_{tx} be the transmitted power. Then the received power assuming an omni-directional antenna at the UE operating at a wavelength λ is given by,

$$P_{rx} = \frac{P_{tx}}{4\pi}\, G(\theta,\varphi)\, \frac{1}{r^2}\, \frac{\lambda^2}{4\pi}. \tag{2.9}$$

$G(\theta, \varphi)$ is the directional power gain of the transmit antenna and given by

$$G(\theta,\varphi) = \frac{4\pi\, U(\theta,\varphi)}{\int\int U(\theta,\varphi)\sin(\theta)d\theta d\varphi} = \alpha^2\, U(\theta,\varphi), \tag{2.10}$$

and $U(\theta, \varphi)$ is the normalized directional radiation intensity. For an isotropic transmitting antenna, $U(\theta,\varphi) = \frac{1}{4\pi}$ and hence $G(\theta, \varphi) = 1$ and when $G(\theta, \varphi)$ is a constant, (2.9) gives Friis' free space equation [22].

Under LoS channel model, two antenna models are supported by the framework: omni-directional antennas and rectangular microstrip patch antennas.

Omni-directional Antenna: For omni-directional antenna, the directional power gain $G(\theta, \varphi) = 1$ in any direction. Thus, the deterministic channel gain g is given by

$$g = \frac{\lambda}{4\pi r} e^{-j\frac{2\pi r}{\lambda}} \tag{2.11}$$

Microstrip Patch Antenna: To get the directional power gain of patch antenna, let h, L, and W denote the height, length, and width, respectively, of the patch antenna. Let ε_r and f be the dielectric constant of the substrate and resonant frequency, respectively. Let c be the speed of light. For a given h, the design of the patch antenna is given as follows

$$W = \frac{c}{2f}\sqrt{\frac{2}{\varepsilon_r + 1}}, \tag{2.12}$$

$$L = \frac{c}{2f\sqrt{\varepsilon_{\text{reff}}}} - 2\Delta L, \tag{2.13}$$

where $\varepsilon_{\text{reff}}$ and ΔL are obtained as

$$\varepsilon_{\text{reff}} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left(1 + 12\frac{h}{W}\right)^{-1/2} \tag{2.14}$$

$$\frac{\Delta L}{h} = 0.412\frac{(\varepsilon_{\text{reff}} + 0.3)(\frac{W}{h} + 0.264)}{(\varepsilon_{\text{reff}} - 0.258)(\frac{W}{h} + 0.8)}. \tag{2.15}$$

Using the above design parameters, the gain pattern of the patch antenna is given by

$$G(\theta, \varphi) = \left(\alpha \sin(\theta)\frac{\sin(X)}{X}\frac{\sin(Z)}{Z}\right)^2 \tag{2.16}$$

where

$$X = \frac{\pi h}{\lambda}\sin(\theta)\cos(\varphi), \tag{2.17}$$

$$Z = \frac{\pi W}{\lambda}\cos(\theta), \tag{2.18}$$

$$\alpha^2 = \frac{4\pi}{\int_{\theta=0}^{\pi}\int_{\varphi=-\frac{\pi}{2}}^{\frac{\pi}{2}}\left(\frac{\sin(X)}{X}\frac{\sin(Z)}{Z}\right)^2 \sin^3(\theta)d\theta d\varphi}. \tag{2.19}$$

Note that, the integration limits for φ is from $\frac{-\pi}{2}$ to $\frac{\pi}{2}$ to consider the fact that the antennas are transmitting forward and not back into the walls. With vertical polarization and with no polarization losses the deterministic channel gain g for a microstrip patch antenna is given by

$$g = \frac{\alpha\lambda}{4\pi r} e^{-j\frac{2\pi r}{\lambda}} \sin(\theta)\frac{\sin(X)}{X}\frac{\sin(Z)}{Z}. \tag{2.20}$$

Independent Rayleigh Fading Channel Model

When we have a rich scattering environment, we consider the channel to be Rayleigh distributed with independent fading across antennas. Each channel gain g is distributed as $g \sim \mathcal{CN}(0, \beta)$, where β is the large scale path loss fading coefficient between the transmitter and the receiver antenna. The following micro-cell propagation model used in [1] is considered for the large-scale fading coefficient β:

$$\beta[\text{dB}] = -30.5 - 36.7 \log_{10}\left(\frac{d}{1\text{m}}\right). \tag{2.21}$$

With a Rayleigh fading channel model, we consider the use of omni-directional antennas, to capture all the scatterers.

Rician Fading Channel Model

If there is a strong line of sight component in a rich scattering environment, the channel can be considered Rician distributed with factor κ. The channel gain is then given by

$$g = \sqrt{\frac{\kappa\beta}{\kappa+1}} + \sqrt{\frac{\beta}{\kappa+1}} X, \tag{2.22}$$

where $X \sim \mathcal{CN}(0, 1)$. A Rician fading channel with $\kappa = 0$ reduces to a Rayleigh fading channel.

Custom Channel Model

The framework also supports custom channel models. It can take arbitrary channel matrices generated by a custom channel model as input. This specifically will enable the integration of new channel models for RadioWeaves being developed in work package 1.

2.2.3 Mobility Model

The framework offers the possibility of integrating different mobility models.

Static

With the static model, we consider that users in the system have fixed locations and their locations do not change with time. Thus the coherence interval can be considered very large, if nothing else in the environment moves.

Random walk

In random walk, the users change their location every coherence time interval. They move in a random direction from their current location, and the movement depends only on the previous location.

Custom movement

The framework also supports custom mobility patterns. The movement pattern need to be given as a vector of positions for each user at each time step.

2.2.4 Antenna Array Deployment

The framework supports arbitrary topologies for the deployment of antennas. In what follows, we give some specific examples that are used later in the performance analysis.

Reference Case: Co-Located Array

For the co-located reference case, we consider an indoor environment with a candelabrum type array on the ceiling of the room, see in Fig. 2.4. With directional antennas, this shape guarantees that there are antennas pointing towards every corner of the room.

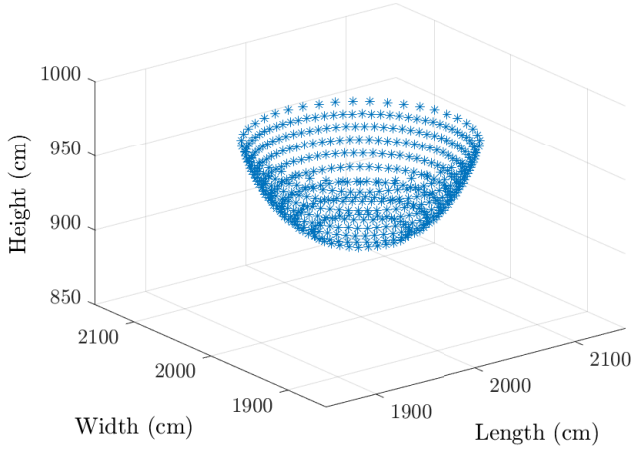


Figure 2.4: Co-located massive MIMO array in the shape of a candelabrum over the ceiling.

Multiple Radio Stripes on the Walls

In the following examples, the service antennas are arranged in multiple radio “stripes” mounted on the walls and are user configurable to consider different deployment options. We consider the following configurations:

1. Single stripe on 1 wall: The antennas are deployed horizontally over a single wall.
2. Single stripe on 4 walls: The antennas are deployed horizontally over all walls.
3. Double stripes on 1 wall: The antennas are deployed horizontally as two separate stripes on a single wall.
4. Double stripes on 4 walls: The antennas are deployed horizontally as two separate stripes over all walls.
5. Quadruple stripes on 1 wall: The antennas are deployed horizontally as four separate stripes on a single wall.
6. Quadruple stripes on 4 walls: The antennas are deployed horizontally as four separate stripes over all walls. A typical scenario is shown in Fig. 2.5.

In general, when we consider stripes on the walls, we do not restrict the antenna spacing to be $\lambda/2$. In the performance evaluation code framework, the spacing of the antennas and the height at which the stripes are placed are user configurable.

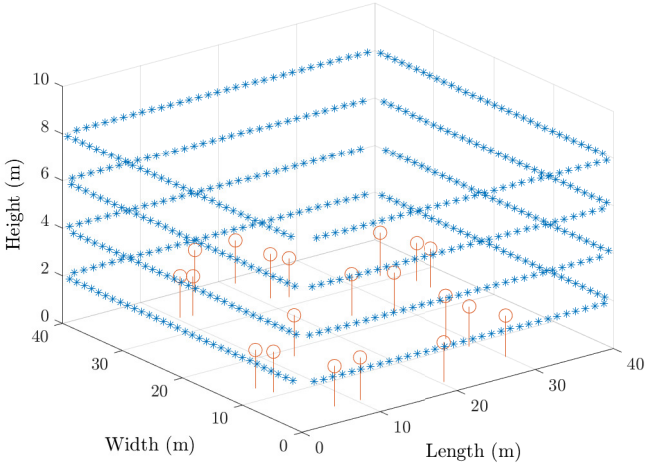


Figure 2.5: Quadruple strip deployed on all four walls of the room with 8λ spacing between adjacent antenna elements and each stripe is placed at 2m vertical distance.

Uniform Linear Array

Another deployment scenario considers multiple uniform linear array (ULA) placed on different walls. Within each ULA, antennas are placed such that they have $\lambda/2$ spacing. The framework supports the placement of ULAs on a single wall as well as four walls. Each ULA is placed in the middle of each wall of the room and a typical scenario is shown in Fig. 2.6 where four ULAs are placed on all four walls. All the ULAs are connected to EPU through backhaul network and act as a single large network. Moreover, we could consider a ULA as a single strip of radio elements placed with $\lambda/2$ spacing. Considering the implementation standpoint, the configurable parameters for ULA scenario is limited only to height at which ULA is placed.

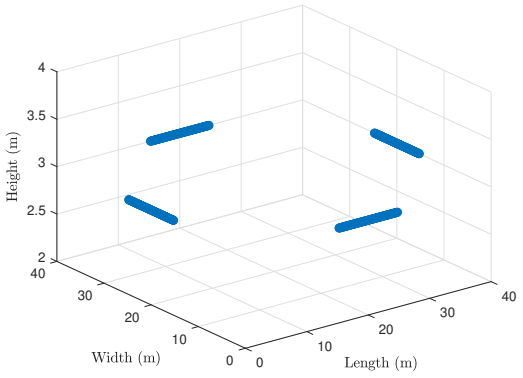


Figure 2.6: Multiple ULA deployed over multiple walls at a vertical height of 3m above ground.

Uniform Planar Array

Another deployment scenario considered is multiple uniform planar array (UPA) placed on different walls. Within each UPA, antennas are placed such that they have $\lambda/2$ spacing. The code framework supports placing of UPA on one or four walls. Each ULA is placed in the middle of each wall of the room and a typical scenario is shown in Fig. 2.6 where four UPAs are placed on all

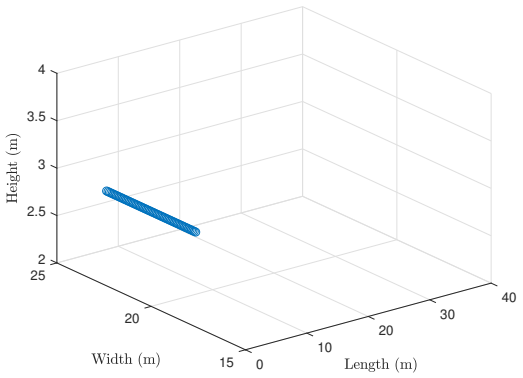


Figure 2.7: Single ULA deployed on single wall.

four walls. In the code framework, the height at which the ULA is placed and planar configuration are user configurable.

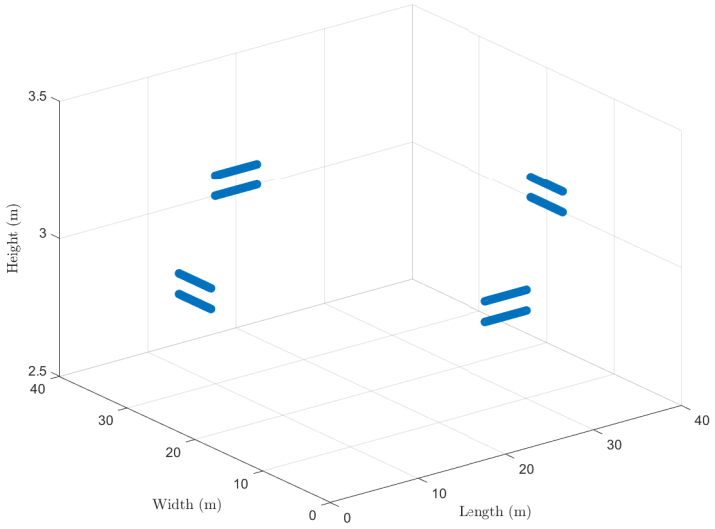


Figure 2.8: Multiple ULA deployed over four walls at vertical height of 3m with a planar configuration of 2×64.

Custom Antenna Configuration

The performance evaluation code framework supports a user defined antenna configuration and the same can be provided to the code as a file containing antenna locations.

2.2.5 Channel Estimation

In practice, obtaining perfect channel state information (PCSI) is difficult and we need to do channel estimation. The channel estimation technique depends on the channel model under consideration, i.e., whether it is deterministic or stochastic. Accordingly, we categorize the channel estimation into two parts: one for line-of-sight-channels, and the other for the Rayleigh fading

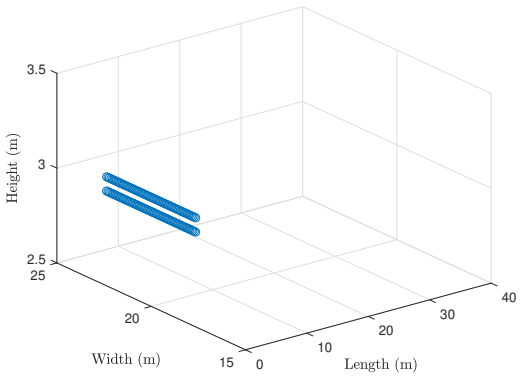


Figure 2.9: Single UPA deployed on single wall with a planar configuration of 2×64.

channel. Towards this, let τ_c denote the length of the coherence interval (see, e.g., Chapter 2 of [23]) and we assume there are τ_p mutually orthogonal pilot sequences of τ_p length such that $\tau_c \geq \tau_p \geq K$. We employ the following pre-processing of the received pilot signal before estimating the channel.

Let $\boldsymbol{\phi}_k \in \mathbb{C}^{\tau_p \times 1}$ be the pilot sequence associated with user k, which is the k^{th} column of the $\tau_p \times K$ unitary matrix $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^{\mathsf{H}}\boldsymbol{\Phi} = \mathbf{I}$. All UEs collectively transmit $\mathbf{X}_p = \sqrt{\tau_p}\boldsymbol{\Phi}^{\mathsf{H}}$ over τ_p symbols and the signals received at the SP $\mathbf{Y}_p \in \mathbb{C}^{M \times \tau_p}$ can be written using (2.6) as

$$\mathbf{Y}_p = \sqrt{\rho_{\mathsf{ul}}}\mathbf{G}\mathbf{X}_p + \mathbf{W}_p, \tag{2.23}$$

where ρ_{ul} is a scalar to scale the uplink transmit power and $\mathbf{W}_p$ is the noise matrix with i.i.d elements $\mathcal{CN}(0, N_0)$. De-spreading operation is performed at the base station on the received signal $\mathbf{Y}_p$ and is given by

$$\mathbf{Y}'_p = \mathbf{Y}_p\boldsymbol{\Phi} = \sqrt{\rho_{\mathsf{ul}}\tau_p}\mathbf{G} + \mathbf{W}'_p. \tag{2.24}$$

Since $\boldsymbol{\Phi}$ is a unitary matrix, the elements of $\mathbf{W}'_p = \mathbf{W}_p\boldsymbol{\Phi}$ are i.i.d. $\mathcal{CN}(0, N_0)$. Let $\hat{\mathbf{G}} \in \mathbb{C}^{M \times K}$ be the estimated channel matrix whose $(m,k)^{th}$ element is $\hat{g}_{mk}$, where $\hat{g}_{mk}$ is the channel estimate of g_{mk}. Also, let $\hat{\mathbf{g}}_k = [\hat{g}_{1k}, \ldots, \hat{g}_{Mk}]^{\mathsf{T}}$ is the k^{th} column of the channel estimation matrix $\hat{\mathbf{G}}$. The following sections provide commonly used methods to compute $\hat{\mathbf{G}}$.

Least Squares (LS) Channel Estimation

The least squares method is commonly employed when no prior statistical knowledge of the channel is available. The LS estimate of the channel g_{mk} between k^{th} user and m^{th} antenna is given by

$$\hat{g}^{\mathrm{LS}}_{mk} = \frac{[\mathbf{Y}'_p]_{mk}}{\sqrt{\rho_{\mathsf{ul}}\tau_p}} \tag{2.25}$$

where $[\mathbf{Y}'_p]_{mk}$ is the $(m,k)^{th}$ element of $\mathbf{Y}'_p$. The LS method applies to channel models without prior knowledge such as LoS channel.

Minimum Mean Square Error (MMSE) Channel Estimation

The MMSE channel estimation is one of the most widely employed estimation techniques in the massive MIMO literature. This is because, among its various attractive benefits, the important

property that the channel estimate and the channel estimation error are independent enables computing closed-form spectral efficiency (SE) expression in many scenarios. Let the $(m,k)^{th}$ element of $\mathbf{G}$ be $g_{mk} \sim \mathcal{CN}(0,\beta_{mk})$ where β_{mk} is the large-scale fading coefficient between user k and antenna m. We assume that the prior distribution of g_{mk} is known at the receiver. Then the MMSE channel estimate is

$$\begin{aligned}\hat{g}_{mk}^{\text{MMSE}} &= \mathbb{E}\left\{g_{mk}|\mathbf{Y}_p^{'}\right\} \\ &= \frac{\sqrt{\rho_{\text{ul}}\tau_p}\beta_{mk}}{1+\tau_p\rho_{\text{ul}}\beta_{mk}}[\mathbf{Y}_p^{'}]_{mk}\end{aligned} \tag{2.26}$$

The mean-square of the channel estimate is denoted by γ_k^m and given by

$$\begin{aligned}\gamma_{mk} &= \mathbb{E}\left\{|\hat{g}_{mk}|^2\right\} \\ &= \frac{\tau_p\rho_{\text{ul}}\beta_{mk}^2}{1+\tau_p\rho_{\text{ul}}\beta_{mk}}.\end{aligned} \tag{2.27}$$

The channel estimation error $\tilde{g}_{mk}$ is given as

$$\tilde{g}_{mk} = \hat{g}_{mk} - g_{mk}. \tag{2.28}$$

The mean-square estimation error is given by

$$\begin{aligned}\mathbb{E}\left\{|\tilde{g}_{mk}|^2\right\} &= \frac{\beta_{mk}}{1+\tau_p\rho_{\text{ul}}\beta_{mk}} \\ &= \beta_{mk} - \gamma_{mk}.\end{aligned} \tag{2.29}$$

The MMSE channel estimates is applicable in with and without prior knowledge of the channel model. However, it is more suitable in case with prior knowledge of the channel such as Rayleigh fading channel.

2.2.6 Precoding/Combiners

Uplink Receiver Combiners

The uplink receive combiners should be selected taking into consideration the performance metric of interest. The most commonly employed receivers are maximum-ratio combining (MRC) and zero-forcing (ZF).

The MRC receiver matrix defined by $\mathbf{V}^{\text{MR}} = [\mathbf{v}_1^{\text{MR}},\ldots,\mathbf{v}_K^{\text{MR}}]^H$ where

$$\mathbf{v}_k^{\text{MR}} = \frac{\hat{\mathbf{g}}_k}{\|\hat{\mathbf{g}}_k\|}. \tag{2.30}$$

The ZF receiver is given by

$$\mathbf{V}^{\text{ZF}} = \left(\hat{\mathbf{G}}^{\text{H}}\hat{\mathbf{G}}\right)^{-1}\hat{\mathbf{G}}^{\text{H}}. \tag{2.31}$$

Downlink Precoders

The downlink precoders can also be selected considering different performance targets. The most commonly used precoders are maximum-ratio transmission (MRT) and ZF.

The MRT matrix defined by $\mathbf{A}^{\mathsf{MR}} = [\mathbf{a}_1^{\mathsf{MR}}, \ldots, \mathbf{a}_K^{\mathsf{MR}}]$ where

$$\mathbf{a}_k^{\mathsf{MR}} = \frac{\hat{\mathbf{g}}_k}{\|\hat{\mathbf{g}}_k\|}. \tag{2.32}$$

The ZF precoder is in turn given by

$$\mathbf{A}^{\mathsf{ZF}} = \hat{\mathbf{G}}^{\mathrm{H}} \left(\hat{\mathbf{G}} \hat{\mathbf{G}}^{\mathrm{H}} \right)^{-1}. \tag{2.33}$$

2.2.7 Power Control

Uplink Power Control

In uplink, the power control coefficient for user k should satisfy the condition $0 \leq \eta_k \leq 1, \forall k$. In the framework, the power control coefficient for user k is given by [10, Ch. 7]

$$\eta_k = \frac{\left(\sum_{m=1}^{M} \beta_{mk} \right)^{\nu}}{\max_{i=\{1,2,\cdots,K\}} \left(\sum_{m=1}^{M} \beta_{mi} \right)^{\nu}} \tag{2.34}$$

where $\nu \in [-1, 1]$. When $\nu = 0$, we have full power allocation and when $\nu = -1$, we will have max-min fairness power control. See also [9], where these two special allocation policies were first derived and analyzed. The parameter ν is provided to the user as a configuration parameter.

Downlink Power Control

For the downlink, the power control coefficients should satisfy the condition $0 \leq \sum_{k=1}^{K} \eta_k \leq 1$. In the framework, it is common to consider a max-min fairness power control policy which results in rather computationally heavy optimization problems [9]. We consider a simpler policy, described in [10, Ch. 7]

$$\eta_k = \frac{\left(\sum_{m=1}^{M} \beta_{mk} \right)^{\nu}}{\sum_{i=1}^{K} \left(\sum_{m=1}^{M} \beta_{mi} \right)^{\nu}} \tag{2.35}$$

where $\nu \in [-1, 1]$, with interpretations similar to uplink power control policy. This policy, or more accurately a minor variation of it, was first proposed in [11] and works well in practice; however, it cannot yield the max-min fairness solution exactly.

2.3 SINR Generation

In this section, we outline the signal-to-interference-plus-noise ratio (SINR) computation for both uplink and downlink scenarios in the SINR generation block (Fig. 2.2). Computations shown below are per one narrow band channel where the channel remains same.

Downlink

In downlink, the collective signal received at all the users is given by

$$\mathbf{y} = \sqrt{\rho_{\mathsf{dl}}} \mathbf{G}^{\mathsf{H}} \mathbf{A} \mathbf{D}_{\boldsymbol{\eta}}^{\frac{1}{2}} \mathbf{q} + \mathbf{w}. \tag{2.36}$$

Table 2.2: Simulation Parameters for SINR analysis

Frequency of operation, f	2 GHz
Antenna Deployment	ULA over four walls
Signal bandwidth	20 MHz
Subcarrier bandwidth	200 kHz
Mobility	Static
Base station power	1 mW
Base station noise figure	5 dB
User power	1 μW
Pilot power	20 μW
User noise figure	9 dB
Channel type	LOS
Channel estimator	LS / PCSI
Precoding/Combining	ZF / MR
Dielectric constant, ε_r	10.2
Height of patch antenna, h	0.1588 cm
Temperature of Operation	300 K

Thus at each user, the received signal can be written as

$$y_k = \sqrt{\rho_{\mathsf{dl}}}\mathbf{g}_k^{\mathsf{H}}\mathbf{A}\mathbf{D}_{\boldsymbol{\eta}}^{\frac{1}{2}}\mathbf{q} + w_k \tag{2.37}$$

$$= \sqrt{\rho_{\mathsf{dl}}\eta_k}\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_k q_k + \sum_{i=1, i\neq k}^{K} \sqrt{\rho_{\mathsf{dl}}\eta_i}\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_i q_i + w_k. \tag{2.38}$$

The instantaneous SINR at user k, is then given by

$$\mathsf{SINR}_k^{\mathsf{inst.}} = \frac{\rho_{\mathsf{dl}}\eta_k|\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_k|^2}{\sum_{i=1,i\neq k}^{K}\rho_{\mathsf{dl}}\eta_i|\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_i|^2 + N_0}. \tag{2.39}$$

Note that for this instantaneous SINR to have a rigorous information-theoretic operational meaning, each UE needs to know the instantaneous value of $\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_k$ as well as the instantaneous SINR. These can be estimated to a good degree of accuracy by using techniques from for example [24] and for the purpose of obtaining an approximate performance prediction we will here assume that they are known.

Figs. 2.10 and 2.11 show the downlink SINR variations as CDFs in a RadioWeave deployment in two different scenarios. The used simulation setup is provided in Table 2.2. From the figure, it can be seen that ZF performs much better than MR as ZF can suppress interference from other users. Moreover, the impact of imperfect channel estimates is large in the case of ZF compared to MR. The gain from the patch antennas can be exploited well by the ZF approach, while MR is not able to exploit the antenna gain as the interference term dominates. The randomness in this simulation is due to the random spatial distribution of the users, the channel estimation errors, and variations of the channel response (in LOS) among the subcarriers. An important observation is that the CDFs are very stable (i.e., the distributions have small variance). This means that the probability of having a scenario (in terms of coordinates of the users) or seeing a realization of the estimation errors that result in poor performance is miniscule. Specifically, even with the simplest MRT precoder, the probability of obtaining an SINR below 0 dB is much less

than 1%. Note that, this SINR corresponds to a gross spectral efficiency of 1 bit/s/Hz, which is by virtue of the stableness of the CDF almost guaranteed for all terminals simultaneously. (Spectral efficiencies of similar orders of magnitude were also considered in the case studies in [23] when applying a uniform quality-of-service power control policy.)

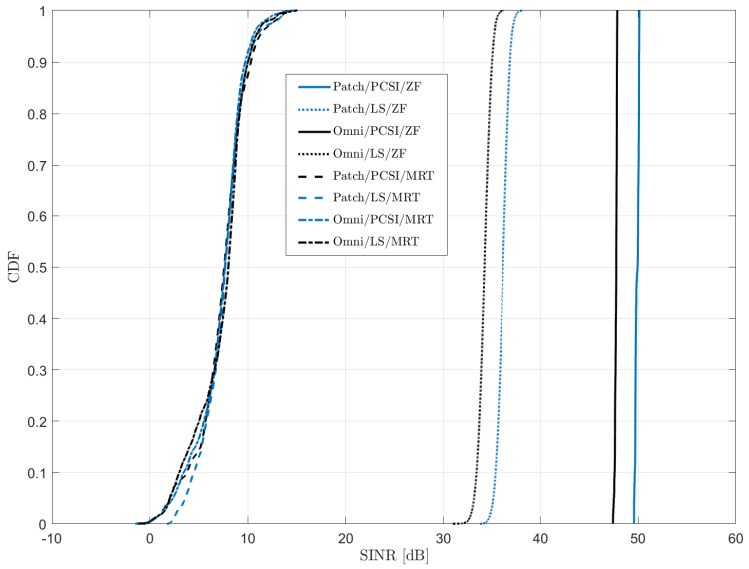


Figure 2.10: Downlink SINR for a room with 40 m $\times$ 40 m $\times$ 10m size, $M = 512$, and $K = 100$. The legend is to be read as antenna Type/Estimation Method/Precoder

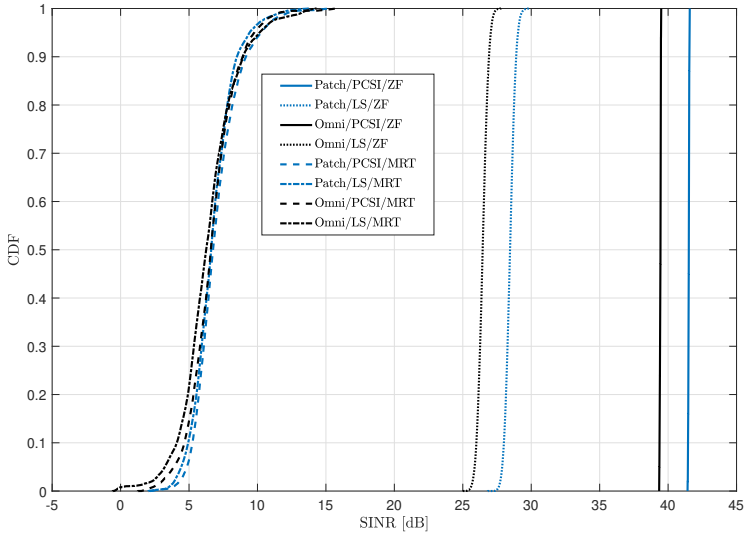


Figure 2.11: Downlink SINR for a room with 140 m $\times$ 70 m $\times$ 15m room size (the production hall in Page 46 deliverable 1.1), $M = 1024$, $K = 200$.

Uplink

In the uplink, the collective signal received from all the users at the EPU is given by

$$\mathbf{y} = \sqrt{\rho_{\text{ul}}}\mathbf{G}\mathbf{D}_{\boldsymbol{\eta}}^{\frac{1}{2}}\mathbf{q} + \mathbf{w}. \tag{2.40}$$

At the EPU a combining operation by a matrix $\mathbf{V} \in \mathbb{C}^{K \times M}$ (Sec. 2.2.6) is done and the processed signal is given by

$$\mathbf{V}\mathbf{y} = \sqrt{\rho_{\text{ul}}}\mathbf{V}\mathbf{G}\mathbf{D}_{\boldsymbol{\eta}}^{\frac{1}{2}}\mathbf{q} + \mathbf{V}\mathbf{w}. \tag{2.41}$$

Therefore, the kth component of the processed signal at EPU is given by

$$\mathbf{v}_k^{\mathsf{H}}\mathbf{y} = \sqrt{\rho_{\text{ul}}}\mathbf{v}_k^{\mathsf{H}}\mathbf{G}\mathbf{D}_{\boldsymbol{\eta}}^{\frac{1}{2}}\mathbf{q} + \mathbf{v}_k^{\mathsf{H}}\mathbf{w} \tag{2.42}$$

$$= \sqrt{\rho_{\text{ul}}\eta_k}\mathbf{v}_k^{\mathsf{H}}\mathbf{g}_k q_k + \sum_{i=1, i\neq k}^{K} \sqrt{\rho_{\text{ul}}\eta_i}\mathbf{v}_k^{\mathsf{H}}\mathbf{g}_i q_i + \mathbf{v}_k^{\mathsf{H}}\mathbf{w}. \tag{2.43}$$

As the combining vector $\mathbf{v}_k$ is unit-norm, the instantaneous SINR at user k at the EPU is

$$\text{SINR}_k^{\text{inst.}} = \frac{\rho_{\text{ul}}\eta_k |\mathbf{v}_k^{\mathsf{H}}\mathbf{g}_k|^2}{\sum_{i=1, i\neq k}^{K} \rho_{\text{ul}}\eta_i |\mathbf{v}_k^{\mathsf{H}}\mathbf{g}_i|^2 + N_0}. \tag{2.44}$$

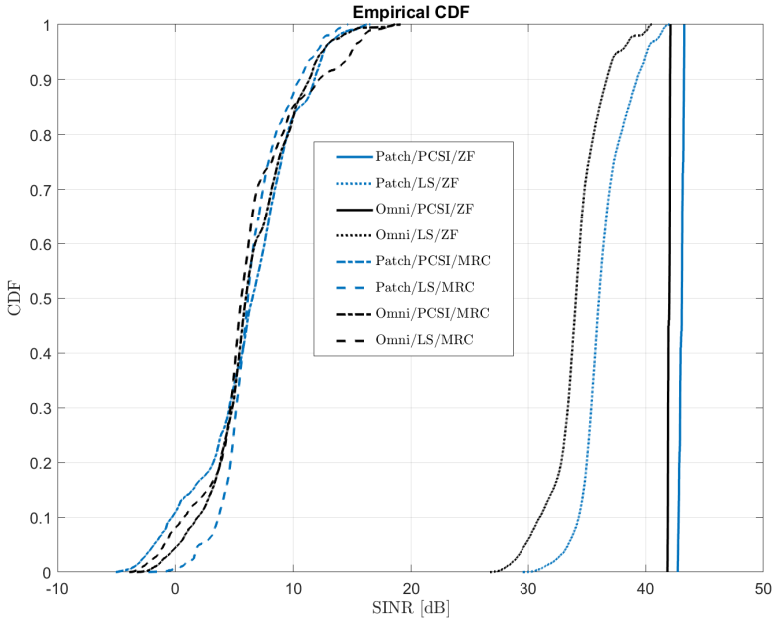


Figure 2.12: Uplink SINR for a room with 40mx40mx10m size, $M = 512$, $K = 100$.

Fig. 2.12 shows the uplink SINR variations for a RadioWeave deployment. We make similar observations as in the downlink case.

To obtain the link-level performance, we compute the SINR for each coherence block and forward a vector of the so-obtained SINRs to the performance mapping block, which is explained in the next section.

2.4 Performance Mapping

In this section, the performance mapping block, which is briefly introduced in Section 2.1.2, will be explained in detail. As explained earlier, the performance mapping accurately predicts the link-level performance in a computationally efficient manner and helps speed up the system-level

simulations. For this reason in literature, this abstraction model has been used in system-level simulations of many systems and standards, including LTE [25], WiMAX [7] and 5G-NR [8].

As shown in Figure 2.3, the performance mapping block contains two stages, namely, *SINR compression* and *quality mapping.* The SINR compression block takes in a vector of SINRs corresponding to subcarriers per channel use for each UE and outputs a scalar, which is known as effective SINR. The quality mapping block maps this effective SINR to the PER. To understand the accuracy of the mapping, its predicted performance is compared with actual performance obtained through link-level simulations. First, we will explain the concept of effective SINR mapping and different mappings studied in the literature. We then elaborate on a specific mapping called the exponential effective SINR mapping (EESM), which is shown to be accurate and has a computationally simple form.

2.4.1 SINR Compression

Here, the vector of SINRs a packet has seen is compressed into a scalar effective SINR. This results an equivalent AWGN channel whose SNR that captures the effect of the vector of SINRs seen by the packet. Subsequently, for a given MCS, the quality mapping step can then map the effective SINR onto a PER.

Let N denote the number of different SINRs seen by the packet. Let $\gamma = [\gamma_1, \ldots, \gamma_N]^{\mathsf{T}}$ denote the vector of SINRs and γ_{eff} denote the effective SINR. Then, the general form of mapping is given by [7, 26]

$$\gamma_{\text{eff}} = \beta f^{-1}\left(\frac{1}{N}\sum_{n=1}^{N} f\left(\frac{\gamma_n}{\beta}\right)\right), \tag{2.45}$$

where $f(\cdot)$ is a application dependent mapping function [26, 27] and β is an MCS-dependent parameter. This parameter β should be calibrated empirically to achieve accurate prediction. The accuracy of this mapping is, as it turns out, very robust and a value of β optimized for one scenario can also be used in many other, quite different scenarios. See Section 2.4.2 for more specifics on this point. However, importantly, each MCS has a different associated β.

We now briefly describe some mapping functions considered in the literature.

Average Value Interface [26, 28]: Here, the mapping function is given by $f(\gamma_n) = \gamma_n$ and the effective SINR is the arithmetic average of the vector of SINRs. It is given by

$$\gamma_{\text{eff}} = \frac{1}{N}\sum_{n=1}^{N}\gamma_n. \tag{2.46}$$

This is generally used to model flat-fading channels and is not an accurate measure for other channels such as frequency selective channels [26, 28]. It is also independent of the MCS used.

Capacity Based Mapping [26, 29, 30]: Here, the AWGN channel capacity formula is used as the mapping function, i.e., $f(\gamma_n) = \log_2(1+\gamma_n)$. Therefore, the effective AWGN SINR is computed based on the channel capacity of parallel channels with SINRs $\gamma_1, \ldots, \gamma_N$ resulting in

$$\gamma_{\text{eff}} = \beta\left[\prod_{n=1}^{N}\left(1+\frac{\gamma_n}{\beta}\right)\right]^{\frac{1}{N}} - \beta. \tag{2.47}$$

2.4.2 Exponential Effective SINR Mapping (EESM)

With the background just given, we now present EESM which has very good accuracy and is easy to calibrate, and has low computational complexity. Here, the mapping function $f(\gamma_n) = 1 - \exp(-\gamma_n)$ [26, 31, 32]. Therefore, the effective SINR mapping is given by

$$\gamma_{\text{eff}} = -\beta \log\left(\frac{1}{N}\sum_{n=1}^{N}\exp\left(-\frac{\gamma_n}{\beta}\right)\right). \tag{2.48}$$

Due to the shape of the mapping function, EESM is mostly impacted by lower SINR values. This emphasis given to low SINR values allows it to accurately predict the PER.

The EESM has been shown to achieve a good trade-off between accuracy and computational complexity. For this reason, it is widely used in physical-layer abstraction models. It is also shown to be applcable to systems with inter-cell interference [7, 33] and to OFDM systems with hybrid automatic repeat request (HARQ) [34]. We, therefore, focus on EESM due to its simplicity and proven accuracy. This mapping is obtained by generalizing the PER of an uncoded BPSK transmission over an AWGN channel. This generalization is done by introducing an MCS-dependent parameter β. The steps involved in the derivation of EESM are given in [26, 35].

EESM has been validated to perform accurately for several wireless standards such as WiMax and LTE. In Figure 3.1 of [28], it is shown how to predict the performance accurately for MCS employed in IEEE 802.16 WiMAX [36]. A similar study of EESM is done for LTE in [5, 6] and 5G NR in [8].

Calibration of β

The parameter β has to be calibrated for each MCS. The calibration procedure, which is described in [28], is as follows. We generate N_r realizations of the vector of SINRs. This set includes the channel realizations of different channel models supported, which makes the calibration apply to various channel models [8]. Including realizations of different channel models allows to generalize the calibration to all channels belonging to these classes. Then, link-level simulations of RadioWeaves are performed to obtain the actual PER for each vector of SINRs. Let P_i denote the actual PER of the i^{th} vector of SINRs. These P_i values are then mapped to the equivalent AWGN SNR by using the corresponding AWGN channels PER curve. Let $\mathbf{\Gamma}_{\text{A}} = [\gamma_{\text{A},1}, \ldots, \gamma_{\text{A},N_r}]$ denote the vector of AWGN equivalent SINRs of the N_r realizations of the vector of SINRs.

Let $\gamma_{\text{eff},i}(\beta)$ denote the effective SINR value for the i^{th} vector of SINRs for a given β obtained using (2.48). Let $\mathbf{\Gamma}_{\text{eff}}(\beta) = [\gamma_{\text{eff},1}(\beta), \ldots, \gamma_{\text{eff},N_r}(\beta)]$ denote the vector of effective SINRs. We then calibrate the value of β to obtain the optimized value β^* that minimizes the squared error between $\mathbf{\Gamma}_{\text{A}}$ and $\mathbf{\Gamma}_{\text{eff}}(\beta)$. It is given by

$$\beta^* = \underset{\beta}{\operatorname{argmin}} \, ||\mathbf{\Gamma}_{\text{A}} - \mathbf{\Gamma}_{\text{eff}}(\beta)||^2. \tag{2.49}$$

This calibration only needs to be performed once for each MCS with the resulting values for β^* is stored in a lookup table. Table 2.3 and Table 2.4 show the calibrated values of β^* for RadioWeaves for different MCS with LDPC coding and polar coding, respectively. With polar coding, cyclic redundancy check (CRC) bits, interleaving and rate matching are taken into consideration. Also, as stated in Section 2.3, the PER simulations assume that the receiver has knowledge of the instantaneous SINRs presented in the SINR generation section and also the corresponding

Table 2.3: Calibrated β^* values for RadioWeaves with LDPC codes

MCS index	Modulation	Code rate	β^*
0	BPSK	1/2	0.78
1	QPSK	1/2	1.55
2	16-QAM	1/2	4.16

Table 2.4: Calibrated β^* values for RadioWeaves with polar codes

MCS index	Modulation	Code rate	β^*
1	QPSK	1/2	0.624

effective channel gain, for instance in the downlink it is $\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_k$. Note that SINR contains the information of the interference and these (SINR and the effective channel) are required to compute the bit log-likelihood ratio (LLR) that required for channel decoders like LDPC.

Figure 2.13 shows the simulated PER of RadioWeaves on the vertical axis (shown as red x) versus the effective SINR $\gamma_{\text{eff}}(\beta^*)$ (obtained through the EESM mapping with calibrated β^*) on the horizontal axis, for different realizations of the vector of SINRs. This is done for different MCS's to illustrate the effectiveness of calibration. The calibrated β values for different MCSs, which are tabulated in Table 2.3, are used in generating the plot. Also shown is the PER curve of AWGN channel with SNR equal to the effective SINR, and for the same choices of MCS. We see that the markers, which represent the actual performance of the RadioWeaves system, match well with the AWGN PER values, which represent the predicted values. This establishes that the prediction accuracy of EESM is very good.

Robustness of β: To verify the robustness of the calibrated parameter β, for a given MCS, we repeated the calibration with different values of antenna elements and subcarriers. Table 2.5 shows the calibrated β values with different parameters. This is in line with the conclusions in [8], where the variation in β^* values is shown to be independent of the subcarrier spacing and other 5G-NR numerology.

Furthermore, we compared the simulated PER and the predicted PER for different parameters values for which calibration is not performed. Here, the simulated PER is obtained through the RadioWeaves link-level simulations and the predicted PER is obtained by the EESM-based performance mapping. Table 2.6 compares the simulated PER with predicted PER for two values of β^* from Table 2.5. We see that the predicted values are in general very close to the simulated values. However, there are exceptions, for example, for a PER of 0.0015 the prediction of 0.0009 is 40% off, and quite overconfident. We also see that the predicted value changes only by a small value as β varies slightly.

Extreme SINR profile: We now consider an extreme, artificial example in order to demonstrate

Table 2.5: Calibrated β^* values for different parameter values

MCS index	Modulation	Code rate	Number of antennas	Number of subcarriers	β^*
1	QPSK	1/2	48	2	1.54
1	QPSK	1/2	48	18	1.55
1	QPSK	1/2	64	2	1.54
1	QPSK	1/2	64	18	1.55

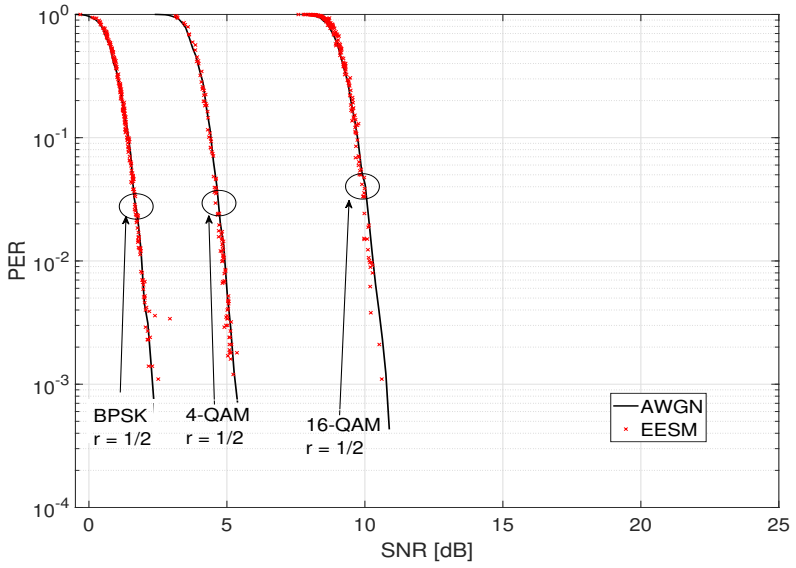


Figure 2.13: PER for AWGN channel and also link-level simulation with vector of SINRs is plotted against AWGN SNR (same axis for effective SINR). This plot provides the validation of EESM

Table 2.6: Comparison of simulated PER with predicted PER for different values of β^* (M=72, QPSK, and rate=0.5)

Simulated PER	Predicted PER (β^*=1.54)	Predicted PER (β^*=1.55)
0.0008	0.0006	0.0005
0.0015	0.0009	0.0009
0.0027	0.0039	0.0038
0.2760	0.2714	0.2693
0.2920	0.3081	0.3059
0.5040	0.4849	0.4830

Table 2.7: Comparison of simulated PER with predicted PER for extreme SINR profile shown in Figure 2.14 for M=72, QPSK, and rate=0.5.

Simulated PER	Predicted PER (β^*=1.54)
0.0118	0.0162
0.0138	0.0156
0.0145	0.0163
0.0187	0.0173
0.0222	0.0188
0.0251	0.0254
0.0255	0.0212
0.0256	0.0199

how robust the EESM mapping technique is. We specifically study the behavior of the performance mapping for an SINR profile that was synthesized only for the purpose of this robustness check, and not used for calibration of β (and not obtained from an actual RadioWeaves simulations). This SINR profile has a set of SINR values that are extremely low, as shown in Figure 2.14. Table 2.7 compares the simulated PER with the predicted PER for β^*=1.54. We see that the predicted PERs are reasonably close to the simulated PER values even for this extreme SINR profile. We conclude that the EESM mapping is a highly robust way of mapping a vector of instantaneous SINRs to link-level performance in terms of PERs for a variety of scenarios, where the specific scenario selected for the calibration is not particular important.

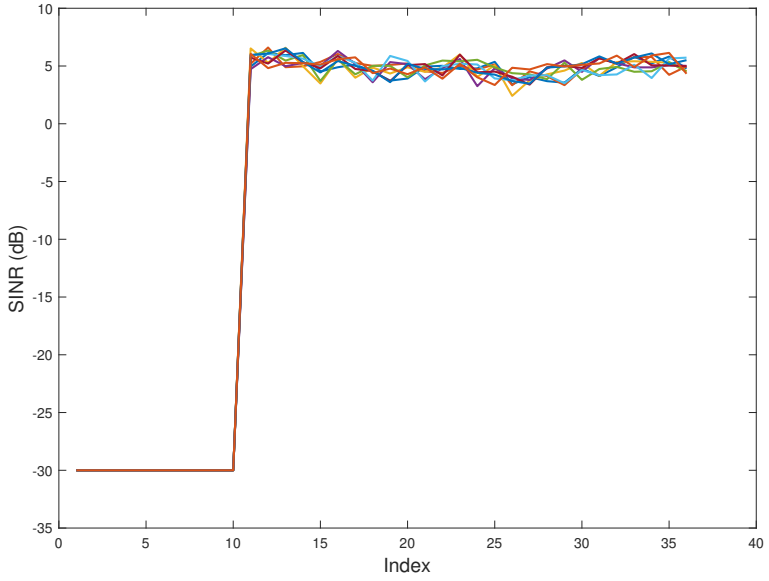


Figure 2.14: Extreme SINR profile as a function of sub-carrier index. Here, different curves correspond to different realizations for which PER's are shown in Table 2.7.

2.5 Analysis of Required Transmit Power in RadioWeaves

In this section, we consider the performance of the RadioWeaves system in terms of transmit power required. The basic theory here follows [37], and we will consider downlink operation in an

LOS scenario.

Power Required with Zero Forcing

We consider zero-forcing precoding [38] in the downlink in order to give all the users similar rates. In this subsection, we compute the power spent at the base station for zero forcing operation. Let $q_k \sim \mathcal{CN}(0,1)$ be the downlink signal intended for user k and let $\mathbf{q} = [q_1 \; q_2 \; \ldots \; q_K]^T \in \mathbb{C}^{K\times 1}$. The zero-forcing precoded signal $\mathbf{x} \in \mathbb{C}^{M\times 1}$ is given by

$$\mathbf{x} = \hat{\mathbf{G}}(\hat{\mathbf{G}}^{\mathsf{H}}\hat{\mathbf{G}})^{-1}\mathbf{q}. \tag{2.50}$$

The power spent at the transmitter to achieve the zero-forced signal is given by

$$P = \|\mathbf{x}\|^2 = \mathsf{Tr}((\hat{\mathbf{G}}^{\mathsf{H}}\hat{\mathbf{G}})^{-1}). \tag{2.51}$$

Downlink Spectral Efficiency

In each coherence block, up to $\tau_c - \tau_p$ symbols can be used for downlink. We consider zero-forcing precoding using the channel estimates for downlink, and the signal from base station to all users is given by

$$\mathbf{x} = \mathbf{A}\mathbf{q}, \tag{2.52}$$

where $\mathbf{A} = \frac{1}{\sqrt{\mathsf{Tr}((\hat{\mathbf{G}}^{\mathsf{H}}\hat{\mathbf{G}})^{-1})}}\hat{\mathbf{G}}(\hat{\mathbf{G}}^{\mathsf{H}}\hat{\mathbf{G}})^{-1} = [\mathbf{a}_1 \; \mathbf{a}_2 \; \ldots \; \mathbf{a}_K]$ such that $\mathbb{E}\{\|\mathbf{x}\|^2\} = 1$. The collective signals received at the user terminals is given by (2.7). Now, the signal received at each user k, can be written as

$$y_k = \sqrt{\rho_{\mathsf{dl}}}\sum_{i=1}^{K}\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_i q_i + w_k, \tag{2.53}$$

which can be equivalently written as

$$\begin{aligned} y_k =& \sqrt{\rho_{\mathsf{dl}}}\mathbb{E}\{\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_k\}q_k + \sqrt{\rho_{\mathsf{dl}}}(\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_k - \mathbb{E}\{\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_k\})q_k \\ &+ \sqrt{\rho_{\mathsf{dl}}}\sum_{i=1,i\neq k}^{K}\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_i q_i + w_k. \end{aligned} \tag{2.54}$$

The terms in (2.54) are uncorrelated, and thus using the use and forget bound technique [23, Ch. 2, Sec. 2.3.2], the signal-to-interference-plus-noise-ratio (SINR) at user k is given by

$$\mathsf{SINR}_k = \frac{\rho_{\mathsf{dl}}|\mathbb{E}\{\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_k\}|^2}{\sigma^2 + \rho_{\mathsf{dl}}\sum_{i=1,i\neq k}^{K}\mathbb{E}\{|\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_i|^2\} + \rho_{\mathsf{dl}}\mathsf{var}\{\mathbf{g}_k^{\mathsf{H}}\mathbf{a}_k\}}. \tag{2.55}$$

In (2.55), all the randomness is in the channel estimation errors and the ergodicity is with respect to the channel estimation errors. Thus, the achievable rate for user k is given by

$$R_k = \left(1 - \frac{\tau_p}{\tau_c}\right)\log(1 + \mathsf{SINR}_k). \tag{2.56}$$

Table 2.8: Simulation Parameters

Frequency of operation, f	2 GHz
Wavelength, λ	15 cm
Room size	40 m x 40 m x 10 m
Number of transmit antennas, M	512
Number of users, K	200
Mobility	Static
Noise figure, F	9 dB
Bandwidth, B	20 MHz
Noise power, σ^2	-92 dBm
Dielectric constant, ε_r	10.2
Height of patch antenna, h	0.1588 cm

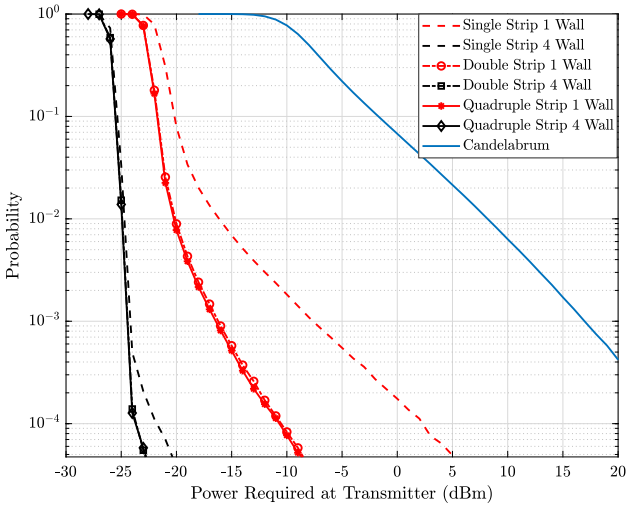


Figure 2.15: Probability of required transmit power to achieve minimum spectral efficiency of 4 bit/s/Hz per user.

Case Study

As an example of a simple scenario that can be modeled in the general framework, we consider a simulation setup consisting of an indoor room of size $40 \times 40 \times 10$ m^3. The simulation parameters are summarized in Table 2.8 with all the UEs are assumed to be static. In this study, radio stripes over the walls are considered and the antenna spacing is adjusted so as to have 512 antennas are spreaded uniformly. For example, for single stripe over 1 wall, 512 antennas are placed at $\lambda/2$ spacing, while for single stripe over 4 walls the antenna spacing is 2λ. Thus quadruple stripe over 4 walls will have an antenna spacing of 8λ.

Simulations show that the total downlink power required when we deploy RadioWeave technology is very low compared to the co-located deployment when zero-forcing precoding is used at the base station. As the channel estimation quality depends on the uplink power, for the comparison of required downlink power among different topologies, we consider an uplink power of 0dBm, such that the channel estimation errors are negligible. From (2.51), we can see that the required transmit power P depends on the channel matrix $\hat{\mathbf{G}}$ and hence is random depending on user locations. For simulations, we consider 10^6 random user locations and as a metric for comparison,

we plot the probability of required transmit power, i.e., Pr$\{P \geq$ power$\}$ to achieve a minimum downlink spectral efficiency requirement of 4 bit/s/Hz per user. Fig. 2.15 shows the probability of required transmit power for different antenna deployment topologies keeping the total number of transmit antennas the same. It can be seen that the power required in the co-located deployment is higher compared to all the RadioWeaves deployment scenarios considered in the paper. From Fig. 2.4, it can be observed that not all antennas will be able to serve each user, as some antennas point in the wrong direction. Hence, in the co-located case, we are not able to achieve the full multiple antenna power gain. The transmit power in the RadioWeaves deployment topologies considered is lower since the users can be spatially separated as all the transmit antennas can see all users, thus achieving a higher power gain from coherent transmission. Also, we observe that, if the antennas are divided over all four walls, the required transmit power is much less compared to keeping all the antennas on the same wall. This means that the capability of the antenna array to spatially resolve each user improves as the antennas are spread over all the walls. Thus, for a RadioWeaves deployment, distributing the antennas over all the walls of the room is preferred. Also, the power savings or spatial resolution capability of the array is negligible when the array is deployed vertically over the walls. From the plot, it can be seen that RadioWeaves with deploying antennas over all the four walls of the indoor building can provide a minimum downlink rate of 4 bit/s/Hz for 200 users with a total required transmit power of -23 dBm.

In the antenna array literature, it is known that the larger than half the wavelength antenna element spacing deployments are not desirable due to the grating lobes. This is true when we consider an equal aperture array and compare the performance between antenna elements spaced half a wavelength and antenna elements spaced more than half a wavelength by removing antenna elements. In the latter case, we get grating lobes which reduces the spatial resolution of users. However, if we put the removed antenna elements at another location or on a different wall, thereby allowing a larger array, we get an improvement in spatial resolution that outweigh the losses caused by the grating lobes. As the antenna separation is increased, the total angular window covered when pointing the main lobe into a particular direction remains more or less unchanged even though the lobes get narrower and more grating lobes appear. Thus when all the antenna elements are used and deployed all around the indoor space, the grating lobes have no effect whatsoever on the performance.

Moreover, we show that with low power levels high rates are achievable with RadioWeaves technology. Fig. 2.16 shows the 99.9% likely rates users can achieve with different uplink and downlink power levels. From (2.25), the channel estimates depend on the uplink power. At low uplink power levels, the channel estimates are not good and the downlink rate reduces. As the uplink power is increased, first we see a gradual improvement in rates after which the channel estimates quality becomes close to perfect channel state and the rate saturates. As the downlink power is increased, the rates increases gradually. This simulations support the assumption that the RadioWeave deployment can provide high rates to the users by spending very little power.

2.6 Summary

To summarize, in this chapter, we described the performance evaluation framework for RadioWeaves. The framework supports different channel models and antenna gain patterns. It also supports different antenna deployment configurations including ULA and UPA. These antenna configurations can be placed on a single wall or they can be spread on all four walls. Furthermore, the framework supports different channel estimators such as LS and MMSE and

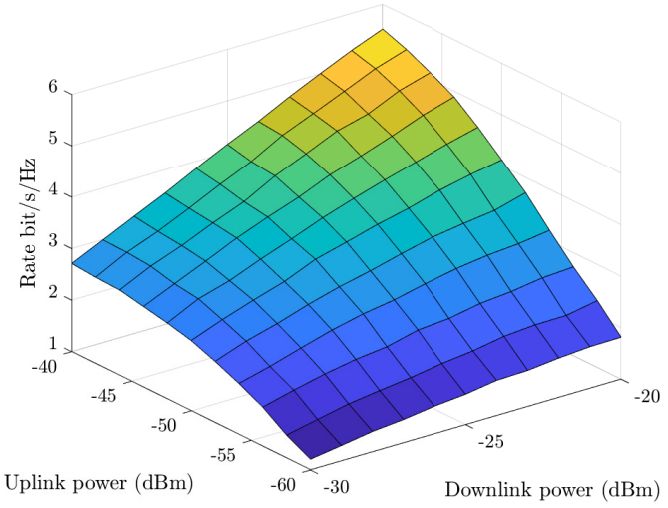


Figure 2.16: 99.9% likely achievable rate per user in a “double strip on 4 walls” configuration under different uplink downlink power levels.

precoders such as ZF and MR. Finally, we showed that the presented performance mapping model enables the prediction of the RadioWeaves physical layer performance in a computationally efficient manner with high accuracy. It can especially be used to speed up system-level simulations of RadioWeaves. We showed that the EESM can achieve a good trade-off between the PER prediction accuracy and the computational complexity for performance evaluation of RadioWeaves. Our simulations also showed that EESM performs well for different MCSs. The study has also shown that the calibration parameter β is mainly dependent on the MCS used and that once β is obtained for a particular MCS, it can be used across a wide range of scenarios with different parameters such as the number of antennas and the number of subcarriers. We have also shown that performance prediction is accurate even for extreme SINR profiles, which confirms the robustness of the physical-layer abstraction model studied.

Chapter 3

Performance Comparison of Centralized vs. Decentralized Algorithms

3.1 Introduction

In this chapter, the existing combining/precoding algorithms are compared in terms of latency and computational complexity. Also, the most common topologies that are considered in the literature are presented.

Note that all the fronthaul signaling and computational complexity calculation of the combining/precoding vectors are valid only for one coherence block which is a frequency/time grid over which the channel vector of a particular User Equipment (UE) to the antennas remains constant. In one coherence block which has τ_c samples, τ_p samples are assigned to pilot training, τ_u and τ_d samples for uplink and downlink transmission, respectively. Furthermore, as most of the existing methods consider Contact Service Points (CSPs) with more than one antenna, assume that we have L CSPs, each having N antennas (Total number of antennas in the network is $M = N \times L$) and serving K UEs in the rest of this chapter.

3.1.1 Centralized vs. distributed processing

The existing algorithms can be mainly categorized into two main groups: distributed vs. centralized.

- In centralized processing, all the computation for combining/precoding is done in a PE, and the CSPs act as relays, sending the pilot and data signal to the PE/UEs. Even though great performance improvement can be achieved using centralized processing, the amount of data to be exchanged between different nodes (front haul signaling) as well as a huge number of computations lower the motivation of using these algorithms in real-life networks. Most of the centralized algorithms are implemented using the star-like architecture shown in Fig. 3.1b.
- In distributed processing, the main objective is to lower the fronthaul signaling as well as computation by using smarter CSPs that can contribute to the processing locally, of course without compromising performance much.

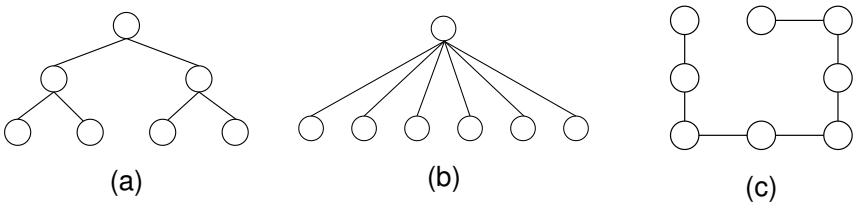


Figure 3.1: Different connection topologies for cell-free networks. Topology (a) is a tree with depth two. Topology (b) is star topology is tree with depth 1. Topology (c) is a tree with depth L.

3.1.2 Network topology

In many existing works on centralized and distributed processing, the connection between different entities can be modeled as a tree. For example, the topology considered in [1], [2], is a star topology which is a tree with the depth of one. On the other hand, the radio stripe sequential processing considered in [3] is a tree with depth L. The larger the depth of the tree, the larger is the delay of propagating information between nodes which are CSPs and PEs in our case. Different topologies are depicted in Fig. 3.1.

In Fig. 3.1, we do not specify the role of the nodes as the aim is to show the possible topologies considered in the literature. However, root nodes are usually processing nodes and the nodes at the branches are the CSPs which can be a sophisticated radio unit, able to do combining/precoding in the uplink/downlink, or a simple radio unit receiving and sending data to the PE and UEs.

The advantage of a tree topology with a smaller depth is the smaller delay imposed on the signal processing of UEs both in uplink and downlink. However, the number of the links connected to the PE is large as well as the rate of the data entering or exiting PE in the uplink and downlink. On the other hand, a tree with a large depth, such as sequential topology in Fig. 3.1c, imposes a large delay because of sequential fronthaul, but it requires just one connection to the PE. Also, generally, tree topologies have the disadvantage of low fault tolerance, in case of a broken link, one or multiple CSPs will be disconnected from the rest of the network. It is worth mentioning that authors in [4] consider a closed daisy chain topology rather than a tree topology between entities which is more fault-tolerant in comparison with the other methods mentioned.

In networks with single-antenna CSPs, the star topology requires the PE to handle a lot of connections at the same time. Also, the pure sequential topology may not be the best solution as some services are delay-sensitive. The reasonable topology can be a mixture of star and sequential topology, as it is shown in Fig. 3.1a. In the next section, some of the existing works are elaborated on and compared in terms of computational cost as well as fronthaul signaling.

3.2 Centralized processing

Most of the works on centralized processing consider a star topology [1], [2].

3.2.1 Centralized MMSE combining/precoding (CMMSE) in [1]

In [1], the authors consider a centralized algorithm for uplink combining/downlink precoding using the minimum mean square error method (MMSE) in the PE. Each CSP first estimates the local channel vectors to all the UEs and then sends the channel estimates to the PE at the end of the

pilot training phase. The PE calculates the combining/precoding vectors based on the received CSI and uses them in the uplink detection/downlink precoding. The MMSE combining matrix (with its columns be the combining vector of the corresponding UEs) calculated in the PE is formulated as follows:

$$\mathbf{V}^{\mathrm{MMSE}} = (\hat{\mathbf{G}}\hat{\mathbf{G}}^{\mathsf{H}} + \sum_{i=1}^{K} \mathbf{C}_i + \frac{\sigma^2}{\rho_{\mathsf{ul}}}\mathbf{I}_M)^{-1}\hat{\mathbf{G}}, \tag{3.1}$$

Where $\mathbf{V}^{\mathrm{MMSE}} \in C^{M\times K}$ and $\mathbf{C}_i$ is the correlation matrix of UE i's channel estimation error. Specifically

$$\tilde{\mathbf{G}} = \mathbf{G} - \hat{\mathbf{G}}, \tag{3.2}$$

$$\mathbb{E}\{[\tilde{\mathbf{G}}]_{:,i}[\tilde{\mathbf{G}}]_{:,i}^{\mathsf{H}}\} = \mathbf{C}_i, \tag{3.3}$$

$[\tilde{\mathbf{G}}]_{:,i}$ is the i^{th} column of matrix $\tilde{\mathbf{G}}$. $\mathbf{C}_i$ is usually a block diagonal matrix, in which each block is the channel estimation error of UE i to the corresponding CSP's antennas.

Fronthaul signaling

Considering a star topology, assume that the links connecting the CSPs have a bandwidth of R $\frac{\text{real scalars}}{\text{second}}$.

- Uplink: To be more specific about the number of real scalars that need to be sent to the PE in the uplink, we consider pilot training and uplink data transmission separately. Then the total amount of delay caused by fronthaul signaling is calculated.
  - After pilot training by the UEs in the uplink, the CSPs estimate the local channel vectors of the UEs and send their local estimates to the PE, which is in total $2MK$ real scalars (MK complex scalars).
  - After reception of the UEs' data signals, the CSPs send the received signal vectors to the PE for every sample that is allocated to the uplink transmission which is $2M\tau_u$ real scalars in total.

  So in total, the total number of real scalars to be transmitted between PE and CSPs in the uplink is $2MK + 2M\tau_u$. As the CSPs send these scalars in parallel in a star topology, the delay associated with these transmissions will be $\frac{2MK+2M\tau_u}{LR}$ in a star topology.
- Downlink: In the downlink, the PE can first calculate the precoded signal vectors that should be transmitted by every CSP to the UEs, and then send the precoded vectors to the corresponding CSPs. Therefore, the total number of real scalars to be transmitted from the PE to the CSPs is $2M\tau_d$. So, the delay caused by this transmission will be $\frac{2M\tau_d}{LR}$ in a star topology.

In both directions, fronthaul signaling scales linearly with the number of antennas which can be large in a dense network. So such a topology is out of the question for ultra-dense practical implementations.

Computational complexity

The computational complexity of combining vectors is mainly dominated by the $M \times M$ matrix inversion which has a complexity order of $\mathcal{O}(M^3)$.

3.2.2 Parallel Interference Cancellation (PIC) combining/precoding in [2]

In [2], the authors considered a different approach than MMSE to come up with combining/precoding vectors for the UEs. To avoid the large matrix inversion, their method is based on the simple maximum ratio combining method. For any particular UE k, the combining/precoding vector is calculated as follows:

$$\mathbf{v}_k^{\mathrm{PIC}} = [\hat{\mathbf{G}}]_{:,k} - \sum_{i \neq k} \frac{\mathsf{SINR}_i^{\mathsf{inst,MR}}}{1 + \mathsf{SINR}_i^{\mathsf{inst,MR}}} \frac{[\hat{\mathbf{G}}]_{:,k}^{\mathsf{H}} [\hat{\mathbf{G}}]_{:,i}}{[\hat{\mathbf{G}}]_{:,i}^{\mathsf{H}} [\hat{\mathbf{G}}]_{:,i}} [\hat{\mathbf{G}}]_{:,i} \tag{3.4}$$

The effect of the other interfering UE such as UE i on combining/precoding vector direction of UE k depends on two aspects:

- $\mathsf{SINR}_i^{\mathsf{inst,MR}}$ which is the instantaneous signal to interference plus noise ratio of UE i when using maximum ratio combining:

$$\mathsf{SINR}_i^{\mathsf{inst,MR}} = \frac{|[\hat{\mathbf{G}}]_{:,i}^{\mathsf{H}} [\hat{\mathbf{G}}]_{:,i}|^2}{\sum_{j \neq i} |[\hat{\mathbf{G}}]_{:,i}^{\mathsf{H}} [\hat{\mathbf{G}}]_{:,j}|^2 + \frac{\sigma^2}{\rho_{\mathsf{ul}}} [\hat{\mathbf{G}}]_{:,i}^{\mathsf{H}} [\hat{\mathbf{G}}]_{:,i}} \tag{3.5}$$

- The similarity of the estimated channel vector of UE i to the estimated channel vector of UE k: the bigger is their similarity (inner product), the more weight is given to the estimated channel vector of UE i in the combining vector of UE k.

Fronthaul signaling

In [2], the amount of signaling between the PE and the CSPs and consequently the associated delay are same as in the case of the centralized approach in [1], so it is omitted here.

Computational complexity

In [2], the computation complexity of computing combining vectors is $\mathcal{O}(KM)$ which is way less than in the case of the centralized MMSE approach in [1].

3.3 Distributed processing

Most distributed processing algorithms in the literature consider a star or sequential topology which has high fronthaul signaling/connections or large fronthaul delay respectively.

3.3.1 Local MMSE (LMMSE) combining/precoding in [1]

The LMMSE combining/precoding scheme proposed in [1] assumes that the CSPs calculate the local combining/precoding vectors using the MMSE approach locally. In the uplink, after the pilot training phase, the CSPs estimate the local channels of UEs. Then, based on the local CSI, they calculate a local combining vector using the MMSE method for each of the UEs. This algorithm is implemented using a star topology but it can also be implemented using other types of fronthaul topologies. The local combining vectors of the UEs at CSP l is calculated as:

$$\mathbf{V}_l^{\mathrm{LMMSE}} = (\hat{\mathbf{G}}_l \hat{\mathbf{G}}_l^{\mathsf{H}} + \sum_{i=1}^{K} \mathbf{C}_{il} + \frac{\sigma^2}{\rho_{\mathsf{ul}}} \mathbf{I}_N)^{-1} \hat{\mathbf{G}}_l \tag{3.6}$$

where $\hat{\mathbf{G}}_l$ is composed of the rows of matrix $\hat{\mathbf{G}}$ which correspond to the antennas of CSP l and $\mathbf{C}_{il}$ is the l^{th} diagonal block of matrix $\mathbf{C}_i$ which has been defined in equation 3.3.

Fronthaul signaling

- Uplink: using these local combining vectors, CSPs estimate the signal of the UEs locally. Then, they send the local estimates of the UEs' signal to the PE, in which the local estimates of a particular UE will be combined, e.g, via weighted averaging, to come up with the global estimate of the UE's signal. Therefore, unlike in the centralized processing approach in which the CSPs send the local channel estimates as well as the received vectors to the PEs, in the local MMSE approach, the CSPs send only the local estimates of the UEs' signal to the PE in the uplink which is $2KL\tau_u$ real scalars in total. Assuming the same link rate in fronthaul links, the delay imposed because of fronthaul signaling is $\frac{2K\tau_u}{R}$.
- Downlink: in the downlink, the PE only sends the UEs' signals to the CSPs which are in total $2KL\tau_d$ real scalars, causing the delay of $\frac{2K\tau_d}{R}$ seconds, and then, the CSPs precode the UEs' signal using their local precoding vectors. While in the centralized methods, PE could precode and then send the vectors to the CSPs that could be immediately transmitted to the UEs. Consequently, the fronthaul signaling from the PE to the CSPs in the downlink may be lower in centralized schemes when L is large or, in other words, antennas are more distributed. Therefore, using LMMSE when having completely distributed antennas, e.g, $L = M$, makes the fronthaul requirements of the LMMSE method more demanding than the centralized MMSE method in the downlink.

Computational complexity

Similar to the centralized MMSE approach described in the previous section, the computational complexity associated with LMMSE combining/precoding vector computation is mainly caused by the matrix inversion and is in total of the order $\mathcal{O}(LN^3)$.

3.3.2 Sequential combining/precoding using Radio Stripe (SRS) in [3]

Another topology considered in distributed processing is radio stripe which is a tree topology with depth L. In [3], the authors proposed a distributed processing algorithm in the uplink assuming that the antennas are connected via a radio stripe. CSP $l-1$ computes the local combining vector for each UE based on the local CSI and the effective scalar channel information it receives from the previous CSP. Then it sends this local information to CSP l and this procedure goes on for all L CSP. For the sake of simplicity, only the algorithm describing the method in [3] is shown in algorithm 1. Note that the correlation matrix of interference plus noise at CSP l, $\mathbf{\Sigma}_l$, is:

$$\mathbf{\Sigma}_l = \rho_{\text{ul}} \sum_{i=1}^{K} \mathbf{C}_i + \sigma^2 \mathbf{I}_N. \tag{3.7}$$

In fact, in the radio stripe sequence, CSP $l-1$ will act as an additional antenna for the CSP l, when computing the combining vector of the UEs in CSP l.

Algorithm 1 Sequential processing using radio stripe [3].

Compute normalized LMMSE combining vector at CSP 1,$\mathbf{V}_1 \in C^{N\times K}$ given $\hat{\mathbf{G}}_1$ and $\mathbf{\Sigma}_1$
Estimate UEs' signal locally, $\hat{\mathbf{s}}_1 = \mathbf{V}_1^{\mathsf{H}}\mathbf{y}_1$ and initialize $\hat{\mathbf{G}}'_{k1} = \hat{\mathbf{G}}_1$, $\mathbf{\Sigma}'_{k1} = \mathbf{\Sigma}_1$, $\forall k \in \{1, 2, \ldots, K\}$
for $l = 2 : L$ **do**
    for $k = 1 : K$ **do**
        Compute LMMSE combining vector $[\mathbf{V}_l]_{:,k}$ for UE k given augmented
        channel estimates, $\hat{\mathbf{G}}'_{kl} = [\hat{\mathbf{G}}'^{\mathbf{H}}_{k(l-1)}[\mathbf{V}_{(l-1)}]_{:,k}, \hat{\mathbf{G}}_l^{\mathsf{H}}]^{\mathsf{H}}$,
        and augmented noise covariance matrix, $\mathbf{\Sigma}'_{kl} = \mathrm{diag}([\mathbf{V}_{(l-1)}]^{\mathsf{H}}_{:,k}\mathbf{\Sigma}'_{k(l-1)}[\mathbf{V}_{(l-1)}]_{:,k}, \mathbf{\Sigma}_l)$
        Compute $\hat{s}_{kl} = [\mathbf{V}_l]^{\mathsf{H}}_{:,k}[\hat{s}_{k(l-1)}\mathbf{y}_l^{\mathsf{T}}]^{\mathsf{T}}$
    end
end

Fronthaul signaling

- Uplink: Besides the local estimates of UEs' signal which are $2K\tau_u$ real scalars, CSP $l-1$ sends $2K^2$ real scalars as effective channel estimates and K real scalars as the statistics of the effective channels, to CSP l in one coherence block. In total, the amount of data transferred between the CSPs and form CSP L to the PE add up to $(2K^2 + K + 2K\tau_u)L$. As the signaling between the CSPs is sequential, the delay imposed by fronthaul signaling will be $L\frac{(2K^2+K+2K\tau_u)}{R}$.
  This algorithm reduces the amount of data to be transmitted to the PE to $3K^2+2K\tau_u$ which is way less than the alternative centralized algorithm described in previous sections.
- Downlink: The authors did not specifically consider the downlink, but because of channel reciprocity, each CSP can use their local combining vector to locally precode the UEs' data. Only the PE should send the intended signal for the UEs to the CSP L and then each CSP should communicate them to the next CSP in the sequence which requires transmitting of $2KL\tau_d$ real scalars in total, equal to the LMMSE algorithm in [1]. But because of the sequential nature of the fronthaul, the imposed delay due to downlink fronthaul signaling will be $L\frac{2K\tau_d}{R}$, which is L times larger than the corresponding delay in the LMMSE algorithm [1].

Computational complexity

Each CSP uses a local normalized MMSE approach to compute the combining vectors of UEs. For one particular UE, the complexity of computing the combining vector in one particular CSP is $\mathcal{O}(N^3)$, which is $\mathcal{O}(LN^3)$ in total for all L CSPs. However, unlike the centralized and local method proposed in [1], the computation of one particular UE combining vector is not reusable for other UEs due to the use of different effective scalar channels.

3.3.3 Decentralized massive MIMO (DmMIMO) combining/precoding using Daisy chain architecture [4]

In [4], the authors proposed a decentralized approach for uplink detection/downlink precoding with minimal intervention of a PE. They start with the least square (LS) error formulation and state that solving the LS problem using gradient descent (GD) is not possible in a completely decentralized way and therefore, a PE should be appointed for some matrix operations. Thus, they tried to

approximate the gradient decent solution in a decentralized way. CSP l contributes sequentially in the calculation of the gradient function by estimating the UEs' data locally and sends one $K \times K$ matrix to the next CSP $l+1$, so it can compute/update UEs' local combining/precoding vectors based on that. The off-diagonal elements of this matrix that get updated in each CSP, show the amount of interference between any two UEs and these off-diagonal elements get smaller each time the matrix is updated in a CSP. Although [4] considered the sequential processing to happen among the antennas of a CSP with a large number of antennas, it can be generalized to multiple CSP case. The algorithm of updating the combining/precoding vector in the CSPs are shown in Algorithm 2. Note that for the sequential procedure to happen in one CSP with multiple antennas,

Algorithm 2 Decentralized GD based combining/precoding vector [4].

1: Input $\hat{\mathbf{G}}$
2: Initialize $\mathbf{A}_0 = \mathbf{I}_K$
3: **for** $l = 1 : L$ **do**
4: $\quad \mathbf{V}_l = \mu_l \hat{\mathbf{G}}_l \mathbf{A}_{l-1}$
5: $\quad \mathbf{A}_l = \mathbf{A}_{l-1} - \hat{\mathbf{G}}_l^{\mathsf{H}} \mathbf{V}_l$
6: end

the iteration in the algorithm will be over the antennas of the CSP and therefore, matrix $\mathbf{G}_l$ will be replaced by a vector, representing the channel estimates of the users to one antenna. The algorithm can have multiple iterations over the CSPs (closed-loop) which will result in more inter-UE interference cancellation (smaller off-diagonal elements in matrix $\mathbf{A}$). It is worth mentioning that, in the uplink, instead of sending a $K \times K$ matrix, the CSPs can only communicate their local estimates of the UEs' signal vector and keep updating only this signal vector from AP 1 until CSP L which can announce the final estimates. However, as the CSPs also need to know the vectors for downlink precoding, the computation of combining vectors is only possible by communicating matrix $\mathbf{A}$.

Fronthaul signaling

- Uplink: In the uplink, each CSP sends a $K \times K$ symmetric matrix as well as the local UEs' signal estimates to the next CSP in the sequence, which is in total $K^2 + 2K\tau_u$ real scalars. This happens between any two CSPs and the last CSP and the PE. So in total, the fronthaul signaling in the uplink would be $(K^2 + 2K\tau_u)L$ and the delay imposed due to uplink fronthaul signaling is $L\frac{K^2+2K\tau_u}{R}$.
- Downlink: In the downlink, the PE sends the UEs' signal to CSP L and then each CSP sends the signal vector to the next CSP. Also, each CSP precodes the data by the UEs' local precoding vectors and sends the precoded data to the UEs. So in every chunk of the chain, $2K\tau_d$ real scalars are transmitted which sum up to $2KL\tau_d$ in total. The imposed delay is $L\frac{2K\tau_d}{R}$.

Computational complexity

The computational complexity of the distributed sequential algorithm 2 is due to the multiplication of two $K \times K$ matrices which is of order $\mathcal{O}(K^3)$.

3.4 Summary

In this chapter, some existing centralized/distributed algorithms and fronthaul topologies for uplink combining/downlink precoding in massive MIMO networks are reviewed. It is observed that as opposed to the star topology, the sequential fronthaul topology reduces the number of connections as well as the amount of fronthaul signaling to the PE at the cost of increased delay in UEs' signal detection. Moreover, the mentioned distributed algorithms aim to trade performance for reduced computational complexity and fronthaul signalling, when compared to the centralized MMSE approach [1]. In Table 3.1, the fronthaul signaling of different combining/precoding methods are compared.

Method	Topology	Uplink FS	Downlink FS
CMMSE [1]	star	$2MK + 2M\tau_u$	$2M\tau_d$
PIC [2]	star	$2MK + 2M\tau_u$	$2M\tau_d$
LMMSE [1]	star	$2KL\tau_u$	$2KL\tau_d$
SRS [3]	sequential	$(2K^2 + K + 2K\tau_u)L$	-
DmMIMO [4]	sequential (closed)	$(K^2 + 2K\tau_u)L$	$2KL\tau_d$

Table 3.1: Fronthaul signaling comparison of different combining/precoding methods using different fronthaul topologies.

Chapter 4

User and Computational Resource Scheduling

Algorithms used to deliver services to users/devices in RadioWeaves need to adapt to changes both in user traffic and the computational resources available across the deployed RadioWeaves infrastructure. While scheduling of communication resources between users is a well known topic, the simultaneous scheduling of computational resources for the associated (distributed) processing algorithms across an infrastructure of the RadioWeaves type is an entirely new area of research.

User and computational resource scheduling for RadioWeaves is in its general form an optimization problem, where energy consumption of the entire RadioWeaves infrastructure should be minimized under a set of restrictions, such as service requirements, available distributed algorithm types, available computational resources, and the topology of the infrastructure. While this optimization problem remains to be more precisely formulated and addressed in the REINDEER project, it is expected that sub-optimal ad-hoc solutions will be employed in any practical setting.

Development of distributed algorithms for RadioWeaves services should, for the above reasons, be done with energy consumption in mind and under the awareness of basic mechanisms governing the scheduling of computational resources for their execution. Below we present a first set of measures that influence energy consumption when executing distributed algorithms and high-level observations about the scheduling challenges, which the processing algorithms themselves are subject to. Algorithms should be designed so that their computational resources can be re-scheduled without excessive cost or interruption of the services they are providing.

4.1 Measures That Influence Energy Consumption

Table 4.1 lists the high-level power consumption model of different hardware components in RadioWeave systems, including analog processing, digital processing, memory systems, interconnections, and synchronization systems. The power consumption model is a mixture of configurable design parameters and results extracted from implementation examples.

The choice of design parameters depends on many aspects. For instance, the PA output power, the resolution and sampling rate of data converters depend on the system setup (e.g., number of antennas, system bandwidth), the use cases (e.g., number of users, the required quality of service), and the channel environments. The general trend is that we can use less powerful

Table 4.1: Hardware components in RadioWeave and power consumption model.

Analog processing	
Power amplifier	(P_{in}+P_{out}/η W) $\times$ # transmitters[a] [39]
ADC	($\alpha 2^b f$W) $\times$ # digital paths[b] [40]
DAC	($\beta 2^b f$ W)$\times$ # digital paths[c]
Digital computation and storage	
Operation (multiplier/MAC)	3.1pJ $\times$ # operation per second (32bit mult., 45nm) [41]
On-chip SRAM memory access	5pJ $\times$ # SRAM accesses per second (32bit for 32KB SRAM) [41]
Off-chip DRAM memory access	640pJ $\times$ # DRAM accesses per second (32bit) [41]
Data transfer and interconnection	
Local links	5.34pJ/bit $\times$ # bps (SerDes, 65nm) [42]
Back-plane	45pJ/bit $\times$ # bps (100G Ethernet, 65nm, PHY only, no switch/router)[d] [43]
	5.2nJ/bit $\times$ #bps (100G Ethernet, switch) [44]
Synchronization	
Clock distribution	2.2W $\times$ # white rabbit nodes [45] (Can be partially merged with back-plane)

[a] P_{in} is the PA input power, P_{out} is the PA output power, η is the PA efficiency, e.g., 0.15-0.25 for class AB.
[b] α is ADC FoM factor (fJ/conv-step), e.g., 165 for 28nm 10bit, 5GS/s ADC; 44 for 28nm 600MS/s, 12bit ADC, b is effective bit resolution, f is conversion rate.
[c] β is DAC FoM factor, b is effective bit resolution, f is conversion rate.
[d] Similar results in the power consumption (1-3W) of QSFP module for 100G Ethernet.

PAs and lower resolution data converters when increasing the number of antennas. The power on computation and memory access highly depend on the processing algorithm selected, the architecture (e.g., topology) of the RadioWeave, as well as the frame structure. The power consumption of data transfer between different hardware units inside RadioWeave can be significant, especially when the units are distributed over long distances. The data rate on the interconnections will be decided by the RadioWeave architecture and how the processing is distributed. The synchronization system power consumption will depend on the frequency band the system is operating at, the geometrical distribution of RadioWeaves, and also the system-level requirement on synchronization.

4.2 Scheduling of computational resources for distributed algorithms

Distributed execution of algorithms on the RadioWeaves infrastructure, in terms of which computational resources are allocated and how processing data is exchanged between computational nodes (typically contact service points/CSPs, as defined in D2.1), will depend on the choice of RadioWeaves topology. The topology choice may differ between different deployments and what services a certain deployment is expected to deliver. For the most critical services, where we have strict requirements on latency, reliability and/or a high density of devices served, mesh-like topologies may be required, while for less critical services cost benefits of topologies with fewer connections between nodes with computational resources may suffice. On top of this, the federations formed to deliver different services may be dynamic in several ways. Computational load on the CSPs forming a federation may vary over time, depending on traffic variations inside the federation's coverage area. This is illustrated in Fig. 4.1, where served users move inside the coverage area of a fixed federation of CSPs, changing the traffic and computational resource requirement on individual CSPs in the federation. The federation may also move spatially, following served users as they move in the environment, in the sense that new CSPs are included and

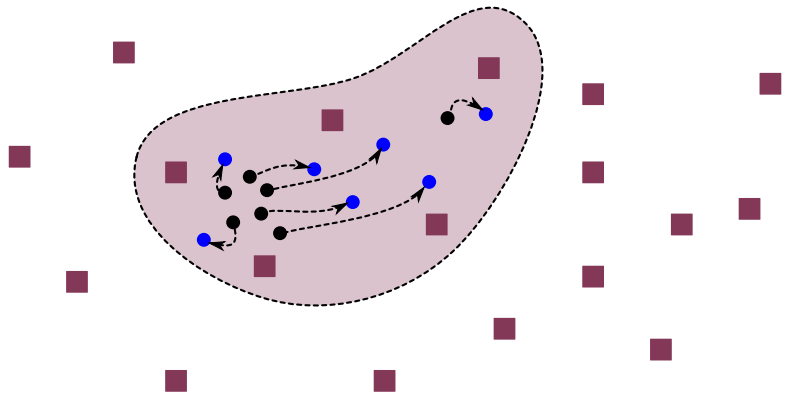


Figure 4.1: Illustration of RadioWeaves CSPs where a shift in user positions (from black to blue dots) influences the computational resources assigned to distributed algorithms at the different CSPs in a federation (light maroon). RadioWeaves CSPs shown as maroon squares.

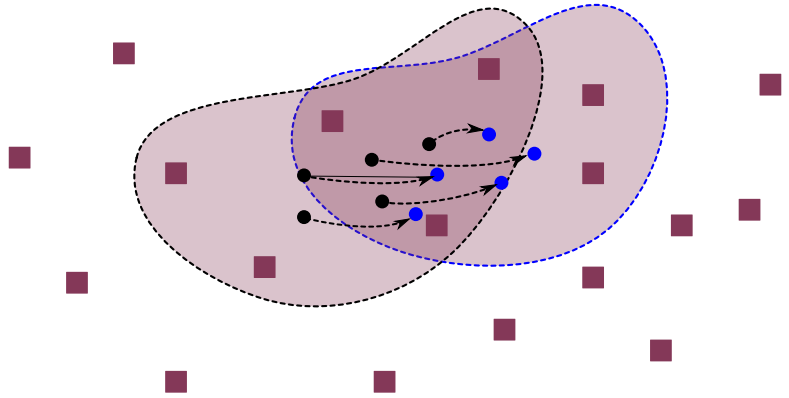


Figure 4.2: Illustration of RadioWeaves CSPs where a shift in user positions (from black to blue dots) results in a move of the federation of CSPs (light maroon with black to blue dashed stroke) serving these users. RadioWeaves CSPs shown as maroon squares.

others removed from the serving federation. This is illustrated in Fig. 4.2, where a group of users move (from black to blue positions), requiring the federation of CSPs to ”move along” to deliver the service. These variations will have to be controlled and computational resources scheduled dynamically in time, adapting to these changing conditions.

An initial discussion about the concept of federations and different topology choices is found in deliverable D2.1, while the detailed analysis is still an ongoing task for the rest of the REINDEER project. There are, however, a number of high-level observations of general interest when developing distributed algorithms for RadioWeaves, all depending on the chosen RadioWeaves topology:

- Latency is introduced when exchanging data between computational nodes, which, through service requirements, limit the number of nodes that can be used for executing distributed algorithms.
- Energy consumption of the RadioWeaves infrastructure depends not only on the amount of operations executed, but also to a high degree on the amount of processing data exchanged between computational nodes and the number of hops data has to traverse reaching its destination.
- Re-scheduling of computational resources assigned to algorithms will be necessary when

traffic characteristics change or federations move. Algorithms should be able to handle this re-scheduling, without excessive cost or interruption of services.

The above observations and related concerns are of particular interest when service requirements include low latency and/or high reliability in combination with mobility.

Chapter 5

Efficient Algorithm Candidates for Resource Allocation

5.1 Uplink D-MIMO Processing Using Kalman Filtering

Four different levels of D-MIMO (distributed multiple input multiple output) uplink processing cooperation, from a fully centralized method to locally distributed methods, were introduced in [1] and [10]. The fully centralized approach, which is denoted level 4 implementation, uses global minimum mean-square error (MMSE) combining, and hence it provides the best performance in terms of spectral efficiency (SE) and equivalently SINR. However, level 4 implementation is extremely difficult to realize in practical implementations, due to a very high fronthaul connection requirement and computational complexity at the D-MIMO central processing unit (EPU). To reduce the computational complexity at the EPU, three alternative methods, denoted level 1-3, first process the signals at each Access Point (SP) based on local information, and then pass them to the EPU for final combining and decoding. Level 1-3 do reduce the fronthaul requirement, however, these methods still require centralized combining, which has an impact on the processing latency. The performance of level 1-3 reaches below 40% of the upper bound performance provided by level 4. To find the possibility to reach the upper bound performance, we propose to use a decentralized partial level-4 method by defining a subset of SPs for each UE. The subset can be defined from one SP to all SPs that are connected to a EPU. When the subset consists of only one SP, which we call sub-1, it turns D-MIMO network into the “small cell” network; when the subset consists of all SPs connected to the EPU, it is equivalent to the level 4. We show in this section that the subset method provides a scalable trade-off between complexity and performance. Moreover, we propose to use a Kalman filter when adding new SPs into the subset since the Kalman filter [46] implementation provides the combining vector that minimizes estimation error (MSE) at each step of incrementation, from a subset of one SP to a subset of all SPs. We also show that the performance with the subset consisting of all SPs is equivalent to the upper bound performance of a full centralized level 4 implementation.

The rest of the section is organized as follows, we introduce the D-MIMO network in Section 5.1.1. A brief description of the four uplink processing combining levels are provided in Section 5.3. The decentralized methods together with the Kalman filter implementation is described in Section 5.4. The implementation cost and requirements are discussed in Section 5.6.

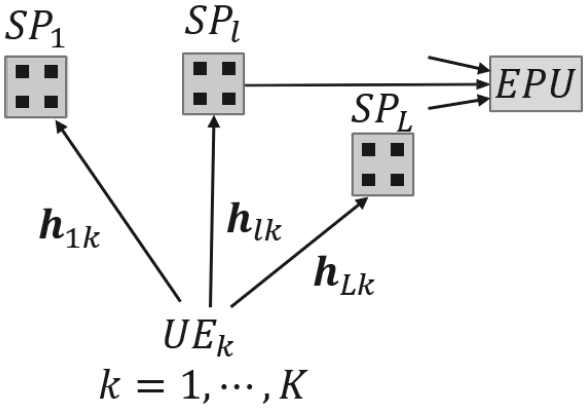


Figure 5.1: Uplink transmission from UE and to receiving SPs.

5.1.1 D-MIMO Network

A D-MIMO network consists of L geographically distributed SPs, each equipped with N antenna elements. The total number of antenna elements in the network is $N \times L$. The SPs are connected via fronthaul links to a EPU, which facilitates the coordination among SPs. The SPs cooperate to serve K UEs (User Equipments) in the coverage area jointly by coherent transmission in the downlink and coherent reception in the uplink, see Fig. 5.1.

The focus of this work is on the the uplink processing. In the uplink, as shown in Fig. 5.1, each SP receives two types of signals from UEs, the pilot signals for channel estimation and data signals for data transfer. When the UEs transmit their pilot signals, the received pilot signal $Z_l \in C^{N \times \tau_P}$ at SP l is

$$Z_l = \sum_{k=1}^{K} \sqrt{p_k}\, \mathbf{h}_{lk} \phi_{t_k}^{\mathsf{T}} + N_l, \tag{5.1}$$

where $p_k \geq 0$ is the transmit power of UE k, $N_l \in \mathbb{C}^{N \times \tau_P}$ is the receiver noise with independent $\mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ entries, and σ^2 is the noise power. The channel between UE k and SP l is denoted by $\mathbf{h}_{lk} \in \mathbb{C}^N$. The elements of the channel vector are modelled by a complex Gaussian distribution, $\mathbf{h}_{lk} \sim \mathcal{N}_{\mathbb{C}}(0, R_{lk})$ where $R_{lk} \in \mathbb{C}^{N \times N}$ is the spatial correlation matrix. Assume that the D-MIMO network has τ_P mutually orthogonal pilot signals $\phi_1, \ldots, \phi_{\tau_P}$ with $\|\phi_t\|^2 = \tau_P$ for $t = 1, \ldots, \tau_P$, which are used for channel estimation. The pilots are assigned to the UEs in a deterministic but arbitrary way. In a network with $K > \tau_P$ several users can be assigned to the same pilot. We denote the index of the pilot assigned to UE k as $t_k \in \{1, \ldots, \tau_P\}$ and $\mathcal{P}_k \subset \{1, \ldots, K\}$ the subset of UEs that use the same pilot as UE k. The interference generated by the pilot-sharing UEs causes pilot contamination which degrades the system performance. To estimate the channel, each SP correlates the received signal with the associated normalized pilot signal $\frac{\phi_{t_k}}{\sqrt{\tau_p}}$ to obtain

$$Z_{lt_k} \triangleq \frac{1}{\sqrt{\tau_p}} Z_l \phi_{t_k}^*. \tag{5.2}$$

Applying the MMSE to estimate the channel $\mathbf{h}_{lk}$, we get

$$\widehat{\mathbf{h}}_{lk} = \sqrt{p_k \tau_p} R_{lk} \psi_{lt_k}^{-1} Z_{lt_k} \tag{5.3}$$

where

$$\psi_{lt_k} = \mathbb{E}\left\{ Z_{lt_k} Z_{lt_k}^{\mathsf{H}} \right\} = \sum_{i \in \mathcal{P}_k} \tau_P p_i R_{li} + I_N \tag{5.4}$$

is the correlation matrix of the received signal. The estimate $\widehat{\mathbf{h}}_{lk}$ and estimation error $\widetilde{\mathbf{h}}_{lk} = \mathbf{h}_{lk} - \widehat{\mathbf{h}}_{lk}$ are independent vectors distributed as $\widehat{\mathbf{h}}_{lk} \sim \mathcal{N}_{\mathbb{C}}\left(0, p_k \tau_P R_{lk} \psi_{lt_k}^{-1} R_{lk}\right)$ and $\widetilde{\mathbf{h}}_{lk} \sim \mathcal{N}_{\mathbb{C}}\left(0, C_{lk}\right)$ with the covariance matrix

$$C_{lk} = \mathbb{E}\{\widetilde{\mathbf{h}}_{lk} \widetilde{\mathbf{h}}_{lk}^{\mathsf{H}}\} = R_{lk} - p_k \tau_P R_{lk} \psi_{lt_k}^{-1} R_{lk}. \tag{5.5}$$

During the uplink data transmission, all SPs will receive a superposition of the signals sent from all UEs. The received signal $\mathbf{y}_l \in \mathbb{C}^N$ at SP l can be written as

$$\mathbf{y}_l = \sum_{k=1}^{K} \mathbf{h}_{lk} s_k + \mathbf{n}_l, \tag{5.6}$$

where $s_k \sim \mathcal{N}_{\mathbb{C}}(0, p_k)$ is the signal transmitted by UE k with a power $p_k = \mathbb{E}\{|s_k|^2\}$ during the uplink data transmission. The independent receiver noise is $\mathbf{n}_l \sim \mathcal{N}_{\mathbb{C}}(0_N, \sigma^2 I_N)$.

The performance metrics used here is the achievable Spectral Efficiency (SE) [bit/s/Hz] based on the Shannon formula, which is the same as it was used in [1] and [10].

5.2 Decentralized sub-set combining for UL processing

We propose a decentralized processing and combining method that utilizes sub-sets of SPs for processing of uplink signals. Instead of sending all signals or estimated signals to the EPU to determine the combining vector, a subset of SPs is selected for each UE, $\mathcal{L} \subset \{1, \ldots, L\}$, and one of the SPs in the subset is assigned to be the aggregating SP for this UE. This enables the fronthaul communication to send information to different aggregation SPs, using different sections of the fronthaul, all at the same time. It further enables parallel processing of uplink signals from multiple UEs in different aggregating SPs. The processing for each UE is also significantly reduced when a sub-set of SPs is used for reception compared to the full set of SPs.

We denote the subset by sub-ℓ when ℓ out of L SPs are selected. The minimum subset, sub-1, can be selected based on the best path gain for each UE. The maximum subset, sub-L, consists of all SPs that are connected to the same EPU. As more SPs are selected for processing uplink signals from a UE, the better performance and the larger complexity in processing we obtain. There is a trade-off between the performance and complexity that need not be identical for each UE active in an uplink transmission time interval (TTI). In Fig. 5.2, a subset of $\ell = 3$ is used to illustrate the sub-3 selection. In this example, the subset for UE 1 is $\mathcal{L}_{\mathrm{UE}_1} = \{4, 5, 8\}$ where SP 5 is assigned to be the aggregating SP where antenna signals from other two SP 4 and SP 8 are collected. The subset for UE 2 is $\mathcal{L}_{\mathrm{UE}_2} = \{3, 6, 7\}$ where SP 6 is assigned to be the aggregating SP and the subset for UE 3 is $\mathcal{L}_{\mathrm{UE}_3} = \{8, 9, 11\}$ where SP 9 is assigned to be the aggregating SP.

In [47], the square-root implementation of the Kalman filter is proposed to find the optimal solution for mean square error. It is shown in [47] that such implementation provides the same result as the MMSE method used in [1] and [10]. However, the square-root Kalman filter is shown to be more efficient and numerically sound when inverting the covariance matrix.

To apply the Kalman filter, we rewrite the signal and the channel between all UEs and SP l in more compact vector or matrix forms in the following. The received data signal at SP l can be written as $\mathbf{y}_l = H_l \mathbf{s} + \mathbf{n}_l$, where the channel is $H_l = \left[\mathbf{h}_{l1} \cdots \mathbf{h}_{lK}\right]$, and transmitted signals from all UEs is the state vector $\mathbf{s}^{\mathsf{T}} = [s_1, \ldots, s_K]$. The Kalman filter provides the estimate of the state vector $\hat{\mathrm{s}}$ at SP l as

$$\hat{\mathbf{s}} = \hat{\mathbf{s}}_0 + \mathcal{K}\left(\mathbf{y}_l - H_l \hat{\mathbf{s}}_0\right) \tag{5.7}$$

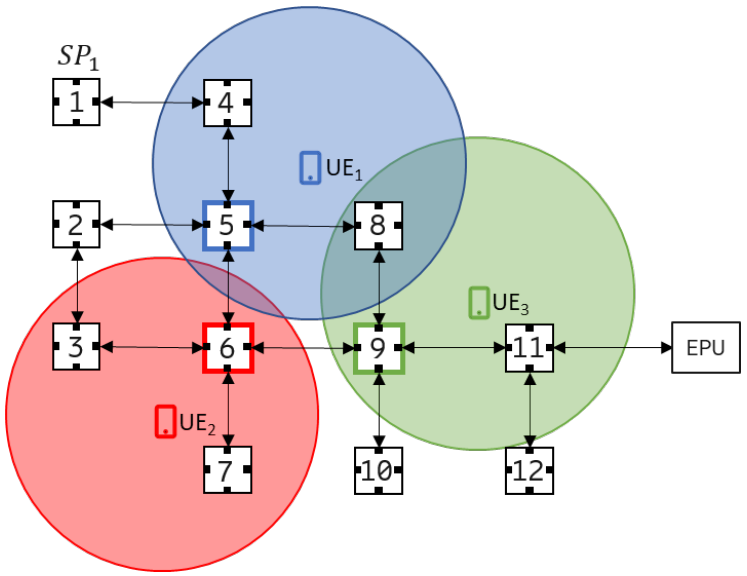


Figure 5.2: Subset that consists of three SPs for each UE.

which minimizes the mean-square error MSE $= \mathbb{E}\{|\mathbf{s} - \mathcal{K}\mathbf{y}_l|^2\}$ by the Kalman filter gain $\mathcal{K}$. The initial estimate state vector $\hat{\mathbf{s}}_0 = 0$ and the initial covariance matrix $P_0 = \text{diag}(p_1, \ldots, p_K)$.

The combining vector $\mathbf{v}_k$ given by (5.19) assumes that the channel H_l is unknown, but the estimate of the channel can be done locally at each SP based on known pilot signals transmitted from UEs. Let $\hat{H}_l = [\widehat{\mathbf{h}}_{l1}, \ldots, \widehat{\mathbf{h}}_{lK}]$ be the estimate of channel H_l, see [48] for a detailed description of channel estimation. Applying the estimation of the channel, the estimate of the state vector (5.24) can then be written as

$$\hat{\mathbf{s}} = \hat{\mathbf{s}}_0 + \mathcal{K}(\mathbf{y}_l - \hat{H}_l\hat{\mathbf{s}}_0), \quad l = 1, \ldots, L \tag{5.8}$$

and covariance matrix becomes

$$P = P_0 - P_0\hat{H}_l^{\mathsf{H}}(\hat{H}_l P_0 \hat{H}_l^{\mathsf{H}} + R_l)^{-1}\hat{H}_l P_0. \tag{5.9}$$

The Kalman filter gain is given by

$$\mathcal{K} = P_0\hat{H}_l^{\mathsf{H}}(\hat{H}_l P_0 \hat{H}_l^{\mathsf{H}} + R_l)^{-1}. \tag{5.10}$$

It is shown in [47] that the Kalman filter gain $\mathcal{K}$ is equivalent to the combining vector $\mathbf{v}_k$ given by (5.19). The covariance matrix P in (5.26) has two parts, the initial value $P_0 = \text{diag}\,[p_1, \ldots, p_K]$ and the updating part, which can be written as $\mathcal{K}\hat{H}_l P_0$ by using $\mathcal{K}$ in (5.27). Hence (5.26) can be written as

$$P = P_0 - \mathcal{K}\hat{H}_l P_0. \tag{5.11}$$

Proposition 1. *The performance measure SINR$_k$ for each UE k is equivalent to the inverse of the covariance matrix given by the Kalman filter*

$$\textit{SINR}_k = \frac{p_k}{P_{kk}} - 1, \tag{5.12}$$

where p_k is the power of UE k to transmit data signal and P_{kk} is the diagonal element of covariance matrix P.

Proof. The SINR represents the ratio between desired signal power and undesired total signal power, which is often called the interference and noise. For UE k, it can be rewritten as

$$\text{SINR}_k = \frac{s_k}{A_k - s_k} = \frac{\frac{s_k}{A_k}}{I - \frac{s_k}{A_k}}, \tag{5.13}$$

where s_k is desired signal power and A_k is the total sum of received signal power. The ratio of desired signal to the total sum is a part of the covariance matrix as

$$\frac{s_k}{A_k} = p_k \widehat{\mathbf{h}}_{lk}^{\mathsf{H}} (\hat{H}_l P_0 \hat{H}_l^{\mathsf{H}} + R_l)^{-1} \widehat{\mathbf{h}}_{lk} = \mathbf{v}_k \widehat{\mathbf{h}}_{lk}. \tag{5.14}$$

From (5.11) we get

$$\mathcal{K}\hat{H}_l = (P_0 - P)P_0^{-1}, \tag{5.15}$$

and hence

$$\mathbf{v}_k \widehat{\mathbf{h}}_{lk} = \mathbf{u}_k^{\mathsf{T}} \mathcal{K} \hat{H}_l \mathbf{u}_k = \mathbf{u}_k^{\mathsf{T}} (P_0 - P) P_0^{-1} \mathbf{u}_k \tag{5.16}$$

where $\mathbf{u}_k^{\mathsf{T}} = [0 \dots 1 \dots 0]$ is a vector with 1 at k^{th} element.

$$\text{SINR}_k = \frac{\frac{s_k}{A_k}}{1 - \frac{s_k}{A_k}} = \frac{\mathbf{u}_k^{\mathsf{T}} (P_0 - P) P_0^{-1} \mathbf{u}_k}{1 - \mathbf{u}_k^{\mathsf{T}} (P_0 - P) P_0^{-1} \mathbf{u}_k} = \frac{p_k}{P_{kk}} - 1. \tag{5.17}$$

□

Proposition 2. *The upper bound performance provided by the fully centralized level 4 can be achieved by sub-L when all SPs in the same cluster are added to the subset $\mathcal{L}$.*

Proof. The estimate of the state vector (5.25) can be used for any aggregating SP, from sub-1 update to sub-2, and then from sub-2 update to sub-3, when a new SP is added into the subset. Each new SP brings the new measurement into the Kalman filter. Each update reduces the estimation error hence reduces the covariance matrix. As SINR can be written as the inverse of the covariance matrix

$$\text{SINR}_k = \frac{p_k}{P_{kk}(\mathcal{L})} - 1. \tag{5.18}$$

Each update of the covariance with a new SP increases SINR. When all SPs are added in the subset $\mathcal{L}$, the minimum covariance is reached, and hence the maximum SINR is reached. The best performance is achieved by sub-L, which has the same performance as fully centralized method level 4, given by (5.20) and (5.21). □

The advantage of using the Kalman filter in this context is that the same maximum SINR or equivalently the spectral efficiency (SE) as fully centralized processing and combing can be reached by sub-L, however the processing processing complexity can be adapted for each UE and combining can be both centralized or distributed at any aggregating SPs.

5.3 Centralized Combining

In [1], four levels of receiver cooperation were proposed. From the most complex and advanced level, which is denoted level 4 to less complex level 1. We summarize these four levels briefly in this section, more details to be found in [1].

5.3.1 Level 4: Fully centralized Processing

Level 4 is a method with fully centralized processing and combining. It requires that all L SPs that are connected to a EPU of a D-MIMO network send all received pilot signals $\{Z_{lt} : l = 1, \dots, L, t = 1, \dots, \tau_P\}$ and received data signals $\{\mathbf{y}_l : l = 1, \dots, L\}$ to the EPU. For each UE, the EPU estimates the channel $\{\widehat{\mathbf{h}}_{lk} : l = 1, \dots, L, k = 1, \dots, K\}$ using received pilot signals

and channel statistics obtained from SPs. Then EPU selects combining weights $\mathbf{v}_k \in \mathbb{C}^{LN}$ for UE k based on the collective channel estimate

$$\widehat{\mathbf{h}}_k = \begin{bmatrix} \widehat{\mathbf{h}}_{lk} \\ \vdots \\ \widehat{\mathbf{h}}_{Lk} \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_L \end{bmatrix}.$$

The MMSE combining vector for UE k that maximizes the instantaneous SINR minimizes the mean-squared error $\mathrm{MSE}_k = \mathbb{E}\{|s_k - \mathbf{v}_k^{\mathsf{H}}\mathbf{y}|^2 \mid \widehat{\mathbf{h}}_k\}$, see [48] for details, is given by

$$\mathbf{v}_k = p_k \left(\sum_{i=1}^{K} p_i \left(\widehat{\mathbf{h}}_i \widehat{\mathbf{h}}_i^{\mathsf{H}} + C_i \right) + \sigma^2 I_{LN} \right)^{-1} \widehat{\mathbf{h}}_k. \tag{5.19}$$

The maximum value of SINR is

$$\mathrm{SINR}_k^{(4)} = p_k \widehat{\mathbf{h}}_k^{\mathsf{H}} \left(\sum_{i=1, i\neq k}^{K} p_i \widehat{\mathbf{h}}_i \widehat{\mathbf{h}}_i^{\mathsf{H}} + \sum_{i=1}^{K} p_i C_i + \sigma^2 I_{LN} \right)^{-1} \widehat{\mathbf{h}}_k. \tag{5.20}$$

Using level 4 combining, an achievable SE of UE k is shown to be

$$\mathrm{SE}_k^{(4)} = \left(1 - \frac{\tau_P}{\tau_c}\right) \mathbb{E}\left\{ \log_2 \left(1 + \mathrm{SINR}_k^{(4)} \right) \right\}. \tag{5.21}$$

The level 4 provides the highest SE, however the computational complexity is very high since it requires first the computation of $LN \times LN$ matrix inverse and then a matrix-vector multiplication.

5.3.2 Level 1-3: Local Processing and Centralized Combining

Level 1-3 are the methods based on local processing and centralized combining. Instead of sending the N-dimensional vector $\{\mathbf{y}_l : l = 1, \ldots, L\}$ and channel estimates to the EPU, each SP pre-processes its signals by computing the local estimates of the data that are then passed to the EPU for further combining. The local estimate for UE k at SP l is $\breve{mathbfs}_{kl} = \mathbf{v}_{lk}^{\mathsf{H}} \mathbf{y}_l$ where the local MMSE combining vector is

$$\mathbf{v}_{lk} = p_k \left(\sum_{i=1}^{K} p_i \left(\widehat{\mathbf{h}}_{li} \widehat{\mathbf{h}}_{li}^{\mathsf{H}} + C_{li} \right) + \sigma^2 I_N \right)^{-1} \widehat{\mathbf{h}}_{lk}. \tag{5.22}$$

The maximum value of SINR with the local MMSE combining (5.22) is given by

$$\mathrm{SINR}_{kl}^{(1)} = p_k \widehat{\mathbf{h}}_{lk}^{\mathsf{H}} \left(\sum_{i=1, i\neq k}^{K} p_i \widehat{\mathbf{h}}_{li} \widehat{\mathbf{h}}_{li}^{\mathsf{H}} + \sum_{i=1}^{K} p_i C_{li} + \sigma^2 I_N \right)^{-1} \widehat{\mathbf{h}}_{lk}. \tag{5.23}$$

Different from level 4, SP l uses only its own local channel estimates $\{\widehat{\mathbf{h}}_{lk} : k = 1, \ldots, K\}$ for the design of $\mathbf{v}_{lk}$. The local estimates $\{\breve{\mathbf{s}}_{kl} : l = 1, \ldots, L\}$ are then sent to the EPU where they are combined in three different methods, which are defined in [1]:

Level 1: $\hat{\mathbf{s}}_k = \underset{\breve{\mathbf{s}}_{kl}}{\mathrm{argmax}} \mathrm{SINR}_{kl}^{(1)},\ l = 1, \ldots, L$

Level 2: $\hat{\mathbf{s}}_k = \frac{1}{L}\sum_{l=1}^{L} \breve{\mathbf{s}}_{kl}$

Level 3: $\hat{\mathbf{s}}_k = \sum_{l=1}^{L} a_{lk}^{*} \breve{\mathbf{s}}_{kl}$

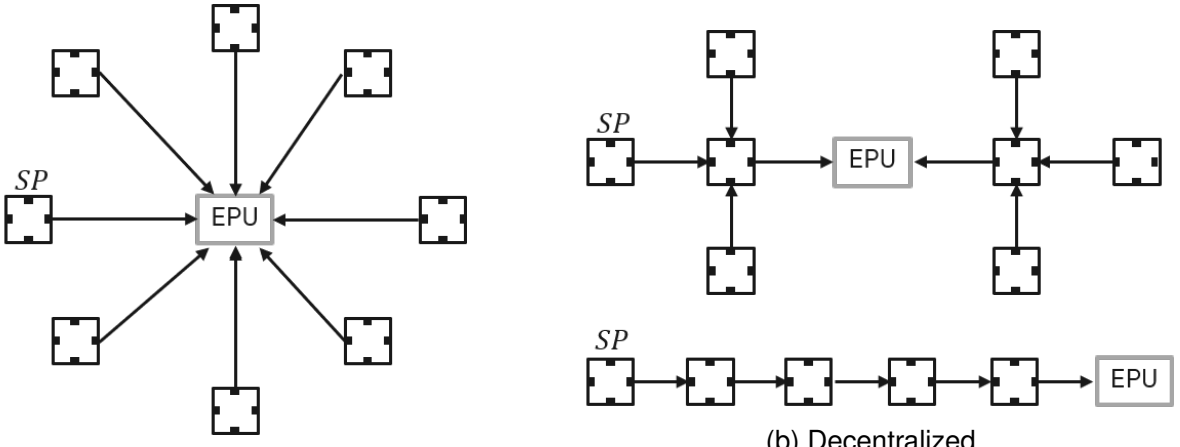
The weighting coefficients a_{lk} at level 3 can be obtained based on the channel statistics, see [1] for more details.

5.4 Decentralized Combining

In [49] a decentralized processing and combining method was proposed. Instead of sending all signals and statistical parameters to the EPU for centralized combining, see Fig. 5.3 (a), they are sent to predefined aggregating SPs, or local processing unit (LPU) for decentralized processing and combining first and then forwarded to the EPU for final decoding. The predefined aggregating SP consists of a subset of SPs, which is selected for each UE, $\mathcal{L} \subset \{1, \ldots, L\}$. The subset is denoted by sub-ℓ where ℓ out of L SPs are selected. The subset can process the signals sequentially if SPs are serially connected; the subset can also process the signals in parallel, as illustrated at the top and bottom of Fig. 5.3 (b) respectively.

5.5 Semi-localized processing for weight computation

Based on internal report description.



(a) Centralized

(b) Decentralized

Figure 5.3: Examples of centralized and decentralized combining methods.

We propose to use the square-root implementation of the Kalman filter in [50] for decentralized processing and combining at aggregating SPs. The square-root implementation is known to be more numerically sound when inverting the covariance matrix, as it always assures the covariance matrix to be symmetric and positive semi-definite.

To enable the formulation of the Kalman filter, we rewrite the signal and the channel between all UEs and SP l in more compact vector or matrix forms in the following. The received data signal at SP l can be written as $\mathbf{y}_l = H_l s + n_l$, where the channel is $H_l = \left[\mathbf{h}_{l1} \cdots \mathbf{h}_{lK}\right]$, and transmitted signals from all UEs is the state vector $\mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_K \end{bmatrix}$. The Kalman filter provides the estimate of the

state vector $\hat{\mathbf{s}}$ at SP l as

$$\hat{\mathbf{s}} = \hat{\mathbf{s}}_0 + \mathcal{K}\left(\mathbf{y}_l - H_l\hat{\mathbf{s}}_0\right), \tag{5.24}$$

which minimizes the mean-square error MSE $= \mathbb{E}\{|\mathbf{s} - \mathcal{K}\mathbf{y}_l|^2\}$ by the Kalman filter gain $\mathcal{K}$. The initial estimate state vector $\hat{\mathbf{s}}_0 = 0$ and the covariance matrix $P_0 = \text{diag}(p_1, \ldots, p_K)$.

Proposition 3. *The combining weights for all UEs can be calculated by the Kalman filter gain $\mathcal{K}$, which is equivalent to the MMSE combining vector given by* (5.19)*.*

Proof. The MMSE combining vector $\mathbf{v}_k$ given by (5.19) assumes that the channel H_l is unknown, but the estimate of the channel can be done locally at each SP based on known pilot signals transmitted from UEs. Let $\hat{H}_l = [\widehat{\mathbf{h}}_{l1}, \ldots, \widehat{\mathbf{h}}_{lK}]$ be the estimate of channel H_l, see [48] for a detailed description of MMSE channel estimation. Applying the estimation of the channel, the estimate of the state vector (5.24) can then be written as

$$\hat{\mathbf{s}} = \hat{\mathbf{s}}_0 + \mathcal{K}(\mathbf{y}_l - \widehat{\mathbf{h}}_l\hat{\mathbf{s}}_0), \quad l = 1, \ldots, L \tag{5.25}$$

and the covariance matrix becomes

$$P = P_0 - P_0\hat{H}_l^{\mathsf{H}}(\hat{H}_l P_0 \hat{H}_l^{\mathsf{H}} + R_l)^{-1}\hat{H}_l P_0. \tag{5.26}$$

The Kalman filter gain is given by

$$\mathcal{K} = P_0\hat{H}_l^{\mathsf{H}}(\hat{H}_l P_0 \hat{H}_l^{\mathsf{H}} + R_l)^{-1}, \tag{5.27}$$

where R_l is the covariance matrix including both the correlation matrix of the channel estimation error and receiver noise, $R_l = \sum_{k=1}^{K} p_k C_{lk} + \sigma^2 I$. The equation (5.27) provides the combining weights the minimizes the MSE for all UEs, $\mathcal{K} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_K \end{bmatrix}$. Thus, for UE k the combining vector $\mathbf{v}_k$ is the same as it is given in (5.19). □

Remark 1. *In the square-root implementation of the Kalman filter the covariance matrix is replaced by its square-root, S, such that $P = SS^{\mathsf{T}}$.*

Corollary 1. *The performance measure SINR for each UE k is equivalent to the inverse of the covariance matrix given by the Kalman filter*

$$\textit{SINR}_k = \frac{p_k}{P_{kk}} - 1, \tag{5.28}$$

where p_k is the power of UE k use for transmitting data signals and P_{kk} is the k^{th} diagonal element of the covariance matrix P.

Proof. The proof is to be found in [49]. □

We have introduced the Kalman filter to estimate the data signals by (5.25)-(5.27). They can be applied to any SPs locally. They can also be applied to any aggregating SPs or local processing unit (LPU), such as the subset, sub-l, or at EPU where signals from other SPs are collected as shown in Fig. 5.3(a). Furthermore, when SPs are connected sequentially where one SP forwards the estimates to another SP, the update of the state vector (5.25) can be generalized with the proposition below.

Proposition 4. *The estimate of a data signal at any aggregating SP can be obtained by the estimation of the state vector from the Kalman filter whenever a new measurement from an additional SP l becomes available.*

Proof. Let $\mathcal{L}$ be the subset consisting of ℓ SPs, and the new measurement from an additional SP l be y_l, $l \notin \mathcal{L}$. The subset of ℓ SPs is denoted by sub-ℓ. The new subset $\mathcal{L} \cup l$ has $\ell + 1$ SPs, sub-$(\ell + 1)$. Applying the Kalman filter, the estimate of the state vector is given by

$$\hat{\mathbf{s}}(\mathcal{L} \cup l) = \hat{\mathbf{s}}(\mathcal{L}) + \mathcal{K}(y_l - \hat{H}_l \, \hat{\mathbf{s}}(\mathcal{L})) \tag{5.29}$$

and the covariance matrix is given by

$$P(\mathcal{L} \cup l) = P(\mathcal{L}) - P(\mathcal{L})\hat{H}_l^{\mathsf{H}}(H_l P(\mathcal{L})\hat{H}_l^{\mathsf{H}} + R_l)^{-1}\hat{H}_l P(\mathcal{L}). \tag{5.30}$$

The Kalman filter gain that minimizes the mean-square errors between the state and estimated state is given by

$$\mathcal{K} = P(\mathcal{L})\hat{H}_l^{\mathsf{H}} \left(\hat{H}_l P(\mathcal{L})\hat{H}_l^{\mathsf{H}} + R_l\right)^{-1}. \tag{5.31}$$

The equation (5.29) updates the state vector from $\hat{\mathbf{s}}(\mathcal{L})$ to $\hat{\mathbf{s}}(\mathcal{L} \cup l)$ with the new measurement $\mathbf{y}_l$ at the aggregating SP and the equation (5.30) updates the covariance matrix from $P(\mathcal{L})$ to $P(\mathcal{L} \cup l)$ when the SP l is added to the subset. The Kalman filter gain (5.31) updates the estimate of the state vector which is the new combining weights that minimizes the MSE for all UEs. □

Corollary 2. *The best performance in terms of SINR or equivalently SE is achieved by sub-L when all SPs in the same cluster are added to the subset $\mathcal{L}$.*

Proof. By using the Kalman filter, the estimate of the state vector is updated by (5.24) when a new SP is added into the subset. Each new SP brings the new measurement into the Kalman filter and reduces the estimation error hence the corresponding covariance matrix is decreased. Each update increases SINR as SINR is the inverse of the covariance matrix as shown in *Corollary 1*. Hence the best performance in terms of SINR or equivalently SE is achieved by sub-L when all SPs in the same cluster are added to the subset $\mathcal{L}$. □

Remark 2. *If the new measurements in Proposition 4 are collected from several SPs, $l_1, \ldots, l_r \notin \mathcal{L}$. The proposition is still valid by defining $\mathcal{L}_{new} = \{l_1, \ldots, l_r\}$ and replacing l by $\mathcal{L}_{new}$, the new subset $\mathcal{L} \cup \mathcal{L}_{new}$ has thus $\ell + r$ SPs and is denoted as sub-$(\ell + r)$.*

$$\hat{\mathbf{s}}(\mathcal{L} \cup \mathcal{L}_{new}) = \hat{\mathbf{s}}(\mathcal{L}) + \mathcal{K}(\hat{\mathbf{s}}(\mathcal{L}_{new}) - \hat{\mathbf{s}}(\mathcal{L})) \tag{5.32}$$

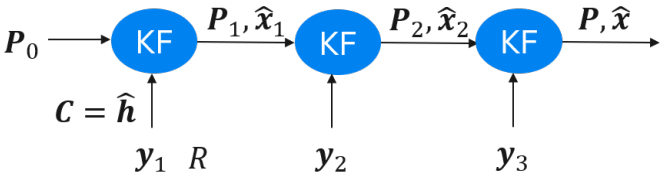
and the covariance matrix is given by

$$P(\mathcal{L}_{\cup}\mathcal{L}_{new}) = P(\mathcal{L}) - P(\mathcal{L})(P(\mathcal{L}) + P(\mathcal{L}_{new}))^{-1}P(\mathcal{L}). \tag{5.33}$$

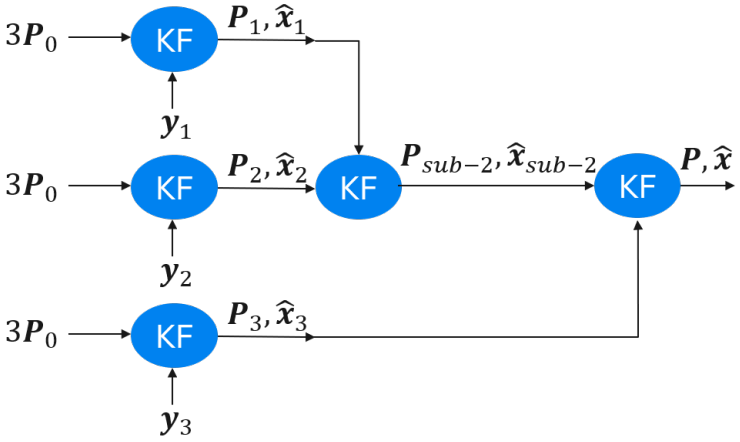
The Kalman filter gain that minimizes the mean-square errors between the state and estimated state is given by

$$\mathcal{K} = P(\mathcal{L})\left(P(\mathcal{L}) + P(\mathcal{L}_{new})\right)^{-1}. \tag{5.34}$$

The advantage of using the Kalman filter in this context is that the same maximum SINR or SE as fully centralized processing and combing (level 4) can be reached by sub-L. However the processing of sub-L can be done either centralized or decentralized at any aggregating SPs by using (5.25)-(5.27); it can also be done by distributing the processing at aggregating SPs and



(a) Serially connected.



(b) Parallelly connected.

Figure 5.4: Equivalent implementation of the Kalman filter.

push the estimate to another SP, as shown in Fig. 5.4(a). The aggregation can also be done when SPs are connected in parallel, as shown in Fig. 5.4(b). Note that the initial covariance matrix needs to be scaled accordingly with the number of parallel connected SPs.

Proposition 5. *The Kalman filter provides the equivalent results no matter if the signals are collected and proceessed centralized at an EPU or decentralized at aggregating SPs, or distributed at SPs.*

Proof. Since the Kalman filter minimizes the estimation error between the signal and estimate of the signal based on the available inputs and measurements at the current stage, the estimate of the signal contains the same amount of information that is available to the Kalman filter at any subset. They can be forwarded to another SP or aggregating SP without loss of the information of that subset. □

Remark 3. *With Proposition 5, the Kalman filter can be applied to any topology of SPs, e.g. star, serial, parallel, grid connections. The estimate of data signal at any aggregating SP is the same whether new measurements are collected from several SPs or sequentially aggregated from the first SP to the aggregating SP.*

5.6 Implementation requirements

We analyse two important aspects that need to be considered when it comes to the implementation of D-MIMO systems.

Firstly, we analyze the fronthaul connection capacity requirement, i.e., the number of signals and measurements that are required to send from SPs to EPU in order to calculate the combining

weights. The four different implementation levels process the signals either at EPU (level 4) or locally at SPs (level 1-3). All signals y_l or the estimated signals $\breve{s}_{kl}$ are collected at the EPU, as shown in Fig. 5.3(a), to be able to make the centralized combining decisions as described in Section 5.3. In contrast, the decentralized subset method based on the Kalman filter implementation has the flexibility to estimate the signals and aggregate the estimated signal, sequentially or partially at aggregating SPs or the EPU, as shown in Fig. 5.3(b).

Table 5.1: The fronthaul capacity requirement.

	signal name	elements of signal	to get covariance matrix	total elements
level 4	$\mathbf{y}_l$	$\tau_C LN$	R_{lk}	$LKN^2/2$
level 3	$\breve{\mathbf{s}}_{kl}$	$(\tau_C - \tau_P)LK$	$\mathbf{a}_{lk}$	$LK + \frac{L^2K^2+LK}{2}$
level 1-2	$\breve{\mathbf{s}}_{kl}$	$(\tau_C - \tau_P)LK$	-	-
subset	$\hat{\mathbf{s}}(\mathcal{L})$	$(\tau_C - \tau_P)LK$	$S(\mathcal{L})$	$LK^2/2$

In Table 5.1, we summarize the fronthaul capacity requirement. The column *signals* can be the received signals or the estimated signals at each SP depending on the implementation methods. The column *total signals* contains the total sum of signals sent from of all SPs to the EPU for final decoding. Level 3 and subset with sub-L have no dependency on the number of antenna elements at each SP, however, level 4 does.

The second aspect to consider is the time delay, i.e. how much time is required for processing and estimating the signals. The time delay depends on computational processing time and the time for collecting the signals. Here, the highest computational demanding operation is to calculate the inverse of the covariance matrix and the sizes or the number of elements in the covariance matrix determines the computational complexity. The fully centralized level 4 method requires collecting all signals and estimating a full-sized covariance matrix at EPU before the processing can start, the number of operations is in the order of $\sim (LN)^3$. Level 4 thus has very high requirement on the hardware processing capabilities. However it has very low requirement on the SPs since SPs simply push everything to the EPU. Level 4 is difficult to realize in practice partly because the time delay for collecting all measurements and partly high computational complexity to invert the very large covariance matrices. Level 3 process the signals locally at each SP, which put some requirement on the processor at SPs. The number of operations is in the order of $\sim LN^3$. The combining vector based on level 1-3 is calculated at the EPU which still requires to collect estimated signals from all SPs. By using the Kalman filter in the decentralized subset combining method, the signals can be processed locally at each SPs or collected at aggregating SPs. The aggregating SPs that are not overlapping can process the signals in parallel, which reduces the delay for processing and collecting the estimated signals. However, the aggregating SPs has certain requirements on processing capabilities. With the square-root implementation of the Kalman filter, see *Remark* 1, the number of the operations is in order of $\sim LNK(N+K)$.

Corollary 3. *Applying the Kalman filter implementation, the required hardware capacity can be gathered at EPU or distributed among SPs.*

Proof. In *Proposition* 5 it is shown that the Kalman filter provides equivalent results whether the signals are collected at EPU or at SPs. This implies that the required hardware capacity to process the signals has different requirements whether the signals are processed centralized at EPU or distributed at SPs. The Kalman filter implementation enables parallelization of the processing to many less capable processors at the distributed SPs. □

Chapter 6

Conclusions

This deliverable presented a RadioWeaves physical layer performance evaluation framework. The abstraction model based on effective SINR equivalent to the AWGN channel is shown to be implemented in computationally efficient manner. This deliverable established the robustness of EESM based effective SINR mapping for RadioWeaves. It also showed that the dependency of calibration parameter on parameters such as number of antennas and number of subcarriers is negligibly small. They study also showed that the proposed physical layer abstraction model performs well even with extreme SINR profiles for which the calibration is not performed. These observations, which are not known earlier, shows that the physical layer performance evaluation of RadioWeaves can be predicted accurately in a computationally efficient manner. Furthermore, this abstraction model, which depends on the MCS dependent parameter, can be used to implement system-level simulations of RadioWeaves in a time efficient manner. The comparison of fronthaul signaling of algorithms implemented on different fronthaul topologies has shown that the deeper the tree topology, the fewer real scalars need to be transmitted to the EPU. Moreover, the number of connections to the EPU is decreased in deep-tree topologies. However, increasing the depth of the tree will impose a delay in the users' data detection.

In Chapter 4 we discuss preliminary energy consumption measures that can be used as a starting point for evaluation and optimization of energy consumption in the scheduling of users and compute resources for distributed algorithms on an infrastructure of the RadioWeaves type. High-level observations about specific algorithm design challenges are also provided, related to re-scheduling of compute resources assigned to algorithms, when traffic load changes or a service providing federation moves. These discussions are preliminary and will be further detailed in later stages of the REINDEER project.

In Chapter 5 we introduce a decentralized subset method for the receiver combining in a D-MIMO network. The proposed sub-set combining method is scalable in the sense that the more SPs you deploy the more distributed processing resources and the more fronthaul segments you get. This enables simultaneous and parallel forwarding of antenna data over different fronthaul segments. It also enable simultaneous and parallel processing of UE uplink signals in different aggregation SPs. The proposed method provides both good performance and low complexity compared to single SP decoding (Level 1) or sequential SP aggregation (Level 2 or Level 3). The method approaches the performance of fully centralized processing (Level 4) as the size of the cooperating set expands. Diminishing returns once the SPs closest to the UE are included in the sub-set imply that for most UEs a small sub-set will provide sufficient performance. This ensures that only the most relevant antenna elements in the D-MIMO system are processed for each UE.

This leads to less demanding processing (e.g. smaller matrix inversions) and less forwarding of information over the fronthaul, resulting in reduced power consumption and reduced hardware cost. Furthermore, Chapter 5 introduces the Kalman filter to estimate the uplink signals in a D-MIMO network. The Kalman filter provides an optimal estimate by minimizing the estimation error. It can be applied both centralized and decentralized processing and combining. When applied to the decentralized combining methods it provides the flexibility to aggregate the estimates in different topologies, star, serial, parallel or grid. Applying the Kalman filter, the performance upper bound can be approached by incrementing the subset size stepwise. With a Kalman filter implementation, it is simple to decide the most cost efficient trade-off between performance and implementation cost.

Chapter 7

List of Abbreviations

Abbreviation	Translation
5G-NR	Fifth generation new radio
EC	European Commission
AP	Access point
AWGN	additive white Gaussian noise
BS	Base station
BPSK	Binary phase shift keying
CE	Channel estimation
CSI	Channel state information
CPU	Central processing unit
LoS	Line-of-sight
MIMO	Multiple-input multiple-output
MISO	Multiple-input single-output
MRT	Maximum ratio transmission
MRC	Maximum ratio combining
MCS	Modulation and coding scheme
OFDM	Orthogonal frequency division multiplexing
HARQ	Hybrid automatic repeat request.
EESM	Exponential Effective SINR Mapping
ZF	Zero forcing
MR	Maximal ratio
MMSE	Minimum mean square error

Abbreviation	Translation
LMMSE	Linear minimum mean square error
LTE	Long term evolution
LS	Least squares
NR	New radio
PER	Packet error rate
RF	Radio frequency
SINR	Signal-to-interference-plus-noise ratio
SNR	signal-to-noise ratio
TDD	Time division duplexing
UE	user equipment
ULA	uniform linear array
UPA	uniform planar array
WiMAX	Worldwide Interoperability for Microwave Access

Bibliography

[1] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 77–90, 2019.

[2] R. Mosayebi, M. M. Mojahedian, and A. Lozano, "Linear interference cancellation for the cell-free c-ran uplink," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1544–1556, 2021.

[3] Z. H. Shaik, E. Björnson, and E. G. Larsson, "Cell-free massive MIMO with radio stripes and sequential uplink processing," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[4] J. Rodríguez Sánchez, F. Rusek, O. Edfors, M. Sarajlić, and L. Liu, "Decentralized massive MIMO processing exploring daisy-chain architecture and recursive algorithms," *IEEE Transactions on Signal Processing*, vol. 68, pp. 687–700, 2020.

[5] J. Fan, Q. Yin, G. Y. Li, B. Peng, and X. Zhu, "MCS selection for throughput improvement in downlink LTE systems," in *Proc. ICCCN*, Aug. 2011, pp. 1–5.

[6] R. Jain, C. So-In, and A. K. A. Tamimi, "System-level modeling of IEEE 802.16e mobile WiMAX networks: Key issues," *IEEE Wireless Commun.*, vol. 15, no. 5, pp. 73–79, Oct. 2008.

[7] R. Srinivasan, J. Zhuang, L. Jalloul, R. Novak, and J. Park, "IEEE 802.16m evaluation methodology document (EMD)," Tech. Rep. IEEE 802.16m-08/004r2, 2008.

[8] S. Lagen, K. Wanuga, H. Elkotby, S. Goyal, N. Patriciello, and L. Giupponi, "New radio physical layer abstraction for system-level simulations of 5G networks," in *IEEE International Conference on Communications (ICC)*, Jun. 2020, pp. 1–7.

[9] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, 2017.

[10] Ö. T. Demir, E. Björnson, and L. Sanguinetti, "Foundations of user-centric cell-free massive MIMO," *Foundations and Trends® in Signal Processing*, vol. 14, no. 3-4, pp. 162–472, 2021.

[11] G. Interdonato, E. Björnson, H. Q. Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 197, 2019.

[12] Z. H. Shaik, E. Björnson, and E. G. Larsson, "MMSE-Optimal Sequential Processing for Cell-Free Massive MIMO With Radio Stripes," *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7775–7789, 2021.

[13] E. Bertilsson, O. Gustafsson, and E. G. Larsson, "A scalable architecture for massive MIMO base stations using distributed processing," in *50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 864–868.

[14] C. Jeon, K. Li, J. R. Cavallaro, and C. Studer, "Decentralized equalization with feedforward architectures for massive MU-MIMO," *IEEE Transactions on Signal Processing*, vol. 67, no. 17, pp. 4418–4432, 2019.

[15] A. Shirazinia, S. Dey, D. Ciuonzo, and P. Salvo Rossi, "Massive MIMO for decentralized estimation of a correlated source," *IEEE Transactions on Signal Processing*, vol. 64, no. 10, pp. 2499–2512, 2016.

[16] K. Li, R. R. Sharan, Y. Chen, T. Goldstein, J. R. Cavallaro, and C. Studer, "Decentralized baseband processing for massive MU-MIMO systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 4, pp. 491–507, 2017.

[17] A. Burr, M. Bashar, and D. Maryopi, "Cooperative access networks: Optimum fronthaul quantization in distributed massive MIMO and cloud RAN - invited paper," in *IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–5.

[18] M. Sadeghi, C. Yuen, and Y. H. Chew, "Sum rate maximization for uplink distributed massive MIMO systems with limited backhaul capacity," in *IEEE Globecom Workshops (GC Wkshps)*, 2014, pp. 308–313.

[19] K. Li, J. McNaney, C. Tarver, O. Castañeda, C. Jeon, J. R. Cavallaro, and C. Studer, "Design trade-offs for decentralized baseband processing in massive MU-MIMO systems," in *53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 906–912.

[20] S. Bassoy, H. Farooq, M. A. Imran, and A. Imran, "Coordinated multi-point clustering schemes: A survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 743–764, 2017.

[21] J. G. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill, 2007.

[22] H. T. Friis, "A note on a simple transmission formula," *Proceedings of the IRE*, vol. 34, no. 5, pp. 254–256, 1946.

[23] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge University Press, 2016.

[24] G. Interdonato, H. Q. Ngo, P. Frenger, and E. G. Larsson, "Downlink training in cell-free massive MIMO: A blessing in disguise," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5153–5169, 2019.

[25] "Feasibility study for OFDM for UTRAN enhancement," 3GPP, Tech. Rep. 25.892 v2.0.0 (2004-06), 2004.

[26] S. S. Tsai and A. C. K. Soong, "Effective-SNR mapping for modeling frame error rates in multiple-state channels," 3GPP2, Tech. Rep. 3GPP2-C30-20030429-010, 2003.

[27] M. Moisio and A. Oborina, “Comparison of effective SINR mapping with traditional AVI approach for modeling packet error rate in multi-state channel,” in *Proc. Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN)*, Jun. 2006, pp. 461–473.

[28] J. Francis, “Wideband rate adaptation and scheduling in ofdm cellular systems: Modeling, analysis, and base station-side estimation,” Ph.D. dissertation, Indian Institute of Science, Bangalore, Aug. 2017.

[29] J. Kim, A. Ashikhmin, A. J. van Wijngaarden, E. Soljanin, and N. Gopalakrishnan, “On efficient link error prediction based on convex metrics,” in *Proc. VTC (Fall)*, Sep. 2004, pp. 4190–4194.

[30] K. Brueninghaus, D. Astely, T. Salzer, S. Visuri, A. Alexiou, S. Karger, and G.-A. Seraji, “Link performance models for system level simulations of broadband radio access systems,” in *Proc. PIMRC*, Sep. 2005, pp. 2306–2311.

[31] Ericsson, “System-level evaluation of OFDM – further considerations,” TSG-RAN WG1 #35, Tech. Rep. R1-031303, 2003.

[32] A. Karim, “Exponential effective signal to noise ratio mapping (EESM) computation for WiMAX physical layer,” Master’s thesis, Washington Univ., May 2007.

[33] E. M. G. Stancanelli, F. R. P. Cavalcanti, and Y. C. B. Silva, “Revisiting the effective SINR mapping interface for link and system level simulations of wireless communication systems,” in *Proc. LatinCOM*, Oct. 2011, pp. 1–6.

[34] B. Classon, P. Sartori, Y. Blankenship, K. Baum, R. Love, and Y. Sun, “Efficient OFDM-HARQ system evaluation using a recursive EESM link error prediction,” in *Proc. WCNC*, Apr. 2006, pp. 1860–1865.

[35] E. Westman, “Calibration and evaluation of the exponential effective SINR mapping (EESM) in 802.16,” Master’s thesis, The Royal Institute of Technology (KTH), Stockholm, Sweden, Sep. 2006.

[36] “Air interface for fixed broadband wireless access systems,” IEEE Standard for local and Metropolitan Area Networks, Tech. Rep. IEEE 802.16-2004, Part 16, Oct. 2004.

[37] U. K. Ganesan, E. Björnson, and E. G. Larsson, “RadioWeaves for Extreme Spatial Multiplexing in Indoor Environments,” in *54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 1007–1011.

[38] A. Wiesel, Y. C. Eldar, and S. Shamai, “Zero-forcing precoding and generalized inverses,” *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4409–4418, 2008.

[39] S. Muneer, L. Liu, O. Edfors, H. Sjöland, and L. V. der Petre, “Handling pa nonlinearity in massive mimo: What are the tradeoffs between system capacity and power consumption,” in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 974–978.

[40] M. Sarajlić, L. Liu, and O. Edfors, “When are low resolution adcs energy efficient in massive mimo?” *IEEE Access*, vol. 5, pp. 14 837–14 853, 2017.

[41] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, “Eie: Efficient inference engine on compressed deep neural network,” *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 243–254, 2016.

[42] H. Okuhara, A. Elnaqib, D. Rossi, A. Di Mauro, P. Mayer, P. Palestri, and L. Benini, "An energy-efficient low-voltage swing transceiver for mw-range iot end-nodes," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.

[43] J. L. Wei, D. G. Cunningham, R. V. Penty, and I. H. White, "Study of 100 gigabit ethernet using carrierless amplitude/phase modulation and optical ofdm," *Journal of Lightwave Technology*, vol. 31, no. 9, pp. 1367–1373, 2013.

[44] S. Aleksic, "Power efficiency of 40 gbit/s and 100 gbit/s optical ethernet," in *2009 11th International Conference on Transparent Optical Networks*, 2009, pp. 1–1.

[45] "White rabbit node reference design," https://ohwr.org/projects/white-rabbit/wiki/WRReferenceDesign, accessed: 2021-11-26.

[46] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans, ASME, J. Basic Eng.,*, vol. 82, pp. 35–45, March 1960.

[47] K. W. Helmersson, P. Frenger, and A. Helmersson, "Optimal uplink D-MIMO processing using Kalman filtering," 2022, submitted for publication.

[48] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO network: Spectral, energy and hardware efficiency," *Foundations and Trends ®in Signal Processing*, vol. 11, pp. 154–655, 2017.

[49] K. W. Helmersson, P. Frenger, and A. Helmersson, "Uplink D-MIMO with decentralized subset combining," 2022, submitted for publication.

[50] J. E. Potter and R. G. Stern, "Statistical filtering of space navigation measurements," in *Proc. of the 1963 AIAA Guidance and Control Conference, AIAA*, New York, 1963, pp. 333–1 – 333–13.