Systems with Massive Number of Antennas: Distributed Approaches

Rodríguez Sánchez, Jesús

2022

Document Version:
Publisher's PDF, also known as Version of record

Systems with Massive Number of Antennas: Distributed Approaches

Jesús Rodríguez Sánchez

Lund 2022

Department of Electrical and Information Technology
Lund University
Box 118, SE-221 00 LUND
SWEDEN

This thesis is set in Computer Modern 10pt
with the LaTeX Documentation System

Series of licentiate and doctoral theses
No. 148
ISSN 1654-790X
ISBN 978-91-8039-228-0 (print)
ISBN 978-91-8039-227-3 (electronic)

Printed in Sweden by *Tryckeriet i E-huset*, Lund.
January 2022.

To my family

Popular Science

It is still in our memories those times where mobile phones were only used to make voice calls. Current smartphones have become a very powerful computer platform capable of delivering many other services such as video calls, music player, gaming, camera, texting, among others. Most of these applications are based on connectivity, and that implies the device transmits and receives wireless signals. Not only phones, but other devices such as: laptops, drones, VR glasses, robots, sensors, etc, are or will be part of our lives in the near future, and all of them rely on wireless connectivity. When these devices are outdoors, they are typically connected to a base-station. But, what is a base-station? It is those antennas at the top of the buildings or in high masts.

Those new devices will bring new applications and connectivity demands, mostly in connection speed and latency[1]. Apart from communications, there will be applications based on localization and sensing. It means that by using wireless signals the base-station will be able to determine your location and even gestures or movements. Another important requirement is the need to consume less energy in the base-stations, which is important from a sustainability point of view.

Unfortunately, these requirements are not supported with the current base-stations, and substantial modifications are needed. In recent years there have already been changes to accommodate for this growing demand, and one of them is the addition of a large number of antennas, in what is called Massive Multiple-Input Multiple-Output (MIMO). While current base-stations are being designed with less than 100 antennas, it is envisioned to reach to hundreds or even thousands of them in the near future, in order to support the new applications. Those antennas will not have to be at the same location as in current base-stations, but can be spread throughout a certain area. This is important for two reasons:

- Antennas allow us to focus transmitted radiated energy into a certain

[1] Latency is the time the data takes to go from the transmitter to the receiver.

small area, where the device is, similar to a lens allows to concentrate the light into a point. The more antennas we have, the more we can focus the transmitted energy. This enable us to send a signal to the device without wasting energy in the surroundings, which helps to reduce energy consumption.

- By spreading the antennas in the area, some of them are placed closer to the devices. This, in turn, also means less transmitted energy to reach the device and therefore lower energy consumption.

In order to move these ideas from theory to reality it is necessary to overcome numerous implementation challenges. There is a need to guide and support the design and implementation process, by giving recipes, evaluating solutions and estimating the cost of different options: this is the main contribution of this thesis. We describe briefly the main points addressed in the thesis:

- Description of mathematical operations needed to accomplish communication and localization. This is referred to as *algorithms*.
- Study of the performance of these algorithms in a simulated environment, in order to determine whether they meet the requirements, and be able to compare with other existing algorithms.
- Propose a certain *topology* for the system, which consists of describing how the different antennas in the system are interconnected.
- Evaluation of the cost of selecting a certain algorithm together with a topology. This cost is measured in different forms: amount of data needed to be shuffled from antennas to different parts of the system, number of mathematical operations to be performed, latency in the processing, among others.

During chapters 1 and 2, and more in detail throughout the included articles, we cover different parts of the system design and the multiple challenges we face, together with promising directions to overcome them. Distribution of the baseband processing through the system, specially close to the antennas, may alleviate the implementation issues of these type of systems. The included articles present specific techniques for processing distribution for the applications of communications and localization. The obtained results indicate that these techniques can alleviate the aforementioned challenges and move forward the implementation of such systems.

Abstract

As 5G is entering maturity, the research interest has shifted towards 6G, and specially the new use cases that the future telecommunication infrastructure needs to support. These new use cases encompass much higher requirements, specifically: higher communication data-rates, larger number of users, higher accuracy in localization, possibility to wirelessly charge devices, among others.

The radio access network (RAN) has already gone through an evolution on the path towards 5G. One of the main changes was a large increment of the number of antennas in the base-station. Some of them may even reach 100 elements, in what is commonly referred as Massive MIMO. New proposals for 6G RAN point in the direction of continuing this path of increasing the number of antennas, and locate them throughout a certain area of service. Different technologies have been proposed in this direction, such as: cell-free Massive MIMO, distributed MIMO, and large intelligent surface (LIS). In this thesis we focus on LIS, whose conducted theoretical studies promise the fulfillment of the aforementioned requirements.

While the theoretical capabilities of LIS have been conveniently analyzed, little has been done in terms of implementing this type of systems. When the number of antennas grow to hundreds or thousands, there are numerous challenges that need to be solved for a successful implementation. The most critical challenges are the interconnection data-rate and the computational complexity.

In the present thesis we introduce the implementation challenges, and show that centralized processing architectures are no longer adequate for this type of systems. We also present different distributed processing architectures and show the benefits of this type of schemes. This work aims at giving a system-design guideline that helps the system designer to make the right decisions when designing these type of systems. For that, we provide algorithms, performance analysis and comparisons, including first order evaluation of the interconnection data-rate, processing latency, memory and energy consumption. These numbers are based on models and available data in the literature. Exact values depend on the selected technology, and will be accurately determined after

building and testing these type of systems.

The thesis concentrates mostly on the topic of communication, with additional exploration of other areas, such as localization. In case of localization, we benefit from the high spatial resolution of a very-large array that provides very rich channel state information (CSI). A CSI-based fingerprinting via neural network technique is selected for this case with promising results. As the communication and localization services are based on the acquisition of CSI, we foresee a common system architecture capable of supporting both cases. Further work in this direction is recommended, with the possibility of including other applications such as sensing.

The obtained results indicate that the implementation of these very-large array systems is feasible, but the challenges are numerous. The proposed solutions provide encouraging results that need to be verified with hardware implementations and real measurements.

Preface

This doctoral thesis summarizes my research contributions during the time as a doctoral student at the department of Electrical and Information Technology (EIT), Lund University, Sweden. It is comprised of two parts: The first part introduces the reader into the topic covered in the rest of the thesis, including the need of very-large antenna array systems, the potential implementation issues, motivation for distribute processing and different considerations to alleviate these limitations.

The second part of the thesis consists of a collection of original scientific publications written during my doctoral studies. My personal contribution to them is detailed as follows:

Paper I

Jesús Rodríguez Sánchez, Fredrik Rusek, Muris Sarajlić, Ove Edfors and Liang Liu, "Fully Decentralized Massive MIMO Detection Based on Recursive Methods," *IEEE International Workshop on Signal Processing Systems (SiPS)*, 2018.

Personal Contributions: I was the main contributor to the paper, developing the idea of mapping existing algorithms in the literature to a daisy-chain topology, in order to enable distributed processing in Massive MIMO uplink baseband processing. I also performed the writing, performance evaluation and analysis, with the guidance and support of the rest of co-authors.

Paper II

Jesús Rodríguez Sánchez, Juan Vidal Alegría and Fredrik Rusek, "Decentralized Massive MIMO Systems: Is There Anything to be Discussed?," in *IEEE International Symposium on Information Theory (ISIT)*, 2019.

Personal Contributions: This work covers the channel estimation problem in distributed systems. I took the lead in deriving expressions, and evaluate

results, while the initial concept, writing and simulations were shared among co-authors.

Paper III

Jesús Rodríguez Sánchez, Fredrik Rusek, Ove Edfors, Muris Sarajlić and Liang Liu, "Decentralized Massive MIMO Processing Exploring Daisy-chain Architecture and Recursive Algorithms," *IEEE Transactions on Signal Processing*, vol. 68, pp. 687-700, 2020.

Personal Contributions: Extension of the Paper I, with more theoretical support, and extensive analysis. I was the main contributor of the paper, taking the lead in the writing part. My contributions also include the derivation of closed-form expressions for SIR and SINR (proofs included), proposing multiple iterations through the array, simulations, deriving expressions for complexity, latency and memory consumption, and evaluation of results and analysis. This was possible thanks to the guidance and support of the rest of the co-authors. I was also responsible of developing a simulation environment for this.

Paper IV

Jesús Rodríguez Sánchez, Ove Edfors, Fredrik Rusek and Liang Liu, "Processing Distribution and Architecture Tradeoff for Large Intelligent Surface Implementation," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020.

Personal Contributions: Preliminary work in distributed processing for Large Intelligent Surfaces. I was the main contributor in the mapping of the IIC algorithm to mathematical operations that can be implemented in hardware based on singular value decomposition (SVD). I developed an entire new simulation framework to evaluate LIS-based systems, in terms of performance, computational complexity, inter-connection bandwidth and latency. I was responsible for writing the paper and simulating the different scenarios, under the guidance and support of the rest of co-authors.

Paper V

Jesús Rodríguez Sánchez, Fredrik Rusek, Ove Edfors and Liang Liu, "Distributed and Scalable Uplink Processing for LIS: Algorithm, Architecture, and Design Trade-offs," to appear in *IEEE Transactions on Signal Processing*.

Personal Contributions: This work represents an extension of Paper IV, including a more advance solution for the interconnection network, more extensive evaluation and analysis. I upgraded the simulator to support the extended

architecture and the new scenarios to simulate. I was the main responsible for writing, simulating and evaluating the results with the great support of the rest of co-authors. Apart from that, I collaborated in the creation of the IIC algorithm, and its mapping onto the daisy-chain topology and the panelized LIS system.

Paper VI

Jesús Rodríguez Sánchez, Ove Edfors and Liang Liu, "Positioning for Distributed Large Intelligent Surfaces using Neural Network with Probabilistic Layer," in *IEEE Global Communications Conference Workshops (Globecom Workshops)*, 2021.

Personal Contributions: This publication represents a preliminary work in the localization problem using LIS. I developed a novel method for wireless-based localization using deep learning, which provides a measure of uncertainty, and enables probability fusion. I wrote a new simulator from scratch to be able to evaluate this method and analyze the results. I was the main contributor to the paper under guidance of the other co-authors.

During the course of my doctoral studies, I had the opportunity to contribute to the following publications, which are not included in this thesis:

Paper VII

Muris Sarajlić, Fredrik Rusek, Jesús Rodríguez Sánchez, Liang Liu and Ove Edfors, "Fully Decentralized Approximate Zero-forcing Precoding for Massive MIMO Systems", *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 773-776, 2019.

Paper VIII

Juan Vidal Alegría, Jesús Rodríguez Sánchez, Fredrik Rusek, Liang Liu and Ove Edfors, "Decentralized Equalizer Construction for Large Intelligent Surfaces," *IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019.

Paper IX

Juan Vidal Alegría, Fredrik Rusek, Jesús Rodríguez Sánchez and Ove Edfors, "Modular Binary Tree Architecture for Distributed Large Intelligent Surface," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

Paper X

Juan Vidal Alegría, Fredrik Rusek, Jesús Rodríguez Sánchez and Ove Edfors, "Trade-offs in Quasi-decentralized Massive MIMO," *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020.

Paper XI

Mohammad Attari, Jesús Rodríguez Sánchez, Liang Liu and Steffen Malkowsky, "An Application Specific Vector Processor for CNN-based Massive MIMO Positioning", *IEEE International Symposiumon Circuits and Systems (ISCAS)*, 2021.

Acknowledgements

During the last 5 years I enjoyed one of the most amazing journeys of my life. A fascinating experience that I benefited from in many ways, personally and professionally. This would not have been possible without the help, friendship and support of many persons.

First and foremost, my PhD supervisors, Associate Prof. Liang Liu and Prof. Ove Edfors that believed in me and offered me the opportunity to start this journey. I will be always thankful.

I would like to express my gratitude to my main supervisor, Associate Prof. Liang Liu, for the unconditional support, discussions, availability, continuous feedback, and setting the quality bar very high. Thank you for making this thesis possible, and for giving me the freedom to explore and follow my interests. I feel I learned a lot from you. Thank you Liang!

I am also indebted with Prof. Ove Edfors for the time, enthusiasm, and vast knowledge in wireless communications, and other topics, which made the discussions really fruitful. It was great working with you. Thank you Ove!

My gratitude also goes to Associate Prof. Fredrik Rusek, from whom I learnt a lot. Fredrik, you showed me that making world-class research can be also really fun. Thank you for your time with me and with others, your sense of humor and love for research. Thank you Fredrik!

I would also like to thank all my colleagues at EIT for all the good times we spent together throughout this journey: Hemanth, Steffen, Sara, Pepe, Juan, Arturo, Muris, Joao, Ilayda, Masoud, Sidra, Lucas, Aleksei, Xuhong, Umar, Guoda, Erik, and many more. Especially I would like also to express my gratitude to Associate Prof. Anders Johansson, Prof. Buon Kiong Lau (Vincent), Prof. Fredrik Tufvesson, and Associate Prof. Joachim Rodrigues. Thank you all for your support, discussions, positive energy and friendship. I feel grateful for having such a good colleagues and friends. Thank you!

I would like also to extend my gratitude to all the EIT administrative and technical stuff that make our work possible. Thanks also to Tryckeriet (printer shop) in the E-building, especially to Jessica, to make this book physically

possible. Thank you!

I would like also to express my gratitude to Alberto Jimenez Felström, who recommended Lund University when I had the idea to pursue a PhD. He convinced me that it is the place to be for a PhD in wireless communications. Thank you Alberto (you were right)!

Of course, I would not be here without my parents. They supported me from the beginning. Thank you for being always there for me. Thank you for being my parents. Thank you for all the love!

Ania, I have no words to describe your unconditional support, love and care. I know that sometimes it has not been easy, but you have been always by my side, helping me and encouraging me throughout the last part of the PhD. This means a lot to me. Thank you for being in my life. Thank you kochanie!

Jesús Rodríguez Sánchez
Lund, June 2022

List of Acronyms and Abbreviations

5G fifth generation. 3

6G sixth generation. 3

AFE analog front end. 11, 22, 23

AP access point. 9, 42

AR augmented reality. 5, 16

CD coordinate descent. 33, 37, 42

CPU central processing unit. 11, 12

CSI channel state information. viii, 29, 36, 39, 42, 43, 42, 43, 44

DFE digital front end. 11, 22, 23

GNSS Global Navigation Satellite Systems. 5

IIC iterative interference cancellation. 34, 35, 42

LIS large intelligent surface. vii, 8, 10, 11, 21, 34, 47

LS least squares. 36

MIMO multiple-input multiple-output. 34

ML machine learning. 42

MMSE minimum mean square error. 30, 32, 33, 32, 36, 37

MRC maximum ratio combining. 30, 32

MRT maximum ratio transmission. 30, 32

NN neural network. 42

NR New Radio. 3

OFDM orthogonal frequency-division multiplexing. 14, 30, 31

PA power amplifier. 7, 13

RAN radio access network. vii, 6, 7, 8, 9, 10, 42

RB resource block. 14, 15

RF radio frequency. 11, 13

RLS recursive least squares. 36

RSSI received signal strength indicator. 42

SINR signal-to-interference-plus-noise ratio. 32

SNR signal-to-noise ratio. 32

SVD singular value decomposition. x

TDD time division duplexing. 14, 30

UE user equipment. 32

WPT wireless power transfer. 3

ZF zero forcing. 30, 32, 33, 34, 36, 42

Contents

Part I

Introduction

Chapter 1

Motivation and Outline

We are living in the digital era, where the presence of electronic devices such as laptops or phones is ubiquitous in our daily lives. These devices allow us to communicate quickly and efficiently in many forms by the use of wireless connectivity. New and more demanding applications are being envisioned for the near future, with the intention to increase our productivity, enhance our well-being and facilitate different tasks in hand. Future sixth generation (6G) networks should be able to support these applications, and for that, the current telecommunication infrastructure needs to undergo a transformation in order to fulfill the high demanding requirements that these new applications require.

In this chapter, we describe a certain number of expected use cases for the near future, together with their requirements from a wireless connectivity point of view. Additionally we also provide a motivation to explore different system architectures for future base stations, that allows the infrastructure to adapt for such requirements, without sacrificing hardware resources and energy consumption.

1.1 6G wireless networks: new use cases and requirements

The fifth generation (5G) New Radio (NR) is reaching maturity with the first networks already deployed. The 5G access interface presents high bandwidth and massive antenna arrays, that not only provides connectivity with a higher data rate than the current communication-based applications demand, but also has potential for accurate localization and sensing. It is expected that this trend will continue into the future, where the radio access infrastructure be-

comes physically larger by the introduction of distributed arrays with coherent processing. The larger arrays will allow a further and richer exploitation of the spatial dimension, that will not only increase communication throughput but also push localization accuracy limits even further, opening the door to new and exciting applications [1, 26, 27]. Many of these use cases are expected to appear in the fields of entertainment, manufacturing and healthcare. 6G networks will offer wireless connectivity that provides low latency, high reliability and high capacity links required by these applications. Furthermore, wireless power transfer (WPT)[1] techniques will offer the capability of wireless power supply of energy-neutral devices, supporting a whole range of new applications. As an illustration of envisioned wireless-based applications, Fig. 1.1 depicts a future fully automated factory, where robots perform different tasks with the help of sensors and cameras that provide the needed information for a correct monitoring and control of the entire process. Different antenna arrays, distributed across the area, provide the required wireless connectivity to those elements, together with accurate localization of the robots within the factory.

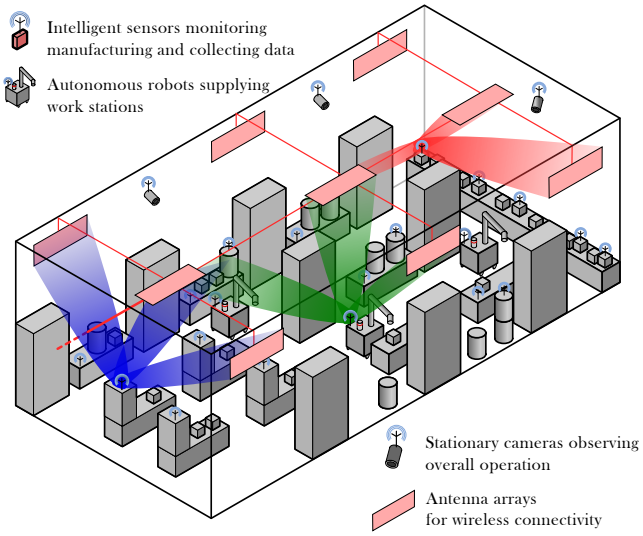


Figure 1.1: Envisioned smart factory.

Here we list and describe briefly four main areas where new and disruptive applications are expected in the near future [1]:

1. **Robotization:** The use of robots is becoming more and more common,

[1] Presumably, WPT will be a key technology in the future wireless infrastructure. However it is not covered in this thesis.

due to advances in electronics and artificial intelligence, allowing them to complete more complex tasks in less time. Factories, warehouses, retail and care homes are the areas where more changes in this direction are expected. Robotization of the care homes, where robots will interact with patients, and take care of them is a relevant area. Robotization of the vehicles is another growing trend, where intelligent vehicles are able to communicate with each other and with the network, in a plan to make driving fully autonomous and safer. The control of, and communication with, robots will come with strict demands of high data reliability and low latency. End-to-end latency (E2E) is expected to be as low as 1ms, while the packet error rate (PER) should not exceed 10^{-6} [1].

2. **Augmented reality:** There is a new wave of wireless devices emerging in the augmented reality (AR) space [2], that will create an immersive experience, making it even simpler for us to enjoy and share experiences in real time with family and friends, who are physically apart. Gaming, celebrations, sport events, etc, will be shared instantaneously with others, together with the addition of virtual interactive elements to enrich the experience. From a professional point of view, these devices will also bring a wide range of possibilities, including an enhanced remote work experience. From a technical point of view, this will impose high throughput demands on individual connections, ranging from 5Mbps to 3Gbps, and a large traffic volume, going up to 50Mbps/m^2. E2E requirements are restricted to be under 10ms [1].

3. **Sensors:** Sensors will be ubiquitous in our daily lives. Applications are ranging from wearables, and in-body sensors for an efficient (and remote) patient monitoring, to smart home automation where sensors could bring temperature, light, sound, and air quality measurements. An infrastructure capable of wireless power transfer enables the use of energy-neutral devices which brings more opportunities, as they do not require battery, and therefore reducing the acquisition and maintenance cost. The density of devices is envisioned to be as high as 100/m^2 for certain applications, potentially reaching 50,000 simultaneous connections [1].

4. **Positioning:** Positioning information will be required to enable many of the envisioned applications. Patient tracking in hospitals and care centers brings more freedom of movement. People tracking in large venues or shopping areas can provide personalized patterns for marketing purposes or emergency indications if needed. Tracking of robots and UVs is also expected. Global Navigation Satellite Systems (GNSS) services will not be enough for these applications, as some of them will happen in indoor

Table 1.1: System level requirements for use cases covered in this thesis, based on the RadioWeaves [1] vision, to support new applications. * indicates requirements that are not directly covered in this thesis.

Requirement	Typical range
Device density (m^2)*	0.1-100
Number of simultaneous devices	2-50k
User experience data rate (Mbps)	<3000
Mobility (m/s)*	0-10
Positioning Accuracy (m)	0.1-1.0
Reliability (packet loss)	10^{-6}-10^{-2}
End to end latency (ms)	1-1000
Traffic volume density (Mbps/m^2)*	<100

environments (where GNSS is not available) and the required accuracy may exceed the one provided by satellite-based services. In this context, future wireless systems will provide another source of localization. It is expected to reach an accuracy as good as 0.1m [1].

Within these four areas, the number of foreseen applications is large and diverse [1], which translates into a wide range of system-level requirements, as shown in Table 1.1. To sum up, we can envision applications where the number of simultaneous connected devices is very large (thousands), each with relatively low to moderate data-rate demands, while other applications require very high data-rates for a much smaller number of users. Apart from this, strict requirements in reliability, mobility and latency will have a drastic impact on the infrastructure architecture and cost. A fixed solution may not fit well all these applications. Therefore, the future 6G infrastructure should be flexible and scalable, as well as be able to adapt to the specific application demands.

Energy consumption is another important requirement for the future infrastructure. It is expected that these systems will be more energy-efficient[2] than current ones. As we want to transition towards a sustainable and green future, the global energy consumption in the telecommunications networks must be revised, especially the RAN. RAN forms the access gate for the users to the network, comprising the infrastructure and the radio signaling that is exchanged between both parties and supports the wireless-based services. It is known that around 80% of the energy consumed in the RAN has been traditionally used in base stations, of which about 80% is consumed in Power Amplifiers (which is proportional to the total radiated energy) [3]. As the baseband algo-

[2] We define more energy-efficiency as the use of less energy to perform the same task or achieve the same result.

rithms are growing in computational complexity, it is expected that they will account for about 50% of the energy consumption in 5G base stations [23].

The absolute volume of information to transmit through the RAN is expected to grow in the next years, and therefore the radiated energy would also grow accordingly with the current infrastructure. As the base stations operate with more and more antennas, the radiated energy per bit is decreasing (as the energy can be more focused towards users), and may counterbalance the growth of transmitted data from the point of view of total radiated energy (and therefore the one also consumed by power amplifiers (PAs)). On the other hand, increasing the number of antennas contributes to a substantial increment of the total energy consumption, due to three factors: 1) the additional circuitry added (PAs, analog front-end, DAC/ADC, etc), 2) higher inter-connection data-rate[3], and 3) more baseband processing. The first factor is technology dependent, and scales linearly with the number of antennas. The second scales linearly with the amount of data to exchange in the front-haul which, as we will see, depends on the system architecture. The third factor scales in a superlinear fashion with respect to the the number of spatially multiplexed users (or layers)[4]. A reduction in the base station energy consumption requires significant effort in these three areas, in order to increase the total energy efficiency. This thesis focuses on the second and third ones since they are both interconnected.

While existing networks have been evolving substantially until now, the explosion of new applications and their diverse requirements call for a profound transformation of the existing infrastructure, including the RAN. In the particular case of communications, new ideas and developments have been carried out during these years to accommodate to the growing demand in capacity. The exploitation of the spatial dimension has been a driving force, materialized as an increase of the number of antennas in Multiple-User Multiple-Input and Multiple-Output (MU-MIMO) systems, with Massive MIMO as a result [5, 6].

1.2 Beyond Massive MIMO

Massive MIMO has gone from an initial theoretical concept to a real deployment within a decade. Massive MIMO consists of an extension of traditional cellular base stations with a very large array of antennas (in the order of 100). This increases the spatial multiplexing capabilities, allowing the system to communicate with more devices at the same time and frequency resources, and therefore

[3] Inter-connection is the capability of the system to transfer the data, from where it is generated to where it is consumed. More details can be found in Subsection 1.3.1 [4] Assuming Multiple-User Multiple-Input and Multiple-Output (MU-MIMO) processing with interference cancellation schemes for downlink precoding and uplink detection.

boosting the spectral efficiency. Massive MIMO can be used in relatively low carrier frequencies, such as below 6 GHz. As a result, the hardware technology and radio components are quite mature and inexpensive. Despite the obvious benefits of this technology, there are many implementation challenges involved. While many of them have been already addressed, there are still some other ones to be solved. This thesis covers some of these challenges and propose solutions to overcome them. Specifically we cover: interconnection data-rate, processing distribution and architecture scalability, computationally-efficient algorithms, cost vs performance trade-offs, among others.

Future applications, as shown in Table 1.1, demand higher data throughput and very large number of simultaneous connections, that can not be conveniently supported with current Massive MIMO deployments. In order to address this demand, we envision to extend MU-MIMO technology beyond current Massive MIMO, which implies an increase of the number of antennas and the physical size of the array. This is the idea behind LIS [7]. LIS was born to exploit spatial multiplexing to the fullest, by the use of thousands of antenna elements and fully digital transceivers, with coherent baseband processing capabilities.

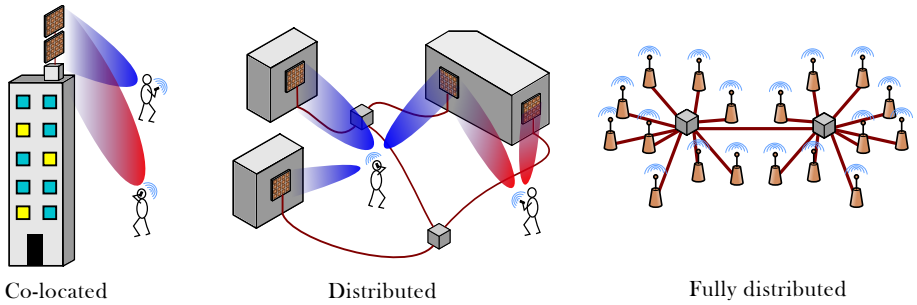


Figure 1.2: Different RAN alternatives.

While Massive MIMO is typically implemented in a co-located fashion, cell-free Massive MIMO can be found usually following a fully distributed scheme. In the case of LIS, the large array is expected to be divided in small panels that are physically distributed in the area of service. These three ways of organizing a very large array are shown graphically in Fig. 1.2:

- **Co-located:** This is the case when all antennas are physically together. In this scenario, each user is served by all antennas in the array. Even though this may represent the simplest case in terms of back-haul, it requires a relatively high transmit energy. This is required in order to cover

the whole area of service (large distance between base-station antennas and users translates into high path loss) and due to the exposure to the shadow fading, which may reduce the coverage probability.

- **Fully distributed:** A fully distributed array with single-antenna access points (APs) is preferred as increases the coverage probability, by exploiting diversity against the shadow fading [29,30,32]. Potential collaboration among APs is beneficial to reduce interference and boost system capacity. As more than one AP may serve each user, the probability of blocking or deep fade gets reduced. However, the back-haul complexity and maintenance cost are higher in terms of the number of interconnection links, and the high operational cost due to having only one antenna per AP[5].

- **Distributed:** Represents a middle point, where the large array is split in sub-arrays or panels (acting as APs). These panels are distributed in the area of service, and cooperate to jointly serve the users. This approach represents a trade-off between exploiting spatial diversity, probability of coverage, and back-haul interconnection.

The carrier frequency is a key element in the system design, which also plays an important role to meet the demand of future applications. We analyze and compare three potential bands for future RAN technologies based on current 5G standardization: low-band (below 1GHz), mid-band (1-6GHz), and high-band or milimiter wave (mmWave) (above 24GHz). Realizing future low-band RANs brings two main potential limitations:

- **Larger array size:** In order to achieve a target array gain (to provide coverage to an area), a certain number of antenna elements are expected in the antenna array. As co-located antennas are separated by $\lambda/2$ (where λ is the wavelength), the physical size of arrays in low-band is expected to be larger, which translates into more volume (and probably weight) for the radio part of the base-station. This potentially can be a problem for operators when extending current sites or finding locations for new ones.

- **Positioning accuracy:** As the wavelength becomes larger in lower frequencies, the positioning accuracy may be severely degraded. According to results in [8], the Cramér-Rao lower bound (CRLB) related to positioning estimation in LIS scales with λ^2 when the physical array size is

[5] By using the term operational cost we refer to the energy consumed for maintaining the operation of the panel or AP, not including the energy consumed in transmission and baseband processing. Examples are the energy lost in the voltage conversion and the active cooling of the system [23].

assumed fixed. For example, transitioning from 700MHz (low-band) up to 3.6GHz (mid-band) translates into 26 times higher positioning accuracy. According to Table 1.1, new applications such as tracking of robots or goods may require high accuracy (0.1m), which may be challenging to achieve in low-band, where this accuracy translates into 0.3λ. This implies relying on spatially distributed and large aperture arrays [3].

On the other hand, high frequencies bring lots of available spectrum to accommodate for the growing demand in terms of capacity. However, even though high-band systems offer many benefits, there is still an interest in developing mid-band solutions for future systems. The main reasons are [10]:

- **Implementation challenges:** High-band solutions have some specific challenges [11], with implementation issues such as phase noise. Increasing the subcarrier spacing and adding phase tracking reference symbols are among the solutions proposed by the 3GPP in order to overcome this challenge. However, as the carrier frequency continues to increase, this may not be sufficient, and advanced functionality such as MU-MIMO techniques and high-order modulation will be less feasible, therefore reducing the spectral efficiency.
- **Digital beamforming:** It is technically feasible to implement full frequency-domain digital beamforming in mid-band as the bandwidths are usually in the order of 100MHz. By allowing a fine control of beamforming in frequency domain (a group of adjacent subcarriers can be beamformed with certain beams, while the next group in frequency may be beamformed with completely different beams), the system is able to exploit the spatial properties of the wireless channel with the goal to maximize the spectral efficiency. This is in contrast to high-band solutions, where analog beamforming (same beams are used for the whole bandwidth) is common practice.
- **Number of elements in the array:** For a certain coverage, mid-band solutions always need less antenna elements than high-band ones, as the latter needs to rely on higher array gains to compensate for the smaller effective antenna aperture in those frequencies. In the case of full digital beamforming, having lower number of transceivers simplifies system design and baseband algorithms.

Based on the previous discussions we focus on mid-band solutions in this thesis, with emphasis on Massive MIMO and specially on LIS as potential future technologies for 6G RAN. While the theoretical performance of LIS in communication and positioning has been already studied [7, 8] with promising

results, little has been done in terms of hardware implementation of fully digital beamforming solutions. This thesis attempts to partially fill that gap by giving design principles and guidelines for the selection of an appropriate architecture, including algorithm and topology (these terms will be introduced in Chapter 2). In the next sections we will introduce the implementation challenges when implementing these type of systems, and motivate for distributed processing as potential solution.

1.3 Implementation challenges

As the number of antenna elements grows in the system, the hardware implementation becomes more challenging. In the case of LIS, with potentially thousands of antennas, implementation requires to leverage new ideas from architectural and algorithmic point of view. As we aim for a distributed LIS, the processing distribution and scalability are considered the key factors for a successful result. Additionally, the distance between panels also increases, which makes it more challenging to move data and ensure tight synchronization among them. In this thesis we focus on physical layer implementation, and are aiming for an unified architecture to support different applications, specifically: communications, localization, and sensing. We will cover the first two in this thesis, while the third is left for future work.

In order to illustrate the potential hardware implementation challenges when it comes to LIS, let us consider a centralized architecture based on a central baseband processing node[6] as illustrated in Fig. 1.3. We consider reception (or uplink) as an example. The antenna array has M elements, and each one is connected to a dedicated analog front end (AFE) with radio frequency (RF) circuitry, followed by an ADC and digital front end (DFE), which includes digital channel filters and downsampling. Several antennas may be operated jointly in a subsystem that share certain common functionality. The other part of the system is the central processing unit (CPU), which may be physically apart from the antenna array. With this architecture in mind, we now list and describe four hardware implementation challenges when it comes to LIS.

It is important to remark that in this chapter we evaluate a **first order approximation** of the system complexity in different areas with numbers that are currently available in literature, in order to motivate the need of searching for alternative architectures. The exact numbers during real deployment will depend on specific system parameters, implementation and silicon technology,

[6] The centralized baseband processing architecture has been already used in Massive MIMO implementations, such as in the LuMaMi testbed [17]

and likely will be optimized for the technology in hand, so may differ with the numbers presented here.

1.3.1 Inter-connection data-rate

In this subsection we focus on the required inter-connection data rate for exchanging digital baseband samples between the antenna subsystems and the CPU. Let us assume for simplicity only uplink direction, as shown in Fig. 1.3. We can imagine that using dedicated physical links between antennas and CPU is not practical, and has a high cost when it comes to adding a new antenna if required. To solve this, we consider a shared bus for connections instead, as shown in the figure.

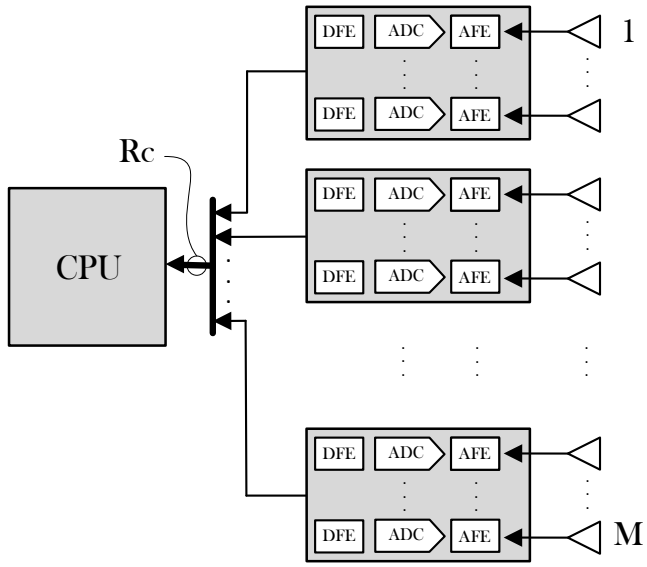


Figure 1.3: System architecture of a base-station with centralized processing.

This architecture requires a very high inter-connection data-rate in the bus, and at the input of the CPU (R_c in the figure). To illustrate that, let us consider a received signal with bandwidth f_B. Then the average interconnection data-rate can be calculated as

$$R_c = 2wMf_B, \tag{1.1}$$

where w is the bit-width for the baseband samples (real/imaginary parts) after DFE. To give a numerical example, we consider a signal bandwidth of 100MHz,

with $M = 1024$ (1.2m $\times$ 1.2m $\lambda/2$-spaced array in the 4GHz band), and $w = 12$ bits. This leads to an aggregated interconnection data rate of $\sim$ **2.5Tb/s**. To put this number into perspective, we compare it with the user information data-rate that the system may handle during the uplink cycle, denoted as R_{i}. This rate can be upper bounded as follows: $R_{\mathrm{i}} \leq C_{\mathrm{r,max}} N_{\mathrm{bs,max}} K f_{\mathrm{B}}$, where $C_{\mathrm{r,max}}$ is the maximum coding rate, and $N_{\mathrm{bs,max}}$ is the maximum number of bits per constellation symbol. The bound is attained if the cyclic prefix overhead is zero, and if the same value of $N_{\mathrm{bs,max}}$ applies to the whole bandwidth. Assuming $K = 150$, $N_{\mathrm{bs,max}} = 8$ (256QAM), and $C_{\mathrm{r,max}} = 1$, this leads to $R_{\mathrm{i}} \leq 120$Gb/s, and therefore indicates that R_{i} is at least 20 times lower than R_{c}. As we can notice, there is a large gap between these two data-rates. Closing this gap by reducing R_{c} is attractive from the energy consumption point of view, especially when considering to cover relatively long distances (in case of distributed LIS). To illustrate this, we can consider SerDes technology for short inter-connections and 100G Ethernet for medium to long distances. For the former case, state-of-the-art solutions offer a merit figure of 5.34pJ/bit [12], which translates to $\sim$ 13W for 2.5 Tb/s. In the case of optical Ethernet, if we consider not only PHY but also Ethernet line cards and switches, the figure can go up to 5.2nJ/bit[7] [13], which leads to a power consumption of 13kW only for data-shuffling.

In order to reduce the gap between R_{c} and R_{i}, and enable scalability, we aim to have an interconnection data-rate that only depends on the number of users and therefore the information data-rate, regardless of the number of antennas. In order to achieve that, part of the baseband processing performed at the CPU side should be migrated close to the antennas, specifically the MIMO processing (beamforming and equalization), acting as a preprocessing. In the next chapter we will cover some of the proposed techniques to achieve this goal.

1.3.2 Computational complexity

Traditionally energy consumption due to computational complexity in a base-station has been considered small compared to other sources, such as the transmission energy[8]. It is usually treated as a fixed term in the energy consumption models, together with other operational contributions such as site cooling [23–25]. With the arrival of 5G and the use of very large arrays, there is a trend to reduce the transmit energy in exchange for an increase in the required computational complexity. Therefore, it should not be surprising that

[7] For 10 Tbit/s Ethernet switch, equipped with 100GbE line cards, each using 4 channels (4x25G). [8] Transmission energy corresponds to the energy used by PAs and RF chains [23], which is proportional to the radiated transmitted energy.

computational complexity can account for 50% of total energy consumption on a 5G Massive MIMO base-station, reaching a power consumption of 800W in high volume traffic scenarios [23]. As the density of base-stations increases, and the distance to the users gets reduced (as is the case in distributed scenarios), the required transmitted energy will tend to reduce, and therefore the computational energy will remain dominant. This trend indicates that this factor will be a very relevant contributor to the energy consumption of the future systems, and should be taken into account as a potential challenge in our analysis. In order to illustrate that, let us consider one baseband processing task as representative factor in our analysis. We select the linear minimum mean square error (L-MMSE) equalizer method, which is a commonly used method for MU-MIMO uplink detection. Even though the method will be described in more detail in the next chapter, we present here its requirement from a computational complexity point of view, followed with an estimate of the demand in terms of energy consumption. There are two phases when considering L-MMSE realization:

Formulation

During this phase, the equalization weights are computed. In the case of the L-MMSE method, these follow the following expression

$$\mathbf{W}_{\text{lmmse}} = \left(\mathbf{H}^H\mathbf{H} + \alpha\mathbf{I}\right)^{-1}\mathbf{H}^H, \tag{1.2}$$

where the $M \times K$ complex matrix $\mathbf{H}$ represents the MU-MIMO wireless channel, K is the number of users in the system, α is a scalar, and $\mathbf{I}$ is the $K \times K$ identity matrix. We assume there is a matrix $\mathbf{H}$ for every coherence-block (frequency and time) of the channel (see Subsection 2.3.2 for more details).

Computation of the weights requires to perform the product $\mathbf{H}^H\mathbf{H}$, which leads to MK^2 complex multiplications and additions. As the matrix to invert is Hermitian, its inversion can be efficiently computed by the means of the Cholesky decomposition, with the need of $\frac{1}{2}K^3$ complex products and additions [14]. Multiplication with the $\mathbf{H}^H$ matrix requires MK^2 complex products, making a total of $K^2(2M+\frac{1}{2}K)$ of multiplyaccumulate (MAC) operations. This procedure should be performed at least once every coherence bandwidth of the channel, while further interpolation may be needed between subcarriers in case of orthogonal frequency-division multiplexing (OFDM) [18]. Let us assume, for simplicity, that one coherence bandwidth equals to a resource block (RB)[9], and that the system is OFDM and time division duplexing (TDD) based, with slot time split equally between uplink and downlink, in a similar way as in [17].

[9] We define a RB as 12 consecutive subcarriers in the frequency domain.

This scheme considers dedicated OFDM symbols for uplink pilots, one per slot. In between two consecutive uplink pilot symbols, all uplink and downlink data with guard symbols need to be accommodated. The weights for precoding need to be computed based on the uplink pilots and be ready before transmission of downlink data symbols, in order to precode the data properly. The required number of MAC (defined as complex multiplication and addition) per second for weights formulation is then determined by

$$C_{\mathrm{w,form}} = K^2 \left(2M + \frac{1}{2}K\right) \frac{N_{\mathrm{PRB}}}{\frac{1}{2}T_{\mathrm{slot}}}, \tag{1.3}$$

where N_{PRB} is the total number of RB allocated for user data transmission, and T_{slot} is the slot duration[10]. Now we give a numerical value with M=1024, K=150, N_{PRB}=275 (referring to 5G NR with 30KHz subcarrier spacing and system bandwidth of 100MHz), T_{slot}=500μs, which leads to $C_{\mathrm{w,form}} \approx$ **52.5 TMAC/s**. To put it value into perspective, this value is three order of magnitude higher than the corresponding in Massive MIMO [4].

Filtering

During filtering, the received signal, containing user data from the antennas is filtered through the weights obtained during the formulation phase. In a similar form, downlink data is precoded using the weights. The filtering process requires KM complex products per subcarrier, with a total number of MACs per second of

$$C_{\mathrm{w,filt}} = KM \frac{N_{\mathrm{sc}}}{T_{\mathrm{OFDM}}} \approx KM f_{\mathrm{B}}, \tag{1.4}$$

where N_{sc} is the number of subcarriers allocated for data transmission, and T_{OFDM} is the OFDM symbol duration. The approximation is valid if the cyclic prefix duration is negligible compared to T_{OFDM}[11]. For our analysis we assume the same values as before, leading to $C_{\mathrm{filt}} \approx$ **15.3 TMAC/s**. The same value is expected for precoding.

Energy considerations

For our energy analysis we take the result obtained during formulation, as it is dominant compared to filtering in terms of computational complexity (in

[10] The spirit of this example is to give an approximative value for computational complexity, that can illustrate potential implementation challenges with LIS. In that context, and for simplification, we do not consider IFFT and channel estimation processing latency. A more accurate analysis would require to give numerical values to each individual processing latency, which is highly dependent on the specific implementation and technology used. [11] As in this case: $T_{\mathrm{OFDM}} \approx \frac{1}{\Delta_f}$, where Δ_f is the subcarrier spacing

our example). If we consider that a complex multiplication can be realized with four real products, and for an energy-efficiency figure of 3.1pJ per MAC [15], then the power consumption during formulation is $\sim$ **651W**. We remark that this is only for weights computation. Other physical layer functionalities such as OFDM modulation/demodulation, upsampling/downsampling, digital predistortion (DPD), channel coding/decoding, etc are not included. However, it can provide a representative value for our analysis.

After presenting the previous results of our analysis, in order to address the expected high computational complexity, we list two directions to explore:

- **Processing distribution:** While processing distribution does not reduce the total computational complexity, it ensures a more uniform allocation of processing resources in the system, and therefore of the energy consumption, facilitating the scalability. Another key property of processing distribution is that by allocating computational resources in the antenna subsystems, only *nearby* users (those received with enough energy) need to be processed[12], reducing the computational complexity significantly due to the $\mathcal{O}(K^3)$ dependency. Other applications, such as AR, also benefit from a distributed processing approach [2].
- **Algorithm complexity:** While the L-MMSE method is known to offer excellent performance, its cost in terms of complexity and energy consumption is relatively high. We look for solutions that can offer a good balance between performance and computational cost, and at the same time the scalability we are looking for.

Both directions can be considered together, as shown in the papers included in this thesis.

1.3.3 Memory capacity

Memory plays a very important role in the energy efficiency and latency of the system. Processors need to access memory to read instructions and data to perform certain tasks. While off-chip memory access is expensive in terms of latency and energy, on-chip access becomes faster and much more energy efficient. To give some perspective, in 45nm CMOS technology, a 32 bit floating point addition requires 0.9pJ, a 32bit SRAM cache access consumes 5pJ, while a 32bit off-chip DRAM memory access demands 640pJ, which is almost three orders of magnitude more expensive than an addition operation [16]. This makes on-chip access very attractive since it is two orders of magnitude more energy efficient than off-chip. However, on-chip memories are very limited in

[12] This is only valid for distributed arrays.

size. The memory requirements in the system, to store the equalizer weights, is indicated as

$$\mathrm{Mem_w} = 2wKMN_{\mathrm{PRB}}. \tag{1.5}$$

To get an idea of the memory requirements of the system regarding weights, let us use the same numerical values as in the previous example, that is, M = 1024, K = 150, N_{PRB} = 275, and w=12. For this case, $\mathrm{Mem_w}$ = 126MB. This is the required memory space only for the weights. Usually there is also a need to buffer received data, which would require additional memory, and would make the integration of such a large on-chip memory very challenging.

Energy considerations

We can give an estimate of the energy consumed by memory accesses due to reading weights during filtering, in order to obtain a representative value of the energy requirements when it comes to memory. We assume no interpolation in weights [18], one memory access per single weight (complex number), and all weights are read within half slot duration. This leads to a power consumption of 0.8W for an on-chip cache, and 108W in an off-chip memory. As we can observe, the difference is substantial, and an on-chip option is highly preferred, as it can be integrated into the processors of each antenna subsystem. However, as shown previously, there are other aspects in the system design that are more relevant from energy consumption point of view.

1.3.4 Synchronization

All elements in a multi-antenna system are required to maintain a certain level of coordination among them, in order to ensure the system works as expected. This coordination can take many forms. One of them is synchronization, which is a critical part of the architecture in distributed systems [28]. It needs to be properly considered in order to avoid systematic errors. Those errors do not average out with the array size, and therefore lead to performance degradation [19]. To get more insight, let us introduce two main types of synchronization in a multi-antenna system:

- **Frequency:** Each antenna branch with its corresponding transceiver needs a high frequency local oscillator (LO) signal for up and down conversion. Even though static or very slowly-varying phase offsets among these LOs can be seen as part of the wireless channel, ideally we prefer to have a common reference shared with all transceivers. This scheme guarantees a tight synchronization under many circumstances. Global

(or centralized) carrier synchronization is considered as the ideal scenario from a performance point of view, but it leads to high energy consumption and scalability limitations. In single-chip arrays with 16-32 transceivers on the die, it is possible to generate a single LO and distribute it to each transceiver in the array. When the array grows and the system becomes multi-chip, or distributed with multiple and distanced panels, routing high frequency LO signals can be very energy consuming. Low frequency reference signal distributed with local LO generation (distributed LO generation) may alleviate theses challenges in spite of introducing slight de-synchronization among LOs, leading to an important design trade-off. Different architectures for LO generation and distribution have been analyzed and compared in [20]. The best trade-off solution seems to point to the distributed LO generation, where each generator or PLL is in charge of a group of transceivers, providing short LO routing and requiring a relatively low number of PLLs. One promising alternative to achieve frequency synchronization is the transmission of RF signals over optical fiber, as explored experimentally in [21]. Similar considerations can be taken into account with the sampling clock frequency and phase in ADC and DAC elements [19, 20].

- **Time:** Synchronization in the time domain is important, and covers two aspects: trigger and timestamp. A trigger signal is used to indicate an event time for the different modules in the system, for example, as a starting time [17]. This allows, for example, to indicate when to start sending samples (in case of downlink) for the first OFDM symbol at the same time. Timestamp is a more evolved form of time synchronization, that allows the system to have an absolute common time-base shared among all nodes in a network. This implies also a common clock frequency and phase.

In real deployments, as the system becomes larger and distributed, ensuring tight synchronization among all nodes may be expensive, energy hungry, and technologically challenging [31]. While there are available technologies to provide synchronization over a large number of nodes and distances such as White Rabbit [22], it is expected that such a tight synchronization level among all of them may not be required in practice, only within the ones that are to cooperate for coherent processing. This implies that those LIS panels (or subarrays) that are beamforming to the same users may need to be tightly synchronized in order to achieve a constructive contribution of the desired signal level at the user. As these panels tend to be physically together, the synchronization is more feasible. This is the principle behind *Federation* concept within the RadioWeaves paradigm [3]. While synchronization is a very important aspect

in the system design, this thesis does not cover it, and perfect synchronization is always assumed among all nodes.

1.4 Thesis structure

The rest of the first part of the thesis contains two more chapters, that address how to exploit scalability through distributed processing techniques, by leveraging topology selection and algorithm design. We specifically focus on communication and positioning services using radio signals, aiming for a unique architecture to support both. In more detail:

- **Chapter 2** gives an introduction to algorithm-topology co-design, topology classification, processing distribution in communications and positioning.
- **Chapter 3** presents the conclusion and future directions.

The second part of the thesis contains six original research papers that cover the areas described in the first part. In more detail:

- **Paper I** proposes three different algorithms for uplink equalization tailored for daisy-chain topology and sequential processing.
- **Paper II** explores channel estimation in decentralized processing systems, including MMSE estimation and performance evaluation.
- **Paper III** extends the results from Paper I by proposing an algorithm for uplink equalization and daisy-chain topology, including closed-form performance expressions, detailed system level analysis and the required hardware resources.
- **Paper IV** is a first look into hardware implementation of LIS, proposing a panel architecture and interconnection topology to exploit processing distribution, and establishing guidelines for system design.
- **Paper V** extends the results from Paper IV by proposing a system architecture, including panels, complete interconnection network topology, and corresponding algorithms. The paper also includes a first-order approximation of the hardware resources needed for system deployment, and establish interesting trade-offs, serving as guidelines for a system designer.

- **Paper VI** explores the positioning problem in distributed LIS systems, by proposing an algorithm and topology solution, that provides high accuracy with relatively low hardware cost.

Chapter 2

Processing Distribution and Algorithm-Topology Co-design

As described in previous chapter, future RAN infrastructures will enable multiple applications under highly demanding requirements. The system designer should translate these into key design decisions, such as: carrier frequency, total number of antennas in the array, system bandwidth, number of processing nodes[1] and how they are connected, algorithms that are executed, etc. These can be formulated as low-level hardware requirements, such as: computational complexity, interconnection data-rate, latency, and memory. This together with implementation decisions, such as the hardware platform to use (FPGA, ASIC, ASIP, etc), technology node, use of accelerators, etc, lead to an estimate of the energy consumption and latency. This process is graphically illustrated in Fig. 2.1. These specifications will constraint the number of available design possibilities, making certain options more appropriate than the others in terms of cost, latency and energy consumption. In summary, there may be multiple solutions that fulfill the applications requirements while offering different hardware cost.

As the number of nodes grows in the system, the system design process becomes more complex. In this thesis, we focus on an infrastructure composed of multiple interconnected nodes (or panels) with local computing capabilities, as was motivated in the previous chapter. In the next sections we describe how

[1] Element in the system with processing capabilities. It is also refered as panel in the context of LIS.

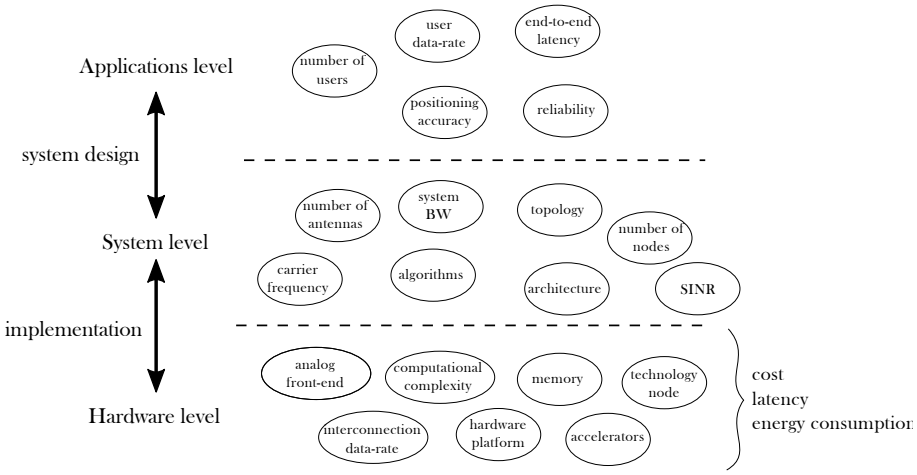


Figure 2.1: Requirements flow during system design and implementation phases. Applications requirements can be translated into system-level design choices, which can then be formulated in terms of low-level hardware requirements and decisions. Different solutions may fulfill the applications requirements while having different cost, latency and energy consumption.

processing is distributed and mapped onto the nodes, and how these are connected. The three important aspects determining the processing distribution of the system are: architecture[2], topology and algorithm. In the next section we will provide a clear definition for these three, that will be applied throughout the rest of the chapter.

2.1 Distributed processing: architecture, topologies and algorithms

Under the distributed processing paradigm, the different nodes in the system should have analog and digital processing capabilities. **Architecture** involves a detailed mapping of this processing, by dividing it in separate parts, each with specific functionality and requirements, and the connection of such parts. One possible internal architecture of a processing node (or panel) is shown in Fig. 2.2, where many of the relevant processing elements have been depicted,

[2] It is important to remark that the concept of architecture here refers only to demand on processing, and not protocols.

together with the respective connections.

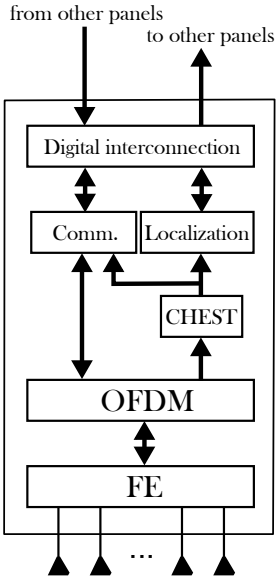


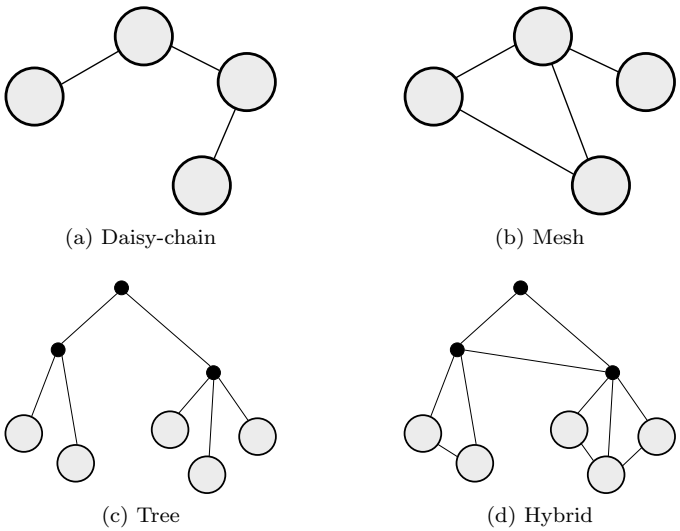
Figure 2.2: Processing architecture of a processing node or panel, capable of providing communications and localizations services, assuming OFDM-based wireless access. CHEST stands for Channel Estimation. Front-End (FE) comprises analog front end (AFE), ADC/DAC, and digital front end (DFE). Digital interconnection includes data exchange and synchronization.

The Front-End (FE) is connected to the antennas. It comprises the AFE with all the analog processing tasks in the panel, the ADC/DAC, and the DFE, with up/downsampling and filtering. In the digital baseband domain, a OFDM block (assuming OFDM-based wireless access) performs frequency-time transformation, followed by channel estimation (CHEST), communication and localization processing.

The respective processing results are collected and transmitted to other panels via the digital interconnection sub-block. At the same time, results from other panels can arrive and be used to refine the ones obtained locally. Even tough the architecture supports this, it is up to the **algorithm** (which is running in the panel) to decide what operations are performed and what data to exchange with other panels. Algorithms will be covered in Section 2.3, with a focus on communication and localization.

Processing nodes can be connected in many different ways, leading to spe-

cific enhancements in certain aspects or potential limitations in others, as we will see in Subsection 2.2. By using the term connection, we refer to logical connection, indicating the way the information flows in the system[3]. The way nodes are interconnected is defined as **topology**. In this section four of the most relevant ones in this context are described and shown in Fig. 2.3. Additionally, we also provide a brief analysis of them and a comparison.



(a) Daisy-chain (b) Mesh

(c) Tree (d) Hybrid

Figure 2.3: Illustration of different topologies covered in this thesis. Grey circles and black dots represent processing nodes. While the former ones typically contain antennas, the latter ones may play the role of *aggregator* with or without antennas.

The third aspect is the **algorithm**, that specifies the operations in terms of information processing to be performed in order to achieve a certain goal. This may lead to the exchange of data with other nodes in the system. As we can imagine, the topology plays an important role here. As the algorithm is mapped onto the topology, the pattern of data traffic becomes defined, together with the interconnection data-rate in the links. This mapping has also influence in the latency. For example, if the algorithm indicates that node 1 should send data to node 4, that may imply three hops in daisy-chain topology, while it may

[3] In this context, logical connection refers to a wired or wireless connection between two nodes where data may be exchanged in both directions. Both nodes do not have to be physically connected, and other nodes without active role in the communication (such as relays) may be in between them.

be only one in mesh. If both nodes need to exchange data at a high rate, it is beneficial to place a dedicated link between them, as that would reduce latency, and possibly routing congestion. The price to pay is to deploy and maintain a dedicated link, whose cost depends on the technology and the distance. In this case, we are questioning the topology by asking: *"Is this topology suitable for the data exchange flow imposed by the algorithm?"*. On the other hand, we can question the algorithm by asking: *"Is the pattern of data exchange between nodes appropriate for the current topology?"*, followed by a further one: *"Is the exchange of data between these two nodes really needed?"*. Unfortunately, usually none of these questions have an easy answer. In general, we could start the design process by choosing an existing infrastructure topology and then select a suitable algorithm for that one, or we could also start with an existing algorithm in literature and proposing a matching topology for it. Probably, none of the approaches lead to the optimal solution from resources and energy consumption point of view. Rather, these two can be seen as part of an iterative method, named **topology-algorithm co-design**, where the goal is to obtain a system design, in which the topology and algorithm are matching, in a way that the algorithm has low computational complexity, and the links available through the topology are in low number and efficiently used[4]. As an example, we refer to the centralized processing scheme illustrated in the previous chapter, where the algorithm and topology were not an optimal choice, and that motivated the search for a distributed approach. Unfortunately, in the topology-algorithm co-design method it is difficult to have a systematic form to approach the solution. Therefore, we will based our analysis in certain heuristics that provide satisfactory solutions for the applications under demand. In the next section we discuss some of these.

In reality, topology-algorithm co-design can be seen as a tool for the system designer in order to find an appropriate topology and algorithm, that match together. Once the system design is completed, the resulting selected topology, algorithms and architecture (among other factors) act as inputs for the hardware implementation. This allows to obtain estimates of cost, latency and energy of the system (as illustrated in Fig. 2.1), which need to fulfill the operator requirements[5]. If those are not met, it is necessary to return to the system design process and to iterate through different design options (as illustrated in Fig. 2.4). The process ends when the applications and operator requirements are mutually met.

[4] Here the term "efficient" is very broad, and includes high occupancy of the link, with low redundancy in the data. Additionally, low redundancy is expected among data carried through different links. [5] Infrastructure operators may impose system level requirements on the equipment such as: maintenance cost, energy consumption, total throughput in the area of service, support of a certain technology, latency, etc.

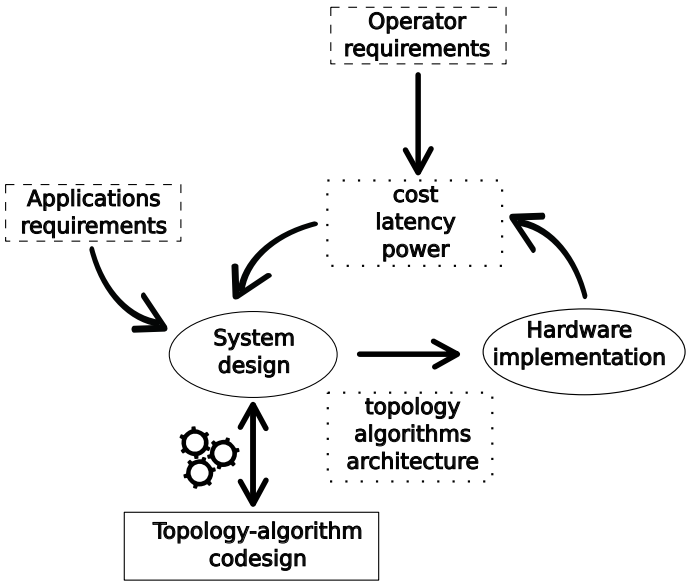


Figure 2.4: System design and hardware implementation cycle. The system design process is assisted by the topology-algorithm co-design method, in order to leverage suitable topology, algorithms, and architecture that has the potential to fulfill the different set of requirements coming from the applications and the operator. Multiple iterations may ne needed to meet the specifications.

In this section we have introduced our methodology, the topology-algorithm co-design. Before entering into details regarding topology, algorithm and architecture, here we would like to provide a short guideline about how these details are structured. In Section 2.2, we will go through several well known topologies that may be used for distributed processing architectures. In Section 2.3 we will introduce algorithms for communications. We will start with standard linear methods, and then we will present several strategies on how to distribute these standard methods into different algorithms. In Section 2.3 we will map the distributed algorithms onto the topologies introduced in Section 2.2.

2.2 Topologies: description and analysis

In this section the different topologies already presented in Fig. 2.3 are briefly described and compared based on four different criterion: scalability, number

of links, latency and reliability. This serves as a useful guideline for the system design [3].

2.2.1 Daisy-chain

In the daisy-chain topology, shown in Fig. 2.3a, each node is connected to two neighboring ones (except both ends) with bidirectional links. There are two benefits of this topology: 1) **scalability**, as adding one additional node into the system is very straightforward, only requiring to establish one or two connections (depending if edge node or not); and 2) low number of **physical connections** compared to other topologies.

On the other hand, there are two main limitations of this topology: 1) relatively high **latency**, as collecting results from each node requires a number of hops equal to the number of nodes in the system (N), which may limit its use in latency-critical applications; and 2) low **reliability**, as a failure in any of the nodes may lead to a partial or total outage of the infrastructure.

As a summary, daisy-chain is a solid candidate in scenarios where latency is not critical, and deployment should be easy (minimum reconfiguration when a new node is added).

2.2.2 2-D mesh

2-D mesh topology, shown in Fig. 2.3b, can be seen as the natural extension of daisy-chain in the 2-D world, where each node is connected to more than one neighbor[6]. These extra connections help to reduce the number of hops between any pair of nodes. In the particular case of a mesh-grid, it scales with $O(\sqrt{N})$, which is a significant reduction compared to the daisy-chain case. Additionally, the new connections improve the reliability issue presented in the 1-D case, enabling re-routing if a node fails.

On the other hand, real implementations may become more complicated as each node may receive data from multiple neighbors, which may require a special attention to the synchronization among the nodes. The different number of connections in the nodes may lead to a diversity of implementations/requirements at the nodes that are difficult to predict in advance, requiring higher degree of reconfigurability. The increment in the number of connections compared to daisy-chain is another disadvantage, which may impose limitations in the deployment over large areas, making this topology more preferable when nodes are physically co-located.

[6] Even tough this definition is very broad and cover all topologies, we restrict to provide a general evaluation for this topology, while for the rest a more specific description is presented.

2.2.3 Multi-level tree

Multi-level tree topology introduces hierarchical levels, in contrast to daisy-chain where all nodes are organized in a flat structure. Tree topology, shown in Fig. 2.3c, establishes a single node as root, connected to a certain number of nodes, which are also connected to other nodes in a recursive fashion. We consider two different types of nodes in this thesis. First, leaf nodes, depicted as circles in the figure, which contain antennas. Second, *aggregators*, depicted as black dots, that may but do not have to contain antennas. The latter ones may collect, process, and deliver data (depends on the algorithm).

A significant reduction in latency is expected in this topology, as the number of hops between two nodes scales now with $O(\log N)$.

The potential limitations come from reliability and imbalanced computational load. The existence of hierarchies, with some nodes serving as *fusion nodes* or *aggregating points*, impose strict requirements and workload in those nodes, leading to potential computational and inter-connection bottlenecks if the algorithm is not carefully selected for this topology. Additionally, failure in one of these nodes may lead to a partial or total outage.

To sum up, multi-level tree is an convenient selection in applications where latency is critical. The algorithm is crucial, as it needs to distribute processing across the nodes and limit the dependence on the aggregation nodes. Examples of such algorithms are shown in the next section.

2.2.4 Hybrid topology

Hybrid is a variant of a previously presented topology, with the aim to enhance it in certain aspects. In this thesis, we consider hybrid topology as a multi-level tree with direct connections between the nodes of the same level. This extra links may help to reduce latency (if the algorithm is properly designed) and increase reliability. It is depicted graphically in Fig. 2.3d.

Table 2.1: Illustration and comparison of the different topologies discussed in this thesis with respect to different characteristics: scalability, number of links, latency and reliability. Symbols definition: $\uparrow$ = high, $\downarrow$ = low, and - = average

	Daisy-chain	Mesh	Tree	Hybrid
Scalability	$\uparrow$	$\downarrow$	-	-
Number of links	$\downarrow$	$\uparrow$	-	-
Latency	$\uparrow$	-	$\downarrow$	$\downarrow$
Reliability	$\downarrow$	$\uparrow$	-	$\uparrow$

In Table 2.1 a summary of the presented topologies is shown, together with the relative strengths in certain areas.

2.3 Algorithms for communications

The design process of an efficient system with processing distribution for communication and localization is not straightforward. One promising heuristic to reach our goal is performing local **dimensionality reduction** in the panel, as it seems to be the right direction to alleviate the implementation issues presented in centralized processing (as shown in Chapter 1). This approach can be combined with another heuristic, the **data locality principle**, according to which data should be consumed as close as possible to where it is generated. These principles, that can be applied to panel baseband processing, can be summarized as follows:

- **Per-user processing:** Performing per-user processing (apart from per-antenna processing) is the first step to reduce dimensionality and enable scalability. In case of communications, the ratio antennas to spatially-multiplexed users can be in the order of 10 or even more, leading to substantial reductions in dimensionality. Moving from antenna-domain to user-domain processing requires to map an equalization and localization algorithm into the processing node (or panel) for the communications and localization cases. A significant dimensionality reduction is expected to be achieved by following this principle.
- **User scheduling:** In spatially distributed and very large arrays as LIS, the system may cover a large geographical area. Therefore, it is expected that a part of the it, such as a panel, receives sufficient signal level from a limited number of users, typically the ones closer. An energy-efficient strategy for resources allocation[7] should focus exclusively on those users. Additionally, in many real scenarios, the users are moving, what demands a dynamic reconfiguration and resource allocation in the system.

CSI is used in communication and localization services, as illustrated in Fig. 2.2 with the panel processing architecture. More specifically, CSI in the communication context allows for a proper equalization of the received symbols during uplink, and support for an adequate beamforming during downlink. On

[7] By resource allocation we refer to radio (frequency and time) and hardware resources (processing and memory).

the other hand, in localization, CSI can be used in machine learning-based fingerprinting methods, such as in **Paper VI** and [33–36]. In these methods, a machine learning algorithm, typically a neural network, is trained to provide an estimate of the user location from the obtained CSI. In both cases, communications and localization, the data locality principle and local per-user processing introduced before can be applied. This implies that each panel has knowledge of local channel information exclusively, and no element in the system has full knowledge of the complete channel information. The above mentioned assumption is the key to ensure scalability and efficient use of resources.

In this section, we first introduce different existing methods in the literature for communications, specifically downlink precoding and uplink equalization, followed by concrete algorithms based on those. In the second part, we discuss algorithms for localization.

2.3.1 System model

Let us consider a distributed LIS infrastructure with M antenna elements and N panels, serving to K single-antenna users, as shown graphically in Fig. 2.5. For simplicity we assume all the panels have same number of antennas, which is: $M_{\mathrm{p}} = \frac{M}{N}$.

The radio access between the LIS and users is assumed to be based on TDD and OFDM [8]. The $M \times K$ channel matrix $\mathbf{H}$ can be written as $\mathbf{H} = [\mathbf{H}_1^T, \mathbf{H}_2^T, \cdot, \mathbf{H}_{\mathrm{N}}^T]^T$, where $\mathbf{H}_i$ is the $M_{\mathrm{p}} \times K$ channel matrix of the i-th panel. In case of uplink, the signal received by the antennas follows the next relationship

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{2.1}$$

where $\mathbf{x}$ is the $K \times 1$ users transmitted signal, and the $M \times 1$ noise vector $\mathbf{n}$ is modeled with Gaussian i.i.d. elements: $\mathbf{n} \sim \mathcal{CN}(0, \sigma_{\mathrm{n}}^2\mathbf{I})$. In the case of communication, the transmitted signals from users may correspond to pilots or data, where the latter one is assumed to be random and mutually independent. For localization, it only consists of pre-determined pilots for channel estimate.

2.3.2 Communications: Equalization and precoding

For simplicity we will reduce our scope to linear methods, including maximum ratio transmission (MRT) and maximum ratio combining (MRC), zero forcing (ZF), and minimum mean square error (MMSE). In case of uplink, for linear equalization methods, the transmitted data vector estimated is obtained as

$$\widehat{\mathbf{x}} = \mathbf{W}\mathbf{y}, \tag{2.2}$$

[8] Further exploiting the spatial domain is a main goal of LIS, and OFDM-TDD is the preferred solution under this assumption.

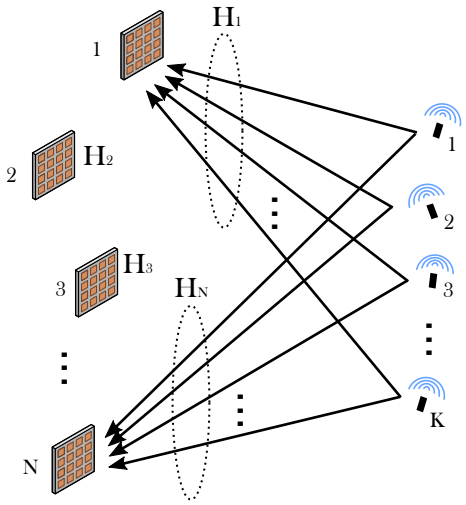


Figure 2.5: System model. K users transmitting simultaneously to a LIS made up of N panels.

where $\mathbf{W}$ is a $K \times M$ complex matrix, which can be written as $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_{\mathrm{N}}]$, where $\mathbf{W}_i$ is the $K \times M_{\mathrm{p}}$ filtering matrix corresponding to the i-th panel. Similarly for $\mathbf{y}$, it can be written as $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \cdots, \mathbf{y}_{\mathrm{N}}^T]^T$.

For downlink precoding, the $M \times 1$ vector $\mathbf{y}$ at the antennas is given by

$$\mathbf{y} = \mathbf{P}\mathbf{x}, \tag{2.3}$$

where $\mathbf{x}$ is the $K \times 1$ data vector to transmit, $\mathbf{P}$ is the $M \times K$ precoder matrix, which can be also written as $\mathbf{P} = [\mathbf{P}_1^T, \mathbf{P}_2^T, \cdots, \mathbf{P}_{\mathrm{N}}^T]^T$, where $\mathbf{P}_i$ is the $M_{\mathrm{p}} \times K$ precoding matrix corresponding to the i-th panel.

Each panel estimates the channel locally based on orthogonal pilots sent by the users. This channel information is then used to compute the filtering matrix for uplink detection and downlink precoding.

In this thesis, and for performance evaluation purposes only, we will assume a block-fading channel model, where channel remains approximately constant over a coherence block (frequency-time). However, in real implementation there may be a need for interpolation in frequency and time domain (across OFDM symbols) in order to estimate channel response in between subcarriers, which may impose more severe constraints from computation time point of view, when calculating the precoder weights [18]. In that sense, there is a need to verify the model with real measurements.

We now revise some of the most common methods for equalization and precoding, followed by different distributed processing strategies, including several algorithms to implement these methods.

Maximum Ratio Transmission (MRT) and Maximum Ratio Combining (MRC)

MRT/MRC is a technique for downlink precoding and data filtering in uplink, where the goal is to maximize signal-to-noise ratio (SNR) at user equipment (UE) (case of MRT), and in the base station receiver (case of MRC). The vector of used weights is calculated directly from the channel estimate, more formally: $\mathbf{W}_i = \mathbf{H}_i^H$, and $\mathbf{P}_i = \alpha \mathbf{H}_i^*$, where α is an scalar for meeting transmit energy constraint.

MRT/MRC are highly suitable under the distributed processing scheme as there is no exchange of data required for interference cancellation, which indeed reduces baseband processing latency and interconnection bandwidth. This enables all baseband processing to be performed locally in the panel, including equalization and precoding.

The downside of this method lies in its limitations to cope with inter-user interference, which is critical in the scenarios with large number of users. Those scenarios are typically limited by interference (instead of noise), and therefore reducing significantly the achievable rate for the users. In conclusion, this method can be a good candidate when the number of users is relatively high but the individual throughput demand is low.

Zero-forcing (ZF)

Zero-forcing (ZF) is another linear method for uplink equalization and downlink precoding, that aims to cancel user interference. Having no interference makes the system capable of operating with a larger number of users and, therefore, considerably increase the capacity of the infrastructure to transmit and receive information. To achieve that, panels need to exchange data (in contrast to MRT/MRC) with the consequent implications in terms of latency and interconnection bandwidth. The filtering matrix in ZF is defined as $\mathbf{W} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$, while the precoding matrix is defined as $\mathbf{P} = \alpha\mathbf{H}^*(\mathbf{H}^H\mathbf{H})^{-1}$, and α is a scalar.

Minimum mean square error (MMSE)

While ZF filtering is able to cancel user interference, at the same time it may enhance the received noise. To mitigate this issue, MMSE[9] can be used, as it presents the best trade off between interference cancellation and noise enhancement, by maximizing post-filtering signal-to-interference-plus-noise ratio (SINR). The filtering matrix is in this case defined as $\mathbf{W} = (\mathbf{H}^H\mathbf{H}+\sigma_n^2\mathbf{I})^{-1}\mathbf{H}^H$, while the precoding matrix is defined as $\mathbf{P} = \alpha\mathbf{H}^*(\mathbf{H}^H\mathbf{H} + \sigma_n^2\mathbf{I})^{-1}$.

The above presented methods require the calculation of the weights first by a procedure called *formulation*, and subsequently use those weights during *filtering/precoding*. In case the block-fading channel model is applicable in the context of LIS, it is possible to reuse the same weights for all transmitted signals under the same channel coherence block (time-frequency).

It is important to notice that different algorithms can be used to implement the methods presented before. We are focused on algorithms that enable distributed processing. In this section we describe some of the available algorithms in the literature. All the algorithms listed below are linear equalizers and follow (2.2):

Channel Gram matrix adder

This algorithm is able to achieve exact ZF/MMSE solution, and it requires the computation of the matrices $\{\mathbf{G}_i\}$, where $\mathbf{G}_i = \mathbf{H}_i^H\mathbf{H}_i$. These matrices are added and the resulting matrix inverted, and therefore obtaining matrix $\mathbf{D}$ as follows

$$\mathbf{D} = (\mathbf{H}^H\mathbf{H} + \mathbf{G}_0)^{-1} = \left(\sum_{i=1}^{N}\mathbf{G}_i + \mathbf{G}_0\right)^{-1}, \tag{2.4}$$

where $\mathbf{G}_0$ is a constant term and $\mathbf{G}_0 = \mathbf{0}$ in case of ZF, and $\mathbf{G}_0 = \sigma_{\mathrm{n}}^2\mathbf{I}$ in case of MMSE. The resulting matrix $\mathbf{D}$ is multiplied with the corresponding channel matrix to obtain the filtering weights as follows

$$\mathbf{W}_i = \mathbf{D}\mathbf{H}_i^H. \tag{2.5}$$

When it comes to processing distribution, this algorithm allows for a mapping where $\mathbf{H}_i$ is obtained locally in panel i, together with the local computation of $\mathbf{G}_i$ and $\mathbf{W}_i$.

[9] It is worth to mention that when data vector $\mathbf{x}$ is assumed to follow a multivariate Gaussian distribution then the linear MMSE estimator described here is actually the MMSE estimator.

Coordinate descent (CD)

Paper III introduces an approximate ZF method based on coordinate descent (CD) for daisy-chain topology in Massive MIMO scenarios. With a very low computational complexity formulation, this algorithm achieves very good interference cancellation properties without the need of explicit matrix inversion. The pseudocode is shown in Algorithm 1, where matrix $\mathbf{A}_i$ is a $K \times K$ complex matrix, and $\mathbf{w}_i$ the corresponds to the $K \times 1$ equalizer vector of i-th antenna. Multiple optional iterations through the array improve the performance, closing the gap to ZF, but at the expense of an increment in latency.

Algorithm 1: Coordinate descent

Input : $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2 \cdots \mathbf{h}_M]^T$
Preprocessing:
$\mathbf{A}_0 = \mathbf{I}_K$
for $i = 1, 2, ..., M$ **do**
  $\mathbf{w}_i = \mu_i \mathbf{A}_{i-1} \mathbf{h}_i$
  $\mathbf{A}_i = \mathbf{A}_{i-1} - \mathbf{w}_i \mathbf{h}_i^H$
end
Output : $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2 \cdots \mathbf{w}_M]^T$

Approximate ZF precoder

An approximate decentralized Massive multiple-input multiple-output (MIMO) ZF precoder for daisy-chain topology is introduced in [38]. The algorithm performs close to ZF in scenarios when the number of antennas is large compared to the number of users. The pseudocode is presented in Algorithm 2, where $\mathbf{p}_i$ represents the i-th row of precoder matrix $\mathbf{P}$, corresponding to the $1 \times K$ precoding vector of the i-th antenna. The selection of the scalar ϵ is detailed in [38]. Although it shares some similarities with the CD algorithm, the downlink requirement that all antennas must transmit the same mean energy calls for a different approach and solution.

iterative interference cancellation (IIC)

Paper V introduces the IIC algorithm suitable for distributed uplink processing in LIS scenarios. This technique exploits user scheduling principle by filtering the received signal from a panel with a certain number of vectors. These ones are related to the singular vectors of the channel matrix associated with such panel. The number of used vectors is related to the number of

Algorithm 2: Approximate ZF precoder

Input : $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2 \cdots \mathbf{h}_M]^T$
Preprocessing:
$\mathbf{A}_0 = \mathbf{I}_K$
for $i = 1, 2, ..., M$ **do**
  $\mathbf{p}_i = \epsilon \frac{\mathbf{h}_i^H \mathbf{A}_{i-1}}{\|\mathbf{h}_i^H \mathbf{A}_{i-1}\|^2}$
  $\mathbf{A}_i = \mathbf{A}_{i-1} - \mathbf{h}_i \mathbf{p}_i$
end
Output : $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2 \cdots \mathbf{p}_M]^T$

nearby users, and should be much lower than the total number of them in the system. Furthermore, certain information is shared throughout the different filtering vectors, in order to mitigate the inter-user interference. The result of this technique is a dimensionality reduction processing pipeline, which is able to achieve good performance with low computational complexity and interconnection bandwidth. The algorithm pseudocode is presented below. $\mathbf{Z}$ is a complex $K \times K$ matrix.

Algorithm 3: Algorithm IIC

Input : $\{\mathbf{H}_i\}$
$\mathbf{Z}_0 = \mathbf{I}_K$
for $i = 1, 2, ..., N$ **do**
  $[\mathbf{U}_z, \mathbf{\Sigma}_z] = \text{svd}(\mathbf{Z}_{i-1})$
  $\widetilde{\mathbf{H}}_i = \mathbf{H}_i \mathbf{U}_z \mathbf{\Sigma}_z^{-1/2}$
  $\widetilde{\mathbf{U}} = \text{svd}(\widetilde{\mathbf{H}}_i)$
  $\mathbf{W}_i = \widetilde{\mathbf{U}}(:, 1 : N_\text{p})$
  $\mathbf{Z}_i = \mathbf{Z}_{i-1} + \rho \mathbf{H}_i^H \mathbf{W}_i \mathbf{W}_i^H \mathbf{H}_i$
end
Output : $\{\mathbf{W}_i\}$

When it comes to processing distribution, this algorithm allows for a mapping where $\mathbf{H}_i$ is obtained locally in panel i, together with the local computation of $\mathbf{W}_i$ and $\mathbf{Z}_i$. $\mathbf{Z}_i$ is passed from the i-th panel to the $i+1$-th panel. Local and fast connections can be used for this purpose (more details in Paper V).

2.3.3 Sequential estimation

The second group of algorithms presented in this section aims to achieve a sequence of estimates during the uplink filtering process, as follows: $\widehat{\mathbf{x}}_1 \rightarrow \widehat{\mathbf{x}}_2 \rightarrow \cdots \rightarrow \widehat{\mathbf{x}}_N$. Each element of the sequence is computed based on the previous element together with local observations from a certain panel, more specifically: $\widehat{\mathbf{x}}_i = f(\mathbf{y}_i, \mathbf{H}_i, \widehat{\mathbf{x}}_{i-1})$. Typically we are interested in linear combiners in the form

$$\widehat{\mathbf{x}}_i = \mathbf{A}_i \mathbf{y}_i + \mathbf{B}_i \widehat{\mathbf{x}}_{i-1}, \tag{2.6}$$

where the matrices $\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ are calculated during the formulation phase and may depend on the local CSI. The algorithms following this approach implement linear equalization methods. More specifically, (2.6) is a special case of (2.2). This can be seen when expanding the recursive expression in (2.6) we get

$$\begin{aligned} \widehat{\mathbf{x}}_i &= \mathbf{A}_i \mathbf{y}_i + \mathbf{B}_i \left(\mathbf{A}_{i-1} \mathbf{y}_{i-1} + \mathbf{B}_{i-1} \widehat{\mathbf{x}}_{i-2} \right) \\ &= \sum_{j=1}^{i} \prod_{k=j+1}^{i} \mathbf{B}_k \mathbf{A}_j \mathbf{y}_j. \end{aligned} \tag{2.7}$$

For the last estimate of the sequence, we have

$$\widehat{\mathbf{x}} = \widehat{\mathbf{x}}_N = \sum_{j=1}^{N} \prod_{k=j+1}^{N} \mathbf{B}_k \mathbf{A}_j \mathbf{y}_j. \tag{2.8}$$

By comparing (2.8) and (2.2) we can observe that

$$\mathbf{W}_i = \prod_{k=i+1}^{N} \mathbf{B}_k \mathbf{A}_i, \tag{2.9}$$

which corresponds to a linear filter. The spirit of the update rule following (2.6) is to develop iterative algorithms that can be executed sequentially. This property can be exploited when mapping the algorithm onto the topology as will be described in the following Section 2.4. These algorithms, as presented below, aim to implement exactly or approximately the equalization methods presented before (ZF and MMSE). We describe two algorithm following this update rule as follows:

Recursive least squares (RLS) and sequential MMSE (S-LMMSE)

Paper I proposes a distributed processing scheme to implement the RLS method, which is a recursive version of the traditional Least squares (LS) solution to the optimization problem $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$. For zero-mean Gaussian

distributed noise, this solution matches with the output of the known ZF filter as shown in Subsection 2.3.2. Similarly, the same idea can be applied to the sequential version of LMMSE (S-LMMSE), described in [40] and defined as

$$\widehat{\mathbf{x}}_i = \widehat{\mathbf{x}}_{i-1} + \mathbf{K}_i(\mathbf{y}_i - \mathbf{H}_i\widehat{\mathbf{x}}_{i-1}), \tag{2.10}$$

where

$$\mathbf{K}_i = \mathbf{M}_{i-1}\mathbf{H}_i^H(\sigma_n\mathbf{I} + \mathbf{H}_i\mathbf{M}_{i-1}\mathbf{H}_i^H)^{-1}, \tag{2.11}$$

and

$$\mathbf{M}_i = (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i)\mathbf{M}_{i-1}, \tag{2.12}$$

where $\mathbf{K}_i$ and $\mathbf{M}_i$ are matrices. To initialize, we take $\widehat{\mathbf{x}}_0 = \mathbf{0}$, and $\mathbf{M}_0 = \mathbf{I}$.

Serial CD

The CD algorithm described before can be expressed in a serial form as presented in **Paper III**. The key idea of this algorithm is to replace the matrix $\mathbf{K}_i$ in (2.10) with another one, which does not require matrix inversion. This provides a significant reduction in computational complexity, in exchange of a performance loss compared to the MMSE method. The update rule is defined as follows, for the particular case of one antenna per iteration step

$$\begin{aligned} \varepsilon_i &= y_i - \mathbf{h}_i^T\hat{\mathbf{x}}_{i-1} \\ \hat{\mathbf{x}}_i &= \hat{\mathbf{x}}_{i-1} + \mu_i\mathbf{h}_i^*\varepsilon_i, \end{aligned} \tag{2.13}$$

where $\mu_i = \frac{\mu}{\|\mathbf{h}_i\|^2}$. This algorithm works on per-antenna basis, where y_i and $\mathbf{h}_i$ are the observation and channel vector of the i-th antenna respectively.

2.4 Mapping algorithms to topologies

In this section we briefly introduce certain considerations when mapping algorithms to topology. As was mentioned in Section 2.1, for a certain processing architecture, the selection of algorithm and topology involves a wide variety of system metrics, raging from communication performance (capacity, bit-error-rate, etc) to energy consumption. This selection should be a joint process within an optimization framework, which was previously described as **topology-algorithm co-design**, and illustrated in Fig. 2.4 as part of the system design process.

There are some considerations to take into account when mapping an algorithm:

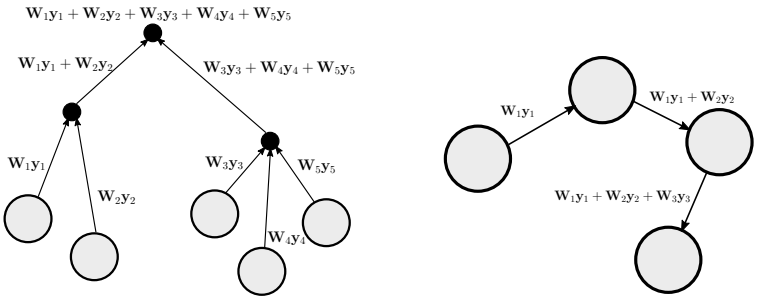
- The algorithm demands a certain level of computational complexity dependent on the type of operations (matrix-matrix product, matrix-vector product, accumulation, etc), and operation rate (MAC/s).
- The processing architecture provides the computational capabilities and interconnection resources (number of I/O ports, data-rate, etc) present in the different nodes in the system.
- The topology tells us how the nodes are connected, which translates to the number of hops needed to move data from one node to another one.

Before entering into details, it is important to realize that all algorithms presented before in Subsection 2.3.2 can be mapped into each of the topologies presented in Subsection 2.2. However, the impact in computational complexity and interconnection data-rate at each node and link is different, which implies that some topologies are more appropriate for certain algorithms. When the algorithm is mapped onto the topology, not only the operations to be performed at each node are determined, but also the interconnection data-rate and data movement in the system (i.e.: transfer data X from node 1 to node 3).

Algorithms that follow the filtering approach in (2.2) can be efficiently mapped to different topologies. The linear equalizer general form in (2.2) can be expressed as

$$\widehat{\mathbf{x}} = \sum_i \mathbf{W}_i \mathbf{y}_i, \tag{2.14}$$

where $\mathbf{y}_i$ is the corresponding received vector at the antennas of the i-th panel. We can conveniently compute locally and store each filtering matrix $\mathbf{W}_i$ at the corresponding panel, which allows to perform the multiplication $\mathbf{W}_i\mathbf{y}_i$ also locally at the i-th panel. These partial results can be aggregated (summed) throughout the infrastructure until the corresponding aggregation point, root node, or CPU, that performs final detection and decoding. This approach allows all panels or nodes to work simultaneously on the same subcarrier during filtering, and once the results are ready, they can be conveniently aggregated throughout the infrastructure. In that regard, it offers high degree of flexibility regarding how the accumulation can be performed, and that includes the topology. In low-latency applications, a tree-based topology can be an ideal candidate due to its adder-tree. The cost to pay is the number of required links, which is larger than in other topologies, such as the daisy-chain. Figures 2.6a and 2.6b represent the aggregation process in the tree and daisy-chain topologies, respectively, when mapped with any of the algorithms under this approach. These two are examples, and combinations of both topologies are also possible.



(a) Tree (b) Daisy-chain

Figure 2.6: Mapping algorithms under the filtering approach in (2.14) onto tree and daisy-chain topologies.

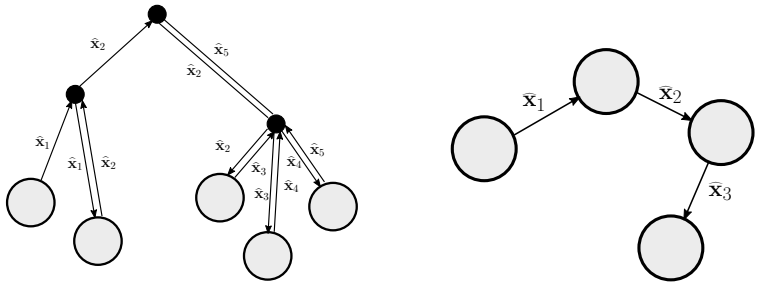
In the case of 2-D mesh, the large number of links increase the reliability of the system to a node failure as presented in Subsection 2.2.

Regarding the algorithms following the sequential filtering format shown in (2.6), the idea is to map each iteration (computation of each estimate in the sequence) to a different physical node or panel. Under this approach, the CSI and the received signals from the antennas do not need to be exchanged with other nodes, and therefore saving in inter-connection data-rate. During filtering, the estimates are passing from node to node, while during formulation the exchange consists of data aiming for interference mitigation. Certain topologies are more suitable for this type of algorithms. Evidently, daisy-chain is a very good fit, while tree may suffer from an increment in interconnection data-rate and latency. This is illustrated in Fig. 2.7a, where the computed estimates need to traverse the tree to follow the sequential processing order, which may imply multiple hops to reach the next processing node.

2.4.1 Initial analysis of hardware requirements

In this subsection we present an initial analysis of the presented algorithms and topologies based on their hardware requirements. Results for uplink filtering are shown in Table 2.2, where for simplicity no dimensionality reduction is assumed (panels process all users, K). Dimensionality reduction techniques such the ones proposed in **Paper V** reduce computational complexity and interconnection data-rate in exchange of potential performance loss. The paper includes a detailed analysis of the hardware requirements.

For simplicity we only cover two topologies, daisy-chain and tree. From Table 2.2 we can observe that sequential algorithms following (2.6) imply twice



(a) Tree (b) Daisy-chain

Figure 2.7: Mapping algorithms under the filtering approach in (2.6) onto tree and daisy-chain topologies.

as much computational complexity as the ones following (2.14), making computational complexity the main drawback of the sequential approaches. Additionally, the sequential tree requires twice as much interconnection data rate compared to the other approach due to the links being used in both directions, as illustrated in Fig. 2.7a.

Daisy-chain and corresponding algorithms seem to be a reasonable choice when latency is not critical, as it simplifies control and enables easy scalability. For use cases where latency is the main concern, the number of elements in the chain has to be kept below a certain value. Another option is to select tree topology instead, as we explain next.

In general, a parallel tree (and a hybrid) seems adequate for latency critical applications with a large number of nodes. It shows the lowest hardware requirements, except for the number of links, which is larger than daisy-chain. However, as the number of children per node increases in the tree (d in Table 2.2), the number of links becomes the same as for a daisy-chain.

Table 2.2: Analysis of hardware requirements for algorithms following the parallel (2.14) and serial (2.6) approaches in daisy-chain and tree topologies. Only uplink filtering is considered. Computational complexity is analyzed in a node/panel, while interconnection data refers to any link in the system. No dimensionality reduction assumed. M_p: number of antennas per node. N: number of nodes. d: maximum number of children per node in the tree. n_b: bit-width of each estimate (real + imaginary) of $\widehat{\mathbf{x}}$. f_B: signal bandwidth (Hz). *a:* Assuming baseband processing latency much larger than routing latency. For the number of links in the tree, N is assumed large. Units: Computational complexity [MAC/s], interconnection data-rate [bps], processing latency [s].

Algorithm - Topology	seq. daisy-chain	parallel daisy-chain	sequential tree	parallel tree
Comp. complexity	$2KM_\mathrm{p}f_\mathrm{B}$	$KM_\mathrm{p}f_\mathrm{B}$	$2KM_\mathrm{p}f_\mathrm{B}$	$KM_\mathrm{p}f_\mathrm{B}$
Inter. data-rate	$n_\mathrm{b}Kf_\mathrm{B}$	$n_\mathrm{b}Kf_\mathrm{B}$	$\approx 2n_\mathrm{b}Kf_\mathrm{B}$	$n_\mathrm{b}Kf_\mathrm{B}$
Processing latency	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)^a$	$\mathcal{O}(\log N)$
Number of links	N	N	$\approx N\left(\frac{d}{d-1}\right)$	$\approx N\left(\frac{d}{d-1}\right)$

2.4.2 Summary of the system design process

In this subsection we refer to the system design and hardware implementation cycle described in Section 2.1 and shown graphically in Fig. 2.4. In order to illustrate the importance of this methodology in the system design process, we describe in detail the procedure that leads to the distributed processing in hybrid-tree topology proposed in **Paper V**.

The starting point (in the case of communication) is the equalization method ZF, described in Subsection 2.3.2. Applying a centralized processing approach here leads to the implementation challenges and limitations presented in **Chapter 1**, which involves a high interconnection data-rate and relatively high computational complexity due to the required matrix inversion. As this solution is not suitable, we iterated in the design cycle by employing a processing distribution approach. At that stage, a serial processing algorithm was selected (CD), together with a convenient topology (daisy-chain) that could alleviate the limitations described before (solution proposed in **Paper III**). As the processing was distributed across the different nodes in the system, a more balanced and scalable system design was achieved. However, the relatively large latency imposed by the daisy-chain topology may result in a limitation for certain applications. This motivated another iteration in the design cycle, resulting in the proposal of the IIC algorithm and hybrid-tree topology in **Paper V**. This selection, not only provide the benefits of the processing distribution, but also experiences low latency due to the tree topology.

This iterative process can continue until a certain list of requirements are met. Those may come from the application to be supported, as described in Section 1.1, or from the operator, as seen in Section 2.1.

2.5 Architecture for distributed positioning

In the case of localization, the system wants to estimate the user's location based on the received signals at the panels. As said previously, the idea is to have a common infrastructure and processing architecture that can support both services: communication and localization, as shown in Fig. 2.2. In general, we are interested in studying how to extract location information from the CSI (which is already available for communication purposes), and how to use it in a way that can be efficiently implemented and deployed in reality.

Localization based on RF signals is a well studied topic with abundant material in the research literature. Its importance is derived from its potential to enable high accuracy localization in indoor and outdoor environments with and without line of sight, becoming a very inexpensive alternative to camera-based systems.

Some known techniques may require the calibration of panels and ensure a tight synchronization between them in order to be able to measure time-differences between RF signal time-of-arrival. Unfortunately this option seems expensive if we take into account the large number of antenna elements and panels involved in the LIS. One alternative that does not need calibration is based on **fingerprinting**, which consists of the (off-line) creation of a data-base of a certain RF-based measurements, such as the received signal strength indicator (RSSI), from different panels and locations [45–48]. During the estimation, real-time RF measurements are compared with the data-base entries, in order to find the closest ones and therefore the user's location. An alternative to RSSI is the CSI[10], which has two main advantages: 1) it is already available for communication demodulation purposes; and 2) contains richer features than RSSI, as it is able to capture not only the amplitude but also the phase relationship of the impinging signals, and herein localization information. Unfortunately, as the number of antenna elements and bandwidth is expected to be large in LIS and future 6G RAN systems, a data-base filled with CSI may not be feasible as it would require very large data-base, slowing down the access and comparison process, and therefore becoming a potential bottleneck [42]. In order to alleviate this limitation, machine learning (ML) techniques, specially neural networks (NNs) have attracted the attention for their generalization capabilities as universal interpolator. Many previous works cover the use of NNs within RF-based localization such as [33–36] among others, with successful results. It is relevant to mention recent works, such as [33], which covers localization in MU-MIMO scenarios with multiple Wi-Fi APs. In [33] each AP computes a probabilistic estimate of the user's location, that can be further merged by convenient techniques based on information fusion. **Paper VI** aims to be a step further in this direction, proposing a novel distributed processing pipeline for localization with probabilistic description of the user's location. In this case, the panels provide a Gaussian probability distribution of the location which can be fused by Gaussian conflation as described in [43]. This scheme is very convenient as it is able to support any topology. As the future RAN systems, as LIS, are expected to grow in number of antennas and become distributed, having a built-in mechanism for information fusion is critical.

Fig. 2.8 shows the system particularized for the localization case, where two panels provide estimates of the user's location based on local CSI, which can be conveniently fused into one estimate. This approach follows the recommendations presented at the beginning of the section:

- By providing the coordinates estimates per panel (or other form of location description), we exploit **per-user processing** or **dimensionality**

[10] Here we consider CSI as complex-values representing the wireless propagation channel in a certain set of time-frequency resources.

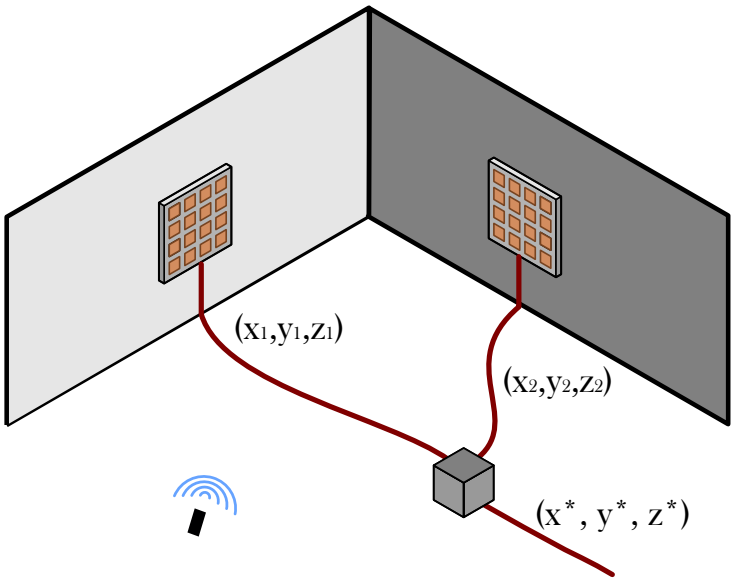


Figure 2.8: System model for localization purposes. Two panels provide an estimate of the user location based on local observations. By using information fusion techniques, these two estimates can be merged into a single one.

reduction, as we avoid the exchange of CSI with the corresponding savings in interconnection data-rate.

- **Data locality principle** is also employed here, as the location estimate is formulated exclusively using local observations (local CSI). There is no need for CSI exchange among panels or nodes.
- **Information fusion** techniques are vital, as we can merge different estimates along the way to the CPU or root node, with important savings in interconnection and routing resources.

In this section, we discussed about positioning using the received RF signals. Specifically how the CSI, already available for communications purposes, can be utilized for positioning. From the topology point of view, the tree has been shown to be very convenient, as it unifies communications and positioning. Processing distribution enables the capability of providing local estimates, that are fused by the use of information fusion techniques. This avoid sending all

data to the CPU, which reduces the inter-connection data-rate. This is highly important as it allows to overcomes the limitation introduced in Chapter 1.

In the next chapter, we will present the conclusions and future directions of this thesis.

Chapter 3

Conclusion and Future Directions

3.1 Summary and conclusions

In this thesis we have introduced some of the envisioned use cases and applications of future wireless communication systems, and translate those to requirements in the telecommunication infrastructure, especially from the base-station point of view. As the consumption of data is expected to keep growing, together with higher accuracy requirements in localization, these systems are evolving towards wider signal bandwidths and very large and distributed arrays, aiming to increase spatial multiplexing capabilities and therefore spectral efficiency. Additionally the capability to beamform energy to smaller volume in space improves radiated energy efficiency and localization accuracy. We studied Massive MIMO and LIS, as relevant technologies in this direction.

Despite of the discussed benefits, such systems present multiple challenges from implementation point of view. Those challenges come from high interconnection and computational aspects, leading to potential bottlenecks in scalability. We addressed these limitations and proposed different solutions, where processing distribution was identified as the key aspect to unlock scalability. We introduced novel algorithms, together with system-level topologies and processing architecture considerations to enable the implementation and deployment of these systems. Additionally, analysis of different hardware resources involved were also presented and compared with centralized solutions as a motivation to pursue these ideas.

Future applications demand not only communications but other services

such as localization and sensing. Supporting all within the same system architecture is the goal. In that context, we explored the localization problem, and proposed a solution that is feasible from implementation point of view and compatible with the system architecture and topology developed for communication purposes.

3.2 Future directions

In this section we list some of the most relevant and promising future directions:

- Several algorithms have been proposed in the included papers to the present thesis. In most cases, an analysis of the demands of such algorithms from hardware point of view has been presented, together with performance evaluation. These analysis are meant to be guidelines for the system designer, giving an estimate of the computational complexity and interconnection data-rate among other metrics. These numbers are very generic and require an specific implementation in order to determine exact values. Implementing such algorithms in a real-time hardware platform is a promising future direction to validate the results of those algorithms, and to give more accurate numbers.

- Most of the included papers are focusing on the uplink aspect of the communication. Further exploration of the downlink, especially precoding, is expected in order to fully validate these algorithms.

- Synchronization, which is not covered in this thesis, is a vital aspect of the system design. As mentioned in 1.3.4, ensure tight synchronization among all panels is extremely challenging from implementation point of view. A future direction consists of the analysis of performance degradation at different levels of synchronization, and therefore obtaining interesting trade-offs.

- A description of the system design and implementation iterative cycle was described in Chapter 2, together with an introduction to the relevance of the topology-algorithm co-design and how to approach it. Due to its complexity, the techniques described in this thesis to approach this are based on heuristics. A future direction may include the investigation of a more systematic method for this purpose.

- Positioning with probabilistic description using machine learning and fingerprinting techniques is very promising, especially because its capacity

to enable information fusion, which becomes vital in distributed processing systems. The method proposed in **Paper VI** pretends to be an initial exploration and analysis of the topic. While the obtained results are promising, further evaluation should be done in this area. Robustness against potential hardware impairments, such as frequency offsets, and uncertainties in the phase and amplitude of the estimated channel are critical to guarantee a correct behavior in real deployments.

- We have covered communication and localization in this thesis. Sensing is another relevant application that is expected to play an important role in future systems. Another research direction could be to explore how to support sensing using the same system architecture proposed in the thesis.

Bibliography

[1] Juan Francisco Esteban *et al.*, "Use case-driven specifications and technical re-quirements and initial channel model", 2021. Zenodo. https://doi.org/10.5281/zenodo.5561844

[2] D. G. Morín, P. Pérez and A. G. Armada, "Toward the Distributed Implementation of Immersive Augmented Reality Architectures on 5G Networks," in *IEEE Communications Magazine*, vol. 60, no. 2, pp. 46-52, February 2022.

[3] Ove Edfors *et al.*, "Initial assessment of architectures and hardware resources for a RadioWeaves infrastructure", 2022. Zenodo. https://doi.org/10.5281/zenodo.5938909

[4] Liang Liu *et al.*, "Distributed and centralized baseband processing algorithms, architectures, and platforms", 2016. MAMMOET: Massive MIMO for Efficient Transmission.

[5] T. L. Marzetta, "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas," in *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590-3600, November 2010.

[6] F. Rusek *et al.*, "Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays," in *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40-60, Jan. 2013.

[7] S. Hu, F. Rusek and O. Edfors, "Beyond Massive MIMO: The Potential of Data Transmission With Large Intelligent Surfaces," in *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2746-2758, 15 May15, 2018.

[8] S. Hu, F. Rusek and O. Edfors, "Beyond Massive MIMO: The Potential of Positioning With Large Intelligent Surfaces," in *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1761-1774, 1 April1, 2018.

[9] A. Fehske, G. Fettweis, J. Malmodin and G. Biczok, "The global footprint of mobile communications: The ecological and economic perspective," in *IEEE Communications Magazine*, vol. 49, no. 8, pp. 55-62, August 2011.

[10] Henrik Asplund *et al.* "Advanced Antenna Systems for 5G Network Deployments: Bridging the Gap Between Theory and Practice". Elsevier. 1st edition. 2020.

[11] R. Hou and B. Gransson, "Millimeter-wave Multi-antenna/MIMO Techniques for 5G NR Base-stations," in *2020 IEEE International Electron Devices Meeting (IEDM)*, 2020, pp. 34.1.1-34.1.4.

[12] Hayate Okuhara *et al.*, "An Energy-Efficient Low-Voltage Swing Transceiver for mW-Range IoT End-Nodes", in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*.

[13] Slavisa Aleksic, "Power efficiency of 40 Gbit/s and 100 Gbit/s optical ethernet", in *2009 11th International Conference on Transparent Optical Networks*.

[14] Aravindh Krishnamoorthy and Deepak Menon, "Matrix Inversion Using Cholesky Decomposition", *ArXiv e-prints*, 2013.

[15] Song Hang *et al.*, "EIE: Efficient inference engine on compressed deep neural network", in *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 243-254, 2016.

[16] Song Han, Huizi Mao and William J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding", in *ArXiv e-prints*, 2016.

[17] S. Malkowsky *et al.*, "The Worlds First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation," in *IEEE Access*, vol. 5, pp. 9073-9088, 2017.

[18] S. Kashyap, C. Mollén, E. Björnson and E. G. Larsson, "Frequency-domain interpolation of the zero-forcing matrix in massive MIMO-OFDM," *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*.

[19] Antonio Puglielli *et al.*, "Design of Energy- and Cost-Efficient Massive MIMO Arrays," in *Proceedings of the IEEE*, vol. 104, no. 3, pp. 586-606, March 2016.

[20] Antonio Puglielli, Borivoje Nikolic, and Elad Alon, "System Architecture and Signal Processing Techniques for Massive Multi-user Antenna Arrays", 2017.

[21] I. C. Sezgin et al., "A Low-Complexity Distributed-MIMO Testbed Based on High-Speed SigmaDelta-Over-Fiber," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 7, pp. 2861-2872, July 2019.

[22] "The White Rabbit Project", https://white-rabbit.web.cern.ch/.

[23] X. Ge, J. Yang, H. Gharavi and Y. Sun, "Energy Efficiency Challenges of 5G Small Cell Networks," in *IEEE Communications Magazine*, vol. 55, no. 5, pp. 184-191, May 2017.

[24] S. Tombaz, K. W. Sung and J. Zander, "On Metrics and Models for Energy-Efficient Design of Wireless Access Networks," in *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 649-652, Dec. 2014.

[25] S. Tombaz, P. Monti, K. Wang, A. Vastberg, M. Forzati and J. Zander, "Impact of Backhauling Power Consumption on the Deployment of Heterogeneous Mobile Networks," *2011 IEEE Global Telecommunications Conference - GLOBECOM*, 2011, pp. 1-5.

[26] C. De Lima *et al.*, "Convergent Communication, Sensing and Localization in 6G Systems: An Overview of Technologies, Opportunities and Challenges," in *IEEE Access*, vol. 9, pp. 26902-26925, 2021.

[27] Andre Bourdoux *et al.*, "6G White Paper on Localization and Sensing," in *ArXiv e-prints*, 2020.

[28] U. Gustavsson *et al.*, "Implementation Challenges and Opportunities in Beyond-5G and 6G Communication," in *IEEE Journal of Microwaves*, vol. 1, no. 1, pp. 86-100, Jan. 2021.

[29] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson and T. L. Marzetta, "Cell-Free Massive MIMO Versus Small Cells," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834-1850, March 2017.

[30] C. Fager, S. Rimborg, E. Rådahl, H. Bao and T. Eriksson, "Comparison of Co-located and Distributed MIMO for Indoor Wireless Communication," in *2022 IEEE Radio and Wireless Symposium (RWS)*, 2022, pp. 83-85.

[31] Z. Ebadi, C. Hannotier, H. Steendam, F. Horlin and F. Quitin, "An over-the-air CFO-assisted synchronization algorithm for TDOA-based localization systems," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, 2020, pp. 1-5.

[32] H. Bao, I. C. Sezgin, Z. S. He, T. Eriksson and C. Fager, "Automatic Distributed MIMO Testbed for Beyond 5G Communication Experiments," in *2021 IEEE MTT-S International Microwave Symposium (IMS)*, pp. 697-700.

[33] E. Gönültaş, E. Lei, J. Langerman, H. Huang and C. Studer, "CSI-Based Multi-Antenna and Multi-Point Indoor Positioning Using Probability Fusion," in *IEEE Transactions on Wireless Communications*, 2021.

[34] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, Deep convolutional neural networks for massive MIMO fingerprint-based positioning, in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1-6.

[35] S. D. Bast, A. P. Guevara, and S. Pollin, CSI-based positioning in massive mimo systems using convolutional neural networks, in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1-5.

[36] P. Ferrand, A. Decurninge, and M. Guillaud, DNN-based localization from channel estimates: Feature design and experimental results, in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1-6.

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

[38] M. Sarajlic, F. Rusek, J. R. Sanchez, L. Liu, and O. Edfors, "Fully decentralized approximate zero-forcing precoding for massive MIMO systems". In *IEEE Wireless Communications Letters*, 2019.

[39] C. Jeon, K. Li, J. R. Cavallaro, and C. Studer, "Decentralized equalization with feedforward architectures for massive MU-MIMO," in *IEEE Transactions on Signal Processing*, 2019.

[40] S.M. Kay, "Fundamentals of Statistical Signal Processing: Estimation Theory". Prentice-Hall PTR, 1993.

[41] Z. H. Shaik, E. Björnson, and E. G. Larsson, "MMSE-optimal sequential processing for cell-free massive MIMO with radio stripes," in *IEEE Transactions on Communications*, 2021.

[42] L. Tang, R. Ghods and C. Studer, "Reducing the Complexity of Fingerprinting-Based Positioning using Locality-Sensitive Hashing," in *53rd Asilomar Conference on Signals, Systems, and Computers*, 2019.

[43] T. P. Hill, "Conflations of Probability Distributions," in *Transactions of the American Mathematical Society*, vol. 363, no. 06, pp. 33513351, aug 2008. [Online]. Available: http://arxiv.org/abs/0808.1808

[44] K. Li, R. R. Sharan, Y. Chen, T. Goldstein, J. R. Cavallaro and C. Studer, "Decentralized Baseband Processing for Massive MU-MIMO Systems," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 4, pp. 491-507, Dec. 2017.

[45] S. Cortesi, M. Dreher and M. Magno, "Design and Implementation of an RSSI-Based Bluetooth Low Energy Indoor Localization System," in *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2021, pp. 163-168.

[46] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications.* Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), 2000, pp. 775-784 vol.2.

[47] Widyawan, M. Klepal and D. Pesch, "Influence of Predicted and Measured Fingerprint on the Accuracy of RSSI-based Indoor Location Systems," in *2007 4th Workshop on Positioning, Navigation and Communication*, 2007, pp. 145-151.

[48] K. N. R. S. V. Prasad, E. Hossain and V. K. Bhargava, "Machine Learning Methods for RSS-Based User Positioning in Distributed Massive MIMO," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8402-8417, Dec. 2018.

Part II

Included Papers

Paper I

Fully Decentralized Massive MIMO Detection Based on Recursive Methods

Algorithms for Massive MIMO uplink detection typically rely on a centralized approach, by which baseband data from all antennas modules are routed to a central node in order to be processed. In case of Massive MIMO, where hundreds or thousands of antennas are expected in the base-station, this architecture leads to a bottleneck, with critical limitations in terms of interconnection bandwidth requirements. This paper presents a fully decentralized architecture and algorithms for Massive MIMO uplink based on recursive methods, which do not require a central node for the detection process. Through a recursive approach and very low complexity operations, the proposed algorithms provide a sequence of estimates that converge asymptotically to the zero-forcing solution, without the need of specific hardware for matrix inversion. The proposed solution achieves significantly lower interconnection data-rate than other architectures, enabling future scalability.

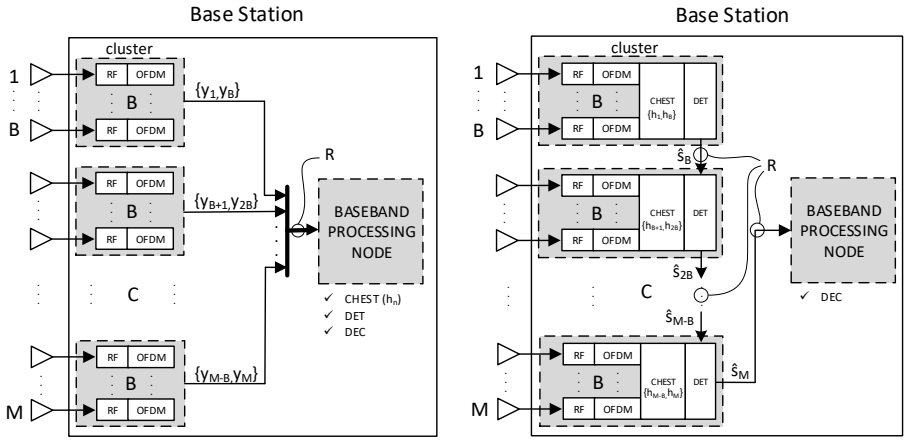
1 Introduction

Massive multi-user (MU) multiple-input multiple-output (MIMO) is one of the most promising technologies in the wireless area [1]. High spectral efficiency and improved link reliability are among the key features of this technology, making it a key enabler to exploit spatial diversity far beyond traditional MIMO systems by employing a large scale antenna array with hundreds or thousands of elements. This allows for unprecedented spatial resolution and high spectral efficiency, while providing simultaneous service to several users within the same time-frequency resource.

Despite all advantages of Massive MIMO, there are challenges from an implementation point of view. Uplink detection algorithms like zero-forcing (ZF) typically rely on a centralized architecture, shown in Figure 1a, where baseband samples and channel state information (CSI) are collected in the central node for further matrix inversion and detection. Dedicated links are needed between antenna modules and central node to carry this data. This approach, that is perfectly valid for a relatively low number of antennas, shows critical limitations when the array size increases, with interconnection bandwidth quickly becoming a bottleneck in the system.

Previous work has been done proposing different architectures for Massive MIMO base-stations [2–6]. All of them conclude by pointing to the interconnection bandwidth as the main implementation bottleneck and a limiting factor for array scalability. Most of them recommend moving to a decentralized approach where uplink detection and downlink precoding can be performed locally in processing nodes close to the antennas. However, to achieve that, CSI still needs to be collected in a central node, where matrix inversion is done and the result distributed back to all modules [2,3,5]. A further step has been made in [6], where CSI is obtained and used only locally (not shared) for precoding and detection. This architecture relies on a central node only for processing partial results. This dependency on a central node limits the scalability of this solution as will be shown in section 4.

In this paper we propose a fully decentralized architecture and recursive algorithms for Massive MIMO uplink detection. Antennas in the array are grouped into clusters. Apart from antennas, clusters contain RF, Analog-to-Digital Converters (ADC), OFDM receiver, channel estimation and detection blocks. The decentralized topology is based on the direct connection of clusters forming a daisy-chain structure as shown in Figure 1b. The proposed algorithms are pipelined so that they run in a distributed way at the cluster nodes, providing a sequence of estimates that converge asymptotically to the zero-forcing solution. We will make use of the following algorithms: Recursive Least Square (RLS), Stochastic Gradient Descent (SGD) and Averaged



(a) Centralized architecture (b) Decentralized architecture

Figure 1: Comparison between base station receiver chain in centralized and fully decentralized architectures for Massive MIMO uplink. Antenna array with M elements is divided into C clusters, each containing B antennas. (a): Centralized architecture. Clusters contain RF amplifiers and frequency down-conversion (RF) elements, analog-to-digital converters (ADC) and OFDM receivers. Each cluster has one link to transfer baseband samples to a central baseband processing node, where the rest of processing tasks are done. (b): Fully decentralized architecture for detection. Clusters performs RF, ADC, OFDM, channel estimation (CHEST) and detection (DET) locally. Decoding (DEC) is centralized. Clusters are connected to each other by uni-directional links. Only one cluster has a direct connection with central node. Proposed algorithms are executed in DET blocks in parallel mode. The points where the interconnection data-rate is estimated are marked by circles and the value is denoted by R.

Stochastic Gradient Descent (ASGD), which are detailed in section 3.

Decentralized architectures overcome bottlenecks by finding a more equal distribution of the system requirements among the processing nodes of the system. Apart from this, data localization is a key characteristic of decentralized architectures. This architecture allows data to be consumed as close as possible to where it is generated, minimizing the amount to transfer, and therefore saving bandwidth and energy. Following this idea, processing nodes need to be

located near the antenna. Further, they perform tasks such as channel estimation and detection locally. Local CSI is estimated and stored locally in each, without any need to share it with any other nodes in the system.

The remainder of the paper is organized as follows. The system model for MIMO uplink is presented in section 2. In section 3 we introduce the proposed algorithms. In 4 we analyze the performance of these algorithms, present the advantages of the daisy-chain topology, and analyze interconnection data-rates. Finally, section 5 presents the conclusions of this publication.

Notation: In this paper, lowercase, bold lowercase and upper bold face letters stand for scalar, column vector and matrix, respectively. The operations $(.)^T$, $(.)^*$ and $(.)^H$ denote transpose, conjugate and conjugate transpose respectively. The vector $\mathbf{s}$ in the nth iteration is $\mathbf{s}_n$. Computational complexity is measured in terms of the number of complex-valued multiplications.

2 System model and detection algorithms

In this section we present the system model for MIMO uplink and introduce the ZF equalizer.

We consider a scenario with K single-antenna users transmitting to an antenna array with M elements. The input-output relation for uplink is

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{v}, \tag{1}$$

where $\mathbf{y}$ is the $M \times 1$ received vector, $\mathbf{s}$ is the transmitted user data vector $(K \times 1)$, $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2\ \cdots\ \mathbf{h}_M]^T$ is the channel matrix $(M \times K)$ and $\mathbf{v}$ samples of noise $(M \times 1)$. Under the Massive MIMO assumption, $M \gg K$.

Assuming time-frequency-based channel access, a Resource Element (RE) represents a slot in the time-frequency grid. Within each RE, the channel model follows (1).

A least-squares (LS) estimate of $\mathbf{s}$ is obtained as

$$\hat{\mathbf{s}}_{\mathrm{ZF}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{y}. \tag{2}$$

This method, commonly referred to as ZF, requires a central architecture as in Figure 1a because the complete matrix $\mathbf{H}$ needs to be collected in the central node before the Gramian matrix $(\mathbf{H}^H\mathbf{H})$ and its inverse can be computed. Decentralized architectures, such as the one shown in Figure 1b, require other type of algorithms.

3 Proposed algorithms

In this section we propose three algorithms for MIMO decentralized uplink detection.

Depending on the situation some algorithms are more appropriate than others. If full knowledge of matrix $\mathbf{H}$ and $\mathbf{y}$ is available at a single node, direct methods such as ZF can be applied (2). However, there are situations when the cost of collecting all knowledge at a single node is too high. For those cases, a different approach has to be used.

The goal of the proposed algorithms for uplink detection is the estimation of the transmitted user data vector, $\mathbf{s}$ in (1), based on knowledge of $\mathbf{H}$ and $\mathbf{y}$ that is distributed among nodes. These algorithms provide a sequence of estimates, which converge to $\hat{\mathbf{s}}_{ZF}$ as more knowledge of $\mathbf{H}$ and $\mathbf{y}$ is obtained. Estimation is done in a sequential manner, by which the estimate is passed from one antenna to the next one, being updated every time based on the previous estimate ($\hat{\mathbf{s}}_{n-1}$), local CSI ($\mathbf{h}_n$) and antenna observation (y_n). This can be summarized as $\hat{\mathbf{s}}_n = f(\hat{\mathbf{s}}_{n-1}, \mathbf{h}_n, y_n)$, which can be seen as a recursive form. This approach is in accordance with the data localization principle, which is a key characteristic of decentralized systems. In the Massive MIMO case, data is consumed close to where it is generated, namely at the antennas. This makes it possible that neither $\mathbf{h}_n$ nor y_n are shared, since only the estimate is.

These algorithms are flexible enough to work in clusters of antennas (see Figure 1b), whose size can vary from 1 up to M, the last case being equivalent to a centralized system.

The first recursive algorithm to be presented is the Recursive Least Square (RLS) method, which is a recursive form of (2). Uplink detection can be also seen as a regression parameter estimation - a problem well studied in the area of stochastic approximation methods. Stochastic Gradient Descent (SGD) and its averaged version (ASGD) fall within this group, and are based on a Gradient Descent algorithm in which the gradient is partially known.

In Section 3.1 we present RLS applied to MIMO uplink detection, which provides approximate ZF performance at the expense of a preprocessing stage. Afterwards, we present the SGD algorithm and its enhanced version, the Averaged SGD (ASGD), which increases robustness of SGD while achieving performance close to ZF for very large arrays.

Before we describe the algorithms we clarify the role played by the variable B, i.e., the number of antennas per cluster. Our algorithms are in fact independent of the value of B, therefore we present them with notation tailored to the choice $B = 1$. However, $B > 1$ is still of importance from an implementation point of view since each cluster may be implemented with a single processing unit. Thus, with $M = 100$ antennas, the choice $B = 1$ requires 100

processing units, while $B = 10$ merely requires 10 such units. Nevertheless, performance of our algorithms remains the same. B therefore takes a trade-off role: The larger the B, the less number of processing units, but meanwhile, the architecture becomes more centralized.

3.1 Recursive Least-Squares (RLS)

RLS is the recursive version of the LS algorithm. It can be shown [7] that the ZF/LS estimate, i.e., the l.h.s. of (2), can be approximated by the RLS as $\hat{\mathbf{s}}_{\mathrm{ZF}} \approx \hat{\mathbf{s}}_M$ where $\hat{\mathbf{s}}_n$ is recursively found as follows

$$\begin{aligned} \varepsilon_n &= y_n - \mathbf{h}_n^T \hat{\mathbf{s}}_{n-1} \\ \mathbf{\Gamma}_n &= \mathbf{\Gamma}_{n-1} - \frac{\mathbf{\Gamma}_{n-1}\mathbf{h}_n^* \mathbf{h}_n^T \mathbf{\Gamma}_{n-1}}{1 + \mathbf{h}_n^T \mathbf{\Gamma}_{n-1} \mathbf{h}_n^*}, \\ \hat{\mathbf{s}}_n &= \hat{\mathbf{s}}_{n-1} + \mathbf{\Gamma}_n \mathbf{h}_n^* \varepsilon_n. \end{aligned} \tag{3}$$

The quality of the approximation depends on the initial value of $\hat{\mathbf{s}}_0$. Nevertheless, for a randomly chosen $\hat{\mathbf{s}}_0$, the impact of $\hat{\mathbf{s}}_0$ quickly fades out over the index n and it can be shown that $s_M \to \hat{\mathbf{s}}_{\mathrm{ZF}}$ as $M \to \infty$ with probability one. In (3), $\hat{\mathbf{s}}_n$ is a $K \times 1$ vector and is the output of cluster n, y_n is the observation at the nth antenna, ε_n is the prediction error and $\mathbf{\Gamma}_n$ is a $K \times K$ matrix. As a side comment, we remark that $\hat{\mathbf{s}}_n$ is an approximate LS solution up to the nth antenna element.

In view of Figure 1b, increasing the iteration number in (3) from n to $n+1$ corresponds to passing on information from cluster n to cluster $n+1$. Each cluster receives an estimate of the transmitted data vector from previous cluster, $\hat{\mathbf{s}}_{n-1}$, and compute a new estimate $\hat{\mathbf{s}}_n$ based on local CSI, $\mathbf{h}_n$, and a local observation, y_n.

Under the block fading channel model, multiple Resource Elements (RE) in a certain region of the time-frequency grid experience identical channels. We name this region Coherence Block (CB), and following this model it is possible to re-use same CSI for all REs in the same CB.

Straightforward implementation of (3) at every RE is not efficient. In fact, a hefty share of the operations associated to (3) can be reused within the CB, namely those associated to computation of $\mathbf{\Gamma}_n$. Defining $\mathbf{\Gamma}_0 = \mathbf{I}_K$ and

$$\begin{aligned} \mathbf{z}_n &= \mathbf{\Gamma}_{n-1}\mathbf{h}_n^* \\ \alpha_n &= \frac{1}{1 + \mathbf{h}_n^T \mathbf{z}_n} \\ \mathbf{\Gamma}_n &= \mathbf{\Gamma}_{n-1} - \alpha_n \mathbf{z}_n \mathbf{z}_n^H, \quad n = 1, 2, \ldots, M \end{aligned}$$

we see that at each RE it suffices to compute

$$\begin{aligned}\varepsilon_n &= y_n - \mathbf{h}_n^T \hat{\mathbf{s}}_{n-1} \\ \hat{\mathbf{s}}_n &= \hat{\mathbf{s}}_{n-1} + \alpha_n \mathbf{z}_n \varepsilon_n, \quad n = 1, 2, \ldots, M\end{aligned}$$

in order to execute (3). It is easily verifiable that the complexity of preprocessing is $\mathcal{O}(K^2)$, whilst the complexity is $\mathcal{O}(K)$ at every RE.

3.2 Stochastic Gradient Descent (SGD)

The setup in SGD [8] is that one intends to solve the unconstrained LS problem

$$\min_{\mathbf{s}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \tag{4}$$

via a gradient descent (GD) approach. The gradient of (4) equals $\nabla_{\mathbf{s}} = \mathbf{H}^H\mathbf{H}\mathbf{s} - \mathbf{H}^H\mathbf{y}$. An immediate consequence is that GD is only feasible in a centralized approach.

SGD is an approximate version that can be operated in a decentralized architecture. It does so by computing, at each cluster, as much as possible of $\nabla_{\mathbf{s}}$ with the information available at the cluster. Then the cluster updates the estimate $\hat{\mathbf{s}}$ using a scaled version of the "local" gradient and passes the updated estimate on to the next cluster.

The above described procedure can, formally, be stated as

$$\begin{aligned}\varepsilon_n &= y_n - \mathbf{h}_n^T \hat{\mathbf{s}}_{n-1} \\ \hat{\mathbf{s}}_n &= \hat{\mathbf{s}}_{n-1} + \mu_n \mathbf{h}_n^* \varepsilon_n,\end{aligned} \tag{5}$$

where $\{\mu_n\}$ is a sequence of scalar step-sizes.

3.3 Averaged Stochastic Gradient Descent (ASGD)

Selection of optimum values μ_n in SGD is not trivial. Even though we take $\mu_n = \mu$ for simplification, the optimum value will depend on M, K and channel properties, where the latter may be unknown in many cases. An inappropriate selection of μ can lead to severe performance degradation depending on the scenario. Averaging a SGD sequence provides an asymptotically optimal convergence rate provided that the noise $\mathbf{v}$ is Gaussian [9], which increases robustness to the step-size selection. In the ASGD algorithm there are three

sequences defined as follows

$$\begin{aligned}\varepsilon_n &= y_n - \mathbf{h}_n^T \hat{\mathbf{x}}_{n-1} \\ \hat{\mathbf{x}}_n &= \hat{\mathbf{x}}_{n-1} + \mu_n \mathbf{h}_n^* \varepsilon_n \\ \hat{\mathbf{s}}_n &= \begin{cases} \hat{\mathbf{x}}_n & \text{if } n < n_0 \\ \frac{1}{n-n_0+1} \sum_{k=n_0}^{n} \hat{\mathbf{x}}_k & \text{if } n \geq n_0, \end{cases}\end{aligned} \tag{6}$$

where $\hat{\mathbf{x}}_n$ takes the role of the SGD output $\hat{\mathbf{s}}_n$ in (5). The ASGD output $\hat{\mathbf{s}}_n$ thereby becomes an averaged SGD sequence, where n_0 determines the onset of the averaging procedure.

The averaged sequence can be written more conveniently as

$$\hat{\mathbf{s}}_n = \begin{cases} \hat{\mathbf{x}}_n & \text{if } n < n_0 \\ \hat{\mathbf{s}}_{n-1} + \frac{1}{n'} \left(\hat{\mathbf{x}}_n - \hat{\mathbf{s}}_{n-1} \right) & \text{if } n \geq n_0, \end{cases} \tag{7}$$

where $n' = n - n_0 + 1$. As will be seen in our numerical results, the ASGD grossly relaxes the need for careful selection of μ.

4 Analysis

In this section we analyze the proposed solution. First, the performance of the presented algorithms will be shown and compared with each other. Second, a few strong points of the daisy-chain topology are given. Finally, an analysis of interconnection bandwidth is presented, followed by a comparison for four different configurations.

4.1 Detection performance

In this section, we present performance results for all algorithms. Reported metrics are Mean-Square-Error (MSE) and Bit-Error-Rate (BER) in block faded Rayleigh channels.

We report MSE, measured between $\hat{\mathbf{s}}$ and $\mathbf{s}$, as a function of the number of iterations (antenna index). The reported signal-to-noise ratio (SNR) is the average receive power at any base station antenna, divided by the noise variance.

MSE results for SGD are shown in Figure 2 for three different step-size values. As can be observed, step-size plays a critical role in the convergence speed of the algorithm. High step-size values provide faster convergence but high steady-state MSE, and low values may not even enter into the steady-state within the array. Given a certain M and K, it is possible to find an optimum step-size which provides the lowest MSE.

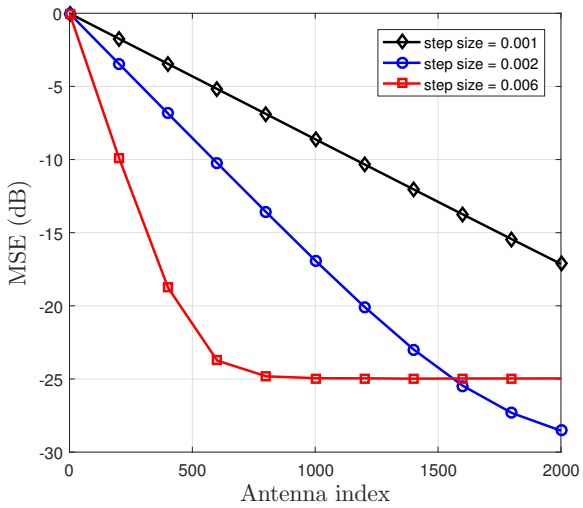


Figure 2: MSE vs antenna index for three different step-size values in SGD.

We now turn our attention towards Figures 3 and 4 which compare RLS, SGD, and ASGD. When the SGD sequence is averaged, MSE and BER curves get closer for different step-sizes, making the algorithm robust against non-optimal step-size selection. The selection of n_0 also has an impact, but less compared to non-optimal step-size in SGD.

As shown in Figure 4, RLS meets ZF (2) performance, as it is optimal for a Gaussian noise source [9]. For large M, performance of RLS and ASGD converge due to ASGD's asymptotically optimal rate property.

4.2 Strengths of daisy-chain topology

Ostensibly, it may come across as if our daisy-chain solution incurs a latency penalty. This is, however, not the case as the detection process over time and/or frequency can be pipelined. While cluster 2 is processing data at subcarrier, say, f_0, cluster 1 can process data at subcarrier $f_0 + 1$. In the next iteration, cluster 2 processes data at subcarrier $f_0 + 1$, etc. See Figure 5 for a graphical visualization of the pipelining procedure.

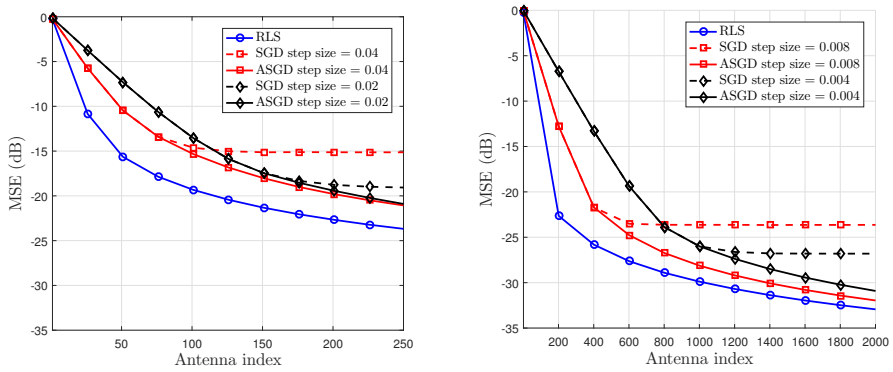


Figure 3: MSE vs antenna index for RLS, SGD and ASGD for different step-size values. Left: M=256. n_0=150 and 75 for μ=0.02 and 0.04 respectively. Right: M=2048. n_0=1000 and 400 for μ=0.004 and 0.008 respectively. K=16 and SNR=12dB in all cases.

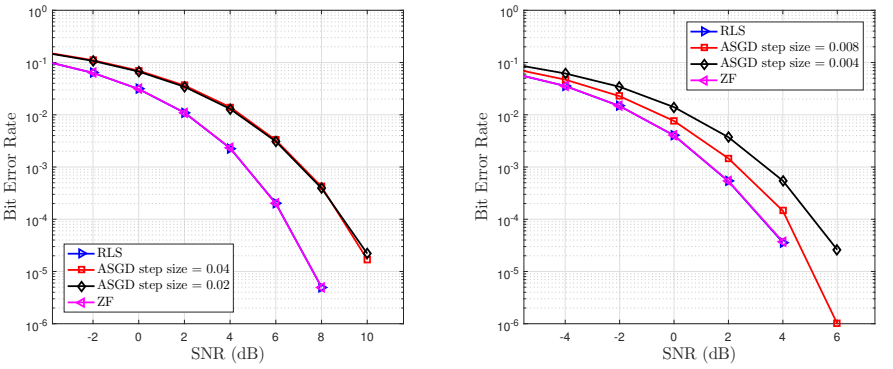


Figure 4: BER vs SNR for RLS, ASGD and ZF. Left: M=256, 16QAM. Right: M=2048, 64QAM. K=16 and SNR=12dB for both cases.

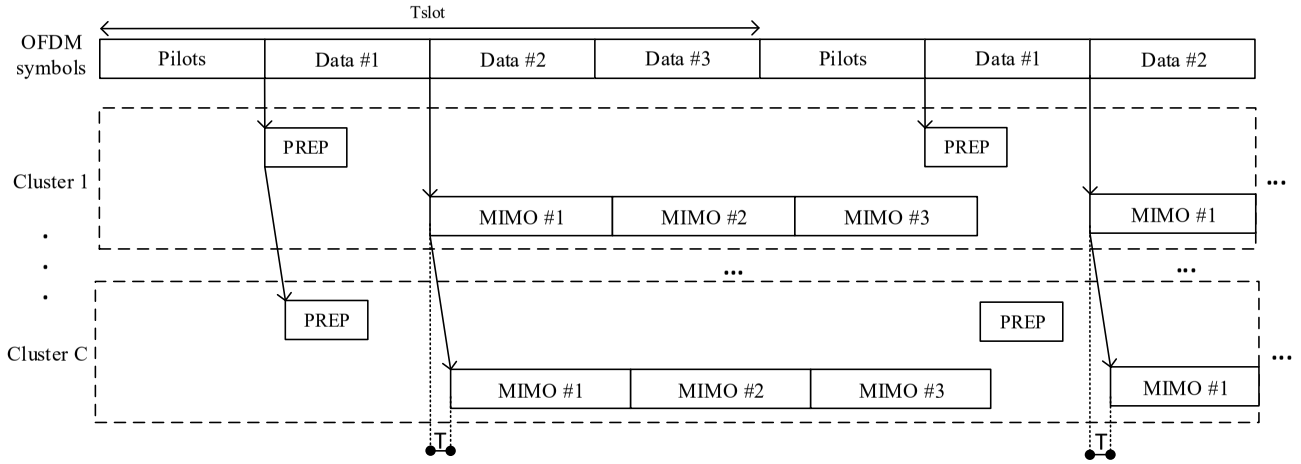


Figure 5: Time diagram representing cluster activities during one uplink slot with 4 OFDM symbol: one pilot and three data symbols. Only cluster 1 and C (last one) are represented for simplicity. Two types of activities are shown per cluster. The first one represents Pre-processing stage (PREP), only if RLS is used. The second one is the MIMO activity, which involves detection. First cluster start processing first RE after complete reception of Data 1. Once such RE is processed, it is then passed to next cluster for further processing and this is repeated successively through all clusters. As it is shown in the figure, there is a delay (T) for the starting time in cluster C compared to first cluster. T needs to be small enough to meet latency constraints.

Further, our daisy-chain solution allows for a power save since if a cluster n regards its incoming estimate to be sufficiently good, then it can do one out of at least two things, 1) set $\hat{\mathbf{s}}_{n+1} = \hat{\mathbf{s}}_n$, or 2) send the incoming estimate $\hat{\mathbf{s}}_n$ to the baseband processing node, thereby terminating the detection procedure. The former has the advantage over the latter that only the last cluster needs to be connected to the baseband processing unit. Further, an indication whether or not the incoming estimate is of sufficiently good quality can be obtained, e.g. for RLS, by the value ε_n in (3).

Finally, our topology is flexible so that additional antenna clusters can be added in a plug-and-play fashion. For example, in order to double the number of antennas, it is merely required to disconnect the cable between the last cluster and the baseband processing unit, connect that very cable to the last cluster of the added antenna array, and connect the two arrays. This will solely impact software scheduling at the baseband processing unit, but not the hardware as would have been the case for the centralized topology in Figure 1a.

4.3 Interconnection data-rate

In order to estimate the expected data-rate in the proposed architecture, we can assume an OFDM-based frame structure based on slots. Each slot is made by N_{slot} consecutive OFDM symbols with duration T_{ofdm}. Each symbol contains N_{u} subcarriers (an RE in OFDM) to carry user data. We can determine the average input/output data rate in the uplink for each of the clusters during a certain slot for SGD as follows

$$\bar{R}_{\text{SGD}} = \frac{K \cdot w_{\text{s}} \cdot N_{\text{u}} \cdot N_{\text{UL}}}{T_{\text{slot}}} = \alpha \cdot \frac{K \cdot w_{\text{s}} \cdot N_{\text{u}}}{T_{\text{ofdm}}}, \tag{8}$$

where T_{slot} is the slot duration, N_{UL} is the number of OFDM symbols allocated for UL data in a slot, w_{s} is the number of bits used to represent each element in the sequence of estimates $(\hat{\mathbf{s}}_n)$ and $\alpha = \frac{N_{\text{UL}}}{T_{\text{slot}}}$ represents the fraction of time spent in UL within the slot, so $0 \leq \alpha \leq 1$. In Figure 1b, $\bar{R}_{\text{SGD}}$ corresponds to R.

This analysis does not take into account the total amount of data that is generated (which depends on M) and needs to move through the structure, but only the data that moves between clusters (which depends on K) because it is the one that imposes physical constraints in the inter-cluster connections and may limit the scalability.

For ASGD, the averaged data rate is expected to be twice the one in SGD, because for each sequence element, two previous elements, $\hat{\mathbf{x}}_n$ and $\hat{\mathbf{s}}_{n-1}$, are needed as can be observed in (7), and therefore

$$\bar{R}_{\text{ASGD}} = 2 \cdot \bar{R}_{\text{SGD}}. \tag{9}$$

For RLS, the data-rate has two components, one due to the preprocessing stage and the other one due to each RE. During the first stage, matrix $\mathbf{\Gamma}$ is passed from cluster to cluster. During the RE processing stage, data rate is the same as in SGD. The averaged data rate for RLS is calculated as

$$\begin{aligned}\bar{R}_{\mathrm{RLS}} &= \frac{N_{\mathrm{CB}} \cdot K^2 \cdot w_\gamma}{T_{\mathrm{slot}}} + \frac{K \cdot w_{\mathrm{s}} \cdot N_{\mathrm{u}} \cdot N_{\mathrm{UL}}}{T_{\mathrm{slot}}} \\ &= \frac{N_{\mathrm{u}} \cdot N_{\mathrm{slot}}}{S_{\mathrm{CB}}} \cdot \frac{K^2 \cdot w_\gamma}{T_{\mathrm{ofdm}} \cdot N_{\mathrm{slot}}} + \alpha \cdot \frac{K \cdot w_{\mathrm{s}} \cdot N_{\mathrm{u}}}{T_{\mathrm{ofdm}}} \\ &= \alpha \cdot \frac{K \cdot w_{\mathrm{s}} \cdot N_{\mathrm{u}}}{T_{\mathrm{ofdm}}} \cdot \left(1 + \frac{\beta}{\alpha} \cdot \frac{K}{S_{\mathrm{CB}}}\right),\end{aligned} \tag{10}$$

where N_{CB} is the number of CBs per slot, S_{CB} the number of REs in each CB, w_γ the number of bits to represent each element in $\mathbf{\Gamma}$ and $\beta = \frac{w_\gamma}{w_{\mathrm{s}}}$. From (10) it can be seen that $\bar{R}_{\mathrm{RLS}} > \bar{R}_{\mathrm{SGD}}$.

We can compare our proposed solution with another cluster-based decentralized architecture, but which relies on a central node to collect partial results, performing a low complexity operation, such as averaging, and broadcasting back the result to the clusters according to an iterative algorithm. This star topology has been proposed in [6]. In this case, the central node will have C bi-directional links with an average aggregated data rate per direction of

$$\bar{R}_{\mathrm{star}} = C \cdot n_{\mathrm{iter}} \cdot \bar{R}_{\mathrm{SGD}}, \tag{11}$$

where n_{iter} is the number of iterations for the selected detection algorithm. From (11) we can observe that typically $\bar{R}_{\mathrm{star}} \gg \bar{R}_{\mathrm{SGD}}$.

In case of a fully centralized architecture as the one in [5], the interconnection data-rate depends linearly on M as follows

$$\bar{R}_{\mathrm{central}} = \frac{M \cdot N_{\mathrm{u}} \cdot N_{\mathrm{UL}} \cdot w_{\mathrm{sc}}}{T_{\mathrm{slot}}} = \alpha \cdot \frac{M \cdot N_{\mathrm{u}} \cdot w_{\mathrm{sc}}}{T_{\mathrm{ofdm}}}, \tag{12}$$

where w_{sc} is the number of bits representing a sample of the received signal $\mathbf{y}$. It is seen that (12) cannot scale easily. $\bar{R}_{\mathrm{central}}$ corresponds to R in Figure 1a. Going from (12) to (8), roughly reduces the data-rate by a factor $\frac{M}{K}$ (typically ≥ 10 in Massive MIMO).

Table 1 shows date-rates for four scenarios. We assume the following parameters: $T_{\mathrm{slot}} = 500\mu s$, $w_{\mathrm{s}} = 16$, $w_{\mathrm{sc}} = 24$, $N_{\mathrm{u}} = 1200$, $N_{\mathrm{slot}} = 7$, $N_{\mathrm{UL}} = 6$, $\alpha = 6/7$, $\beta = 3/2$, $S_{\mathrm{CB}} = 400$ and $n_{\mathrm{iter}} = 3$. We can observe that the analyzed topology and algorithms achieve significantly lower interconnection data-rate than other architectures [5] [6], enabling future scalability. As observed, for very-large arrays, RLS and ASGD require similar data-rates and have similar

Table 1: Data Rate comparison for different topologies / algorithms

M	128	256	512	1024
K	16	32	64	128
C	8	8	16	16
B	16	32	32	64
$\bar{R}_{\text{SGD}}$	439MB/s	879MB/s	1.7GB/s	3.4GB/s
$\bar{R}_{\text{RLS}}$	470MB/s	1.0GB/s	2.2GB/s	5.3GB/s
$\bar{R}_{\text{ASGD}}$	879MB/s	1.7GB/s	3.4GB/s	6.8GB/s
$\bar{R}_{\text{star}}$ [6]	10.3GB/s	10.3GB/s	20.6GB/s	20.6GB/s
$\bar{R}_{\text{central}}$ [5]	5.1GB/s	10.2GB/s	20.4GB/s	40.8GB/s

performance, but RLS requires a pre-processing stage and matrix manipulation that ASGD does not.

5 Conclusions

In this article we have introduced a base station uplink architecture for Massive MIMO and analyzed the main implementation bottleneck, the interconnection data-rate. We have proposed three algorithms and a fully decentralized topology for uplink detection, which alleviate this limitation. One of the algorithms (RLS) achieves approximate zero-forcing performance, while another (ASGD) is an approximation which converges to the former one for very large arrays. All of them are of low-complexity and do not require matrix inversion. An estimate of data-rate is also presented and compared with other architectures for different array-sizes and configurations, showing the benefits of the proposed solution.

Acknowledgment

This work was supported by ELLIIT, the Excellence Center at Linkping-Lund in Information Technlology.

Bibliography

[1] T. L. Marzetta, Noncooperative cellular wireless with unlimited numbers of base station antennas, *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590-3600, November 2010.

[2] C. Shepard *et al.*, Argos: Practical many-antenna base stations, in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (Mobicom)*, New York, NY, USA, 2012, pp. 53-64. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348553

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

[4] A. Puglielli *et al.*, Design of energy- and cost-efficient massive MIMO arrays, in *Proceedings of the IEEE*, vol. 104, no. 3, pp. 586-606, March 2016.

[5] S. Malkowsky *et al.*, The worlds first real-time testbed for massive MIMO: Design, implementation, and validation, *IEEE Access*, vol. 5, pp. 9073-9088, 2017.

[6] 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, Dec 2017.

[7] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. The MIT Press, 1983.

[8] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. Springer, 2003.

[9] B. Polyak and A. B. Juditsky, Acceleration of stochastic approximation by averaging, *SIAM Journal on Control and Optimization*, vol. 30, pp. 838-855, July 1992.

[10] B. Polyak and Y. Z. Tsypkin, Adaptive estimation algorithms (convergence, optimality, stability), *Automation and Remote Control*, vol. 40, pp. 378-390, March 1979.

Paper II

Decentralized Massive MIMO Systems: Is There Anything to be Discussed?

Algorithms for Massive MIMO uplink detection are typically based on a centralized approach, by which baseband data from all antenna modules need to be routed to a central node for further processing. In the case of Massive MIMO, where hundreds or thousands of antennas are expected in the base-station, such architecture requires high interconnection bandwidth between antennas and the central node. Recently, decentralized architectures have been proposed to maintain low interconnection bandwidth, where channel-state-information (CSI) is obtained locally in each antenna node and not shared. Further, Massive MIMO performance is sensitive to CSI quality. However, in the literature, ideal CSI is typically assumed in decentralized systems, which is not only far from reality but also limits the generality of the analysis.

This paper proposes a decentralized (a term that will be defined in the main body of the paper) architecture with the following main features: (i) the channel matrix is not made available at any single node, (ii) there is no inter-communication among antennas, (iii) the architecture used during the payload data phase, is reused to provide a certain statistic to a processing node, (iv) A non-standard channel estimation problem based on said statistic arises, (v) a matrix inversion is needed (in case of zero-forcing) at said processing node.

A hefty share of the paper is devoted to (iv).

Jesús Rodríguez Sánchez, Juan Vidal Alegría and Fredrik Rusek,
"Decentralized Massive MIMO Systems: Is There Anything to be Discussed?,"
in *Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 787-791, 2019.

1 Introduction

Massive MIMO [1] is one of the most relevant technologies in contemporary wireless communications. High spectral efficiency and improved link reliability, achieved by using hundreds or thousands of antenna elements, are among the technology's key features, making it a key enabler to exploit spatial diversity far beyond traditional MIMO systems. Unprecedented spatial resolution is achieved, while simultaneously providing service to multiple users within the same time-frequency resource.

Despite all advantages of Massive MIMO, there are challenges from an implementation point of view. Uplink detection algorithms, like zero-forcing (ZF), typically rely on a centralized architecture, where baseband samples and channel state information (CSI) are collected in a central processing node for further matrix inversion and detection. Physical connections are needed between the antenna modules and the central node to carry the required data. This approach, that is perfectly valid for a relatively low number of antennas, becomes problematic when the number of antennas increases, with the interconnection bandwidth as the main bottleneck in the system.

Initial Massive MIMO prototypes [2] [3] were the first to face this problem and pragmatic solutions were proposed. Later, efforts have been made to study the problem more rigorously, see for example [4] and [5]. In [5], a partial decentralized (PD) solution is put forth, which achieves exactly the same estimates (and therefore performance) as linear detectors, such as maximum ratio combing (MRC), ZF and L-MMSE. There are more recent activities in this area in [6] and [7], where antennas are connected by direct links forming a daisy-chain. It has been shown that such a structure is able to achieve approximate ZF performance under IID channel conditions and perfect CSI with very low inter-connection requirements and fully decentralized detection/precoding processing. In order to achieve this, obtaining CSI is required prior to detection. However, motivated by scalability reasons, decentralized architectures do not allow all baseband samples to be collected at a single point, which limits one's channel estimation capability.

It has been understood that the full benefits of massive MIMO, such as high spectrum efficiency, heavily rely on accurate CSI. Poor channel estimates cannot reach sufficiently good inter-user interference (IUI) cancellation, a problem that amplifies as the number of users grows. Unfortunately, channel estimation is not typically covered in the decentralized debate and ideal CSI is always assumed to be available.

In this work we argue that much of the centralized vs. non-centralized discussion is unnecessary and that non-perfect CSI has to be assumed for the problem not to be trivial. In short, our arguments are as follows: The non-

centralized discussion seems to be revolving around the training phase, more specifically around avoiding the collection of full CSI at any given node. However, there is also a payload data phase, and during this phase it can be assumed that each antenna contributes, at the very minimum, with a scalar value to a central processing unit (responsible for, e.g., error control decoding, HARQ, etc.). In order to maintain low interconnection bandwidth, these antenna contributions are likely to be summed up before being formally presented to the central processing unit. That said, we observe that there is an easy way to transfer a statistic that is sufficient for demodulation purposes, using the same circuitry used during the payload data phase. Such a method can, according to a separate discussion in the next section, be classified as non-centralized since it 1) does not store the full CSI at any given node, and 2) it does not expand the interconnections between antennas and processing unit beyond what is needed for payload data. At the central processing node, channel estimation and channel inversion remains, but we argue that these are fairly minor tasks compared to other baseband tasks needed at such a node. One of our main messages we try to convey is that, in our view, meeting 2) is sufficient for a scheme to be classified as non-centralized (roughly speaking, 1) is a natural consequence of 2)). Further, if perfect CSI is assumed, then the method to be presented is optimal, so not much discussion is needed.

The remainder of the paper is organized as follows. A system model and a review of linear detection methods in Massive MIMO is presented in section 2. In 3 we discuss centralized and decentralized systems and motivate the scope of our article. Our method and a collection of practical channel estimators are presented in 4. Results are presented in 5. Finally, section 6 summarizes our conclusions.

Notation: In this paper, lowercase, bold lowercase and upper bold face letters stand for scalars, column vectors and matrices, respectively. The operations $(.)^T$, $(.)^*$ and $(.)^H$ denote transpose, conjugate and conjugate transpose, respectively. ${}_0\tilde{F}_1$ is the hypergeometrical function of a matrix argument, defined as in [8], $\tilde{\Gamma}_m$ denotes the complex multivariate gamma function. $|\mathbf{A}|$, $\mathrm{tr}(\mathbf{A})$ and $\mathrm{eig}(\mathbf{A})$ represent the determinant, trace, and eigenvalues of a matrix $\mathbf{A}$, respectively. $\mathcal{CN}(\mathbf{0}, \mathbf{A})$ denotes circularly-symmetric multivariate complex-valued Gaussian probability density distribution with covariance $\mathbf{A}$.

2 System model

For uplink detection, we consider a scenario with K single-antenna users transmitting to a base-station (BS) with an antenna array with M elements through

a narrow-band channel with IID Rayleigh fading

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{y}$ is the $M \times 1$ received vector, $\mathbf{x}$ is the transmitted user data vector ($K \times 1$), with uniform unitary power distribution across users such that $\mathbb{E}\{\mathbf{x}\mathbf{x}^H\} = \mathbf{I}$, $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2 \cdots \mathbf{h}_M]^T$ is the channel matrix ($M \times K$) whose rows follows $\mathcal{CN}(\mathbf{0}, \mathbf{C})$, and $\mathbf{n}$ thermal noise ($M \times 1$) at the BS with distribution $\mathcal{CN}(\mathbf{0}, N_0\mathbf{I})$.

2.1 Linear processing in massive MIMO

We consider only linear detectors, because they exhibit close to optimal performance in Massive MIMO while having low complexity. A linear equalizer provides an estimate of $\mathbf{x}$, $\hat{\mathbf{x}}$, by applying an equalizer filter matrix $\mathbf{W}$ to the vector observation, $\mathbf{y}$, as follows

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{y}, \tag{2}$$

where $\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2 \cdots \mathbf{w}_K]^T$ is a $K \times M$ complex matrix.

We limit our exposition to maximum-ratio combining (MRC) and zero-forcing (ZF), with the remark that our analysis would remain essentially unaltered for other choices, whose equalizer matrices are defined as

$$\mathbf{W} = \begin{cases} \mathbf{H}^H & \text{for MRC} \\ \mathbf{G}^{-1}\mathbf{H}^H & \text{for ZF.} \end{cases} \tag{3}$$

where $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ is the Gramian matrix. It is important to note that matrix $\mathbf{W}$ is valid during a Coherence Block (CB) of the channel, representing a time-frequency region where the channel can be considered approximately constant.

MRC represents ideal decentralized processing, allowing each antenna node to obtain an equalization vector from local CSI. ZF, on the other hand, ostensibly requires the system to collect all CSI from all antennas in a central processing node, for further matrix inversion. It is well known that ZF provides superior performance over MRC due to perfect inter-user interference cancellation capabilities at the cost of increased inter-connection bandwidth and processing requirements. However, as we will show, the interconnection bandwidth needs in fact not to increase, leaving the amplified processing as the only issue. If we take all other tasks that a central processing node needs to carry out into account, we argue that the need of a matrix inversion per CB is not the bottleneck of a centralized architecture. Thus, it is the interconnection bandwidth that needs consideration.

3 Decentralized vs centralized: current debate

Linear processing in Massive MIMO was presented in the previous section together with two detection methods. These two schemes can be implemented in centralized or decentralized systems. In centralized architectures, the central node collects CSI from all antenna elements, represented by the matrix $\mathbf{H}$, which allows for optimal construction of the matrix $\mathbf{W}$. Apart from that, the central unit's tasks also include, e.g., demodulation and decoding. Processing at the antenna side comprises RF, ADC and optionally OFDM processing (FFT). The amount of interconnection data-rate between antennas and the central node depends on M, which is an important limitation in Massive MIMO systems where a large number of antenna elements is expected.

There is nowadays a trend towards decentralized systems in order to allow scalability of the system. Decentralized systems perform antenna processing locally, including CSI acquisition and partial detection. One of the key characteristics of this type of systems is that full CSI is not available at any point. The central node carries out remaining parts, such as per-user processing (e.g., symbol de-mapping and decoding).

In both types of architectures, antenna elements need to be connected to the central node. We acknowledge that the definition of a decentralized system is not an easy one to make, but here we define the term as *a system where the volume of the data provided during the training phase to the processing node is independent of M*. Our definition allows for the Gramian to be transfered, since it is of dimension $K \times K$.

We now, briefly, interlude the discussion by considering what tasks are needed during the payload data phase. Since matched filtering is information lossless, we can, without any loss of generality, assume that such operation is performed. This implies that, for each channel observation, each antenna should multiply its observed signal with a $K \times 1$ vector, and that the M resulting $K \times 1$ vectors should be summed and passed to the central node by specific hardware that has to be be in place.

Let us now return to a decentralized system. For K single-antenna users, there are K slots per CB where the payload data phase is inactive[1]. This releases K usages of the hardware mentioned just above. Earlier it was also mentioned that passage of the $K \times K$ Gramian to the central node is within the delineations of a decentralized architecture. Conveniently, such transfer can be done via K transmissions of $K \times 1$ vectors. Since the Gramian can be written as a sum over the M outer products of the antennas' channel vectors, we can conclude that the hardware used in the payload data phase is, 1) sufficient

[1] Thesis: In the original paper it was written "in inactive", while the correct words are "is inactive", referring to the resources needed for orthogonal pilots during channel estimation in training phase.

to transfer the Gramian, and 2) unoccupied for a duration sufficient for the transmission of the Gramian.

Under assumption that matched filtering is applied at each antenna, it remains for the central node during the payload data phase to invert the Gramian, and apply the inverse to its $K \times 1$ input vectors. We claim that this is of non-prohibitive complexity when taking its other tasks into account.

If we further, assume an absolute absence of noise, then the above described method would be optimal. In that case, we do not see much need for any further discussion on decentralized architecture design. Our sentiment is, thus, that non-ideal CSI must be assumed. Consequently, we must study the channel estimation problem that the central node faces, namely, that of the estimation of a Gramian matrix from a noisy version. We study this in some detail in the next section, with outcome that it poses no noteworthy challenges.

4 Channel estimation

In this section we discuss channel estimation of Gramian matrices. However, we commence by first going through all steps prior to the central node facing said estimation problem. An initial channel estimation is performed per-antenna basis, based on pilots sent by users. With a received signal $\mathbf{y}$ given by (1), the goal for each antenna is to obtain an estimate of its corresponding part of $\mathbf{H}$. If users send orthogonal pilots, we can define the $K \times K$ matrix $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_K]$, where $\mathbf{p}_k$ is defined as a $K \times 1$ vector of the following form

$$\mathbf{p}_k(i) \triangleq \begin{cases} p_k & \text{if } k = i \\ 0 & \text{if } k \neq i, \end{cases}$$

where p_k is a complex number known by the BS which represents the pilot from the k-th user. Based on this definition, the $M \times K$ BS observation matrix $\mathbf{Z}$ can be described by

$$\mathbf{Z} = \mathbf{HP} + \mathbf{N}, \tag{4}$$

where $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_M]^T$, $\mathbf{z}_m$ being the $K \times 1$ observation vector at antenna m. $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, ..., \mathbf{n}_M]^T$ represents the noise term as a $M \times K$ matrix and $\mathbf{n}_m$ the $K \times 1$ noise vector related to the same antenna. For antenna m, (4) becomes

$$\mathbf{z}_m = \mathbf{P}\mathbf{h}_m + \mathbf{n}_m,$$

and channel estimation is performed locally based on the observation vector $\mathbf{z}_m$. The Least Squares (LS) estimate of $\mathbf{H}$ is

$$\widehat{\mathbf{H}}_{\text{LS}} = \mathbf{Z}\mathbf{P}^{-1}, \tag{5}$$

and in the simple case $\mathbf{P} = \mathbf{I}$, then $\widehat{\mathbf{H}}_{\mathrm{LS}} = \mathbf{Z}$, or per antenna $\widehat{\mathbf{h}}_{m,\mathrm{LS}} = \mathbf{z}_m$.

The second step in channel estimation is to obtain $\mathbf{R} = \widehat{\mathbf{G}}_{\mathrm{LS}}$, where $\widehat{\mathbf{G}}_{\mathrm{LS}} = \widehat{\mathbf{H}}_{\mathrm{LS}}^H \widehat{\mathbf{H}}_{\mathrm{LS}}$. To do that, each antenna computes the partial term $\mathbf{R}_m = \widehat{\mathbf{h}}_m^* \widehat{\mathbf{h}}_m^T$, a $K \times K$ matrix, and therefore $\mathbf{R} = \sum_{m=1}^{M} \mathbf{R}_m$. This addition can be carried out by exploiting the existing antennas' connections needed for data detection.

In next subsections we obtain the PDF of $\mathbf{R}$ and $\mathbf{G}$, and the results will be used to formulate estimators of $\mathbf{G}$, which outperform the one-step channel estimation based solely on the decentralized LS channel estimate derived from (5).

4.1 Derivation of conditional PDFs

If $\mathbf{Z}$ is an $M \times K$ complex matrix whose rows follow a multivariate normal distribution with covariance matrix $\mathbf{\Sigma}$ and mean $\mathbb{E}(\mathbf{Z}) = \mathbf{H}$, then the conditional distribution of $\mathbf{R} = \mathbf{Z}^{\mathbf{H}}\mathbf{Z}$ is noncentral Wishart and is defined as [8]

$$\begin{aligned} p(\mathbf{R}|\mathbf{G}) = {} & \mathrm{e}^{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{G})}\, {}_0\tilde{F}_1(M; \mathbf{\Sigma}^{-1}\mathbf{G}\mathbf{\Sigma}^{-1}\mathbf{R}) \\ & \times \frac{1}{\tilde{\Gamma}_K(M)|\mathbf{\Sigma}|^M} e^{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{R})} |\mathbf{R}|^{M-K}, \end{aligned} \tag{6}$$

and noted as $\mathbf{R}|\mathbf{G} \sim \mathcal{W}_K(M, \mathbf{\Sigma}, \mathbf{\Sigma}^{-1}\mathbf{G})$. The matrix $\mathbf{\Sigma}^{-1}\mathbf{G}$ is referred to as noncentrality matrix in most literature.

The marginal PDF for matrix $\mathbf{G}$ becomes a central Wishart and its expression is as follows

$$p(\mathbf{G}) = \frac{1}{\tilde{\Gamma}_K(M)|\mathbf{C}|^M} e^{-\mathrm{tr}(\mathbf{C}^{-1}\mathbf{G})} |\mathbf{G}|^{M-K}, \tag{7}$$

and noted as $\mathbf{G} \sim \mathcal{W}_K(M, \mathbf{C})$. The marginal PDF of $\mathbf{R}$ can be obtained as

$$\begin{aligned} p(\mathbf{R}) = {} & \frac{1}{\tilde{\Gamma}_K(M)|\mathbf{\Sigma}|^M} e^{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{R})} |\mathbf{R}|^{M-K} \\ & \times \frac{1}{\tilde{\Gamma}_K(M)|\mathbf{C}|^M} \int e^{-tr[(\mathbf{\Sigma}^{-1}+\mathbf{C}^{-1})\mathbf{G}]} |\mathbf{G}|^{M-K} \\ & \times {}_0\tilde{F}_1(M; \mathbf{\Sigma}^{-1}\mathbf{G}\mathbf{\Sigma}^{-1}\mathbf{R}) \mathrm{d}G. \end{aligned}$$

Introducing a matrix $\mathbf{A}$ such that $\mathbf{A}^{-1}\mathbf{\Sigma}^{-1} = \mathbf{\Sigma}^{-1} + \mathbf{C}^{-1}$, which translates

to $\mathbf{A} = (\mathbf{I} + \mathbf{C}^{-1}\mathbf{\Sigma})^{-1}$, leads to

$$\begin{aligned} p(\mathbf{R}) = & \frac{|\mathbf{\Sigma A}|^M}{\tilde{\Gamma}_K(M)|\mathbf{\Sigma}|^M|\mathbf{C}|^M} e^{-tr[\mathbf{\Sigma}^{-1}\mathbf{R}(\mathbf{I}-\mathbf{A})]}|\mathbf{R}|^{M-K} \\ & \times \int \frac{1}{\tilde{\Gamma}_K(M)|\Sigma\mathbf{A}|^M} e^{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{RA})} e^{-\mathrm{tr}(\mathbf{A}^{-1}\mathbf{\Sigma}^{-1}\mathbf{G})} \\ & \times |\mathbf{G}|^{M-K} {}_0\tilde{F}_1(M; \mathbf{\Sigma}^{-1}\mathbf{R}\mathbf{\Sigma}^{-1}\mathbf{G})\mathrm{d}G, \end{aligned} \tag{8}$$

where we used the fact that $\mathrm{eig}(\mathbf{PXPY}) = \mathrm{eig}(\mathbf{PYPX})$ given any complex matrix $\mathbf{P}$, being $\mathbf{X}$ and $\mathbf{Y}$ two Hermitian complex matrices, and therefore ${}_0\tilde{F}_1(M; \mathbf{PXPY}) = {}_0\tilde{F}_1(M; \mathbf{PYPX})$. In our case $\mathbf{P} = \mathbf{\Sigma}^{-1}$, $\mathbf{X} = \mathbf{G}$ and $\mathbf{Y} = \mathbf{R}$.

We can manipulate $\frac{|\mathbf{\Sigma A}|^M}{|\mathbf{\Sigma}|^M|\mathbf{C}|^M}$ as follows

$$\begin{aligned} \frac{|\mathbf{\Sigma A}|^M}{|\mathbf{\Sigma}|^M|\mathbf{C}|^M} &= |\mathbf{A}|^M|\mathbf{C}^{-1}|^M \\ &= |\mathbf{A}(\mathbf{A}^{-1} - \mathbf{I})\mathbf{\Sigma}^{-1}|^M \\ &= |(\mathbf{I} - \mathbf{A})\mathbf{\Sigma}^{-1}|^M, \end{aligned} \tag{9}$$

where the equality $\mathbf{C}^{-1} = (\mathbf{A}^{-1} - \mathbf{I})\mathbf{\Sigma}^{-1}$ has been used.

By comparing (6) and (8) it is possible to observe that the second part of (8) is actually the integral of a noncentral Wishart PDF and therefore its value must be 1. The final expression for $p(\mathbf{R})$ is a central Wishart as shown below

$$p(\mathbf{R}) = \frac{1}{\tilde{\Gamma}_K(M)|\mathbf{\Sigma}(\mathbf{I} - \mathbf{A})^{-1}|^M} e^{-\mathrm{tr}[(\mathbf{I}-\mathbf{A})\mathbf{\Sigma}^{-1}\mathbf{R}]}|\mathbf{R}|^{M-K}, \tag{10}$$

which we denote as $\mathbf{R} \sim \mathcal{W}_K(M, \mathbf{\Sigma}(\mathbf{I} - \mathbf{A})^{-1})$. Additionally, using $\mathbf{\Sigma}(\mathbf{I} - \mathbf{A})^{-1} = \mathbf{C} + \mathbf{\Sigma}$ then $\mathbf{R} \sim \mathcal{W}_K(M, \mathbf{C} + \mathbf{\Sigma})$.

Finally, the posterior PDF can be obtained by using Bayes' theorem and the results from (6), (7), (9) and (10) as follows

$$\begin{aligned} p(\mathbf{G}|\mathbf{R}) &= \frac{p(\mathbf{R}|\mathbf{G})p(\mathbf{G})}{p(\mathbf{R})} \\ &= e^{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{RA})} {}_0\tilde{F}_1(M; \mathbf{\Sigma}^{-1}\mathbf{R}\mathbf{\Sigma}^{-1}\mathbf{G}) \\ &\times \frac{1}{\tilde{\Gamma}_K(M)|\mathbf{\Sigma A}|^M} e^{-\mathrm{tr}(\mathbf{A}^{-1}\mathbf{\Sigma}^{-1}\mathbf{G})}|\mathbf{G}|^{M-K}. \end{aligned} \tag{11}$$

PDF in equation (11), assuming IID noise samples, is a noncentral Wishart, $\mathbf{G}|\mathbf{R} \sim \mathcal{W}_K(M, \mathbf{\Sigma A}, \mathbf{A}^{-1}\mathbf{\Sigma}^{-1}\mathbf{ARA})$.

Once we have presented the PDFs involved in this study we can introduce the estimators.

4.2 Maximum Likelihood (ML)

Maximum Likelihood (ML) estimate of $\mathbf{G}$ is defined as the matrix $\widehat{\mathbf{G}}_{\mathrm{ML}}$, which maximizes the likelihood as follows

$$\begin{aligned}\widehat{\mathbf{G}}_{\mathrm{ML}} &= \arg\max_{\mathrm{G}}\{p(\mathbf{R}|\mathbf{G})\} \\ &= \arg\max_{\mathrm{G}}\{-\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{G}) \\ &+ \log[{}_0\tilde{F}_1(M;\mathbf{\Sigma}^{-1}\mathbf{G}\mathbf{\Sigma}^{-1}\mathbf{R})]\}.\end{aligned} \tag{12}$$

If $\mathbf{R}$ has an eigenvalue decomposition $\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H$, then we look for solutions whose form is $\mathbf{G} = \mathbf{U}\mathbf{\Omega}\mathbf{U}^H$, where $\mathbf{\Omega}$ and $\mathbf{\Lambda}$ are diagonal matrices containing the eigenvalues of $\mathbf{G}$ and $\mathbf{R}$ respectively. In the particular case that all pilots have equal power, i.e., $|p_k| = |p|, \forall k$, then without loosing generality we can set $|p| = 1$, leading to $\mathbf{\Sigma} = N_0\mathbf{I}$, and (12) can be expressed as

$$\widehat{\mathbf{G}}_{\mathrm{ML}} = \arg\max_{\mathbf{\Omega}}\left\{-\frac{1}{N_0}\mathrm{tr}(\mathbf{\Omega}) + \log\left[{}_0\tilde{F}_1(M;\frac{1}{N_0^2}\mathbf{\Omega}\mathbf{\Lambda})\right]\right\}.$$

After derivation of the argument of the previous expression, the optimality condition can be expressed as

$$\frac{1}{N_0}{}_0\tilde{F}_1(M;\frac{1}{N_0^2}\mathbf{\Omega}^*\mathbf{\Lambda})\mathbf{I} = \frac{\partial_0\tilde{F}_1(M;\frac{1}{N_0^2}\mathbf{\Omega}\mathbf{\Lambda})}{\partial\mathbf{\Omega}}|_{\mathbf{\Omega}=\mathbf{\Omega}^*},$$

where $\widehat{\mathbf{G}}_{\mathrm{ML}} = \mathbf{U}\mathbf{\Omega}^*\mathbf{U}^H$, and from where we do not continue in an analytical form.

4.3 Maximum A Posteriori (MAP)

The Maximum A Posteriori (MAP) estimate is defined as

$$\begin{aligned}\widehat{\mathbf{G}}_{\mathrm{MAP}} &= \arg\max_{\mathrm{G}}\{p(\mathbf{G}|\mathbf{R})\} \\ &= \arg\max_{\mathrm{G}}\log\left[{}_0\tilde{F}_1(M;\mathbf{\Sigma}^{-1}\mathbf{R}\mathbf{\Sigma}^{-1}\mathbf{G})\right] \\ &- \mathrm{tr}(\mathbf{A}^{-1}\mathbf{\Sigma}^{-1}\mathbf{G}) + (M-K)\log|\mathbf{G}|.\end{aligned}$$

If $\mathbf{\Sigma} = N_0\mathbf{I}$ and $\mathbf{C} = h_{\mathrm{pow}}\mathbf{I}$ then $\mathbf{A}^{-1}\mathbf{\Sigma}^{-1} = \left(\frac{1}{N_0} + \frac{1}{h_{\mathrm{pow}}}\right)\mathbf{I}$ and therefore the MAP estimate can be expressed as follows, where $\mu_1, .., \mu_K$ are the eigenvalues of $\mathbf{G}$,

$$\begin{aligned}\widehat{\mathbf{G}}_{\mathrm{MAP}} = \arg &\max_{\mu_1,\mu_2,..,\mu_K} \log\left[{}_0\tilde{F}_1(M;\frac{1}{N_0^2}\mathbf{\Omega}\mathbf{\Lambda})\right] \\ &- \left(\frac{1}{N_0} + \frac{1}{h_{\mathrm{pow}}}\right)\sum_{i=1}^{K}\mu_i + (M-K)\sum_{i=1}^{K}\log(\mu_i).\end{aligned}$$

4.4 Minimum Mean Square Error (MMSE)

The Minimum Mean Square Error (MMSE) estimate is defined as the expectation over the posterior probability. Taking into account that $\mathbf{G}|\mathbf{R}$ is a non-central Wishart, that is $\mathbf{G}|\mathbf{R} \sim \mathcal{W}_K(M, \mathbf{\Sigma A}, \mathbf{A}^{-1}\mathbf{\Sigma}^{-1}\mathbf{ARA})$, the expectation is

$$\widehat{\mathbf{G}}_{\text{MMSE}} = \mathbb{E}(\mathbf{G}|\mathbf{R}) = M\mathbf{\Sigma A} + \mathbf{ARA}. \tag{13}$$

If $\mathbf{C} = h_{\text{pow}}\mathbf{I}$ and $\mathbf{\Sigma} = N_0\mathbf{I}$ then (13) becomes

$$\widehat{\mathbf{G}}_{\text{MMSE}} = \frac{h_{\text{pow}}N_0}{h_{\text{pow}} + N_0} M\mathbf{I} + \left(\frac{h_{\text{pow}}}{h_{\text{pow}} + N_0}\right)^2 \mathbf{R}. \tag{14}$$

When $h_{\text{pow}} \gg N_0$ then $\widehat{\mathbf{G}}_{\text{MMSE}} \approx \mathbf{R}$. On the other hand, when $N_0 \gg h_{\text{pow}}$ then $\widehat{\mathbf{G}}_{\text{MMSE}} \approx h_{\text{pow}}M\mathbf{I}$, which does not depend on $\mathbf{R}$, due to $\mathbf{R}$ conveys almost no information about $\mathbf{G}$ in that case.

5 Equalizer formulation and results

The ZF equalization matrix is made up of two parts (3). The first part (or MRC part), $\widehat{\mathbf{H}}^H$, that is implemented in a decentralized form by the antenna modules, and the second part, $\widehat{\mathbf{G}}^{-1}$, that is performed in a central processing unit. Let's define two equalizers and compare their performance. ZF based on MMSE channel estimation (14) and LS channel estimation (5), is defined as, respectively,

$$\mathbf{W}_{\text{MMSE}} = \widehat{\mathbf{G}}^{-1}_{\text{MMSE}}\widehat{\mathbf{H}}^H_{\text{LS}}, \quad \mathbf{W}_{\text{LS}} = \widehat{\mathbf{G}}^{-1}_{\text{LS}}\widehat{\mathbf{H}}^H_{\text{LS}}.$$

5.1 Channel estimation

In this subsection we compare both channel estimation methods based on the relative error defined as follows

$$\epsilon_{\text{Q}} \triangleq \mathbb{E}\left\{\frac{\|\widehat{\mathbf{G}}_{\text{Q}} - \mathbf{G}\|^2_{\text{F}}}{\|\mathbf{G}\|^2_{\text{F}}}\right\},$$

where $Q \in \{\text{MMSE}, \text{LS}\}$. In scenarios with high SNR, that is $h_{\text{pow}} \gg N_0$, (14) simplifies to $\widehat{\mathbf{G}}_{\text{MMSE}} \simeq \mathbf{R}$, so $\epsilon_{\text{MMSE}} \simeq \epsilon_{\text{LS}}$. At low SNR levels, that is $N_0 \gg h_{\text{pow}}$, $\widehat{\mathbf{G}}_{\text{MMSE}} \approx h_{\text{pow}}M\mathbf{I}$ and ϵ_{MMSE} saturates unlike ϵ_{LS}, that grows with N_0. Figure 1 shows this behavior in simulation[2].

[2] Thesis: In the original paper, the figure included reference to ZF, while the correct method is LS. Also in the caption a reference to $\mathbf{W}$ was made, when this is not required in channel estimation, and therefore removed in the present thesis.

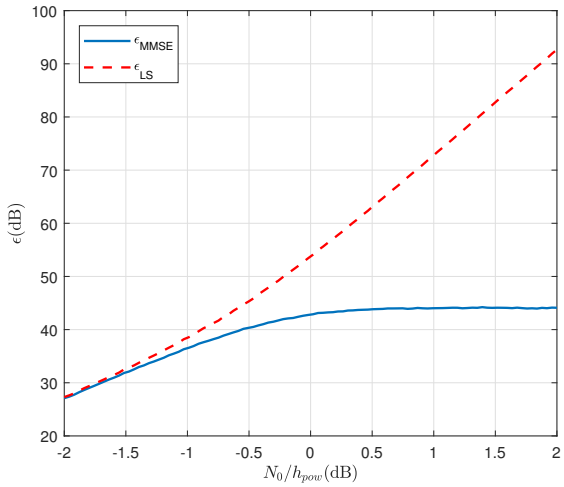


Figure 1: Channel estimation error vs. noise level for both methods, MMSE and LS. M=100. K=16. $\epsilon(\text{dB}) = 10\log_{10}(\epsilon)$.

5.2 Sum-rate

The instantaneous SINR for user k is given by

$$\text{SINR}_k = \frac{|\mathbf{w}_k^H \mathbf{h}_k|^2}{\sum_{i=1, i\neq k}^{K} |\mathbf{w}_k^H \mathbf{h}_i|^2 + N_0 \|\mathbf{w}_k\|^2},$$

and the ergodic sum-rate as follows

$$R = \mathbb{E}\left\{ \sum_{k=1}^{K} \log_2 \left(1 + \text{SINR}_k\right) \right\}. \tag{15}$$

Figure 2 shows the simulation results in terms of ergodic sum-rate for ZF with MMSE and LS channel estimators versus $\frac{K}{M}$. It is observed same performance for relative low K. MMSE outperforms the LS counterpart as K grows which shows greater robustness against inter-user interference.

6 Conclusions

We claim that the Gramian can be collected in a processing node by re-using existing connections without increasing inter-connection data-rate requirements,

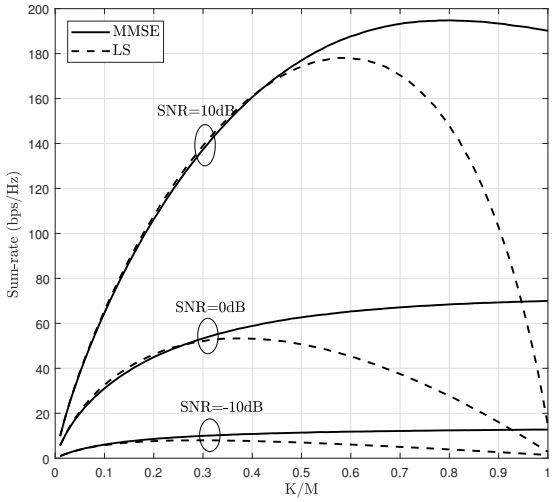


Figure 2: Total sum-rate for ZF detector with both channel estimation methods, LS and MMSE, according to (15) versus $\frac{K}{M}$ for M=100. SNR at base-station receiver antenna. SNR $= h_{\text{pow}}/N_0$.

and that this is sufficiently decentralized for practical purposes. In this paper we have discussed in detail the ensuing channel estimation in decentralized Massive MIMO systems, where channel information is distributed and not fully available in one single point. We proposed a channel estimation based on the Gramian matrix for further ZF detection which outperforms direct methods based uniquely on per antenna processing.

Bibliography

[1] T. L. Marzetta, Noncooperative cellular wireless with unlimited numbers of base station antennas, *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590-3600, November 2010.

[2] C. Shepard *et al.*, Argos: Practical many-antenna base stations, in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (Mobicom)*, New York, NY, USA, 2012, pp. 53-64.

[3] S. Malkowsky *et al.*., The worlds first real-time testbed for massive MIMO: Design, implementation, and validation, *IEEE Access*, vol. 5, pp. 9073-9088, 2017.

[4] 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, Dec 2017.

[5] C. Jeon, K. Li, J. R. Cavallaro, and C. Studer, Decentralized Equalization with Feedforward Architectures for Massive MU-MIMO, *ArXiv e-prints*, Aug. 2018.

[6] J. R. Sanchez, F. Rusek, M. Sarajlic, O. Edfors, and L. Liu, Fully decentralized massive mimo detection based on recursive methods, in *2018 IEEE International Workshop on Signal Processing Systems (SiPS)*, Oct 2018, pp. 53-58.

[7] M. Sarajlic, F. Rusek, J. R. Sanchez, L. Liu, and O. Edfors, Fully decentralized approximate zero-forcing precoding for massive mimo systems, *IEEE Wireless Communications Letters*, pp. 1-1, 2019.

[8] A. T. James, Distributions of matrix variates and latent roots derived from normal samples, *Ann. Math. Statist.*, vol. 35, no. 2, pp. 475-501, 06 1964.

[9] A. Zanella, M. Chiani, and M. Z. Win, On the marginal distribution of the eigenvalues of wishart matrices, *IEEE Transactions on Communications*, vol. 57, no. 4, pp. 1050-1060, April 2009.

[10] M. Kang and M. Alouini, Largest eigenvalue of complex wishart matrices and performance analysis of mimo mrc systems, *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 418-426, April 2003.

[11] K. Gross and D. Richards, Total positivity, spherical series, and hypergeometric functions of matrix argument, *IEEE Journal of Approximation Theory*, no. 59, pp. 229-246, 1989.

[12] A. Gupta, Y. Sheena, and Y. Fujikoshi, Estimation of the eigenvalues of noncentrality parameter matrix in noncentral wishart distribution, *Journal of Multivariate Analysis*, vol. 93, no. 1, pp. 1-20, 2005.

[13] P. Koev and A. Edelman, The efficient evaluation of the hypergeometric function of a matrix argument, *Math. Comp*, p. 2006.

[14] Y. Sheena, A. K. Gupta, and Y. Fujikoshi, Estimation of the eigenvalues of noncentrality parameter in matrix variate noncentral beta distribution, *Annals of the Institute of Statistical Mathematics*, vol. 56, no. 1, pp. 101-125, Mar 2004.

[15] A. K. Gupta and D. K. Nagar, Matrix Variate Distributions. *Monographs and Suerveys in Pure and Applied Mathematics.* 104. Chapman & Hall/CRC, 2000.

Paper III

Decentralized Massive MIMO Processing Exploring Daisy-chain Architecture and Recursive Algorithms

Algorithms for Massive MIMO uplink detection and downlink precoding typically rely on a centralized approach, by which baseband data from all antenna modules are routed to a central node in order to be processed. In the case of Massive MIMO, where hundreds or thousands of antennas are expected in the base-station, said routing becomes a bottleneck since interconnection throughput is limited. This paper presents a fully decentralized architecture and an algorithm for Massive MIMO uplink detection and downlink precoding based on the Coordinate Descent (CD) method, which does not require a central node for these tasks. Through a recursive approach and very low complexity operations, the proposed algorithm provides a good trade-off between performance, interconnection throughput and latency. Further, our proposed solution achieves significantly lower interconnection data-rate than other architectures, enabling future scalability.

Jesús Rodríguez Sánchez, Fredrik Rusek, Ove Edfors, Muris Sarajlić and Liang Liu, "Decentralized Massive MIMO Processing Exploring Daisy-chain Architecture and Recursive Algorithms,"
in *IEEE Transactions on Signal Processing*, vol. 68, pp. 687-700, 2020.

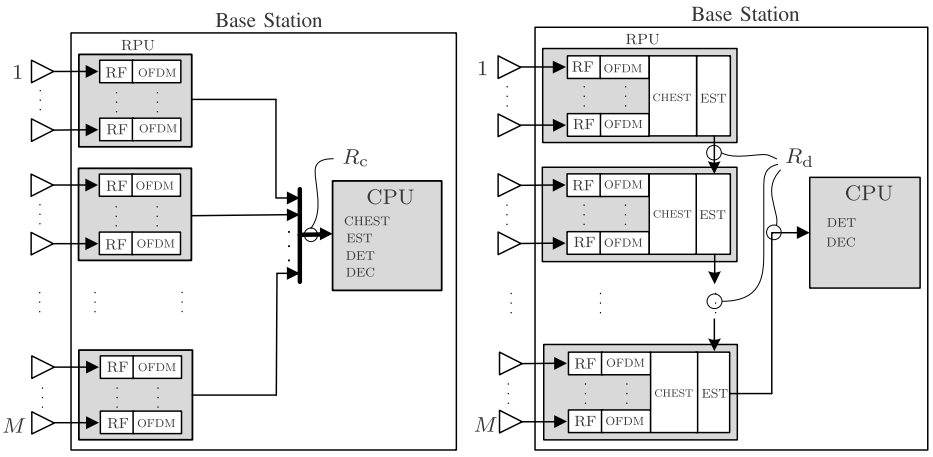
1 Introduction

Massive MIMO is one of the most relevant technologies in wireless communications [3, 4]. Among the key features of this technology are high spectral efficiency and improved link reliability, making it a key enabler for 5G. Massive MIMO exploits spatial diversity far beyond traditional MIMO systems by employing a large scale antenna array in the base-station (BS) with hundreds or possibly even thousands of elements. This large number of elements allows for unprecedented spatial resolution and high spectral efficiency, while providing simultaneous service to several users within the same time-frequency resource.

Despite all the advantages of Massive MIMO, there are still challenges from an implementation point of view. One of the most critical ones is sending data from the BS antennas to the central processing unit (CPU) and vice-versa, and the high interconnection throughput it requires. In current set-ups, uplink detection algorithms based on zero-forcing (ZF) equalizer typically rely on a centralized architecture, shown in Fig. 1a, where baseband samples are collected in the CPU for obtaining channel state information (CSI) and further matrix inversion, which allows data estimation and further detection. The same argument is valid for downlink precoding. In order to avoid dedicated links between antenna modules and CPU, a shared bus is typically used to exchange this data. In case of LuMaMi testbed [5, 6], the shared bus was reported to support an aggregated data-rate of 384Gps, which exceed base-station internal interface standards such as eCPRI [7]. Additionally, the pin-count of integrated circuits (IC) limits the number of links the IC can handle simultaneously and thus the throughput. Due to this high data-rate, the power appears as another potential limitation. This combination of factors are considered as the main bottleneck in the system and a clear limitation for array scalability. In this paper we will address the inter-connection throughput limitation by decreasing its value per link and consequently reducing the impact of the other two (pin-count and power).

The inter-connection bottleneck has been noted in several previous studies on different architectures for Massive MIMO BSs [5, 8–13]. As a solution, most of these studies recommend moving to a decentralized approach where uplink estimation and downlink precoding can be performed locally in processing nodes close to the antennas (final detection can still be done in a CPU). However, to achieve that, CSI still needs to be collected in the CPU, where matrix inversion is performed [5, 8, 9], imposing an overhead in data shuffling.

The CSI problem is addressed in [11], where CSI is obtained and used only locally (not shared) for precoding and estimation, with performance close to MMSE. However, this architecture relies on the CPU for exchanging a certain amount of consensus information between the nodes, and this exchange nega-



(a) Centralized architecture (b) Decentralized architecture

Figure 1: Comparison between base station receiver chain in centralized and fully decentralized architectures for Massive MIMO uplink. Antenna array with M elements is divided into RPUs, each containing a set of antennas. (a): Centralized architecture. Each RPU has one link to transfer baseband samples to the CPU, where the rest of processing tasks are done. (b): Fully decentralized architecture for detection. Each RPU performs RF, ADC, OFDM, channel estimation (CHEST) and data estimation (EST) locally. Detection (DET) and decoding (DEC) is centralized. RPUs are connected to each other by uni-directional links. Only one RPU has a direct connection with the CPU. Proposed algorithms are executed in EST blocks in parallel mode. The points where the interconnection data-rate is estimated are marked by circles and the value is denoted by R_c and R_d for centralized and decentralized respectively. The goal is to have $\mathrm{R}_d \ll \mathrm{R}_c$ without compromising performance and latency.

tively impacts the processing latency and throughput [12], and therefore limits the scalability of this solution. In order to solve these problems, feedforward architectures for detection [13] and precoding [12] have been proposed recently, where the authors present a partially decentralized (PD) architecture for detection and precoding, which achieves the same results as linear methods (MRC, ZF, L-MMSE), and therefore becomes optimal when M is large enough. Partial Gramian matrices from antennas are added up before arriving to a processing

unit where the Gramian is inverted.

In [8], a flat-tree structure with daisy-chained nodes was presented. The authors propose conjugate beamforming as a fully decentralized method with the corresponding penalty in system capacity. In the same work it is also pointed out that by following this topology the latency was being severely compromised. The more detailed analysis on latency is thus needed to evaluate the algorithm.

In this article we propose a fully decentralized architecture and a recursive algorithm for Massive MIMO detection and precoding, which is able to achieve very low inter-connection data-rate without compromising latency. The proposed algorithm is pipelined so that it runs in a distributed way at the antenna processing units, providing local vectors for estimation/detection that approximate to the zero-forcing solution. We make use of the Coordinate Descent (CD) algorithm, which is detailed in Section 4, to compute these vectors.

There is previous work based on CD, such as [14]. The main difference is that the coordinate update in [14] is done per user basis, i.e., a different user index is updated every iteration, while in our proposed method the coordinate update is done per antenna basis, updating all users at once.

We extend the work presented in [1] and [2], which are also based on decentralized daisy-chain architecture. The novelties of the present work compared to these two is as follows:

- A common strategy for downlink precoding and uplink equalization is presented, in contrast to [1] and [2], which only covers uplink and downlink separately.
- The algorithm has been modified that serial processing is only needed when new CSIs are estimated. The corresponding filtering phase can be conducted in parallel to reduce latency, in contrast to [1], where serial processing is always needed, which increases the latency.
- A recommended step-size is provided, in contrast to [1].
- An analytical expression for resulting SINR and a complete performance analysis is presented in this paper.
- Complexity analysis from a general point of view (not attached to any specific implementation) is provided, which includes: inter-connection data-rate, memory size and latency. In [1], only inter-connection data-rates are analyzed.

Decentralized architectures, as shown in Fig. 1b, have several advantages compared to the centralized counterpart, as shown in Fig. 1a. For example,

they overcome bottlenecks by finding a more equal distribution of the system requirements among the processing nodes of the system. Apart from this, data localization is a key characteristic of decentralized architectures. In uplink, the architecture allows data to be consumed as close as possible to where it is generated, minimizing the amount to transfer, and therefore saving throughput and energy. To achieve data localization, processing nodes need to be located near the antenna, where they perform processing tasks locally such as channel and data estimation. Local CSI is estimated and stored locally in each, without any need to share it with any other nodes in the system. This approach has been suggested previously in [8–13], and we take advantage of it in the proposed solution.

The remainder of the paper is organized as follows. In Section 2 the preliminaries are presented, comprising the system model for uplink and downlink, together with an introduction to linear processing and the ZF method. Section 3 is dedicated to a comparison between the centralized and decentralized architectures and reasoning why the latter one is needed, together with an overview of the daisy-chain topology. The proposed algorithm, based on CD, is presented in Section 4. In 5 closed-form expressions of the SIR and SINR are provided for this algorithm, together with interconnection data-rates, latency and memory requirements of the proposed solution. Finally, Section 6 summarizes the conclusions of this publication.

Notation: In this paper, lowercase, bold lowercase and upper bold face letters stand for scalar, column vector and matrix, respectively. The operations $(.)^T$, $(.)^*$ and $(.)^H$ denote transpose, conjugate and conjugate transpose respectively. The i-th element of vector $\mathbf{h}$ is denoted as h_i. A vector $\mathbf{w}$ and a matrix $\mathbf{A}$ related to the mth antenna is denoted by $\mathbf{w}_m$ and $\mathbf{A}_m$, respectively. $A_{i,j}$ denotes element (i,j) of $\mathbf{A}$. $\mathbf{A}_m(i,j)$ denotes element (i,j) of the m-th matrix in the sequence $\{\mathbf{A}_m\}$. The kth coordinate vector in $\mathbb{R}^K$ is defined as $\mathbf{e}_k$. Kronecker delta is represented as δ_{ij}. Probability density function and cumulative density function are denoted respectively as $f_{\mathbf{X}}(x)$ and $F_{\mathbf{X}}(x)$. Computational complexity is measured in terms of the number of complex-valued multiplications.

2 Background

2.1 System model

For uplink, we consider a scenario with K single-antenna users transmitting to a BS with an antenna array with M elements. Assuming time-frequency-based channel access, a Resource Element (RE) represents a unit in the time-

frequency grid (also named subcarrier in OFDM) where the channel is expected to be approximately flat. Under this scenario, the input-output relation is

$$\mathbf{y}^{\mathrm{u}} = \mathbf{H}\mathbf{x}^{\mathrm{u}} + \mathbf{n}^{\mathrm{u}}, \tag{1}$$

where $\mathbf{y}^{\mathrm{u}}$ is the $M \times 1$ received vector, $\mathbf{x}^{\mathrm{u}}$ is the transmitted user data vector $(K \times 1)$, $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \cdots \mathbf{h}_M]^T$ is the channel matrix $(M \times K)$ and $\mathbf{n}^{\mathrm{u}}$ an $M \times 1$ vector of white, zero-mean complex Gaussian noise. The entries of $\mathbf{H}$ are i.i.d. zero-mean circularly-symmetric complex-gaussian entries, with rows $\mathbf{h}_i \sim \mathcal{CN}(0, \mathbf{I})$ for all i. The noise covariance at the receiver is $N_0\mathbf{I}$. The average transmitted power is assumed to be equal across all users and we assume, without any loss of generality, a unit transmit power. SNR is defined as $\frac{1}{N_0}$ and represents the average "transmit" signal-to-noise ratio.

For downlink, if Time Division Duplex (TDD) is assumed, then according to channel reciprocity principle and by employing reciprocity calibration techniques [15], it is assumed that within the same coherence time, the channel matrix is the same as in the uplink case, and the system model follows

$$\tilde{\mathbf{x}}^{\mathrm{d}} = \mathbf{H}^T\mathbf{y}^{\mathrm{d}} + \mathbf{n}^{\mathrm{d}}, \tag{2}$$

for a RE, where $\mathbf{y}^{\mathrm{d}}$ is the $M \times 1$ transmitted vector, $\tilde{\mathbf{x}}^{\mathrm{d}}$ is the received data vector by users $(K \times 1)$, and $\mathbf{n}^{\mathrm{d}}$ samples of noise $(K \times 1)$.

Once the system model is established, we introduce the linear processing fundamentals used for downlink precoding and uplink estimation.

2.2 Linear processing

In this article we focus on linear estimators and precoders, because they show close to optimal performance in Massive MIMO regime while requiring low complexity.

A linear estimator provides $\hat{\mathbf{x}}^{\mathrm{u}}$, which is an estimate of $\mathbf{x}^{\mathrm{u}}$, by applying an equalizer filter matrix $\mathbf{W}$ to the vector of observations, $\mathbf{y}^{\mathrm{u}}$:

$$\begin{aligned} \hat{\mathbf{x}}^{\mathrm{u}} &= \mathbf{W}^H\mathbf{y}^{\mathrm{u}} \\ &= \sum_{m=1}^{M} \mathbf{w}_m^* y_m^{\mathrm{u}}, \end{aligned} \tag{3}$$

where $\mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2 \cdots \mathbf{w}_M]^T$ is an $M \times K$ matrix, $\mathbf{w}_m$ is a $K \times 1$ filter vector related to antenna m and y_m^{u} the observation at antenna m. As it can be seen the estimate $\hat{\mathbf{x}}^{\mathrm{u}}$ is computed by the sum of M partial products. If $\mathbf{w}_m$ is obtained and stored locally in the mth antenna module, then the partial

products can be computed with local data only, reducing the amount of data to exchange between nodes. From implementation point of view, the linear estimator relies on the accumulation of all partial results according to (3), which can be done centrally (fusion node) or distributed.

For downlink, the data vector intended to the users, $\mathbf{x}^{\mathrm{d}}$, is precoded with matrix $\mathbf{P}$ as

$$\mathbf{y}^{\mathrm{d}} = \mathbf{P}\mathbf{x}^{\mathrm{d}}, \tag{4}$$

where $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2\ \cdots\ \mathbf{p}_M]^T$ is an $M \times K$ matrix, which fulfills a power constraint $\|\mathbf{P}\|_F^2 \leq P$, such that P is the maximum transmitted power. Particularly for antenna m we have

$$y_m^{\mathrm{d}} = \mathbf{p}_m^T \mathbf{x}^{\mathrm{d}}. \tag{5}$$

Similarly to uplink, if $\mathbf{p}_m$ is obtained and stored locally at the mth antenna module, then y_m^{d} can be computed only with local data after $\mathbf{x}^{\mathrm{d}}$ is broadcasted to all antennas.

The zero-forcing (ZF) equalizer, which is one type of linear estimator, constitutes a reference in our analysis. It is defined for uplink estimation as

$$\mathbf{W}_{\mathrm{ZF}}^H = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H, \tag{6}$$

and $\mathbf{P}_{\mathrm{ZF}} = \mathbf{W}_{\mathrm{ZF}}^*$ for the downlink precoding.

ZF is able to completely cancel inter-user interference (IUI) and reach the promised spectral efficiency of Massive MIMO. However, as ZF is performed in a central processor, the Gramian matrix $\mathbf{H}^H\mathbf{H}$ needs to be collected and inverted, which increases the average inter-connection data-rate. The computational load is also increased due to the matrix inversion and posterior matrix multiplication during estimation phase. Taking this into consideration, we look for methods with IUI-cancellation capabilities but with lower requirements for the system.

2.3 Uplink & downlink reciprocity

Substituting (1) into (3) leads to

$$\hat{\mathbf{x}}^{\mathrm{u}} = \mathbf{E}_{\mathrm{u}}\mathbf{x}^{\mathrm{u}} + \mathbf{z}^{\mathrm{u}} \tag{7}$$

for uplink, where $\mathbf{E}_{\mathrm{u}} = \mathbf{W}^H\mathbf{H}$ is a $K \times K$ matrix containing the equivalent uplink channel with IUI information and $\mathbf{z}^u$ is the $K \times 1$ post-equalization noise term.

On the other hand, in the downlink, substituting (4) into (2) leads to

$$\tilde{\mathbf{x}}^{\mathrm{d}} = \mathbf{E}_{\mathrm{d}}\mathbf{x}^{\mathrm{d}} + \mathbf{n}^{\mathrm{d}}, \tag{8}$$

where $\mathbf{E}_{\mathrm{d}} = \mathbf{H}^T\mathbf{P}$ is a $K \times K$ matrix containing the equivalent downlink channel with IUI information. For the particular case that $\mathbf{P}^T = \mathbf{W}^H$, we have $\mathbf{E}_{\mathrm{d}} = \mathbf{E}_{\mathrm{u}}^T$, meaning that both equivalent channels are transposed, and therefore experiment the same IUI cancellation properties. From this result it is clear that once an equalization matrix $\mathbf{W}$ is obtained for uplink detection, it can also be applied for downlink precoding with no extra effort. It is interesting to note that, since $\mathbf{P}^T = \mathbf{W}^H$, it follows that $\mathbf{p}_i = \mathbf{w}_i^*$, so each antenna node can re-use same vector for detection and precoding, ideally reducing complexity and storage needs by half. Said that, in this article we focus mainly on uplink estimation without limiting the results to downlink. In reality, there is a downlink power constraint as the total transmitted power, which is addressed in 5.

3 Centralized vs decentralized

In this section we describe the differences between centralized and decentralized Massive MIMO processing and the justification to study the later one.

Uplink estimation based on ZF equalization has two components that should be multiplied: $\mathbf{W}_{\mathrm{ZF}}$ and $\mathbf{y}^{\mathrm{u}}$. The former includes a $K \times K$ matrix inversion, which typically is done in one place, and for that, CSI from all antennas needs to be collected. Apart from that, the observation data vector, $\mathbf{y}^{\mathrm{u}}$, is also needed for estimation. This vector is $M \times 1$, increasing considerably the amount of data to transfer and limiting the scalability of the array. Based on those considerations, we can think of two possible architectures for the Massive MIMO base-station: centralized and decentralized.

Fig. 1a presents an architecture based on a central baseband processing node, where baseband samples are exchanged between Remote Processing Units (RPUs) and CPU. Each antenna is connected to a receiver and transmitter circuitry, which involves: RF front-end, ADC/DAC and OFDM processing. For simplicity, only uplink is represented in this figure. We can identify some common tasks among these processing elements across different antennas, such as: time synchronization, automatic gain control, local oscillator generation, carrier frequency and sampling rate offset estimation, phase noise compensation, among others. Therefore, a few antennas (together with corresponding receivers/transmitters) can be grouped into one RPU for efficient implementation of such common tasks. However, for simplicity, in this work we only analyze the case where each RPU manages one antenna.

Dedicated physical links would easily exceed the number of I/O connections in current standards, in addition to the increment of the cost of adding a new RPUs when needed. To overcome this, we consider that RPUs are connected

to the CPU node by a shared bus as shown in Fig. 1a.

Even though, this approach can support ZF detection (and precoding) from a functionality point of view, from the implementation point of view, it requires a very high inter-connection data-rate in the bus and at the input of the CPU (R_{c} in the figure). As an example, consider a 5G NR-based system with 128 antennas and OFDM as an access technology, then the average data-rate can be calculated as

$$R_{\mathrm{c}} = \frac{2wMN_{\mathrm{u}}}{T_{\mathrm{OFDM}}}, \tag{9}$$

where N_{u} is the number of active subcarriers, w is the bit-width for the baseband samples (real/imaginary parts) after FFT, and T_{OFDM} is the OFDM symbol duration. For $N_{\mathrm{u}} = 3300$, $w = 12$ and $T_{\mathrm{OFDM}} = 1/120\mathrm{kHz}$ then $R_{\mathrm{c}} = 1.2\mathrm{Tbps}$. This result clearly exceed the limit data-rate for common interfaces, such as eCPRI [7] and PCIe, and furthermore, it is proportional to M, which clearly limits the scalability of the system.

As a solution to this limitation, we propose the fully-decentralized architecture for baseband detection and precoding shown in Figure 1b. We can observe that channel estimation and estimation/precoding have been moved from the CPU to the RPUs, with detection and decoding as a remaining task in the CPU from physical layer point of view. The benefit of this move is manifold. Firstly, the inter-connection data-rate scales with K instead of M. Secondly, the high complexity requirement in the CPU for channel estimation and data estimation/precoding is now equally distributed among RPUs, which highly simplifies the implementation and overcomes the computational bottleneck and, additionally, CSI is obtained and consumed locally in each RPU without the need for exchange, with the consequent reduction in the required inter-connection data-rate. In addition to the advantages already mentioned, which are common to other decentralized schemes, the proposed architecture presented in this work achieves an unprecedented low inter-connection data-rate by the direct connection of RPUs forming a daisy-chain, where the CPU is at one of the ends.

In the daisy-chain, depicted in Fig. 1b, nodes are connected serially to each other by a dedicated connection. All elements in the chain work simultaneously in pipeline mode, processing and transmitting/receiving to/from the respective next/previous neighbor in the chain. The data is passed through the nodes sequentially, being updated at every RPU. There is an unique connection to the root node where the last estimate is transmitted and therefore been detected by the CPU. An important remark is the average inter-connection data-rate between nodes is the same regardless of the number of elements in the chain. This topology was proposed in [8] and further studied in [1] and [2] with specific algorithms designed for this topology.

When the decentralized architecture in Fig. 1b needs to be deployed, antennas can be collocated in the same physical place or distributed over a large area. These antennas and therefore their corresponding RPUs can behave as nodes in the chain, whilst the CPU remains as the root node. There may be multiple chains in a network. The selection of the RPUs to form a chain may depend on the users they are serving. RPUs which serve the same set of users should be in the same chain, so they can work jointly to cancel IUI. This concept fits very well with the distributed wireless communication system [16], the recent cell-free Massive MIMO concept [17] and the promising large intelligent surface [18].

Decentralized architectures, such as the one shown in Fig. 1b, require other type of algorithms compared to Fig. 1a. In the next section we introduce our proposed algorithm, which is a method for obtaining $\mathbf{w}_m$ and $\mathbf{p}_m$ as the equalization and precoding vectors, respectively.

4 Coordinate Descent

Our proposed algorithm is an iterative algorithm based on the gradient descent (GD) optimization, in which the gradient information is approximated with a set of observations in every step. From this, each antenna can obtain its own equalization/precoding vector sequentially in a coordinate descent approach. The main advantage of this method is that it does not require access to all observations at each iteration, becoming an ideal choice for large scale distributed systems.

4.1 Preliminaries

From (7) we know that in the non-IUI case, $\mathbf{E}_{\mathrm{u}}$ is a diagonal matrix, which is the case when zero-forcing (ZF) is applied. In the general case, IUI is not zero and as consequence $\mathbf{E}_{\mathrm{u}}$ contains non-zero entries outside the main diagonal.

The objective is to find a matrix $\mathbf{W}$, which cancels IUI to a high extent ($\mathbf{E}_{\mathrm{u}} \approx \mathbf{I}$), while fulfilling the following conditions:

- Uses daisy-chain as a base topology, so we exploit the advantages seen in Section 3.
- No exchange of CSI between nodes. Only local CSI.
- Limited amount of data to pass between antenna nodes. It should depend on K instead of M, to enable scalability.

- Limit the dependency on the central processing unit in order to reduce data transfer, processing and memory requirements of that unit. One consequence of this is to avoid matrix inversion in the central unit.

4.2 Algorithm formulation

The algorithm setup is that one intends to solve the unconstrained Least Squares (LS) problem in the uplink

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \tag{10}$$

via a GD approach. The gradient of (10) equals $\nabla_{\mathbf{x}} = \mathbf{H}^H\mathbf{H}\mathbf{x} - \mathbf{H}^H\mathbf{y}$. Even though $\mathbf{H}^H\mathbf{H}$ and $\mathbf{H}^H\mathbf{y}$ can be formulated in a decentralized way, the selection of $\mathbf{x}$ and the product with $\mathbf{H}^H\mathbf{H}$ is preferably done in a central processing unit to limit latency and inter-connection data-rates. Following the fully-decentralized approach and the intention to off-load the equalization/precoding computation from the CPU to the RPUs, we propose a different approach.

The proposed method can be derived as an approximate version of GD that can be operated in a decentralized architecture with minimum CPU intervention. It does so by computing, at each antenna, as much as possible of $\nabla_{\mathbf{x}}$ with the information available at the antenna. Then the estimate $\hat{\mathbf{x}}$ is updated by using a scaled version of the "local" gradient and the antenna passes the updated estimate on to the next antenna.

The above described procedure can, formally, be stated as

$$\begin{aligned} \varepsilon_m &= y_m - \mathbf{h}_m^T\hat{\mathbf{x}}_{m-1} \\ \hat{\mathbf{x}}_m &= \hat{\mathbf{x}}_{m-1} + \mu_m\mathbf{h}_m^*\varepsilon_m, \end{aligned} \tag{11}$$

for antenna m, where μ_m is a scalar step-size. The update rule in (11) corresponds to the Kaczmarz method [19], whose step-size is according to [20]

$$\mu_m = \frac{\mu}{\|\mathbf{h}_m\|^2}, \tag{12}$$

where $\mu \in \mathbb{R}$ is a relaxation parameter. In case of consistent systems, this is $\mathbf{y} = \mathbf{H}\mathbf{x}$ (if SNR is high enough or there is no noise), $\mu = 1$ is optimum and the method converge to the unique solution. Otherwise, when the system is inconsistent, μ give us an extra degree of freedom, which allows to outperform the $\mu = 1$ case, as we will see in Section 5.

After M iterations of (11) we have

$$\hat{\mathbf{x}}_M = \prod_{m=1}^{M} \left(\mathbf{I}_K - \mu_m \mathbf{h}_m^* \mathbf{h}_m^T\right) \hat{\mathbf{x}}_0 + \sum_{m=1}^{M} \prod_{i=m+1}^{M} \left(\mathbf{I}_K - \mu_i \mathbf{h}_i^* \mathbf{h}_i^T\right) \mu_m \mathbf{h}_m^* y_m.$$

If we assume $\hat{\mathbf{x}}_0 = \mathbf{0}_{K\times 1}$ [1], then it is possible to express $\hat{\mathbf{x}}_M$ as linear combination of $\mathbf{y}$, in the same way as (3), and identify $\mathbf{w}_m$ (the equalization vector associated to antenna m) as

$$\mathbf{w}_m = \left[\prod_{i=m+1}^{M} \left(\mathbf{I}_K - \mu_i \mathbf{h}_i \mathbf{h}_i^H\right)\right] \mu_m \mathbf{h}_m. \tag{13}$$

If (11) is applied in reverse antenna order ($m = M \cdots 1$), then we obtain a different estimation. The expression for $\mathbf{w}_m$ when using the alternative approach is

$$\mathbf{w}_m = \mu_m \mathbf{A}_{m-1} \mathbf{h}_m, \tag{14}$$

where matrix $\mathbf{A}_m$ is defined as

$$\mathbf{A}_m = \prod_{i=1}^{m} \left(\mathbf{I}_K - \mu_i \mathbf{h}_i \mathbf{h}_i^H\right). \tag{15}$$

It is important to remark that both approaches lead to different $\mathbf{w}_m$ sequences, however the overall performance should be the same if CSI in all antennas shows same statistical properties (stationarity across antennas).

4.3 Algorithm design and pseudocode

In this subsection we derive an equivalent and more attractive form for the calculation of the weights of the algorithm in (14) in an easy and low-complexity way, suitable for hardware implementation.

The algorithm description is shown in Algorithm 1. The vector $\mathbf{w}_m$ is computed in each antenna, while the matrix $\mathbf{A}_{m-1}$ gets updated according to the recursive rule: $\mathbf{A}_m = \mathbf{A}_{m-1} - \mathbf{w}_m \mathbf{h}_m^H$. Then, $\mathbf{w}_m$ is stored for the detection and precoding phase, and $\mathbf{A}_m$ is passed to the next antenna node for further processing.

From Algorithm 1 we can observe that after M steps we achieve the following expression: $\mathbf{A}_M = \mathbf{I}_K - \mathbf{E}_{\mathrm{u}}^*$. Then, if perfect IUI cancellation is achieved,

[1] If prior information of $\mathbf{x}$ is available, it can be used here.

Algorithm 1: Proposed algorithm

Input : $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2 \cdots \mathbf{h}_M]^T$

Preprocessing:

$\mathbf{A}_0 = \mathbf{I}_K$

for $m = 1, 2, ..., M$ **do**

  $\mathbf{w}_m = \mu_m \mathbf{A}_{m-1} \mathbf{h}_m$

  $\mathbf{A}_m = \mathbf{A}_{m-1} - \mathbf{w}_m \mathbf{h}_m^H$

end

Output : $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2 \cdots \mathbf{w}_M]^T$

$\mathbf{E}_{\mathrm{u}} = \mathbf{I}_K$ and therefore $\mathbf{A}_M = \mathbf{0}$. As a consequence we can take $\|\mathbf{A}_m\|^2$ as a metric for residual IUI. The interpretation of Algorithm 1 is as follows. $\|\mathbf{A}_m\|$ is reduced by subtracting from $\mathbf{A}_m$ a rank-1 approximation to itself. In order to achieve that, $\mathbf{A}_m$ is projected onto $\mathbf{h}_m$ to obtain $\mathbf{w}_m$, therefore $\mathbf{w}_m \mathbf{h}_m^H$ is the best rank-1 approximation to $\mathbf{A}_m$, having $\mathbf{h}_m$ as vector base. Ideally, if the channel is rich enough, vectors $\mathbf{h}_m$ are weakly correlated and assuming M is large (Massive MIMO scenario) then IUI can be canceled out to a high extent[2].

The role of step-size μ is to control how much IUI is removed at every iteration. High values will tend to reduce IUI faster at the beginning when the amount to remove is high, but will lead to oscillating or unstable residual IUI after some iterations because the steps are too big, so the introduced error dominates. Low values for μ will ensure convergence of the algorithm and a relatively good IUI cancellation at the expense of a slower convergence.

4.4 Multiple-iterations along the array

Recalling from Section 4.3, Algorithm 1 reduces the norm of $\mathbf{A}$ at each step, providing as a result $\mathbf{A}_M$, which contains the residual IUI after the algorithm is run along the array. It is possible to expand the algorithm and apply $\mathbf{A}_M$ as initial value, $\mathbf{A}_0$ for a new iteration through the array, with the intention of decreasing even more the norm of $\mathbf{A}$. The pseudocode of the expanded version is shown in Algorithm 2, with n_{iter} iterations, and as it can be seen, an increment of $\mathbf{w}_m$ is computed at each iteration. From topology point of

[2] The selection of Coordinate Descent as our method's name is because we consider the vectors $\{\mathbf{w}_i\}$ as the outcome of the method, and these can be seen as coordinates of a cost function to minimize. Such optimization problem can be written as: $\mathbf{w}_m = \arg\min_z f(\mathbf{w}_1, \cdots, \mathbf{w}_{m-1}, \mathbf{z}, \mathbf{w}_{m+1}, \cdots, \mathbf{w}_M)$, where $f = \|\mathbf{A}_{m-1} - \mathbf{z}\mathbf{h}_m^H\|_F^2$, and $\mathbf{A}_{m-1} = \mathbf{I}_K - \sum_{i \neq m} \mathbf{w}_i \mathbf{h}_i^H$. Each antenna solves this optimization problem in a sequential fashion, obtaining one coordinate as a result, while keeping the rest fixed. This is valid for single and multiple iterations to the array, which is presented in the next subsection.

view, it requires an extra connection between last and first RPUs, closing the daisy-chain and becoming a ring. It is expected to improve the performance at the expense of increasing the latency.

Algorithm 2: Proposed algorithm multiple iterations

Input : $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2 \cdots \mathbf{h}_M]^T$
Preprocessing:
$\mathbf{A}_{0,1} = \mathbf{I}_K$
$\mathbf{w}_{m,1} = \mathbf{0}, m = 1, ..., M$
for $n = 1, 2, ..., n_{iter}$ **do**
  for $m = 1, 2, ..., M$ **do**
    $\mathbf{w}_{m,n} = \mathbf{w}_{m,n-1} + \mu_m \mathbf{A}_{m-1,n} \mathbf{h}_m$
    $\mathbf{A}_{m,n} = \mathbf{A}_{m-1,n} - \mathbf{w}_{m,n} \mathbf{h}_m^H$
  end
  $\mathbf{A}_{0,n+1} = \mathbf{A}_{M,n}$
end
Output : $\mathbf{W} = [\mathbf{w}_{1,n_{iter}}, \mathbf{w}_{2,n_{iter}} \cdots \mathbf{w}_{M,n_{iter}}]^T$

5 Analysis

In this section we present an analysis of the proposed solution. The main points are:

- Performance analysis of the presented solution based on SIR, SINR and BER evaluation, and comparison with other methods.
- Complexity and timing analysis, including computational complexity, inter-connection throughput, memory requirement and latency.

As was commented in the Introduction, the analysis presented in this section is quite general and not dependent on any specific hardware implementation. The idea is to provide high level guidelines on algorithm-hardware trade-offs, system parameter selections, and hardware architectures. A more specific analysis can be performed when one has decided the dedicated implementation strategy.

5.1 Performance

In this subsection we obtain and present different metrics to evaluate and compare the performance of the proposed algorithm. The analysis we present is

divided as follows: Derivation of SIR and SINR closed form expressions, bit-error-rate (BER) analysis of the proposed algorithm based on ideal and measured channels and comparison with other methods, such as MF and ZF. The performance analysis that follows is focused on uplink, but it can be extended to downlink.

SIR & SINR

Specifically for user k, (7) is reduced to

$$\hat{\mathbf{x}}_k^{\mathrm{u}} = E_{k,k} x_k^{\mathrm{u}} + \sum_{i=1, i\neq k}^{K} E_{k,i} x_i^{\mathrm{u}} + z_k,$$

where the first term represents the desired value to estimate (scaled version), the second one is the interference from other users and the third one is due to noise. The signal-to-interference ratio (SIR) for user k is defined as

$$\mathrm{SIR}_k = \frac{\mathbb{E}|E_{k,k}|^2}{\mathbb{E}\left\{\sum_{i=1,i\neq k}^{K} |E_{k,i}|^2\right\}}. \tag{16}$$

And for the signal-to-interference-and-noise ratio (SINR) we have

$$\mathrm{SINR}_k = \frac{\mathbb{E}|E_{k,k}|^2}{\mathbb{E}\left\{\sum_{i=1,i\neq k}^{K} |E_{k,i}|^2\right\} + \mathbb{E}|z_k|^2}. \tag{17}$$

A list of parameters and their corresponding values are presented in Table 1, which are used in the following propositions.

Table 1: Parameters

Parameter	Description
α	$1 - \frac{2\mu}{K} + \frac{\mu^2}{K(K+1)}$
β	$\frac{\mu^2}{K(K+1)}$
ν	$1 - \frac{\mu}{K}$
ϵ	$1 - \frac{2\mu}{K} + \frac{\mu^2}{K}$

From (16) it is possible to obtain a closed-form expression of the SIR as follows:

Proposition 1 *With perfect* CSI *and channel model as defined in Section 2,* SIR *per user in uplink with* CD *algorithm for estimation is*

$$\text{SIR} = \frac{1 - 2\nu^M + \alpha^M \left(1 - \frac{1}{K}\right) + \epsilon^M \frac{1}{K}}{\left(1 - \frac{1}{K}\right) \cdot (\epsilon^M - \alpha^M)}, \tag{18}$$

which can be simplified in case of relatively large M, K, and $\frac{M}{K}$, which is the case of Massive MIMO, as

$$\text{SIR} \approx e^{\mu(2-\mu)\frac{M}{K}}. \tag{19}$$

Proof: See Appendix-A. ∎

The maximum value of (19) is achieved for $\mu = 1$ and the SIR value only depends on the ratio $\frac{M}{K}$ in an exponential fashion, showing how fast the IUI is canceled as M grows, and therefore ZF is approached. As an example, for a target value of SIR = 40dB, $\frac{M}{K} = 10$ meets the requirement, which is a typical ratio in Massive MIMO regime.

Regarding SINR, it can be derived based on previous results as

Proposition 2 *With perfect* CSI *and channel model as defined in Section 2,* SINR *per user in uplink with* CD *algorithm for estimation is given by*

$$\text{SINR} = \frac{1 - 2\nu^M + \alpha^M \left(1 - \frac{1}{K}\right) + \epsilon^M \frac{1}{K}}{\left(1 - \frac{1}{K}\right)(\epsilon^M - \alpha^M) + \frac{N_0}{K-1}\left(\frac{\mu}{2-\mu}\right)(1 - \epsilon^M)}, \tag{20}$$

which can be simplified in case of relatively large M, K, and $\frac{M}{K}$, which is the case of Massive MIMO, as

$$\text{SINR} \approx \left[e^{-\mu(2-\mu)\frac{M}{K}} + \frac{1}{K \cdot \text{SNR}}\left(\frac{\mu}{2-\mu}\right)\right]^{-1}. \tag{21}$$

Proof: See Appendix-B. ∎

The first term in (21) represents SIR containing IUI information, while the second one takes into account the post-equalized noise power. For high SNR, the first term is dominant and SINR $\rightarrow e^{\mu(2-\mu)\frac{M}{K}}$, which depends on $\frac{M}{K}$ and μ, but not on SNR. On the other hand, when SNR is low, the second term is dominant and SINR $\rightarrow$ SNR $\cdot K \left(\frac{2-\mu}{\mu}\right)$ as M grows, which grows linearly with SNR and K (up to certain value). This linear dependency on K is due to the post-equalization noise is equally distributed among the users. While the noise power per antenna remains constant, the portion assigned to each user decays as K grows, so the SINR per user grows linearly. However, as K increases

the IUI does so (first term in (21) grows), and both effects cancel out at some point, being IUI dominant afterwards, with the corresponding decay of SINR.

The optimal value of μ, denoted as μ^*, depends on M, K, and the specific channel. For the i.i.d. case, defined in Section 2, it is possible to obtain μ^* by numerical optimization over (20). An approximate value, denoted as μ_0, is presented as follows.

Proposition 3 *A recommended value for μ_0, in the vicinity of μ^*, under CD and i.i.d. channel as defined in Section 2, is given by*

$$\mu_0 = \frac{1}{2}\frac{K}{M}\log(4M \cdot \mathrm{SNR}). \tag{22}$$

Proof: See Appendix-C. ■

As a side result, from the analysis performed in this section, we can extract interesting properties of the matrix $\mathbf{W}$, such the following one:

Proposition 4 *The equalization matrix $\mathbf{W}$ as result of* CD *algorithm satisfies the next properties for $\mu \in [0, 2)$*

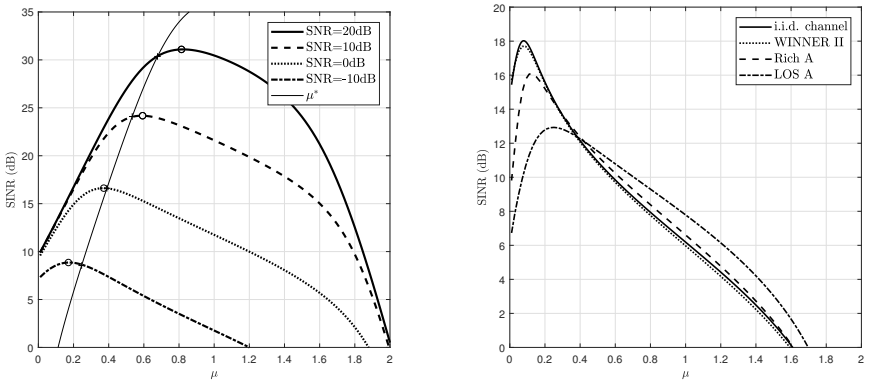
$$\mathbb{E}\|\mathbf{W}\|_F^2 = \frac{K}{K-1} \cdot \frac{\mu}{2-\mu} \cdot \left(1 - \epsilon^M\right). \tag{23}$$

Proof: See Appendix-D. ■

This result is relevant in downlink, where a transmission power budget is needed. Expression in (23) is a monotonically growing function of μ. It can be shown that total transmitted mean power is bounded by $4\frac{M}{K}$, reaching this value at $\mu = 2$. However, as we will see in next section, optimal μ for i.i.d. Gaussian channel is within the range $(0, 1]$, therefore for a large enough K, we have $\mathbb{E}\|\mathbf{W}\|_F^2 \leq 1$, which does not depend on M, therefore ensure the scalability of the proposed solution.

Expression (20) is plotted in Figure 2a showing SINR vs μ for CD under different SNR values and step-size according to (12). As expected, optimal μ approaches 1 as SNR grows. Simulation results shows a good match with (20). The curve with μ_0 values obtained from (22) is also plotted for a wide range of SNR. It is observed how the μ_0 value is reasonably close to the optimum for the SNR range depicted. Furthermore, the result is much closer to ZF than MRC values, which are $\{40.5, 30.5, 20.5, 10.5\}$dB and $\{9.0, 9.0, 8.8, 6.8\}$dB respectively for the different SNR values used in the figure.

Figure 2b shows simulation results for the CD algorithm performance under different channels. For some of them we use a model (i.i.d and WINNER II) and others are based on real measurements (Rich A and LOS A). For this



(a) (b)

Figure 2: a) SINR vs μ under different SNR. M=128 and K=16. b) SINR vs μ under different channels. M=128 and K=5. SNR=0dB.

comparison we use different $\frac{M}{K}$ ratio and the step-size according to (12). Rich A is non-line-of-sight (NLOS) channel, rich in scatters, while LOS A is predominantly line-of-sight (LOS) channel. WINNER II is obtained from a NLOS scenario with a uniform linear array at the BS, with M elements separated by $\lambda/2$. Users are randomly located in a 300m×300m area, with the BS at the center. Carrier frequency is 2GHz. It is noticed how rich channels (i.i.d and WINNER II) provide better performance. SINR levels reached by ZF are {20.9, 20.9, 19.8, 17.6}dB and for MRC they are {14.3, 15.2, 7.8, 4.8}dB, in both cases for the i.i.d., WINNER II, Rich A and LOS A channels, respectively. It is also noticed that CD performance lies in between ZF and MRC for these scenarios.

Figure 3 shows SINR versus $\frac{M}{K}$ for $M = 128$ and SNR = 0dB. SINR for CD is shown comparing the effect of using μ^* and μ_0 according to (22). We observe that $\frac{M}{K} \approx 10$ (equivalent to $K \approx 12$) is the preferred working point, where SINR reaches the maximum value and μ_0 gives the same result as μ^*. We also compare the performance with ZF and MRC algorithms.

As presented in Subsection 4.4, the algorithm can be extended to perform multiple iterations through the array, in order to increase the performance. Figure 4 shows SINR versus μ for a different number of iterations through the array together with ZF for comparison. From the figure we can notice that the maximum SINR increases after each iteration, approaching to ZF. It is also relevant to note that μ^* changes with the number of iterations.

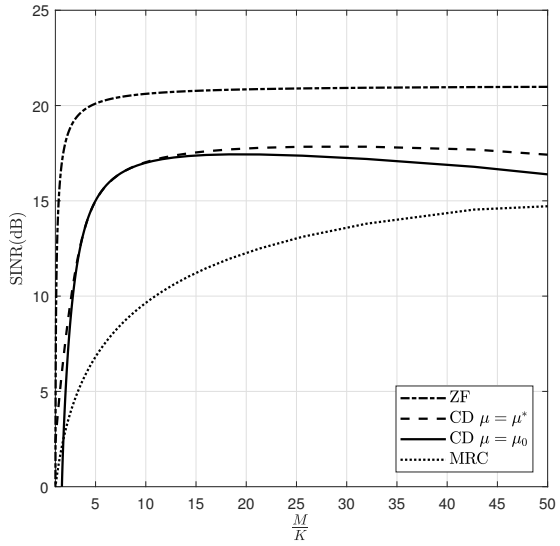


Figure 3: SINR (dB) versus $\frac{M}{K}$ for SNR=0dB and M=128. SINR for CD is plotted in the case of μ^* (dashed) and μ_0 (solid) are used. i.i.d. channel.

BER

BER versus SNR is shown in Figure 5 under i.i.d. channel for three different methods: CD, ZF and MRC. CD is shown using two different values for μ: 1 and μ^*. It is noticeable the great impact of the selected μ and therefore the importance of selecting an appropriate value.

The effect of non-ideal CSI in the BER is shown in Figure 6 for ZF and CD (for μ^*). The non-ideal CSI is modeled as an ideal-CSI with a noise contribution (complex normal distributed) with a variance equal to N_0, therefore it depends inversely on SNR. No boosting in pilots is used. As it can be observed, for SNR<0dB the SNR gap is very small and increases as long as SNR increases too, in a similar fashion as the ideal CSI case. For SNR>0 the SNR gap in both cases is similar.

5.2 Complexity & timing

In this subsection we analyze the complexity of the proposed solution from three different domains: computational complexity (data processing), inter-

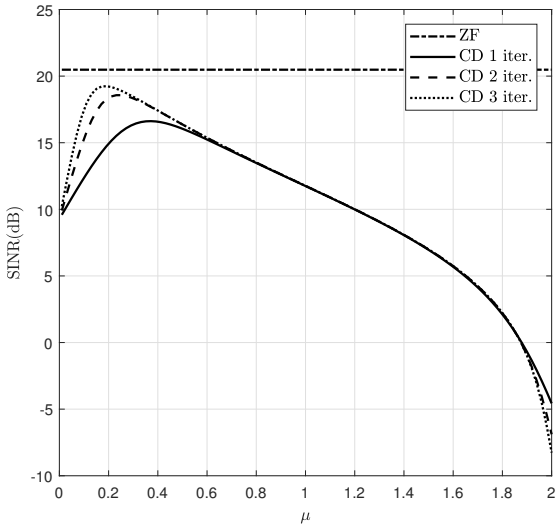


Figure 4: SINR vs. SNR for M=128, $K = 16$. 16QAM. i.i.d. channel. SNR=0dB. SINR after a certain number of iterations through the array. ZF added for comparison.

connection throughput (data movement) and memory (data storage). Timing in the form of total system latency is also analyzed.

For this analysis we assume a frame structure based on OFDM, which contains one dedicated OFDM symbol per frame for channel estimation based on orthogonal pilots, so each one is dedicated to one of the users in a consecutive way. The other symbols convey users' data. Under the TDD assumption, some of them are used for DL and others for UL. We also assume that all RPUs perform IFFT/FFT in parallel with an output data-rate of $\frac{N_{\mathrm{u}}}{T_{\mathrm{OFDM}}}$.

We can exploit channel correlation based on the Physical Resource Block (PRB) concept in 3GPP. A PRB is a region in frequency-time domain where the channel response is assumed to be approximately constant across all subcarriers within that PRB. Within an OFDM symbol, the number of subcarriers in each PRB and the number of PRB per symbol, defined as $N_{\mathrm{sc,PRB}}$ and N_{PRB} respectively, are related as follows: $N_{\mathrm{u}} = N_{\mathrm{PRB}} N_{\mathrm{sc,PRB}}$. We define T_{PRB} as the time needed by $N_{\mathrm{sc,PRB}}$ consecutive subcarriers to come out the FFT.

For each PRB we have a different channel matrix and also MIMO model as in (1) and (2). Then, it is required to have a unique set of vectors $\mathbf{w}_m$

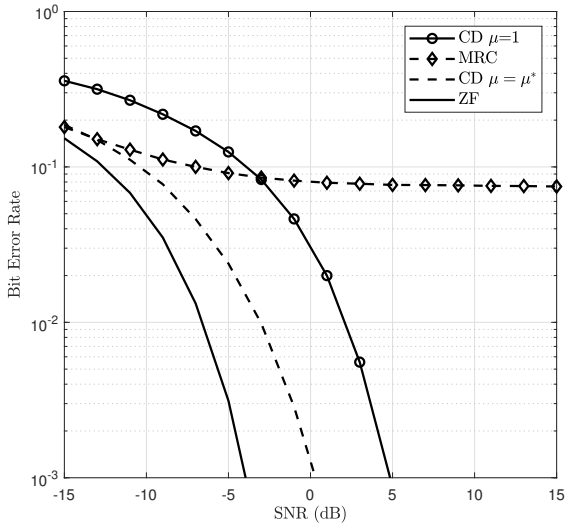


Figure 5: BER vs. SNR for M=128, $K = 16$. 16QAM. i.i.d. channel.

and $\mathbf{p}_m (m = 1...M)$ per antenna, as in (3) and (5), for uplink detection and downlink precoding respectively. The phase where these vectors are computed is named *formulation*, while the phase where user's data is processed is named *filtering* and *precoding* for UL and DL respectively. To minimize data buffering, formulation needs to be completed before filtering/precoding starts. This imposes the constraint that the formulation phase needs to be finished within one OFDM symbol, or in other words, all antennas need to obtain these vectors and the matrix $\mathbf{A}$ needs also to pass through the array within one OFDM symbol. A diagram of the main activities involved and their timing relationship is shown in Figure 7. The analysis assumes that the processing and data transmission are pipelined in each RPU so they concurrently operate.

Computational complexity

- Formulation phase: The number of complex multiplications needed to formulate one precoding/filtering vector per antenna are $C_{\text{form}} \approx 2K^2$, which represents the matrix-vector product to obtain $\mathbf{w}_m$ and the outer product to update $\mathbf{A}_m$ according to algorithm 1. Other possible required operations such as norm, square root or division are assumed to be negligible.

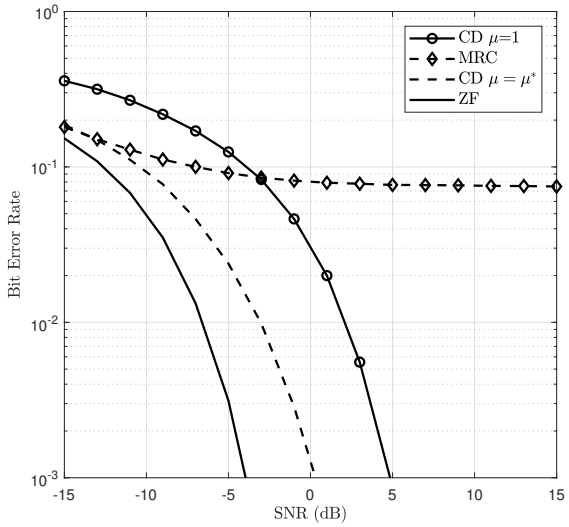


Figure 6: BER vs. SNR for M=128, $K = 16$. 16QAM. i.i.d. channel. Comparison between ideal and non-ideal CSI.

- Filtering phase: During the filtering phase, each RPU performs the required operations for UL detection. Vectors $\mathbf{w}_m$ are applied to all observations (data subcarriers), y_m^{u}, under the same PRB. The complexity measured in number of complex multiplications per antenna and per $N_{\mathrm{sc,PRB}}$ subcarriers is $C_{\mathrm{filt}} = KN_{\mathrm{sc,PRB}}$.
- Precoding phase: During the precoding phase, each RPU performs the operations required by (5). Similarly to the filtering case, the same vector $\mathbf{p}_m$ is applied to all data vectors x_m^{d} under same PRB. The complexity measured in number of complex multiplications per antenna and PRB is $C_{\mathrm{prec}} = KN_{\mathrm{sc,PRB}}$.

Inter-connection data-rate

- Formulation phase: The average inter-connection data-rate during formulation can be calculated assuming that the average time to complete a transfer of a matrix $\mathbf{A}$ is T_{PRB}, which leads to an average rate of

$$R_{\mathrm{d,form}} = \frac{2w_{\mathbf{A}}K^2N_{\mathrm{PRB}}}{T_{\mathrm{OFDM}}},$$

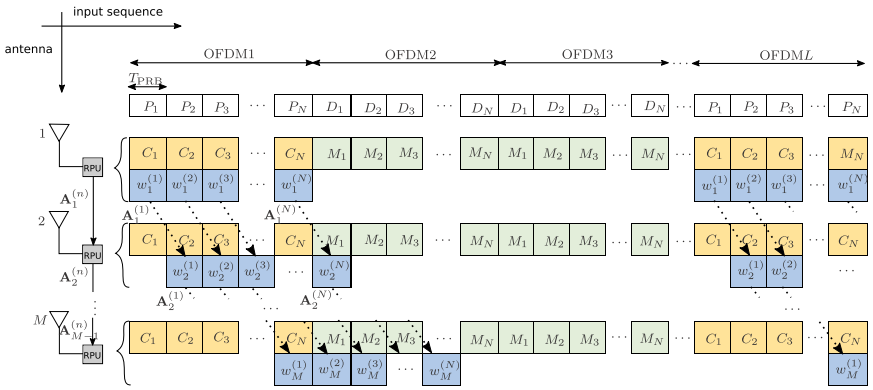


Figure 7: Time diagram representing formulation and filtering/precoding activities performed in the antenna modules. Each OFDM symbol is split into N_{PRB} blocks (N in the figure) in the same order as data come out of any of the receiver FFT. Those blocks which contains pilots are shown as P_i, while those carrying data are denoted as D_i. Channel estimation is performed during C_i blocks, while formulation is done in $\mathbf{w}_i$ blocks. Filtering/precoding data is carried out during the MIMO processing blocks, named M_i. As it can be observed, all antennas perform their tasks simultaneously, while formulation is done sequentially as a matrix $\mathbf{A}^{(n)}$ passes through the array. In total, N matrices are passed sequentially through antenna m, corresponding to $\mathbf{A}_m^{(n)}, n = 1 \cdots N$. $\mathbf{w}_i$ vectors need to be available in the antenna modules before the corresponding data comes out of the receiver FFT so it can be properly processed. Daisy-chain topology exploits the parallelism of the operations by allowing the pipeline of the operations and the fully usage of all dedicated links simultaneously.

where the numerator represents the amount of bits to transfer (all matrices $\mathbf{A}$ in a symbol) and $w_{\mathbf{A}}$ is the bit-width of $\mathbf{A}$ entries (real/imaginary parts).

- Filtering phase: Partial filtering results from each RPU are added up through the chain. The average inter-connection data-rate per dedicated link can be calculated as

$$R_{\mathrm{d,filt}} = \frac{2 w_{\mathrm{d}} K N_{\mathrm{u}}}{T_{\mathrm{OFDM}}},$$

where w_{d} is the bit-width of baseband samples exchanged among RPUs.

- Precoding phase: In the precoding phase, the data vectors $\mathbf{x}^{\mathrm{d}}$ are passed through the array for processing. Each node receives a vector which is passed to next node without any required pause (broadcasting). This leads to the same data-rate as in the filtering case.

Latency

The processing latency in the formulation phase for one antenna is given from next expression

$$\begin{aligned} T_{\mathrm{proc,form}} &= \frac{C_{\mathrm{form}} T_{\mathrm{CLK}}}{N_{\mathrm{mult}}} \\ &\approx \frac{2K^2 T_{\mathrm{CLK}}}{N_{\mathrm{mult}}}, \end{aligned}$$

where N_{mult} is the number of multipliers available in each RPU that can be used in parallel, T_{CLK} is the clock period and we assume that one complex multiplication can be done within one T_{CLK}. Total latency is expressed as

$$Lat_{form} = M \cdot T_{\mathrm{proc,form}} + (N_{\mathrm{RPU}} - 1) \cdot T_{\mathrm{trans}},$$

where N_{RPU} is the number of RPUs in the system, and T_{trans} is the transmission latency between two consecutive RPUs. As said before, formulation needs to be finished within one T_{OFDM}, therefore the formulation latency is constrained as $Lat_{form} < T_{\mathrm{OFDM}}$. This leads to an upper limit for M as

$$M < \frac{T_{\mathrm{OFDM}} + T_{\mathrm{trans}}}{T_{\mathrm{proc,form}} + \frac{T_{\mathrm{trans}}}{M_{\mathrm{RPU}}}},$$

where $M_{\mathrm{RPU}} = \frac{M}{N_{\mathrm{RPU}}}$ is the number of antennas per RPU, which is considered as a design parameter. We can consider another limit, slightly lower than previous one but easier to extract conclusions as follows

$$M < \frac{T_{\mathrm{OFDM}}}{T_{\mathrm{proc,form}} + \frac{T_{\mathrm{trans}}}{M_{\mathrm{RPU}}}}.$$

We analyze three scenarios:

- $T_{\mathrm{proc,form}} \to 0$: When processing time is reduced, by increasing N_{mult} or decreasing T_{CLK}, then transaction time becomes dominant and a reduction in the number of links allow for higher values of M. Formally, the upper value for M scales proportionally to M_{RPU} as follows

$$M < M_{\mathrm{RPU}} \cdot \frac{T_{\mathrm{OFDM}}}{T_{\mathrm{trans}}}.$$

- $T_{\text{trans}} \to 0$: By decreasing the transaction time the upper limit of M converges to a certain value, which is inversely proportional to the processing time as follows
$$M < \frac{T_{\text{OFDM}}}{T_{\text{proc,form}}}.$$
- $M_{\text{RPU}} \gg \frac{T_{\text{trans}}}{T_{\text{proc,form}}}$. When M_{RPU} increases beyond a certain value, processing time becomes dominant and we obtain the same limit as previous point.

In case of filtering, its related processing is done in parallel as soon as data comes out of the FFT. However, partial results needs to be accumulated through the array from RPU 1 to N_{RPU}. This latency is uniquely due to data transfer through the dedicated links, then

$$\begin{aligned} Lat_{\text{filt}} &= (N_{\text{RPU}} - 1) \cdot T_{\text{trans}} \\ &< Lat_{\text{form}} \\ &< T_{\text{OFDM}}. \end{aligned} \tag{24}$$

Memory

In terms of memory requirement, a centralized architecture requires to store the channel matrix $\mathbf{H}$ fully at the CPU, previous to the inversion. There is a channel matrix per PRB, so CSI storage requires $M_{\text{H}} = 2w_{\text{h}}MKN_{\text{PRB}}$ bits, where $w_{\mathbf{h}}$ represents the bit-width of $\mathbf{H}$ entries (real/imaginary parts), and in order to store the resulting square matrix, $(\mathbf{H}^H\mathbf{H})^{-1}$ requires $M_{\text{inv}} = 2w_{\text{h}}K^2N_{\text{PRB}}$ and therefore the total requirement is: $M_{\text{central}} = M_{\text{H}} + M_{\text{inv}} \approx M_{\text{H}}$.

In the decentralized architecture, each antenna module needs to store the corresponding $\mathbf{h}$, which gets replaced by $\mathbf{w}$ after formulation. Both of them requires the same amount of memory if same bit-width is assumed, which is $M_{\text{w}} = 2w_{\text{h}}KN_{\text{PRB}}$, and the total amount of memory in the system is: $M_{\text{daisy}} = M \cdot M_{\mathbf{w}} \approx M_{\text{central}}$. Therefore, the total amount of memory required for $\mathbf{H}$ and $\mathbf{W}$ is the same in both systems, however the daisy-chain allows a uniform distribution of the memory requirements across all antenna modules, reducing design complexity, time and cost. As a drawback, we point out the need for data buffering during the filtering phase due to latency in the transfer of partial results, as discussed in the previous subsection (Latency). The buffer size for the RPU closest to the CPU (worst case) can, based on (24), be obtained as

$$M_{\text{buffer}} = \frac{2w_{\text{d}}KN_{\text{u}}Lat_{\text{filt}}}{T_{\text{OFDM}}},$$

which is shared by all antennas belonging to that RPU.

Table 2: Inter-connection data-rate comparison for different system parameters $[Gb/s]$

Scenario				
M	32	64	128	256
K	4	8	12	12
$R_{\mathrm{d,form}}$	12.67	50.69	114.05	114.05
$R_{\mathrm{d,filt/prec}}$	38.02	76.03	114.05	114.05
R_{c}	304.13	608.26	1216.51	2433.02

Table 3: Computational complexity comparison for different system parameters $[GOPS]$

Scenario				
M	32	64	128	256
K	4	8	12	12
$C_{\mathrm{d,ant}}$	1.58	3.17	4.75	4.75
C_{c}	50.69	202.75	608.26	1216.51

5.3 Comparison

Table 2 shows a comparison of interconnection data-rate between daisy-chain and centralized architecture for different scenarios of M and K. It is important to remark that R_{c} corresponds to the aggregated data/rate at the shared bus, while R_{d} is the average data/rate in each of the RPU-RPU dedicated links. For the centralized case, (9) is used, while for the daisy-chain case, data-rates are detailed according to the different tasks (formulation, filtering and precoding) as described in Section 5.2. For the numerical results we employ $T_{\mathrm{CLK}} = 1\mathrm{ns}$ and $w = 12$. The rest of system parameters are as follows according to worst case in 5G NR: $N_{\mathrm{u}} = 3300$, $N_{\mathrm{PRB}} = 275$, $N_{\mathrm{sc,PRB}} = 12$ and $T_{\mathrm{OFDM}} = \frac{1}{120\mathrm{KHz}}$. We observe that for $M = 128$ case, daisy-chain requires $\sim 10\%$ of the interconnection data-rate needed by the centralized case. This number can even decrease as $\frac{M}{K}$ grows. As it is observed, daisy-chain requires much lower interconnection data-rates than the centralized counterpart. We remark that if we take into account the total inter-connection data-rate in the decentralized case, which is $N_{\mathrm{RPU}} R_{\mathrm{d,form}}$, may easily exceed the centralized counterpart R_{c}, however the decentralized architecture is able to distribute this data-rate equally across all links, reducing considerably the requirements for each of them.

Table 3 shows a computational complexity comparison between centralized

Table 4: Latency comparison for different system parameters

Scenario				
M	32	64	128	256
K	4	8	12	12
$Lat(\mu s)$	0.83	2.52	7.71	15.52
Lat/T_{OFDM}	0.10	0.30	0.92	1.86

and decentralized architectures. $C_{\mathrm{d,ant}}$ represents complex multiplications per second and per antenna in the decentralized case, while C_{c} is the computational complexity required by CPU in centralized system. In both cases, only filtering/precoding is taken into account because formulation depends on how often channel estimation is available. The result of the comparison is meaningful. Even tough, the total complexity in the decentralized system is approximately equal to the centralized counterpart, this is $M \cdot C_{\mathrm{d,ant}} \approx C_{\mathrm{c}}$, our decentralized solution is able to divide equally the total computational complexity among all existing RPUs, relaxing considerably the requirements compared to the CPU in centralized case. The relatively low number obtained for the daisy-chain allows the employment of cheap and general processing units in each RPU, in opposite to the centralized architecture where the total complexity requirement is on the CPU.

Numerical results for latency are shown in table 4 for $N_{\mathrm{mult}} = 8$, $T_{\mathrm{trans}} = 100ns$ and $N_{\mathrm{RPU}} = \frac{M}{4}$. These design parameters meets the constraint $Lat < T_{\mathrm{OFDM}}$ up to $M = 128$. For larger arrays there are different solutions: allows the latency to increase and buffer the needed input data (need for larger memory), group more antennas in each RPU (which reduces the number of links but increase the complexity of the CPU controlling each RPU), and/or employ low-latency link connections (reducing T_{trans} at the expense of higher cost). It is relevant to note that T_{OFDM} value in the table is the worst case $1/120KHz$.

In table 5 a comparison between both systems from memory perspective is shown. If $w_{\mathbf{h}} = 12$ and $N_{\mathrm{PRB}} = 275$ are assumed, then for the $M = 128$ case, each antenna module in the daisy-chain only needs $\sim$ 80kbits of memory and each RPU needs at maximum 354kbits for buffering, while in the centralized architecture, the central processor requires $\sim$ 11Mbits, which is a challenging number for a cache memory. The memory requirement grows proportionally to M in the centralized system, while that does not happen in $M_{\mathbf{w}}$. In order to reduce the buffer size we can group more antennas in each RPU, so all of them share the same buffer memory.

Table 5: Memory requirement comparison for different system parameters [$kbits$]

Scenario				
M	32	64	128	256
K	4	8	12	12
$M_{\mathbf{w}}(ant)$	26.4	52.8	79.2	79.2
$M_{\text{buffer}}(RPU)$	26.6	114.1	353.6	718.5
M_{H}	844.8	3379.2	10137.6	20275.2
M_{inv}	105.6	422.4	950.4	950.4

6 Conclusions

In this article we proposed an architecture for Massive MIMO base-station for uplink detection and downlink precoding, which is based on the fully distribution of the required baseband processing across all antenna modules in the system. The main goal is to reduce the inter-connection data-rate needed to carry out the processing tasks and enable the scalability needed in Massive MIMO. We continued our previous work in this topic [1] [2] by a detailed introduction to the CD algorithm and its application to the Massive MIMO case. We also presented an extensive analysis of the expected performance of the system, the inter-connection data-rate, complexity, latency and memory requirements. The results show that there is a performance loss compared to ZF, but unlike MF, our proposed method does not have an error floor, from which we can not recover, while the inter-connection data-rate is distributed avoiding the aggregation of the centralized approach. At the same time, complexity and memory requirements per antenna module are easy to meet with commercial off-the-self hardware, which proves the scalability of this solution.

Appendix

In the appendix we present two propositions which are going to support the proof of propositions 1 and 2 seen in Section 5. We start with some important considerations.

Let's define the random matrix $\mathbf{Q}_i$ as

$$\mathbf{Q}_i \triangleq \mathbf{I}_K - \mu_i \mathbf{h}_i \mathbf{h}_i^H, \tag{25}$$

where $\mathbf{h}_i \sim \mathcal{CN}(0, \mathbf{I})$ and independent CSI is assumed between antennas, this is $\mathbb{E}\{\mathbf{h}_i^H \mathbf{h}_j\} = \delta_{ij}, \forall i, j$. Additionally, based on (25) we can rewrite (15) as

$$\mathbf{A}_m = \mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_m, \tag{26}$$

as well as (14), which can be expressed in the following form

$$\mathbf{w}_m = \mu_m \mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_{m-1} \mathbf{h}_m. \tag{27}$$

We list in Table 6 some useful properties which are used throughout this section.

Table 6: PROPERTIES

$\mathbb{E}\left\{\frac{\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^2}\right\}$	$\frac{1}{K}\mathbf{I}$,
$\mathbb{E}\left\{\frac{\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\right\}$	$\frac{1}{K(K-1)}\mathbf{I}$
$\left[\mathbb{E}\left\{\frac{\lvert h_k\rvert^2 \mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\right\}\right]_{i,j}$	$\begin{cases} \mathbb{E}\left\{\frac{\lvert h_k\rvert^4}{\|\mathbf{h}\|^4}\right\} = \frac{2}{K(K+1)} & \text{if } k = i = j \\ \mathbb{E}\left\{\frac{\lvert h_k\rvert^2 \lvert h_i\rvert^2}{\|\mathbf{h}\|^4}\right\} = \frac{1}{K(K+1)} & \text{if } k \neq i = j \\ 0 & \text{if } i \neq j, \end{cases}$
$\mathbb{E}\{\mathbf{Q}_m\}$	$\nu\mathbf{I}, \forall m$
$\mathbb{E}\{\mathbf{A}\}$	$\nu^M\mathbf{I}$

The previous properties are based on the following proofs:

- $\mathbb{E}\left\{\frac{\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^2}\right\} = a\mathbf{I}$, where a is a complex number, due to the i.i.d. property among elements in $\mathbf{h}$. Applying the trace operator to both sides of previous equality, it follows that $a = \frac{1}{K}$, which proves the property.
- $\mathbb{E}\left\{\frac{\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\right\} = a\mathbf{I}$, by the same principle as previous property. Applying trace to both sides leads to: $\mathbb{E}\left\{\frac{1}{\|\mathbf{h}\|^2}\right\} = aK$. Let define the random variable $\mathbf{Y} = \|\mathbf{h}\|^2$, then $\mathbf{Y}$ follows a Chi-Square distribution with 2K-degrees

of freedom, this is $\mathbf{Y} \sim \chi^2(2K)$, such that: $f_{\mathbf{Y}}(y) = \frac{1}{\Gamma(K)} y^{K-1} e^{-y}$. Then follows: $\mathbb{E}\left\{\frac{1}{\mathbf{Y}}\right\} = \int_0^\infty y^{-1} f_{\mathbf{Y}}(y) dy = \frac{\Gamma(K-1)}{\Gamma(K)} = \frac{1}{K-1}$, therefore: $a = \frac{1}{K(K-1)}$, and proving the property.

- $\mathbb{E}\left\{\frac{|h_k|^2 \mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\right\}$ is also a diagonal matrix as previous properties. The values of the elements in the main diagonal can be obtained as follows: $1 = \mathbb{E}\left\{\frac{\|\mathbf{h}\|^4}{\|\mathbf{h}\|^4}\right\} = K\mathbb{E}\left\{\frac{|h_k|^4}{\|\mathbf{h}\|^4}\right\} + K(K-1)\mathbb{E}\left\{\frac{|h_k|^2|h_i|^2}{\|\mathbf{h}\|^4}\right\}$, where the next equality has been used: $\|\mathbf{h}\|^4 = \sum_{k=1}^K |h_k|^4 + \sum_{k=1}^K \sum_{i=1, i\neq k}^K |h_k|^2 |h_i|^2$. Then deriving one of the expectations leads to the other one. Let's define the random variable $\mathbf{Z} = \frac{\|\mathbf{h}\|^2}{|h_k|^2}$ as: $\mathbf{Z} = 1 + \frac{\mathbf{Y}}{\mathbf{X}}$, where $\mathbf{X} = |h_k|^2$ and $\mathbf{Y} = \sum_{i\neq k}^K |h_i|^2$. $\mathbf{X}$ follows an exponential distribution, $f_{\mathbf{X}}(x) = e^{-x}$ and $\mathbf{Y} \sim \chi^2(2K-2)$. To obtain $f_{\mathbf{Z}}(z)$, first we express
$F_Z(z) = P(\mathbf{Z} \leq z) = P(\mathbf{Y} \leq \mathbf{X}(z-1)) =$
$\int_0^\infty f_{\mathbf{X}}(x) \int_0^{x(z-1)} f_{\mathbf{Y}}(y) dy dx$. The derivative with respect to z is: $f_{\mathbf{Z}}(z) = \int_0^\infty f_{\mathbf{X}}(x) f_{\mathbf{Y}}(x(z-1)) x dx = \frac{1}{\Gamma(K-1)}(z-1)^{K-2} \int_0^\infty x^{K-1} e^{-xz} dx$ $= \frac{1}{z^2}(1-\frac{1}{z})^{K-2}(K-1)$ for $z \geq 1$ and 0 otherwise, where the definition of gamma function based on improper integral has been used. Finally, $\mathbb{E}\left\{\frac{1}{\mathbf{Z}^2}\right\} = \int_1^\infty z^{-2} f_{\mathbf{Z}}(z) dz = \frac{2}{K(K+1)}$, proving the property.

- $\mathbb{E}\{\mathbf{Q}\} = \mathbf{I} - \mu\mathbb{E}\left\{\frac{\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^2}\right\} = \mathbf{I} - \mu\frac{1}{K}$, due to first property, where $\mu_m = \frac{\mu}{\|\mathbf{h}_m\|^2}$ has been used and the index m in $\mathbf{Q}$ dropped for clarity.

- $\mathbb{E}\{\mathbf{A}\} = \mathbb{E}\left\{\prod_{m=1}^M \mathbf{Q}_m\right\} = \prod_{m=1}^M \mathbb{E}\mathbf{Q}_m$ due to statistical independence among antennas, then proving the property.

Proposition 5 *For a matrix* $\mathbf{Q}$ *defined as in equation* (25) *and* μ *as in* (12), *the next result holds for any deterministic diagonal matrix* $\mathbf{D}$

$$\mathbb{E}\left\{\mathbf{Q}\mathbf{D}\mathbf{Q}^H\right\} = \alpha\mathbf{D} + \beta\,\mathrm{Tr}(\mathbf{D})\mathbf{I}, \tag{28}$$

where α *and* β *are defined in table 1.*

Proof: Let's define a deterministic diagonal matrix as $\mathbf{D} = diag\{d_1, d_2, \cdots, d_K\}$ and a random matrix $\mathbf{Q}$ defined according to (25). Taking into account the

properties in Table 6 we can establish the following

$$\begin{aligned}
&\mathbb{E}\left\{\mathbf{Q}\mathbf{D}\mathbf{Q}^H\right\} \\
&= \mathbb{E}\left\{\mathbf{D} - \mu\mathbf{D}\frac{\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^2} - \mu\frac{\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^2}\mathbf{D} + \mu^2\frac{\mathbf{h}\mathbf{h}^H\mathbf{D}\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\right\} \\
&= \mathbf{D} - 2\frac{\mu}{K}\mathbf{D} + \mu^2\mathbb{E}\left\{\frac{\mathbf{h}\mathbf{h}^H\mathbf{D}\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\right\},
\end{aligned}$$

where

$$\begin{aligned}
\mathbb{E}\left\{\frac{\mathbf{h}\mathbf{h}^H\mathbf{D}\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\right\} &= \mathbb{E}\left\{\frac{\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\left(\sum_{k=1}^{K} d_k\mathbf{e}_k\mathbf{e}_k^T\right)\mathbf{h}\mathbf{h}^H\right\} \\
&= \sum_{k=1}^{K} d_k\mathbb{E}\left\{\frac{|\mathbf{h}_k|^2\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\right\},
\end{aligned}$$

which can be simplified as follows taking into account properties in table 6,

$$\mathbb{E}\left\{\frac{\mathbf{h}\mathbf{h}^H\mathbf{D}\mathbf{h}\mathbf{h}^H}{\|\mathbf{h}\|^4}\right\} = \frac{1}{K(K+1)}\mathbf{D} + \frac{\mathrm{Tr}(\mathbf{D})}{K(K+1)}\mathbf{I},$$

proving the proposition. ■

This proposition leads to a more general one.

Proposition 6 *For a matrix* $\mathbf{A}_m$ *defined as in equation* (26) *the next result holds for any deterministic diagonal matrix* $\mathbf{D}$

$$\mathbb{E}\left\{\mathbf{A}_m\mathbf{D}\mathbf{A}_m^H\right\} = \alpha^m\left[\mathbf{D} - \mathbf{D}_a\right] + \epsilon^m\mathbf{D}_a, \tag{29}$$

where $\mathbf{D}_a = \frac{\mathrm{Tr}(\mathbf{D})}{K}\mathbf{I}$*, and for the particular case of* $\mathbf{D} = \mathbf{I}$ *it reduces to* $\mathbb{E}\left\{\mathbf{A}_m\mathbf{A}_m^H\right\} = \epsilon^m\mathbf{I}$*, and for* $\mathbf{D} = \mathbf{e}_k^T\mathbf{e}_k$ *the following result applies*

$$\mathbf{e}_k^T\mathbb{E}\left\{\mathbf{A}_m\mathbf{e}_k\mathbf{e}_k^T\mathbf{A}_m^H\right\}\mathbf{e}_k = \alpha^m\left(1 - \frac{1}{K}\right) + \epsilon^m\frac{1}{K}.$$

Proof: Let's define a sequence of diagonal matrices $\{\mathbf{D}_m\}_{m=0,\ldots,M}$, which can be defined recursively as

$$\mathbf{D}_m = \begin{cases}\mathbb{E}\left\{\mathbf{Q}_m\mathbf{D}_{m-1}\mathbf{Q}_m^H\right\} & \text{if } m > 0 \\ \mathbf{D} & \text{if } m = 0\end{cases}$$

where $\mathbf{Q}$ is a matrix defined according to (25). From proposition 5 we know that

$$\mathbf{D}_m = \alpha\mathbf{D}_{m-1} + \beta\,\mathrm{Tr}(\mathbf{D}_{m-1})\mathbf{I},$$

and therefore $\mathrm{Tr}(\mathbf{D}_m) = \epsilon\,\mathrm{Tr}(\mathbf{D}_{m-1})$, following that $\mathrm{Tr}(\mathbf{D}_m) = \epsilon^m\,\mathrm{Tr}(\mathbf{D}_0)$, which leads to

$$\begin{aligned}\mathbf{D}_m &= \alpha\mathbf{D}_{m-1} + \mathrm{Tr}(\mathbf{D}_0)\beta\epsilon^{m-1}\mathbf{I} \\ &= \alpha^m\mathbf{D}_0 + \mathrm{Tr}(\mathbf{D}_0)\beta\epsilon^{m-1}\sum_{i=0}^{m-1} r^i\mathbf{I},\end{aligned}$$

for $m > 0$, where $r = \frac{\alpha}{\epsilon} < 1$, and finally taking into account that $\mathbf{D}_m = \mathbb{E}\left\{\mathbf{A}_m\mathbf{D}_0\mathbf{A}_m^H\right\}$ the proposition is proved. ■

A. Proof of Proposition 1

We prove the proposition by derivation of analytical expressions for $\mathbb{E}|E_{k,k}|^2$ and $\mathbb{E}\left\{\sum_{i=1,i\neq k}^{K}|E_{k,i}|^2\right\}$. From the properties shown in Table 6, $\mathbb{E}|E_{k,k}|^2$ is expressed as

$$\begin{aligned}&\mathbb{E}|E_{k,k}|^2 = \mathbb{E}|\mathbf{e}_k^T\mathbf{E}_{\mathrm{u}}\mathbf{e}_k|^2 \\ &= 1 - \mathbf{e}_k^T\mathbb{E}\{\mathbf{A}\}\mathbf{e}_k - \mathbf{e}_k^T\mathbb{E}\{\mathbf{A}^H\}\mathbf{e}_k + \mathbf{e}_k^T\mathbb{E}\{\mathbf{A}\mathbf{e}_k\mathbf{e}_k^T\mathbf{A}^H\}\mathbf{e}_k \\ &= 1 - 2\nu^M + \alpha^M\left(1 - \frac{1}{K}\right) + \epsilon^M\frac{1}{K},\end{aligned} \tag{30}$$

and for the IUI term

$$\begin{aligned}&\mathbb{E}\left\{\sum_{i=1,i\neq k}^{K}|E_{k,i}|^2\right\} = \mathbb{E}\|\mathbf{e}_k^T\mathbf{E}_{\mathrm{u}}\|^2 - \mathbb{E}|E_{k,k}|^2 \\ &= \mathbf{e}_k^T\mathbb{E}\{\mathbf{A}\mathbf{A}^H\}\mathbf{e}_k - \mathbf{e}_k^T\mathbb{E}\{\mathbf{A}\mathbf{e}_k\mathbf{e}_k^T\mathbf{A}^H\}\mathbf{e}_k \\ &= \left(1 - \frac{1}{K}\right)\cdot\left(\epsilon^M - \alpha^M\right),\end{aligned} \tag{31}$$

which proves the first part of the proposition.

In the limit when $M \to \infty$, if the ratio $\frac{M}{K}$ is kept constant, then $\epsilon^M \to e^{-\mu(2-\mu)\frac{M}{K}}$. Similarly, for α we have $\alpha^M \to e^{-2\mu\frac{M}{K}}$ under same conditions. Given that, we have that $\epsilon^M - \alpha^M \to e^{-2\mu\frac{M}{K}}\left[e^{\mu^2\frac{M}{K}} - 1\right]$. If we assume $\frac{M}{K}$ is large enough, such that $\mu^2\frac{M}{K} \gg 0$, then 1 is negligible in the second term (within brackets), and therefore $\left(1 - \frac{1}{K}\right)\left(\epsilon^M - \alpha^M\right) \to e^{-\mu(2-\mu)\frac{M}{K}}$. In the numerator, assuming $\mu\frac{M}{K} \gg 0$, then $1 - 2\nu^M + \alpha^M(1 - \frac{1}{K}) + \epsilon^M\frac{1}{K} \to 1$ when $M \to \infty$ and $\frac{M}{K}$ kept constant. Then, under previous assumptions regarding the ratio $\frac{M}{K}$, SIR $\to e^{\mu(2-\mu)\frac{M}{K}}$ when $M \to \infty$. Based on this limit, we can

establish the following approximation: SIR $\approx e^{\mu(2-\mu)\frac{M}{K}}$ = SIRa for large values of M. To give an idea of the validity of this approximation, we give some numerical values. For example, for $M = 128$, $K = 16$ and $\mu = 1$ leads to SIR(dB) $= 36.2dB$, while SIRa(dB) $= 34.7dB$, resulting in a relative error of 4%. For $M = 256$, $K = 32$ and $\mu = 1$, leads to an error of 2%, while the error goes down to 1% for $M = 512$ and $K = 64$, approaching 0 in the limit, and proving the second part of the proposition.

B. Proof of Proposition 2

The proof of the proposition is based on the corresponding proof to SIR expression (Appendix-A). The noise term, $\mathbb{E}|z_k|^2$, is the only term which has not been analyzed in the proof of Proposition 1[3]. This term can be computed as

$$\mathbb{E}|z_k|^2 = N_0 \mathbf{e}_k^T \sum_{m=1}^{M} \mathbb{E}\left\{\mathbf{w}_m \mathbf{w}_m^H\right\} \mathbf{e}_k. \tag{32}$$

Recalling that $\mathbf{w}_m = \mu \mathbf{A}_{m-1} \frac{\mathbf{h}_m}{\|\mathbf{h}_m\|^2}$ and taking into account properties in Table 6, (32) can continue as

$$\begin{aligned} \mathbb{E}|z_k|^2 &= \frac{\mu^2 N_0}{K(K-1)} \mathbf{e}_k^T \sum_{m=1}^{M} \mathbb{E}\left\{\mathbf{A}_{m-1}\mathbf{A}_{m-1}^H\right\} \mathbf{e}_k \\ &= \frac{N_0}{K-1} \cdot \left(\frac{\mu}{2-\mu}\right) \cdot \left(1-\epsilon^M\right), \end{aligned} \tag{33}$$

where Proposition 6 has been used, and shows that the post-processing noise power per user does not depend on M. This result, together with (17) and Proposition 1 leads to the final expression shown in (20), and proving the first part of the proposition.

In the limit, $\mathbb{E}|z_k|^2 \to \frac{N_0}{K-1} \cdot \frac{\mu}{2-\mu}$ when $M \to \infty$. Based on this result, we can establish an approximation, similarly to the proof of Appendix-A, consisting of: $\mathbb{E}|z_k|^2 \approx \frac{N_0}{K-1} \cdot \frac{\mu}{2-\mu}$ for large values of M. As an example of the validity of this approximation, let's consider a Massive MIMO scenario such as: $M = 128$, $K = 16$, $\mu = 0.4$, and $N_0 = 1$. This leads to a relative error of 0.13% for magnitudes in dB. This approximation, together with the one in Proposition 1, provides (21). To check the validity for this approximation, for the same scenario as before, (20) and (21) provide $16.60dB$ and $16.66dB$ respectively, which leads to a relative error of 0.36%. This completes the proof of current proposition.

[3] Thesis: In the original article it states "Proposition A", while the correct term is "Proposition 1".

C. Proof of Proposition 3

If (21) is denoted as $\widetilde{\text{SINR}}$, then maximizing this value is equivalent to minimizing the inverse value, whose derivative is

$$\frac{\partial \widetilde{\text{SINR}}^{-1}}{\partial \mu} = -2(1-\mu)\frac{M}{K}e^{-\mu(2-\mu)\frac{M}{K}} + \frac{1}{K \cdot SNR}\frac{2}{(2-\mu)^2}$$

and by setting to 0 leads to an expression which does not have closed form. However, which can be further simplified as: $4M \cdot \text{SNR} = e^{2\mu\frac{M}{K}}$, leading to (22) and proving the proposition.

D. Proof of Proposition 4

From (32) and (33) we can derive the exact expression as

$$\begin{aligned} \mathbb{E}\|\mathbf{W}\|_F^2 &= \text{Tr}\,\mathbb{E}\left\{\mathbf{W}^H\mathbf{W}\right\} \\ &= \frac{K}{K-1}\cdot\frac{\mu}{2-\mu}\cdot\left(1-\epsilon^M\right), \end{aligned} \tag{34}$$

proving the proposition.

Bibliography

[1] J. R. Sanchez, F. Rusek, M. Sarajlic, O. Edfors, and L. Liu, Fully Decentralized Massive MIMO Detection Based on Recursive Methods, in *2018 IEEE International Workshop on Signal Processing Systems (SiPS)*, Oct 2018, pp. 53-58.

[2] M. Sarajlic, F. Rusek, J. R. Sanchez, L. Liu, and O. Edfors, Fully Decentralized Approximate Zero-Forcing Precoding for Massive MIMO Systems, *IEEE Wireless Communications Letters*, pp. 1-1, 2019.

[3] T. L. Marzetta, Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas, *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590-3600, Nov 2010.

[4] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays, *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40-60, Jan 2013.

[5] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. C. Wong, F. Tufvesson, V. wall, and O. Edfors, The Worlds First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation, *IEEE Access*, vol. 5, pp. 9073-9088, 2017.

[6] J. Vieira, S. Malkowsky, K. Nieman, Z. Miers, N. Kundargi, L. Liu, I. Wong, V. wall, O. Edfors, and F. Tufvesson, A exible 100-antenna testbed for massive mimo, in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 287-293.

[7] Common public radio interface: ecpri interface specication, 2018. [Online]. Available: http://www.cpri.info/

[8] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, Argos: Practical Many-Antenna Base Stations, in *Proceedings of the 18th*

Annual International Conference on Mobile Computing and Networking, ser. Mobicom 12. New York, NY, USA: ACM, 2012, pp. 53-64. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348553

[9] E. Bertilsson, O. Gustafsson, and E. G. Larsson, A Scalable Architec- ture for Massive MIMO Base Stations Using Distributed Processing, in *2016 50th Asilomar Conference on Signals, Systems and Computers*, Nov 2016, pp. 864-868.

[10] A. Puglielli, N. Narevsky, P. Lu, T. Courtade, G. Wright, B. Nikolic, and E. Alon, A Scalable Massive MIMO Array Architecture Based on Common Modules, in *2015 IEEE International Conference on Communication Workshop (ICCW)*, June 2015, pp. 1310-1315.

[11] 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, Dec 2017.

[12] K. Li, C. Jeon, J. R. Cavallaro, and C. Studer, Feedforward Architec- tures for Decentralized Precoding in Massive MU-MIMO Systems, in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, Oct 2018, pp. 1659-1665.

[13] C. Jeon, K. Li, J. R. Cavallaro, and C. Studer, Decentralized Equal- ization with Feedforward Architectures for Massive MU-MIMO, *IEEE Transactions on Signal Processing*, vol. 67, no. 17, pp. 4418-4432, Sep. 2019.

[14] K. Li, O. Castaeda, C. Jeon, J. R. Cavallaro, and C. Studer, Decentralized Coordinate-Descent Data Detection and Precoding for Massive mumimo, in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2019, pp. 1-5.

[15] J. Vieira, F. Rusek, O. Edfors, S. Malkowsky, L. Liu, and F. Tufvesson, Reciprocity Calibration for Massive MIMO: Proposal, Modeling, and Validation, *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3042-3056, May 2017.

[16] S. Zhou, M. Zhao, X. Xu, J. Wang, and Y. Yao, Distributed Wireless Communication System: A New Architecture for Future Public Wireless Access, *IEEE Communications Magazine*, vol. 41, no. 3, pp. 108-113, March 2003.

[17] 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, March 2017.

[18] S. Hu, F. Rusek, and O. Edfors, Beyond Massive MIMO: The Potential of Data Transmission With Large Intelligent Surfaces, *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2746-2758, May 2018.

[19] S. Kaczmarz, Angenaherte auosung von systemen linearer gleichun- gen, 1937.

[20] Y. Censor, Row-action methods for huge and sparse systems and their applications, *SIAM Review*, vol. 23, no. 4, pp. 444-466, 1981. [Online]. Available: https://doi.org/10.1137/1023097

Paper IV

Processing Distribution and Architecture Tradeoff for Large Intelligent Surface Implementation

The Large Intelligent Surface (LIS) concept has emerged recently as a new paradigm for wireless communication, remote sensing and positioning. It consists of a continuous radiating surface placed relatively close to the users, which is able to communicate with users by independent transmission and reception (replacing base stations). Despite of its potential, there are a lot of challenges from an implementation point of view, with the interconnection data-rate and computational complexity being the most relevant. Distributed processing techniques and hierarchical architectures are expected to play a vital role addressing this while ensuring scalability. In this paper we perform algorithm-architecture codesign and analyze the hardware requirements and architecture trade-offs for a discrete LIS to perform uplink detection. By doing this, we expect to give concrete case studies and guidelines for efficient implementation of LIS systems.

Jesús Rodríguez Sánchez, Ove Edfors, Fredrik Rusek and Liang Liu
"Processing Distribution and Architecture Tradeoff for Large Intelligent Surface Implementation,"
in *Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1-6, 2020.

1 Introduction

The LIS concept has the potential to revolutionize wireless communication, wireless charging and remote sensing [1–4] by the use of man-made surfaces electromagnetically active. In Fig. 1 we show the concept of a LIS serving three users simultaneously. A LIS consists of a continuous radiating surface placed relatively close to the users. Each part of the surface is able to independently receive and transmit electromagnetic (EM) waves with a certain control, so the EM waves can be focused in 3D space with high resolution, creating a new world of possibilities for power-efficient communication.

Apart from LIS, other network architectures have been proposed recently for beyond-5G systems. Some of them can be classified within the smart radio environment paradigm [5], by which the wireless channel can be controlled to facilitate the transmission of information, as opposite to traditional communication systems where the channel is assumed to be imposed by nature, and transmitter and receiver adapt to changes in it. One example of this new trend is the reconfigurable surfaces, known as *intelligent reflecting surfaces* (IRS), *programmable metasurfaces*, *reconfigurable intelligent surfaces*, and *passive intelligent mirrors* among others [1], which consist of electronically passive surfaces with the capability to control how the waves are reflected when hitting their surface. Furthermore, the term LIS has also been used for such a passive surfaces [8–11], with the subsequent risk of confusion. In the common form of these surfaces there is a lack of a receiver chain, therefore not having the possibility to obtain channel state information (CSI) necessary to control the reflected waves for coherence beamforming. This means that the control must come from an external system resulting in a corresponding latency. This is in conflict with the real-time requirements of many communication systems, such as cellular communications, where channel updates are required within typically 1ms. In addition, it is known that conventional MIMO communication is more efficient than IRS-aided transmission in terms of rate [12]. These two limitations lead us to consider LIS as the preferred architecture for beyond-5G systems.

Regarding LIS, there are important challenges from an implementation point of view. It is known [1] that a continuous LIS can be replaced by a discrete one with no practical difference in achieved capacity, and therefore making LIS implementable. This discrete LIS is made up of a large number of antennas with the corresponding receiver (and transmitter) chains producing a huge amount of baseband data that needs to be routed to the Central Digital Signal Processor (CDSP) through the backplane network. As an example, a $2m \times 20m$ LIS contains $\sim 28,500$ antennas in the 4GHz band (assuming spacing

[1] We refer to [6] and [7] for a complete list of surfaces.

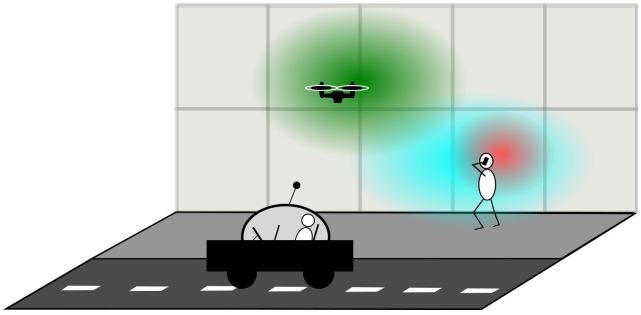


Figure 1: A LIS serving multiple users simultaneously.

of half wavelength), with the corresponding radio frequency (RF) and analog-to-digital converter (ADC) blocks. Then, if each ADC uses 8bits per I and Q, that makes a total baseband data-rate of 45.5Tbps. This is orders of magnitude higher than the massive MIMO counterpart, where this issue has been analyzed [13–16]. In order to ensure feasibility of LIS without compromising the expected benefit over Massive MIMO, in terms of spectral efficiency (mainly due to the greater number of elements and proximity to users) there are two approaches: relax the requirements (antenna density, ADC resolution, hardware quality, etc), and design proper algorithms/architecture allowing modularization and scalability. In this paper we focus on the second approach.

LIS is fundamentally different to massive MIMO due to the potential very large physical size of the surface and the amount of data to be handled, which requires specific processing, resources and performance analysis. [17] is a preliminary work addressing this issue by employing a distributed approach, where panels exchange messages with neighbors in order to build the equalizers. Multiple iterations are expected to be needed until a certain level of convergence is being achieved. The lack of a need of central processing unit (while building the equalizer) in this proposal is the key argument to ensure scalability. Together with the architecture, [17] presents the corresponding performance analysis. However, an evaluation of the required cost, from hardware point of view, is missing. For the best of our knowledge, there is not publication which performs analysis of the processing distribution, performance and the corresponding cost together for LIS.

In this paper, we propose to tackle those challenges leveraging algorithm and architecture co-design. At the algorithm level, we explore the unique features of LIS (e.g., very large aperture) to develop uplink detection algorithms that enable the processing being performed locally and distributed over the surface. This will significantly relax the requirement for interconnection bandwidth. At

the hardware architecture design level, we propose to panelize the LIS to simplify manufacturing and installation. A hierarchical interconnection topology is developed accordingly to provide efficient and flexible data exchange between panels. Based on the proposed algorithm and architecture, extensive analysis has been performed to enable trade-offs between system capacity, interconnection bandwidth, computational complexity, and processing latency. This will provide high-level design guidelines for the real implementation of LIS systems.

2 Large Intelligent Surfaces

In this article we consider a LIS for communication purpose only. Due to the large aperture of the LIS, the users are generally located in the near field. A consequence of this is that the LIS can harvest up to 50% of the transmitted user's power. This is one of the fundamental differences to the current 5G massive MIMO. One consequence of this difference, is that the transmitted power in uplink/downlink is much lower than in traditional systems, opening the door for extensive use of low-cost and low-power analog components.

Another important characteristic of LIS is that users are not seen by the entire surface as shown in Fig. 1, which can be exploited by the use of localized digital signal processing, demanding an uniform distribution of computational resources and reduced inter-connection bandwidth, without significantly sacrificing the system capacity.

2.1 System model

We consider the transmission from K single antenna users to a LIS with a total area A, containing M antenna elements. We assume the antennas are distributed evenly with a distance of half wavelength. The $M \times 1$ received vector at the LIS is given by

$$\mathbf{y} = \sqrt{\rho}\mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{x}$ is the $K \times 1$ user data vector, $\mathbf{H}$ is the $M \times K$ normalized channel matrix such that $\|\mathbf{H}\|^2 = MK$, ρ the SNR and $\mathbf{n} \sim \mathcal{CN}(0, \mathbf{I})$ is a $M \times 1$ noise vector.

Assuming the location of user k is (x_k, y_k, z_k), where the LIS is in $z = 0$. The channel between this user and a LIS antenna at location $(x, y, 0)$ is given by the complex value [1]

$$h_k(x, y) = \frac{\sqrt{z_k}}{2\sqrt{\pi} d_k^{3/2}} \exp\left(-\frac{2\pi j d_k}{\lambda}\right), \tag{2}$$

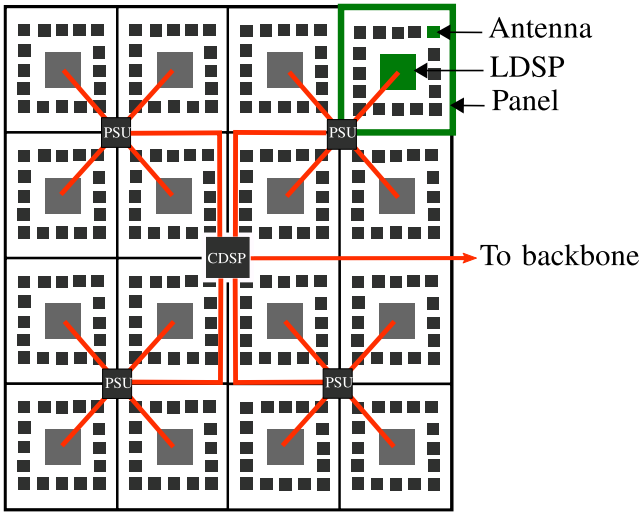


Figure 2: Overview of the LIS processing distribution and backplane interconnection. Backplane interconnection in red.

where $d_k = \sqrt{z_k^2 + (x_k - x)^2 + (y_k - y)}$ is the distance between the user and the antenna, and Line of Sight (LOS) between them is assumed. λ is the wavelength.

2.2 Panelized implementation of LIS

An overview of the processing distribution and interconnection in a LIS is shown in Fig. 2. As it can be seen, we propose that a LIS can be divided into units which are connected with backplane interconnections. We will use the term *panel* to refer to each of these units. Each panel contains a certain number of antennas (and transceiver chains). A processing unit, named Local Digital Signal Processor (LDSP) is in charge of the baseband signal processing of a panel. LDSPs are connected via backplane interconnection network to a Central DSP (CDSP), which is linked to the backbone network. In the backplane network, there are Processing Swiching Units (PSU) performing data aggregation, distribution, and processing at different levels.

Based on the general LIS implementation framework, the number of panels P, the panel area A_p, the number of antennas per panel M_p, the algorithms to be executed in LDSP and CDSP, and the backplane topology are important design parameters we would like to investigate in this paper.

3 Uplink detection algorithms

The LIS performs a linear filtering

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{y} = \sqrt{\rho}\mathbf{W}\mathbf{H}\mathbf{x} + \mathbf{W}\mathbf{n} \tag{3}$$

of the incoming signal to the panels, where $\mathbf{W}$ is the $K \times M$ equalization-filter matrix, and $\hat{\mathbf{x}}$ the estimated value of $\mathbf{x}$.

In this section we introduce two algorithms for uplink detection suitable for the panelized implementation presented in the previous section. The outcome of both is the formulation of the equalizer matrices $\{\mathbf{W}_i\}$ for panels.

3.1 Reduced Matched Filter (RMF)

The Reduced Matched Filter [18] is a reduced complexity version of the full MF, where the N_p *strongest* received users ($N_p \leq K$) by the i-th panel according to their respective CSI are used as filtering matrix, this is

$$\mathbf{W}_{\mathrm{RMF},i} = \left[\mathbf{h}_{k_1}, \mathbf{h}_{k_2}, ..., \mathbf{h}_{k_{Np}}\right]^H, \tag{4}$$

where $\mathbf{W}_{\mathrm{RMF},i}$ is the $N_\mathrm{p} \times M_\mathrm{p}$ filtering matrix of the i-th panel, and $\mathbf{h}_n$ is the $M_\mathrm{p} \times 1$ channel vector for the n-th user, $\{k_i\}$ represents the set of indexes relative to the N_p strongest users. The corresponding strength of user n is defined as $\|\mathbf{h}_n\|^2$.

3.2 Iterative Interference Cancellation (IIC)

IIC is an algorithm that allows panels to exchange information in order to cancel inter-user interference. The detailed description of the algorithm can be found in [18], and the pseudocode for the processing at the i-th panel is shown below, where $\mathbf{H}_i$ is the $M_\mathrm{p} \times K$ local CSI matrix as seen by the i-th panel, $\mathbf{Z}_{i-1}$

Algorithm 1: IIC algorithm steps for i-th panel

Input : $\mathbf{H}_i, \mathbf{Z}_{i-1}$

1 $[\mathbf{U}_z, \mathbf{\Sigma}_z] = \mathrm{svd}(\mathbf{Z}_{i-1})$

2 $\mathbf{H}_{eq} = \mathbf{H}_i \mathbf{U}_z \mathbf{\Sigma}_z^{-1/2}$

3 $\mathbf{U}_{eq} = \mathrm{svd}(\mathbf{H}_{eq})$

4 $\mathbf{W}_i^H = \mathbf{U}_{eq}(1 : N_\mathrm{p})$

5 $\mathbf{Z}_i = \mathbf{Z}_i + \mathbf{H}_i^H \mathbf{W}_i^H \mathbf{W}_i \mathbf{H}_i$

Output : $\mathbf{W}_i, \mathbf{Z}_i$

is the $K \times K$ matrix received from the $(i-1)$-th panel (neighbor), and $\mathbf{W}_i$ the local filtering matrix. $\mathbf{U}_z$ and $\mathbf{\Sigma}_z$ are the left unitary matrix and singular values of $\mathbf{Z}_{i-1}$ respectively. $\mathbf{U}_{eq}$ is the left unitary matrix of $\mathbf{H}_{eq}$, and $\mathbf{W}_i$ is made by the eigenvectors associated to the N_p strongest singular values. Each iteration of the algorithm is performed in a different panel. Matrix $\mathbf{Z}$ is passed from one panel to another by dedicated links.

Ideally we would like to find the set of filtering matrices $\{\mathbf{W}_i\}$ providing the maximum sum-rate capacity for a given channel information set $\{\mathbf{H}_i\}$. Solving this optimization problem in a distributed way is not trivial, so in the IIC approach we solve a local optimization problem in each panel and share the result with neighbor panels. Panel i will calculate $\mathbf{W}_i$ while taking the other matrices in $\{\mathbf{W}_i\}$ as given (fixed and not subject to optimization) in the form of $\mathbf{Z}_{i-1}$. This matrix $\mathbf{Z}_{i-1}$ acts as a noise covariance matrix in the local sum-rate optimization problem carried out locally.

4 Local DSP and hierarchical interconnection

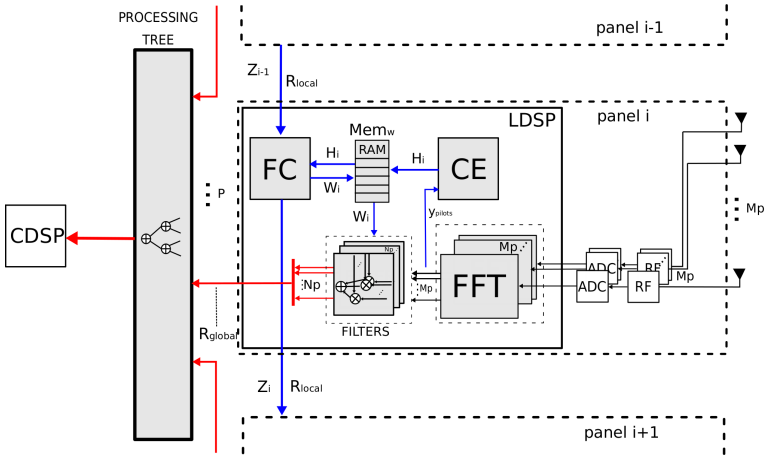
In this session, we describe the corresponding LDSP and backplane architecture that supports both the RMF and IIC algorithms. We assume the OFDM-based 5G New Radio (NR) frame structure and consider uplink detection only.

4.1 Local DSP in each panel

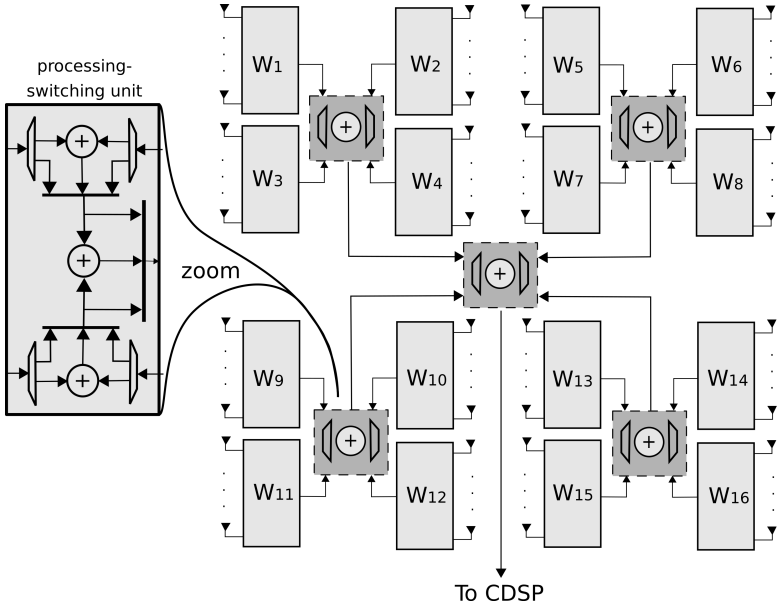
The architecture of the LDSP is depicted in Fig. 3a. After the RF and ADC, FFT blocks perform time-to-frequency domain transformation. The processing of the uplink signal is divided in two phases: formulation and filtering. During the formulation phase, the Channel Estimation block (CE) estimates a new $\mathbf{H}_i$ for each channel coherence interval. In this paper we assume perfect channel estimation. The Filter Coefficient calculation (FC) block receives $\mathbf{H}_i$ and computes the filtering matrix $\mathbf{W}_i$. FC performs complex conjugate transpose in the case of RMF and executes Algorithm 1 in the case of IIC. $\mathbf{W}_i$ is then written to the memory. During the filtering phase, the Filters block reads $\mathbf{W}_i$ and apply it to the incoming data. The Filters block reduces the $M_\mathrm{p} \times 1$ input to a $N_\mathrm{p} \times 1$ output ($N_\mathrm{p} \ll M_\mathrm{p}$), which is sent to the backplane for further processing.

4.2 Hierarchical backplane interconnection

To reduced the required interconnection bandwidth, a hierarchical backplane topology is developed to fully explore the data locality in the proposed algorithms. As shown in Fig. 3a, the backplane is divided into local direct



(a) LDSP architecture and hiarachical backplane interconnection.



(b) Tree-based global interconnection with distributed processing-switching units.

Figure 3: Overview of the local DSP unit in each panel and the backplane interconnection topology.

Table 1: System parameters

Parameter	Definition
M_{p}	number of antennas per panel
A_{p}	panel area
N_{p}	number of filtered outputs per panel
w_{filt}	bit-width of the panel output
K	number of users
f_{B}	signal bandwidth (Hz)
N_{cs}	number of coherent subcarriers

panel-to-panel link (marked in blue) and global interconnection (marked in red and will be described in detail in the next sub-section). The local link is dedicated for low-latency data exchange between two neighboring panels, e.g., the $\mathbf{Z}_{i-1}$ in the IIC algorithm. The global interconnection will aggregate the $N_{\mathrm{p}} \times 1$ filtering result from each panel to CDSP for final decision.

4.3 Tree-based global interconnection and processing

For the global interconnection, we propose to use a tree topology with distributed processing to minimize latency (the latency grows logarithmically with the number of panels), as shown in Fig. 3b. There are several levels of processing switching units (PSU) in the tree to aggregate and/or combine the panel outputs. These hierarchical PSUs can reduce the overall bandwidth requirement of the backplane and also the processing load of CDSP. Fig. 3b also shows the detailed block diagram of a PSU. It is flexible to support both RMF and IIC, and can be extended for other algorithms. Combination and bypass functionalities are used in RMF, while for IIC the streams are bypassed to the CDSP for final decision.

5 Implementation cost and simulation results

In this section, we analyze the implementation cost of the proposed uplink detection algorithms with the corresponding implementation architecture, in terms of computational complexity, interconnection bandwidth, and processing latency. The trade-offs between system capacity and implementation cost is then presented to give high-level design guidelines. For convenience, we summarize the system parameters in Table 1.

Table 2: Computational complexity in MAC/s/m^2.

Method	RMF	IIC
C_{filt}	$\frac{N_{\text{p}} M_{\text{p}} f_{\text{B}}}{A_{\text{p}}}$	$\frac{N_{\text{p}} M_{\text{p}} f_{\text{B}}}{A_{\text{p}}}$
C_{form}	$\frac{K M_{\text{p}} f_{\text{B}}}{N_{\text{cs}} A_{\text{p}}}$	$\frac{f_{\text{B}}(30K^3+bK^2+cK)}{N_{\text{cs}} A_{\text{p}}}$

5.1 Computational complexity

In Table 2, we summarize the required computational complexity for both RMF and ICC algorithms. The complexity includes both formulation phase and filtering phase and are normalized to panel area A_P. In the filtering phase, the operations are the same for RMF and ICC, which is applying a liner filter of size $N_P \times M_P$ to the $M_P \times 1$ input vector.

The formulation phase of RMF includes the computation of $\|\mathbf{h}\|^2$ for each user. For the IIC algorithm, the steps required for the formulation phase are shown in Algorithm 1. For step 1, which consists of of a singular value decomposition (SVD) of the $K \times K$ Gramian matrix $\mathbf{Z}_{i-1}$, complexity is $17K^3$ [19]. Step 2 has a complexity of $(M_{\text{p}}+1)K^2$, step 3 requires a complexity of $4M_{\text{p}}^2K+13K^3$, and step 4 and 5 need $M_{\text{p}}KN_{\text{p}} + N_{\text{p}}K^2$. In Table 2, $b = M_{\text{p}} + N_{\text{p}} + 1$ and $c = 4M_{\text{p}}^2 + M_{\text{p}}N_{\text{p}}$.

5.2 Interconnection bandwidth

The normalized (to panel area) bandwidth requirement for the global interconnection can be formulated as $R_{\text{global}} = \frac{2w_{\text{filt}} N_{\text{p}} f_{\text{B}}}{A_{\text{p}}}$ [bps/m^2]. The corresponding bandwidth requirement for the local panel-to-panel link is (only needed for the IIC algorithm) $R_{\text{local}} = \frac{2w_{\text{W}} K^2 f_{\text{B}}}{N_{\text{cs}} A_{\text{p}}}$ [bps/m^2].

5.3 Processing latency

The processing latency of the filtering phase can be formulated as $L_{filtering} = T_{\text{Filter}} + \log_4(P) T_{\text{PSU}}$, where T_{Filter} is the time needed for performing the linear filtering and T_{PSU} represents the PSU processing time as well as the PSU-to-PSU communication time.

The latency of the formulation phase differs for RMF and IIC. For RMF, the formulation phase is done in parallel in all the panels. The corresponding latency $L_{\text{form,RMF}}$ depends on the computational complexity $C_{\text{form, RMF}}$, the

clock frequency, and the available parallelism in the computation. On the other hand, the latency for IIC includes both computation and panel-to-panel communication. The worst case is $L_{\text{form,IIC}} = PT_{\text{compute,IIC}} + (P-1)T_{\text{panel-panel}}$, where $T_{\text{compute, IIC}}$ is the time for computing the filter coefficient and $T_{\text{panel-panel}}$ is the transmission latency between two consecutive panels.

5.4 Results and trade-offs

The scenario for simulation is shown in Fig. 4. Fifty users ($K = 50$) are uniformly distributed in a $40m \times 45m$ (depth x width) area in front of a $2.25m \times 22.5m$ (height x width) LIS. Signal bandwidth and carrier frequency are 100MHz and 4GHz, respectively.

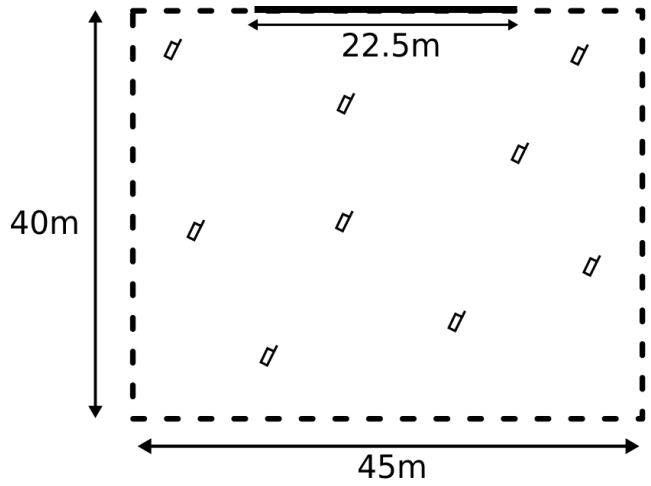
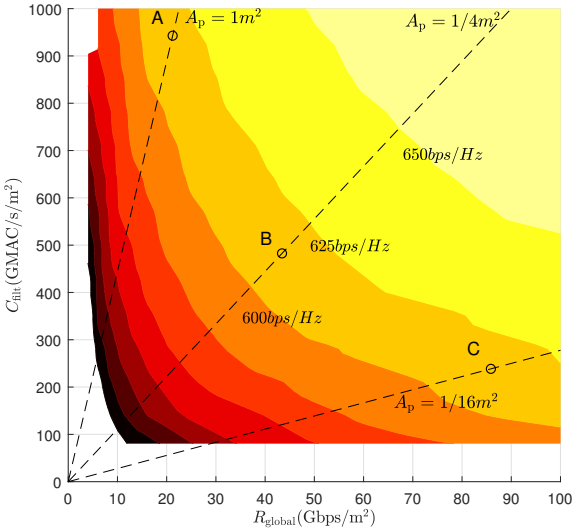
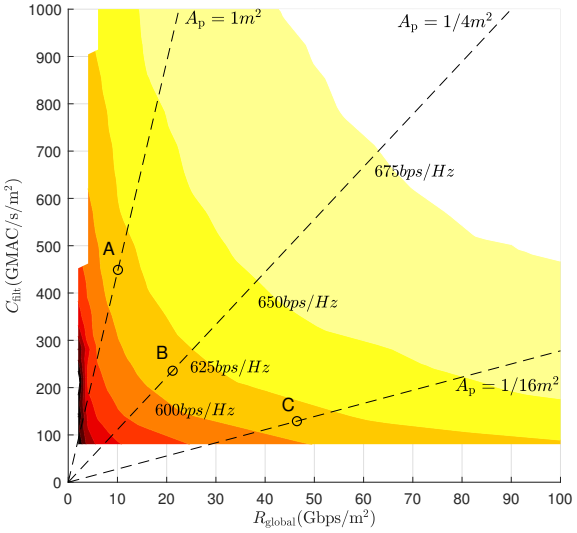


Figure 4: Top view of the simulation scenario.

The average sum-rate capacity at the interface between panels and processing tree for both algorithms is show in Fig. 5. The figures show the trade-offs between computational complexity (C_{filt} in the vertical axis) and interconnection bandwidth (R_{global} in the horizontal axis). Dashed lines represent points with constant panel size A_{p}, which is another design parameter for LIS implementation. To illustrate the trade-off, we marked points A, B, and C in the figures, presenting 3 different design choices to a targeted performance of 610bps/Hz. Comparing the same points in both figures, it can be observed the reduction in complexity and interconnection bandwidth of IIC compared to RMF. We can also observe as small panels (e.g., point C comparing to point A) demand lower computational complexity in expense of higher backplane



(a) RMF method.



(b) IIC method.

Figure 5: Sum-rate contour plot as a function of filtering complexity (C_{filt}) and inter-connection bandwidth (R_{global}). Carrier wavelength (λ) = $7.5cm$, number of users (K) = 50, SNR = $0dB$, signal bandwidth (f_{B}) = $100MHz$, ADC resolution ($\mathbf{w}_{\text{filt}}$) = $8bits$, number of coherence subcarriers (N_{cs}) = 12, and antenna spacing is $\lambda/2$.

bandwidth. Once A_{p} is fixed, the trade-off between system capacity and implementation cost (computational complexity and interconnection data-rate) can be performed depending on the application requirement.

6 Conclusions

In this article we have presented distributed processing algorithms and the corresponding hardware architecture for efficient implementation of large intelligent surfaces (LIS). The proposed processing structure consists of local panel processing units to compress incoming data without losing much information and hierarchical backplane network with distributed processing-switching units to support flexible and efficient data aggregation. We have systematically analyzed the system capacity and implementation cost with different design parameters and provided design guidelines for the implementation of LIS.

As a future direction in our research, we aim for the implementation of a LIS, as a proof-of-concept of this technology.

Acknowledgment

This work was supported by ELLIIT, the Excellence Center at Linköping-Lund in Information Technlology.

Bibliography

[1] S. Hu, F. Rusek, and O. Edfors, Beyond Massive MIMO: The Potential of Data Transmission With Large Intelligent Surfaces, *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2746-2758, May 2018.

[2] S. Hu, F. Rusek, and O. Edfors, The Potential of Using Large Antenna Arrays on Intelligent Surfaces, in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, June 2017, pp. 1-6.

[3] S. Hu, K. Chitti, F. Rusek, and O. Edfors, User Assignment with Distributed Large Intelligent Surface (LIS) Systems, in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2018, pp. 1-6.

[4] S. Hu, F. Rusek, and O. Edfors, Beyond Massive MIMO: The Potential of Positioning With Large Intelligent Surfaces, *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1761-1774, April 2018.

[5] M. D. Renzo, M. Debbah, D.-T. Phan-Huy, A. Zappone, M.-S. Alouini, C. Yuen, V. Sciancalepore, G. C. Alexandropoulos, J. Hoydis, H. Gacanin, J. d. Rosny, A. Bounceur, G. Lerosey, and M. Fink, Smart radio environments empowered by reconfigurable ai meta-surfaces: an idea whose time has come, *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 129, 2019. [Online]. Available: https://doi.org/10.1186/s13638-019-1438-9

[6] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M. Alouini, and R. Zhang, Wireless Communications Through Reconfigurable Intelligent Surfaces, *IEEE Access*, vol. 7, pp. 116 753-116 773, 2019.

[7] C. Huang, S. Hu, G. C. Alexandropoulos, A. Zappone, C. Yuen, R. Zhang, M. D. Renzo, and M. Debbah, Holographic mimo surfaces for 6G wireless networks: Opportunities, challenges, and trends, 2019.

[8] A. Taha, M. Alrabeiah, and A. Alkhateeb, Enabling Large Intelligent Surfaces with Compressive Sensing and Deep Learning, *arXiv e-prints*, p. arXiv:1904.10136, Apr 2019.

[9] Y. Han, W. Tang, S. Jin, C. Wen, and X. Ma, Large Intelligent Surface-Assisted Wireless Communication Exploiting Statistical CSI, *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8238-8242, Aug 2019.

[10] M. Jung, W. Saad, Y. Jang, G. Kong, and S. Choi, Performance Analysis of Large Intelligent Surfaces (LISs): Asymptotic Data Rate and Channel Hardening Effects, *arXiv e-prints*, p. arXiv:1810.05667, Oct 2018.

[11] C. Huang, G. C. Alexandropoulos, A. Zappone, M. Debbah, and C. Yuen, Energy Efficient Multi-User MISO Communication Using Low Resolution Large Intelligent Surfaces, in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec 2018, pp. 1-6.

[12] E. Björnson and L. Sanguinetti, Demystifying the Power Scaling Law of Intelligent Reflecting Surfaces and Metasurfaces, *arXiv e-prints*, p. arXiv:1908.03133, Aug 2019.

[13] 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, Dec 2017.

[14] A. Puglielli, A. Townley, G. LaCaille, V. Milovanovi, P. Lu, K. Trotskovsky, A. Whitcombe, N. Narevsky, G. Wright, T. Courtade, E. Alon, B. Nikoli, and A. M. Niknejad, Design of Energy- and Cost-Efficient Massive MIMO Arrays, *Proceedings of the IEEE*, vol. 104, no. 3, pp. 586-606, March 2016.

[15] J. Rodriguez Sanchez, F. Rusek, O. Edfors, M. Sarajlic, and L. Liu, Decentralized Massive MIMO Processing Exploring Daisy-chain Architecture and Recursive Algorithms, *IEEE Transactions on Signal Processing*, pp. 1-1, 2020.

[16] M. Sarajlic, F. Rusek, J. Rodrguez Snchez, L. Liu, and O. Edfors, Fully Decentralized Approximate Zero-Forcing Precoding for Massive MIMO Systems, *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 773-776, June 2019.

[17] J. V. Alegria, J. Rodriguez Sanchez, F. Rusek, L. Liu, and O. Edfors, Decentralized Equalizer Construction for Large Intelligent Surfaces, in *2019*

IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Sep. 2019, pp. 1-6.

[18] J. Rodriguez Sanchez, F. Rusek, O. Edfors, and L. Liu, An Iterative Interference Cancellation Algorithm for Large Intelligent Surfaces, *arXiv e-prints*, p. arXiv:1911.10804, Nov 2019.

[19] G. H. Golub and C. F. V. Loan, *Matrix Computations*.

[20] O. Ozdogan, E. Bjornson, and E. G. Larsson, Intelligent Reflecting Surfaces: Physics, Propagation, and Pathloss Modeling, *IEEE Wireless Communications Letters*, pp. 1-1, 2019.

Paper V

Distributed and Scalable Uplink Processing for LIS: Algorithm, Architecture, and Design Trade-offs

The Large Intelligent Surface (LIS) is a promising technology in the areas of wireless communication, remote sensing and positioning. It consists of a continuous radiating surface located in the proximity of the users, with the capability to communicate by transmission and reception (replacing base stations). Despite its potential, there are numerous challenges from an implementation point of view, with the interconnection data-rate, computational complexity, and storage the most relevant ones. In order to address these challenges, hierarchical architectures with distributed processing techniques are envisioned to be relevant for this task, while ensuring scalability. In this work we perform algorithm-architecture codesign to propose two distributed interference cancellation algorithms, and a tree-based interconnection topology for uplink processing. We also analyze the performance, hardware requirements, and architecture trade-offs for a discrete LIS, in order to provide concrete case studies and guidelines for efficient implementation of LIS systems.

"Distributed and Scalable Uplink Processing for LIS: Algorithm, Architecture, and Design Trade-offs,"
to appear in *IEEE Transactions on Signal Processing*.

1 Introduction

Large Intelligent Surface (LIS) has been identified as one of the key technologies for beyond 5G [3–6]. In Fig. 1 we show the concept of a LIS serving multiple users simultaneously. The LIS is a continuous radiating surface located in the proximity of the users. Each part of the surface is capable of receiving and transmitting electromagnetic (EM) waves with a certain control, so the EM waves can be focused in 3D space with high resolution, opening the door to a new world of possibilities for power-efficient communication.

Apart from LIS, another type of intelligent surface has been studied in the literature, which can be classified within the smart radio environment paradigm [7]. This consists on a wireless channel that can be controlled to facilitate the transmission of information, as opposed to traditional wireless communication systems, where the channel is imposed by nature, and the transmitter and receiver adapt to changes in it. One example of this new trend is the reconfigurable surfaces, known as *intelligent reflecting surfaces*, *programmable metasurfaces*, *reconfigurable intelligent surfaces (RIS)*, and *passive intelligent mirrors* among others [8–14], which consist of electronically passive surfaces with the capability to control how the waves are reflected when hitting their surface. Furthermore, the term LIS has also been recently used for such passive surfaces [15–17], which further adds to the confusion. While RIS can be seen as part of the radio channel, LIS acts as an active base station/access point. LIS contains full transmitter and receiver chains, together with baseband processing capabilities. A list of the main differences between RIS and LIS is provided in Section 2.2.

Most of the research on LIS has been focused on concept exploration [3–6], system performance [18, 19], and communication modeling [20, 21]. However, the implementation aspects have not been explored yet. This paper aims to cover this area, by identifying and addressing implementation challenges, and providing design guidelines for an efficient implementation of LIS.

The first step to make LIS implementable is to make it discrete (based on discrete antennas). It is known [3] that a continuous LIS can be replaced by a discrete one with no practical difference in achieved capacity. However, an efficient implementation of a discrete LIS is still very challenging, as it is expected to be made up of a very large number of antennas with the corresponding receiver (and transmitter) chains, which translates into a tremendous amount of interconnection data-rate, that needs to be routed to the Central Digital Signal Processor (CDSP) through the backplane network. This centralized approach has already been employed in the LuMaMi Massive multiple-input and multiple-output (MIMO) testbed [22], with a need of 100 bidireccional links, and a total aggregated interconnection bandwidth of 5GB/s. In case of

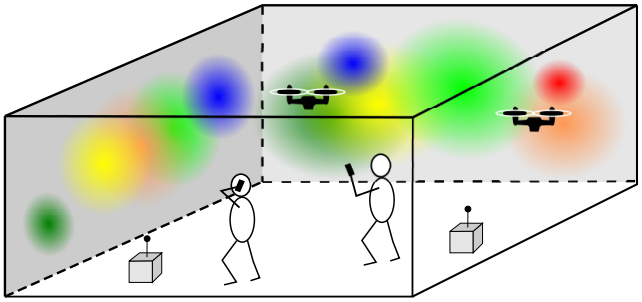


Figure 1: A LIS serving multiple users simultaneously.

LIS, this number is much higher. To illustrate, let's assume a 1.2 m $\times$ 1.2 m array containing 1,024 antennas in the 4GHz band (assuming spacing of half wavelength), with the corresponding radio frequency (RF) and analog-to-digital converter (ADC) blocks. Then, if each ADC uses 12 bits per I and Q, that amounts to a total rate of $\sim$ 48Tb/s[1]. This is three orders of magnitude higher than the massive MIMO counterpart [22], where this issue has been previously addressed [23–26]. Consequently there is a need to come up with specific architectures and algorithms in order to overcome this bottleneck.

We propose to tackle those challenges by algorithm and architecture co-design. At the algorithm level, we explore the unique features of LIS (e.g., very large aperture) to develop distributed algorithms that enable the processing being performed locally, near the antennas. This will significantly relax the requirement for interconnection bandwidth. At the hardware architecture design level, we propose to panelize the LIS in order to facilitate processing distribution, scalability, manufacturing, and installation. A hierarchical interconnection topology is developed accordingly to provide efficient and flexible data processing, and data exchange between panels and CDSP. Based on the proposed algorithm-architecture, extensive analysis has been performed to enable trade-offs between system capacity, interconnection bandwidth, computational complexity, and processing latency. This will provide high-level design guidelines for the real implementation of LIS systems. The contributions of this work are originated from our previous work in [27] and [1], being considerably extended in the present paper. The contributions of this paper compared to previous ones are summarized as follows:

- While the preliminary work presented in [1] covers baseband processing in the panels, and an analysis of the complexity and performance, in the

[1] Assuming 5G-NR standard, and sampling rate of $480,000 \cdot 4,096 \sim$ 2Gs/s.

present work we introduce a new interconnection topology that expands the dimensionality reduction capabilities. Furthermore, in the present work we also perform a more extensive analysis at system level with solid examples.

- Our work in [27] is a preliminary study addressing the processing distribution in LIS, that considers panels exchange messages with all neighbors (2D mesh structure) in an iterative fashion. This is done in a fully decentralized manner with minimal intervention of CDSP. In contrast, in present work we simplify the interconnection protocol for ease of implementation and reduced processing latency. The algorithms presented in this work are fundamentally different from the ones in [27].

This article is organized as follows: Section 2 introduces the LIS concept, then the system model is presented in Section 3. Our proposed algorithms are described in Section 4, and the architecture description in Section 5. Analysis and design trade-offs are discussed in Section 6, and finally conclusions in Section 7.

Notation: In this paper, lowercase, bold lowercase and upper bold face letters stand for scalar, column vector and matrix, respectively. The operations $(.)^T$, $(.)^*$ and $(.)^H$ denote transpose, conjugate and conjugate transpose respectively. $\mathbf{I}_K$ represents the identity matrix of size $K \times K$. Operator diag(.) returns a block diagonal matrix built with the list of matrices in the argument.

2 Large Intelligent Surfaces

This section describes the key features of LIS, by juxtaposing them with the corresponding features of massive MIMO and RIS. We also present the general concept of panelized LIS, which is proposed to ensure scalability and implementation feasibility.

2.1 Differences with Massive MIMO

Multi-antenna technology has evolved in recent years in the form of Massive MIMO, where the number of antennas in the base station (BS) grows up to ~ 100, bringing many benefits from communication and energy consumption points of view [28]. LIS goes even further by increasing the number of antennas by one or two orders of magnitude, which brings gains beyond what massive MIMO can provide. This results in fundamental differences between these two technologies, which are listed as follows:

- LIS aperture is larger in comparison to Massive MIMO, which translates into higher directivity and spatial multiplexing capabilities.
- The users are close to the LIS in relation to its size, being in the near field region, as opposed to massive MIMO (and other cellular access technologies) where users are in the far field region. Being in the near field requires the use of channel models based on spherical waveforms, rather than the planar wave approximation, whose use is generalized in massive MIMO (and other cellular technologies).
- Due to the lower path loss (owing to the close proximity between users and LIS), and the large antenna gain, transmit power is expected to be relatively small for both sides of the communication, opening the door for extensive use of low-cost and low-power analog components.
- Received power distribution from users is not uniform throughout the surface as illustrated in Fig. 1. The same user's signal is received with different signal intensity from different parts of the LIS. This can be exploited by the use of localized digital signal processing, leading to a more efficient use of computational resources, and interconnection bandwidth, without significantly sacrificing the system performance. This is in contrast with Massive MIMO (and other cellular technologies), where users are seen with same power across the antenna array.

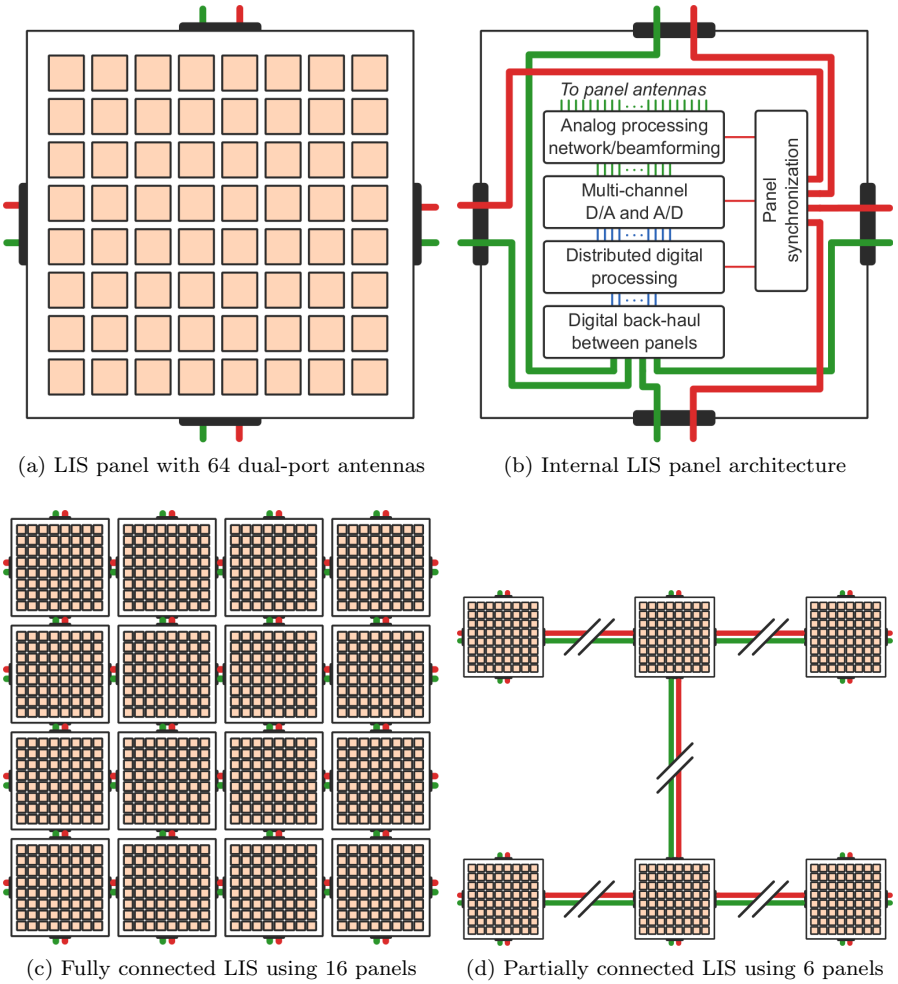
2.2 Differences with RIS

As commented in the Introduction, LIS and RIS are fundamentally different technologies. The main differences are summarized here:

- RIS acts as a programmable reflector between the radio access point and the users, forming part of the channel. Typically it is configured in a way to improve a certain quality metric, such as capacity. LIS acts as a radio access point capable of communicating directly with the users.
- LIS contains full receivers (in contrast to most of RIS) and baseband processing capabilities to obtain channel state information (CSI) from pilots transmitted by users. This allows an accurate calculation of the corresponding equalization matrix, and further detection within LIS.

2.3 Panelized implementation of LIS

Given that LIS is physically large and there is a need for distributed processing close to the antennas, we propose to divide the LIS into square units or panels.



(a) LIS panel with 64 dual-port antennas

(b) Internal LIS panel architecture

(c) Fully connected LIS using 16 panels

(d) Partially connected LIS using 6 panels

Figure 2: LIS architecture components in the form of a) panel, b) each with internal analog and digital processing resources, synchronization, and digital back-haul. Identical panels can be combined in arbitrary configurations, e.g., fully or partially connected. Each panel contributes with its own processing resources, making the available resources for distributed processing fixed per area unit.

Panelization allows the LIS to adapt to a wide range of scenarios by adding, moving, or removing panels as desired, and consequently changing the size and form of the LIS. Different shapes can be achieved by placing the panels in different formations: square, rectangular or distributed (panels not physically together, but covering a certain area). It also simplifies the system design, verification, and fabrication by only focusing on the panel as a building block, instead of covering all possible LIS sizes and forms. Additionally, the installation becomes simpler as the panel weighs less, and hence is easier to lift and mount.

A high level overview of the LIS architecture components, processing distribution, and interconnection is shown in Fig. 2. Panels are composed of a group of antennas forming a square-shaped array as depicted in Fig. 2a. Each panel contains internal processing resources in the analog and digital domains, and interconnection capabilities to connect the panel to other panels (Fig. 2b). As mentioned before, panels provide freedom to assembly the LIS. As an example, Fig. 2c shows 16 panels fully connected, forming a 1024-antenna LIS, while in Fig. 2d, 6 physically distant panels are connected in a distributed fashion (e.g: covering a certain volume in space, such as an office, or a theater).

3 System model

A conceptual view of a discrete LIS system is presented in Fig. 3. We consider K users transmitting to the LIS, which is divided in three parts: *front-end*, *backplane*, and CDSP. The term *front-end* will be used to refer to the per-antenna processing which is performed locally at each panel, and *backplane* to the related processing involving data aggregation, distribution, and processing for further dimensionality reduction. The backplane can be made of multiple levels and processing nodes as we will present in Section 5. The processing unit in the front-end is the Local DSP (LDSP), while the one in the backplane is the Backplane DSP (BDSP). The data are finally collected by the CDSP for detection. A mathematical model for the communication and the LIS-baseband processing is also derived in this section.

We consider the transmission from K single antenna users to the LIS containing M active antenna elements (input dimensionality). The LIS is divided into P square-shaped panels, each with M_p elements, such that $M_\mathrm{p} \cdot P = M$. Each panel has an output dimensionality of N_p, and the total number is N, such that $N = N_\mathrm{p} \cdot P$. Panels are connected to the backplane, which collects and process the output data, and provides the CDSP with K values to ensure proper detection. The data dimensionality is reduced from the antenna elements interface (vector $\mathbf{y} \in \mathbb{C}^M$ in the figure) to the backplane input ($\mathbf{z} \in \mathbb{C}^N$)

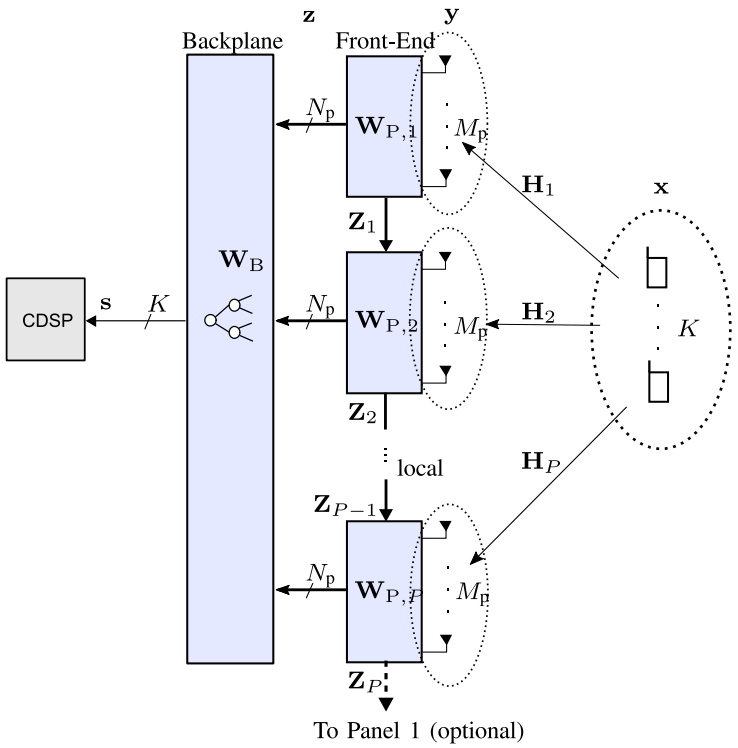


Figure 3: K users transmitting to an M-element discrete-LIS formed by P panels.

due to the front-end, and from this to the CDSP interface ($\mathbf{s} \in \mathbb{C}^K$) due to backplane processing. $M \gg K$ is assumed for the rest of the article.

The $M \times 1$ received vector at the LIS is given by

$$\mathbf{y} = \sqrt{\rho}\mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{x}$ is the transmitted $K \times 1$ user data vector, and $\mathbb{E}\{\mathbf{x}\mathbf{x}^H\} = \mathbf{I}_K$. $\mathbf{H}$ is the channel matrix, and $\mathbf{n} \sim \mathcal{CN}(0, \mathbf{I})$ is an $M \times 1$ noise vector, that it is assumed with identity covariance for simplicity without loss of generality. This convention leaves ρ as the "transmit" signal-to-noise ratio (SNR) and therefore it is dimensionless.

Assuming the location of user k is (x_k, y_k, z_k), where the LIS is at $z = 0$, the channel between this user and a LIS antenna at location $(x, y, 0)$ is given

by the complex value [3]

$$h_k(x,y) = \frac{\sqrt{z_k}}{2\sqrt{\pi}d_k^{3/2}} \exp\left(-\frac{2\pi j d_k}{\lambda}\right), \tag{2}$$

where $d_k = \sqrt{z_k^2 + (x_k - x)^2 + (y_k - y)^2}$ is the distance between the user and the antenna, λ is the wavelength, and Line of Sight (LoS) propagation between them is assumed. The channel matrix can be expressed as

$$\mathbf{H} = [\mathbf{H}_1^T, \mathbf{H}_2^T, \cdots \mathbf{H}_P^T]^T, \tag{3}$$

where $\mathbf{H}_i$ is the $M_\mathrm{p} \times K$ channel matrix of the i-th panel. Each panel is assumed to have perfect knowledge of its local channel.

3.1 Dimensionality reduction: a lossless or lossy process

As stated previously, our LIS architecture can be seen as a system to reduce the dimensionality of the very large incoming signal ($M \times 1$) down to a value required for detection at the CDSP ($K \times 1$). We can classify this process attending to the criteria of preserving information as: lossless and lossy. A lossless process maintains the mutual information between CDSP input and user's data, formally expressed as

$$I(\mathbf{s};\mathbf{x}) = I(\mathbf{y};\mathbf{x}),$$

so the system can achieve channel capacity performance if optimal processing is done at CDSP. Initial progress on the trade-offs of distributed processing for MIMO systems in the lossless approach can be seen in [29], and more recently in [30]. In this regime $N_\mathrm{p} \geq \min\{M_\mathrm{p}, K\}$.

In spite of the attractiveness of achieving optimal performance, the lossless approach imposes a high cost from an implementation point of view, as it requires larger panel output dimensionality, which translates in higher interconnection bandwidth throughout the backplane. In this article we seek to achieve a good compromise between implementation cost and performance, which leads us to explore the case $N_\mathrm{p} \leq M_\mathrm{p}$, and especially $N_\mathrm{p} \ll M_\mathrm{p}$. By selecting this regime it is expected to significantly reduce the interconnection bandwidth at the cost of a loss in performance, which can be expressed formally as

$$I(\mathbf{s};\mathbf{x}) \leq I(\mathbf{y};\mathbf{x}).$$

Our approach is to include enough flexibility into the system in order to obtain sufficient working points to establish a rich trade between implementation

cost and performance, which in fact, allows the system to adapt to a large variety of scenarios during the deployment phase. As we will see in Section 6, it is possible achieve close-to-channel-capacity conditions with significant reduction in implementation cost.

Filtering

In order to achieve dimensionality reduction, linear filtering is employed for the incoming data, and to achieve enough flexibility, separate filters for front-end and backplane are considered.

Let us consider the panelized architecture shown in Fig. 3, where each panel performs local per-antenna processing on the received signal and delivers the result to the backplane. There is no cooperation among panels during front-end filtering, and as a result the filter matrix $\mathbf{W}_{\mathrm{P}}$ has the following structure

$$\mathbf{W}_{\mathrm{P}} = \mathrm{diag}(\mathbf{W}_{\mathrm{P},1}, \mathbf{W}_{\mathrm{P},2}, \cdots, \mathbf{W}_{\mathrm{P},P}) \tag{4}$$

where $\mathbf{W}_{\mathrm{P},i}$ is the $M_{\mathrm{p}} \times N_{\mathrm{p}}$ matrix filter of the i-th panel. Then the front-end output is given by

$$\mathbf{z} = \mathbf{W}_{\mathrm{P}}^{H}\mathbf{y} = \sqrt{\rho}\mathbf{W}_{\mathrm{P}}^{H}\mathbf{H}\mathbf{x} + \hat{\mathbf{n}}, \tag{5}$$

where $\hat{\mathbf{n}} = \mathbf{W}_{\mathrm{P}}^{H}\mathbf{n}$ is the filtered noise. It should be noted that the size of $\mathbf{z}$ is N, and $N \leq M$. Finally, the backplane filters $\mathbf{z}$ in order to obtain $\mathbf{s}$ as

$$\mathbf{s} = \mathbf{W}_{\mathrm{B}}^{H}\mathbf{z}, \tag{6}$$

which is used by CDSP for detection.

3.2 Sum-rate capacity

The mutual information between $\mathbf{z}$ and $\mathbf{x}$ is $I(\mathbf{x};\mathbf{z}) = H(\mathbf{z}) - H(\mathbf{z}|\mathbf{x})$, where $H(.)$ represents entropy. Assuming white Gaussian signaling transmitted by users, the mutual information for a given $\mathbf{H}$ and $\mathbf{W}_{\mathrm{P}}$ can be further expanded as

$$\begin{aligned} I(\mathbf{x};\mathbf{z}) &= \log_2 |\mathbf{\Sigma}_{\mathbf{zz}}| - \log_2 |\mathbf{\Sigma}_{\hat{\mathbf{n}}\hat{\mathbf{n}}}| \\ &= \log_2 |\rho\mathbf{W}_{\mathrm{P}}^{H}\mathbf{H}\mathbf{H}^{H}\mathbf{W}_{\mathrm{P}} + \mathbf{W}_{\mathrm{P}}^{H}\mathbf{W}_{\mathrm{P}}| \\ &\quad - \log_2 |\mathbf{W}_{\mathrm{P}}^{H}\mathbf{W}_{\mathrm{P}}|, \end{aligned} \tag{7}$$

where $\mathbf{\Sigma}_{\mathbf{zz}}$ and $\mathbf{\Sigma}_{\hat{\mathbf{n}}\hat{\mathbf{n}}}$ are the covariance of the multivariate complex Gaussian vector $\mathbf{z}$ and $\hat{\mathbf{n}}$ respectively. If $\mathbf{W}_{\mathrm{P}}$ is a full-rank matrix, and taking into account that $M \geq N$, then $(\mathbf{W}_{\mathrm{P}}^{H}\mathbf{W}_{\mathrm{P}})^{-1}$ exists and (7) can be rewritten as

$$I(\mathbf{x};\mathbf{z}) = \log_2 |\mathbf{I}_K + \rho\mathbf{H}^{H}\mathbf{W}_{\mathrm{P}}(\mathbf{W}_{\mathrm{P}}^{H}\mathbf{W}_{\mathrm{P}})^{-1}\mathbf{W}_{\mathrm{P}}^{H}\mathbf{H}|. \tag{8}$$

We are interested in maximize the sum-rate capacity for this front-end architecture, and it will be the maximum of (8) over all possible $\mathbf{W}_{\mathrm{P}}$ for a given $\mathbf{H}$. If we take into account the block structure of $\mathbf{H}$ and $\mathbf{W}_{\mathrm{P}}$ presented in (3) and (4) respectively, the sum-rate capacity at the $\mathbf{z}$ interface is given by

$$\begin{aligned} C_{\mathbf{z}} &= \max_{\{\mathbf{W}_{\mathrm{P},i}\}} \log_2 |\mathbf{I}_K + \rho \sum_{i=1}^{P} \mathbf{H}_i^H \mathbf{W}_{\mathrm{P},i} (\mathbf{W}_{\mathrm{P},i}^H \mathbf{W}_{\mathrm{P},i})^{-1} \mathbf{W}_{\mathrm{P},i}^H \mathbf{H}_i| \\ &= \max_{\{\mathbf{Q}_i : \mathbf{Q}_i^H \mathbf{Q}_i = \mathbf{I}_{N_{\mathrm{p}}}\}} \log_2 |\mathbf{I}_K + \rho \sum_{i=1}^{P} \mathbf{H}_i^H \mathbf{Q}_i \mathbf{Q}_i^H \mathbf{H}_i|, \end{aligned} \tag{9}$$

where $\mathbf{Q}_i$ is a $M_{\mathrm{p}} \times N_{\mathrm{p}}$ semiunitary matrix, consisting of the N_{p}-first singular vectors of $\mathbf{W}_{\mathrm{P},i}$. For the last expression in (9), it is assumed that all matrices $(\mathbf{W}_{\mathrm{P},i}^H \mathbf{W}_{\mathrm{P},i})$ are full-rank, implying that the inverses exist.

As it will be shown in the next section, selection of $\{\mathbf{W}_{\mathrm{P},i}\}$ is done in a way that each element is semiunitary, which leads to white noise at the front-end output. Therefore, once the front-end filters are selected, they can be seen as part of the channel by the backplane, and we can apply the same reasoning to obtain $\mathbf{W}_{\mathrm{B}}$, leading to[2]

$$C_{\mathbf{s}} = \max_{\mathbf{W}_{\mathrm{B}}} \log_2 |\mathbf{I}_K + \rho \widetilde{\mathbf{H}}^H \mathbf{W}_{\mathrm{B}} (\mathbf{W}_{\mathrm{B}}^H \mathbf{W}_{\mathrm{B}})^{-1} \mathbf{W}_{\mathrm{B}}^H \widetilde{\mathbf{H}}|, \tag{10}$$

where $\widetilde{\mathbf{H}} = \mathbf{W}_{\mathrm{P}}^H \mathbf{H}$ is the equivalent channel.

4 Distributed algorithms for dimensionality reduction

In this section we introduce two algorithms to obtain the filtering matrices $\{\mathbf{W}_{\mathrm{P},i}\}$ and $\mathbf{W}_{\mathrm{B}}$, which are executed in the LDSP and BDSP respectively. The way the algorithms are explained here refers to the panels for simplicity, but can be extended to the backplane by using $\widetilde{\mathbf{H}}$ instead of $\mathbf{H}$, and P equal to the number of processing nodes in backplane. More details about the backplane case can be found in Section 5.

The first of the algorithms is a straightforward approach with relatively low computational complexity based on the known Maximum Ratio Combining or Matched Filter (MF) method, which we select conveniently as a comparison baseline for our proposed algorithm.

[2] A detailed explanation of this process can be found in Section 5.

4.1 Reduced Matched Filter (RMF)

RMF consists of a reduced version of the known MF method. In this case, the filter $\mathbf{W}_i$ is built by the N_p *strongest* columns of $\mathbf{H}_i$. The *strenght* of a column $\mathbf{h}_n$ is defined as $\|\mathbf{h}_n\|^2$. The $M_\mathrm{p} \times N_\mathrm{p}$ filtering matrix of the i-th panel is then expressed as

$$\mathbf{W}_{\mathrm{RMF},i} = \left[\mathbf{h}_{k_1}, \mathbf{h}_{k_2}, ..., \mathbf{h}_{k_{N_\mathrm{p}}}\right], \tag{11}$$

where $\mathbf{h}_n$ is the $M_\mathrm{p} \times 1$ channel vector for the n-th user, and $\{k_i\}$ the set of indexes relative to the N_p strongest users[3].

When RMF is applied at the panel level as local filtering, each output is associated to a certain user. Therefore, nodes in the backplane can combine data coming from the same user, in a similar fashion as in distributed MF [31]. The result of the filtering is available at CDSP input for final detection (hard or soft). It is important to note that in this method front-end processing nodes can work independently, without sharing channel related information. This saving in interconnection bandwidth comes with a performance loss as described in Section 6.

4.2 Iterative Interference Cancellation (IIC)

The IIC algorithm aims to solve the optimization problem described in (9). It is an iterative algorithm based on a variant of the known multiuser water-filling method [32]. The pseudocode is shown in Algorithm 1. The algorithm

Algorithm 1: IIC algorithm pseudocode

Input : $\{\mathbf{H}_i\}$
Preprocessing: $\mathbf{Q}_i = \mathbf{0}, i = 1 \cdots P$
repeat
  for $i = 1, 2, ..., P$ **do**
    $\mathbf{Z}_i = \mathbf{I}_K + \rho \sum_{j=1, j\neq i}^{P} \mathbf{H}_j^H \mathbf{Q}_j \mathbf{Q}_j^H \mathbf{H}_j$
    $\mathbf{Q}_i = arg\, max_{\overline{\mathbf{Q}}_i} |\rho \mathbf{H}_i^H \overline{\mathbf{Q}}_i \overline{\mathbf{Q}}_i^H \mathbf{H}_i + \mathbf{Z}_i|$
    subject to $\overline{\mathbf{Q}}_i^H \overline{\mathbf{Q}}_i = \mathbf{I}_{N_\mathrm{p}}$
  end
until *sum-rate converges*;
Output : $\{\mathbf{Q}_i\}$

[3] This is connected to the non-uniform user power distribution in the LIS, described in Section 2.1, which translates to the fact that a panel may not see all users with same power, which depends on their physical proximity.

splits the joint optimization problem (9) into P small ones, which are solved sequentially. The goal of the algorithm is to calculate the $M_\mathrm{p} \times N_\mathrm{p}$ matrices $\{\mathbf{Q}_i\}$. The product $\mathbf{Q}_i\mathbf{Q}_i^H$ is low-rank, as $N_\mathrm{p} \leq M_\mathrm{p}$, which exploits the fact that only a few users are conveniently seen by each panel (ideally this number is N_p). The fundamental difference between our current algorithm and [32] is due to the low-rank constraint present in our proposed algorithm.

At each iteration of the algorithm, the $K \times K$ matrix $\mathbf{Z}_i$ is obtained as intermediate result, which contains contribution from the rest of the panels, and plays the role of noise covariance in the sum-rate optimization problem formulated in line 4. The algorithm iterates over all panel indexes, as many times as needed until a certain convergence criterion is achieved.

4.3 Processing distribution

It is natural to map each iteration of the IIC algorithm to each panel, as it requires local CSI, while $\mathbf{Z}_i$ can be computed also locally as an update of $\mathbf{Z}_{i-1}$. Therefore, each panel computes and shares $\mathbf{Z}_i$ with the neighbor panel $i+1$, while $\mathbf{Q}_i$ is stored locally for further filter calculation, and not shared.

We propose that the panels are connected by fast local and dedicated connections for the exchange of data related to matrix $\mathbf{Z}$. In general, we can say that the matrix $\mathbf{Z}$ is passed from panel to panel using the dedicated connections depicted in Fig. 3. This decentralized approach is described in Algorithm 2 for a certain panel i[4]. The solution to the local optimization problem at the

Algorithm 2: Decentralized IIC algorithm at i-th panel

Preprocessing: $\mathbf{Z}_0 = \mathbf{I}_K$
Input : $\mathbf{H}_i, \mathbf{Z}_{i-1}$
1 $\mathbf{Q}_i = arg\,max_{\overline{\mathbf{Q}}_i} |\rho \mathbf{H}_i^H \overline{\mathbf{Q}}_i \overline{\mathbf{Q}}_i^H \mathbf{H}_i + \mathbf{Z}_{i-1}|$
2 subject to $\overline{\mathbf{Q}}_i^H \overline{\mathbf{Q}}_i = \mathbf{I}_{N_\mathrm{p}}$
3 $\mathbf{Z}_i = \mathbf{Z}_{i-1} + \rho \mathbf{H}_i^H \mathbf{Q}_i \mathbf{Q}_i^H \mathbf{H}_i$
Output : $\mathbf{Q}_i, \mathbf{Z}_i$

i-th panel is $\mathbf{Q}_i = [\widetilde{\mathbf{u}}_1, \widetilde{\mathbf{u}}_2, \cdots, \widetilde{\mathbf{u}}_{N_\mathrm{p}}]$, where $\widetilde{\mathbf{u}}_n$ is the n-th left-singular vector of $\widetilde{\mathbf{H}}_i = \mathbf{H}_i \mathbf{U}_z \mathbf{\Sigma}_z^{-1/2}$, corresponding to the n-*th* ordered singular value, and $\mathbf{Z}_{i-1} = \mathbf{U}_z \mathbf{\Sigma}_z \mathbf{U}_z^H$ the eigen-decomposition of $\mathbf{Z}_{i-1}$. See Appendix-B for proof.

The pseudocode for the processing at the i-th panel is shown in Algorithm 3, where $\widetilde{\mathbf{U}}$ is the left unitary matrix of $\widetilde{\mathbf{H}}$, and $\mathbf{Q}_i$ is made by the eigenvectors

[4] For simplicity and to limit latency, only one iteration to the set of panels is considered throughout the rest of this article. We are aware that increasing the number of iterations improves the performance.

associated to the N_p strongest singular values.

Algorithm 3: Decentralized IIC algorithm processing steps for i-th panel

Input : $\mathbf{H}_i, \mathbf{Z}_{i-1}$

$[\mathbf{U}_z, \mathbf{\Sigma}_z] = \mathrm{svd}(\mathbf{Z}_{i-1})$

$\widetilde{\mathbf{H}}_i = \mathbf{H}_i \mathbf{U}_z \mathbf{\Sigma}_z^{-1/2}$

$\widetilde{\mathbf{U}} = \mathrm{svd}(\widetilde{\mathbf{H}}_i)$

$\mathbf{Q}_i = \widetilde{\mathbf{U}}(:, 1 : N_\mathrm{p})$

$\mathbf{Z}_i = \mathbf{Z}_{i-1} + \rho \mathbf{H}_i^H \mathbf{Q}_i \mathbf{Q}_i^H \mathbf{H}_i$

Output : $\mathbf{Q}_i, \mathbf{Z}_i$

4.4 Selection of W in IIC algorithm

In the single panel case, the optimal selection of $\mathbf{Q}$ leads to $\mathbf{Q} = \widetilde{\mathbf{U}}_H^H$, where $\widetilde{\mathbf{U}}_H$ is an $M \times N$ semiunitary matrix made by the N-first left singular vectors of $\mathbf{H}$. Then, capacity will be given by the first N largest singular values of $\mathbf{H}$. Once $\mathbf{Q}$ is known, in order to select $\mathbf{W}$, we observe that $\mathbf{W} = \mathbf{Q}\widetilde{\mathbf{\Sigma}}_W \mathbf{V}_W^H$, where $\widetilde{\mathbf{\Sigma}}_W$ is a diagonal $N \times N$ matrix containing the N largest singular values of $\mathbf{W}$. Selection of $\widetilde{\mathbf{\Sigma}}_W$ and $\mathbf{V}_W$ does not play any role in the sum-rate capacity, but the right choice can provide some benefits in other areas. In this work, $\widetilde{\mathbf{\Sigma}}_W = \mathbf{I}_N$ is chosen to make $\mathbf{W}$ a semiunitary matrix, which is beneficial in terms of reduction of interconnection bandwidth, that will be explained in next section. Selection of $\mathbf{V}_W$ can be arbitrary, and for simplicity $\mathbf{V}_W = \mathbf{I}_N$ is selected. However, other unitary matrices are also valid, and could offer some advantages, but this is not covered in the present work.

In the multiple panel case, (9) represents a joint optimization problem among the matrices in the set $\{\mathbf{Q}_i\}$. Similarly to the single panel case, $\mathbf{W}_i = \mathbf{Q}_i \widetilde{\mathbf{\Sigma}}_{W,i} \mathbf{V}_{W,i}^H$. Therefore, once $\mathbf{Q}_i$ is obtained, the selection of $\widetilde{\mathbf{\Sigma}}_{W,i}$ and $\mathbf{V}_{W,i}^H$ will entail identical considerations, this is: $\widetilde{\mathbf{\Sigma}}_{W,i} = \mathbf{I}_{N_\mathrm{p}}$, and $\mathbf{V}_{W,i}^H = \mathbf{I}_{N_\mathrm{p}}$.

5 Interconnection topology and DSP architecture

In this section, the proposed LIS architecture is presented, including interconnection topology, and LDSP internal architecture able to support both the RMF and IIC algorithms.

5.1 Tree-based global interconnection and processing

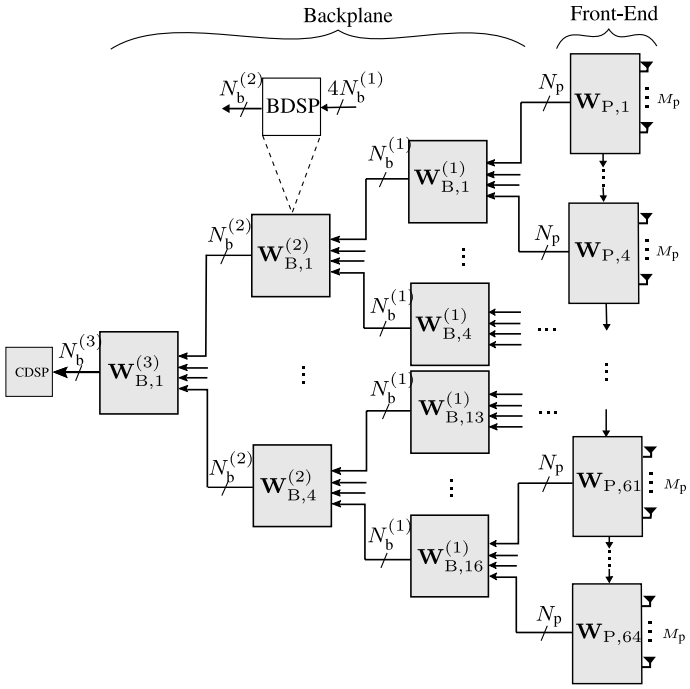


Figure 4: Front-end and backplane tree topology and interconnection for a 64-panels LIS. Each panel contains an LDSP for distributed MIMO processing. Additionally, each node in the tree contains a BDSP unit, which aggregates data from four nodes, processes, and delivers the result to the next node after corresponding dimensionality reduction, this is: $N_{\mathrm{b}}^{(i+1)} \leq 4N_{\mathrm{b}}^{(i)}, i = 1, 2$, and $N_{\mathrm{b}}^{(1)} \leq 4N_{\mathrm{p}}$.

In order to further increase the dimensionality reduction of the incoming data, while performing spatially local processing, we propose a hierarchical interconnection based on tree topology. The tree represents a distributed backplane, where front-end processing nodes are the leaves, and their outputs are combined in backplane nodes through multiple levels, reducing the total interconnection bandwidth each time, until the resulting data is delivered to the CDSP. This process is shown in Fig. 4. The main idea is to enable system scalability by adding levels in the tree as the LIS grows (more panels), while keeping the CDSP resource demands constant (dependent only on K). Another benefit

of the tree topology is its low latency, as the latency grows logarithmically with the number of panels.

As shown in the figure, the LIS backplane constitutes a 4-ary tree, which acts as an adaptation between the panels and the CDSP, introducing an extra dimensionality reduction of the incoming signal down to a level which can be efficiently transfered and handled by the CDSP, but high enough to allow good detection performance. Each node in the backplane contributes to $\mathbf{W}_{\mathrm{B}}$, and aggregates data from four nodes, processes it and delivers the output to the next node. The dimensionality of the output is lower or equal to the input, this is: $N_{\mathrm{b}}^{(i+1)} \leq 4N_{\mathrm{b}}^{(i)}, i = 1, 2$, and $N_{\mathrm{b}}^{(1)} \leq 4N_{\mathrm{p}}$. This reduction is accumulated for the different consecutive levels of nodes that the signal goes through.

Let us assume the panels, during the formulation phase and after they obtain their local filtering matrices $\{\mathbf{W}_{\mathrm{P},i}\}$ (according to the selected algorithm), deliver the products $\{\mathbf{W}_{\mathrm{P},i}^{H}\mathbf{H}_i\}$ (each with size $N_{\mathrm{p}} \times K$) to the corresponding node in the backplane. This can be seen as the result of filtering over the incoming pilot signals, which requires the same amount of data as the filtering phase. These products are the equivalent channel between the panel output and the users. A node aggregating the outputs from four panels ($4N_{\mathrm{p}}$) can see those incoming values as an equivalent channel including the wireless channel and the four panels combined. The dimensionality of this equivalent channel is $4N_{\mathrm{p}}$, which is lower compared to the $4M_p$ at the antenna level, but we expect it to carry most of the captured channel capacity. Taking into account that $\{\mathbf{W}_{\mathrm{P},i}\}$ are selected in the panels as semiunitary matrices according to Subsection 4.4, then the noise will be also white at the panel output. And filtered noise from the four adjacent panels is still white due to the independence property of noise of different antennas/panels. Therefore at any node in the backplane connected to the panels, the same model as in (1) applies with the equivalent channel instead of the wireless channel, and the filtered noise instead of the noise at antennas, but with same covariance (identity matrix[5]). Refer to Appendix-C for proof. For the tree-based backplane proposed here, the corresponding filtering matrices for first level, $\{\mathbf{W}_{\mathrm{B},i}^{(1)}\}$, can then be obtained by following the same procedures described in Section 4. This process can be repeated recursively for all levels of the tree up to the CDSP, which receives the total equivalent $K \times K$ channel matrix between the CDSP input interface and the users[6]. This is used by the CDSP for detection. The general formulation algorithm to be executed at a certain LDSP or BDSP follows the

[5] In case of not using semiunitary matrices, the noise gets colored and the covariance needs to be taken into account for sum-rate capacity optimization, therefore this noise covariance matrix needs also to be transfered between nodes in the tree. Selecting semiunitary matrices for the filters saves from this requirement. [6] We remark that this procedure is a suboptimal form (from performance point of view) of solving (10), while allowing processing distribution and lower interconnection data-rate.

steps shown in Algorithm 4, where $\mathbf{H}_{\text{eq}}$ is the equivalent channel matrix from current node input interface to users[7].

Algorithm 4: General formulation algorithm for tree-based LIS

Input : $\{\mathbf{H}_{\text{eq}}, \mathbf{Z}\}$

if *algorithm == IIC* **then**
  $\{\mathbf{W}, \mathbf{Z}\} = \text{IIC}(\mathbf{H}_{\text{eq}}, \mathbf{Z})$
else
  $\mathbf{W} = \text{RMF}(\mathbf{H}_{\text{eq}})$
end

Output: $\{\mathbf{W}^H\mathbf{H}_{\text{eq}}, \mathbf{Z}\}$

5.2 DSP in panel and backplane nodes

The internal architecture of the panel together with LDSP is depicted in Fig. 5. LDSP comprises all digital signal processing involved in the uplink tasks. After the RF and ADC, digitalized incoming signal is processed by fast Fourier transform (FFT) blocks to perform time-to-frequency domain transformation. During the formulation phase, the channel estimation block (CE) estimates a new $\mathbf{H}_i$ for each channel coherence interval[8]. The spatial processing unit (SPU), and specifically the formulation unit (FU) block, receives $\mathbf{H}_i$ and computes the filtering matrix $\mathbf{W}_{\text{P},i}$ (in the figure the subscript P is omitted for convenience). FU performs complex conjugate transpose in the case of RMF, and follows steps in Algorithm 3 in the case of IIC. $\mathbf{W}_{\text{P},i}$ is then written to the memory. During the filtering phase, incoming data vector ($\mathbf{y}_i$) gets multiplied by $\mathbf{W}_{\text{P},i}$, and its dimensionality reduced from M_{p} to a N_{p} ($N_{\text{p}} \leq M_{\text{p}}$), which is then sent to the backplane for further processing.

The SPU is shown in the figure as part of the LDSP, but it is also present in the BDSP architecture. SPU is in charge of data collection, filtering, and distribution. It also performs matrix filtering calculation and storage. In case of the BDSP architecture, SPU is its main processing element, as in this case FFT and channel estimation are not needed[9]. The filtering matrix can be either $\mathbf{W}^{(j)}_{\text{B},i}$, or $\mathbf{W}_{\text{P},i}$, depending on whether it is part of BDSP or LDSP respectively, and it supports both algorithms. The multiplexers allow to switch between the filtering and formulation phase. It is important to notice that the same input

[7] Our experimental results shows no performance improvement by sharing $\mathbf{Z}$ among backplane nodes. Due to this reason its use is skipped in Fig. 4 [8] In this paper, perfect channel estimation is assumed. [9] Even tough the SPU as a processing unit is identical at each node, data dimensionality may differ from one level to another in the system tree

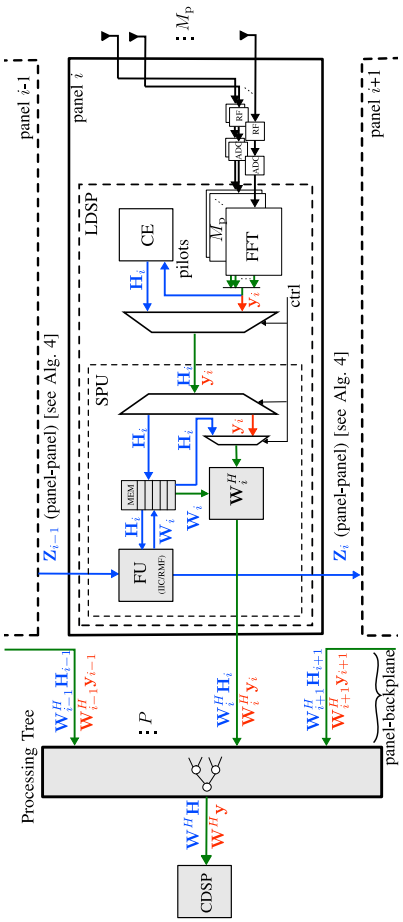


Figure 5: Overview of the Local DSP and spatial processing unit (SPU) in a panel. Panel-panel, and panel-backplane connections are also shown. Blue lines are used only in formulation phase. Blue letters relate to data which is generated/transfered during formulation. Red ones refer to filtering phase. Green lines are used in both phases. In those cases, blue and red data structures are shown above and below the line. ctrl represents control line to switch between formulation and filtering phases. $\mathbf{W}_i$ represents $\mathbf{W}_{\mathrm{P},i}$, and $\mathbf{W}$ the end-to-end filtering matrix, including panels and processing tree.

and output data ports are used during both phases. The dimensionality in both phases is the same. This design decision of using the same SPU architecture throughout the LIS is highly desirable, as it simplifies the design time, verification, and cost of the system considerably. Furthermore, by using the same unit, some or all of the backplane nodes may potentially be mapped onto the panels, therefore reducing the number of physical units in the system (at the expense of increasing the workload in panels).

6 Performance analysis and design trade-offs

In this section, we analyze the performance and implementation cost of the proposed uplink processing pipeline with the corresponding implementation architecture. More in detail:

- Performance is analyzed based on sum-rate capacity.
- Implementation cost in terms of computational complexity, interconnection bandwidth, and processing latency.

The trade-offs between sum-rate capacity and implementation cost are then presented to give high-level design guidelines.

6.1 Performance: optimality and capacity bounds

Closed-form sum-rate expression for multi-panel LIS and IIC algorithm is out of the scope of this work, however we present two upper bounds which provide useful insights. Numerical evaluation of the bounds is shown in the next subsection.

Proposition 1 *For a certain channel realization* $\mathbf{H}$*, an upper bound for* $C_{\mathbf{z}}$ *is given by*

$$C_{\mathbf{z}} \leq \min\{C_{\mathrm{ub1}}, C_{\mathrm{ub2}}\}, \tag{12}$$

where

$$C_{\mathrm{ub1}} = K \log_2 \left(1 + \rho \frac{S_{\mathrm{N}}}{K}\right), \tag{13}$$

and

$$C_{\mathrm{ub2}} = \sum_{n=1}^{K} \log_2(1 + \rho \lambda_n), \tag{14}$$

where $S_{\mathrm{N}} = \sum_{i=1}^{P} \sum_{n=1}^{N_p} \lambda_n^{(i)}$*,* $\lambda_n^{(i)}$ *is the* n*-th eigenvalue of* $\mathbf{H}_i^H \mathbf{H}_i$*, and* λ_n *is the* n*-th eigenvalue of* $\mathbf{H}^H \mathbf{H}$*.* $PN_p \geq K$ *is assumed.*

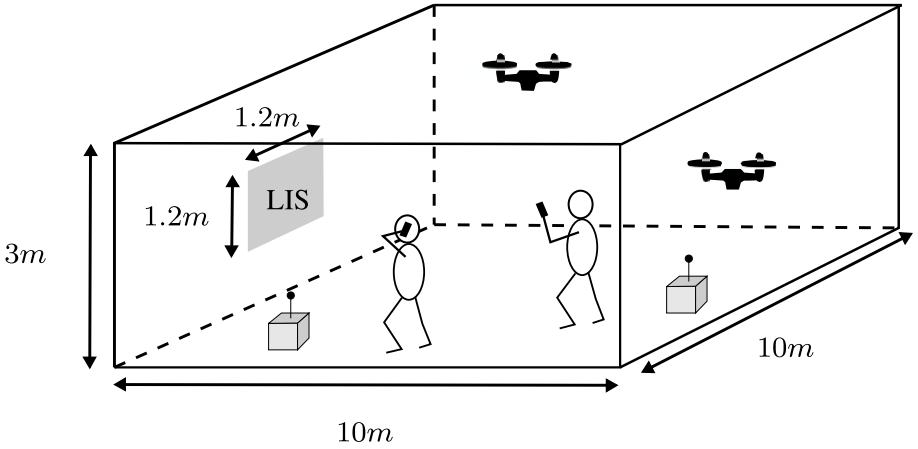


Figure 6: Simulation scenario. A $10m \times 10m \times 3m$ volume, with $1.2m \times 1.2m$ LIS. 64 users uniformly distributed in 3D.

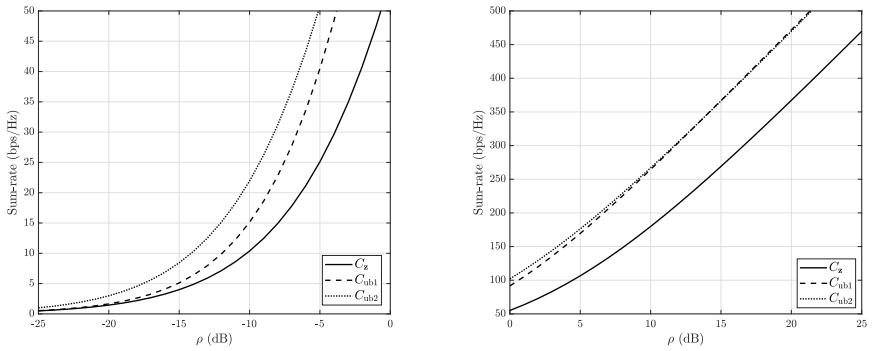
Proof: C_{ub2} corresponds to the single panel case. It acts as an upper bound, and always outperforms the multiple-panel case under the same conditions of M and K. See Appendix-A for proof of C_{ub1}. ■

6.2 Performance: experimental results and simulation

The scenario for simulation is shown in Fig. 6. It consists of 64 users ($K = 64$) uniformly distributed in a $10m \times 10m \times 3m$ (depth $\times$ width $\times$ height) volume, and a $1.2m \times 1.2m$ (height $\times$ width) LIS. Signal bandwidth and carrier frequency are 100MHz and 4GHz, respectively. We assume the orthogonal frequency-division multiplexing (OFDM)-based 5G New Radio (NR) frame structure [33] and consider uplink processing.

To obtain meaningful statistical information, 100 channel realizations are generated by placing the users within the volume following a uniform distribution in the three dimensions. For each realization, sum-rate capacity is calculated at different interfaces[10] of the system, and then averaged across all realizations. The first analysis consists of studying the relation between sum-rate and SNR, and the validity of the bounds in Proposition 1. Averaged $C_{\mathbf{z}}$ for $N_{\text{p}} = 2$ at different SNR values is shown in Fig. 7, which has been divided

[10] Interfaces include: panels output, tree nodes outputs, and CDSP input.



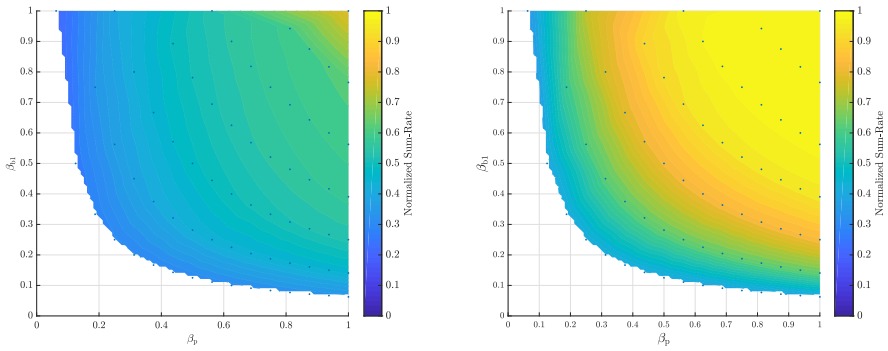
(a) Low SNR (b) High SNR

Figure 7: Average sum-rate capacity at panels output interface vs SNR. Upper bounds in Proposition 1 also shown in low and high SNR regimes. $M = 1024$, $M_{\mathrm{p}} = 16$, $N_{\mathrm{p}} = 2$, and $K = 64$.

in two SNR regions for visual clarity[11]. Selection of $N_{\mathrm{p}} = 2$ allows us to have enough output panel dimensionality [12], specifically: $N = 128 > K$. Averaged values of the bounds are also shown for comparison. C_{ub1} is tight in the low SNR region, while both bounds and $C_{\mathbf{z}}$ follow the same slope (K) at high SNR values, with $\sim 5dB$ offset in this case. C_{ub1} is better bound than C_{ub2} in this scenario.

The sum-rate capacity at CDSP input interface depends on the individual selection of the dimensionality reduction factor at each node in the system, which leads to a considerable number of possibilities. In order to simplify the analysis and show in a clear form how this individual selection affects the system performance, let us consider a tree with 3 levels (as in Fig. 4) and reduction factors as follows: $\beta_{\mathrm{b2}} = \beta_{\mathrm{b3}}$, and $\beta_{\mathrm{b3}}\beta_{\mathrm{b2}}\beta_{\mathrm{b1}}\beta_{\mathrm{p}} = \frac{K}{M}$, where $\beta_{\mathrm{b}i} \triangleq \frac{N_{\mathrm{b}}^{(i)}}{4N_{\mathrm{b}(i-1)}}$, $\beta_{\mathrm{p}} \triangleq \frac{N_{\mathrm{p}}}{M_{\mathrm{p}}}$, and $\beta_{\mathrm{b1}} \triangleq \frac{N_{\mathrm{b}}^{(1)}}{4N_{\mathrm{p}}}$. By doing so, we ensure there is dimensionality K at the CDSP input for every combination. Therefore, β represents the dimensionality reduction at a certain level of the system (all nodes in a certain

[11] The bounds are obtained for the sum-rate at the panels output interface, but are also valid for any other internal interface in the system (such as CDSP input), as sum-rate is reduced after each processing. [12] $N_{\mathrm{p}} > 2$ also meets this requirement, but at the expense of an increased interconnection bandwidth.



(a) RMF (b) IIC

Figure 8: Sum-rate capacity normalized by channel capacity at CDSP interface for different values of β_{b1} vs β_p. $\beta_{b1} = \beta_{b2} = \beta_{b3}$. $M = 1024, M_{\mathrm{p}} = 16, K = 64$, $\rho = 10$. Black dots represent simulated cases. The rest is obtained by linear interpolation.

level are assumed to have the same β for simplicity[13]), and may take values from 0 (total reduction) to 1 (no reduction). Under this constraint, β_{p} and β_{b1} can be freely chosen. Each possible combination provides a different sum-rate at CDSP interface, in exchange of different complexity cost. Fig. 8 shows the relation between these two parameters and the normalized sum-rate (value 1 refers to the channel capacity measured at antenna interface, and consequently it is the same for both algorithms) for RMF and IIC. It is important to note the multiple (β_{p}, β_{b1}) working points on the same contour level provide the same performance. We observe that a high β value (close to 1) leads to high capacity (but high interconnection bandwidth), reaching the maximum (or close to it) for points in the upper right corner in the figure, corresponding to configurations where almost no dimensionality reduction is performed in the first two levels. It is evident that IIC allows higher dimensionality reduction than RMF for same performance, which translates in lower complexity during filtering, in exchange of higher computational complexity and interconnection data-rate in formulation.

[13] We suspect a non-uniform β case can be more adequate for scenarios with non-uniform user distribution, which allows to spend resources where it is needed. This is left for further analysis.

6.3 Computational complexity

We evaluate the number of complex multiply-accumulate operations (MAC) as a metric to measure computational complexity. Our analysis includes both phases, namely formulation and filtering. In the filtering phase, the operations are the same for RMF and ICC, which consist of applying a liner filter in the panels (of size $N_\mathrm{p} \times M_\mathrm{p}$ each), to the $M_\mathrm{p} \times 1$ input vector. Similar considerations apply to the BDSP nodes (with corresponding filter sizes). The total computational complexity for filtering is given by (in MAC/s)

$$C_\mathrm{filt} = \underbrace{f_\mathrm{B} P C_\mathrm{filt}^{(0)}}_{\text{front-end}} + \underbrace{f_\mathrm{B} \sum_{n=1}^{L} N_\mathrm{SPU}^{(n)} C_\mathrm{filt}^{(n)}}_{\text{backplane}}, \tag{15}$$

where f_B is the signal bandwidth, $C_\mathrm{filt}^{(0)} = M_\mathrm{p} N_\mathrm{p}$ is the computational complexity per panel to filter one subcarrier, $C_\mathrm{filt}^{(n)} = 4 N_\mathrm{b}^{(n-1)} N_\mathrm{b}^{(n)}$ is the corresponding computational complexity in a node at level n, L is the number of levels in the tree, $N_\mathrm{SPU}^{(n)}$ is the number of SPUs at level n, which is $N_\mathrm{SPU}^{(n)} = \frac{P}{4^n}$, and $N_\mathrm{b}^{(0)} = N_\mathrm{p}$ for notation convenience.

The formulation phase of RMF includes the computation of $\|\mathbf{h}\|^2$ for each user. For the IIC algorithm, the steps required for the formulation phase are shown in Algorithm 3 for each panel[14]. This algorithm relies on SVD, which we assume is based on two steps: the Householder bidiagonalization and the QR method by Givens rotations. Bidiagonalization is dominant in terms of complexity, so the total complexity of SVD of an $L \times T$ complex matrix can be approximated by $2L^2T$. For step 1 of the Algorithm 3, SVD of a $K \times K$ Gramian matrix $\mathbf{Z}_{i-1}$ is required, with a complexity of $2K^3$. Step 2 has a complexity of $(M_\mathrm{p}+1)K^2$, and step 3 combined with step 4 require a complexity of $2N_\mathrm{p}d_0^2$, where $d_0 = \max\{K, M_\mathrm{p}\}$. $\mathbf{H}_{eq}^H = \mathbf{W}^H\mathbf{H}$ consists of $N_\mathrm{p}M_\mathrm{p}K$ products, and step 5 requires $N_\mathrm{p}K^2$ products. We evaluate formulation over the whole bandwidth, and for that, we assume one channel estimate per physical resource block (PRB), and therefore one filtering matrix calculation per PRB. The total computational complexity (in MACs) for IIC is given by

$$C_\mathrm{form,IIC} = \underbrace{N_\mathrm{PRB} P C_\mathrm{form}^{(0)}}_{\text{front-end}} + \underbrace{N_\mathrm{PRB} \sum_{n=1}^{L} N_\mathrm{SPU}^{(n)} C_\mathrm{form}^{(n)}}_{\text{backplane}} \tag{16}$$

[14] For backplane there is no exchange of $\mathbf{Z}$ as explained in Section 5, so the computational complexity is highly reduced.

where $C_{\text{form}}^{(0)} = (2K + M_{\text{p}} + N_{\text{p}} + 1)K^2 + N_{\text{p}}M_{\text{p}}K + 2N_{\text{p}}d_0^2$ is the computational complexity per panel and PRB during formulation, while $C_{\text{form}}^{(n)} = 4N_{\text{b}}^{(n)}N_{\text{b}}^{(n-1)}K + 2N_{\text{b}}^{(n)}d_n^2$ is the one per node at level n, and $d_n = \max\{K, 4N_{\text{b}}^{(n-1)}\}$. For RMF, expression (16) also applies, with $C_{\text{form}}^{(0)} = M_{\text{p}}K$, and $C_{\text{form}}^{(n)} = 4N_{\text{b}}^{(n-1)}K$. Fig. 9a shows normalized sum-rate capacity versus computational complexity during filtering for both algorithms and different panel sizes. We observe that IIC achieves better performance than RMF for the same panel size, while large panels are key to harvest most of the capacity, with $M_{\text{p}} \geq 64$ reaching channel capacity in our simulations. Fig. 9c shows sum-rate capacity versus computational complexity during filtering for different LIS sizes (IIC and $M_{\text{p}} = 64$ assumed). It is interesting to observe that the same performance (for example 200) can be achieved by both $M = 4096$ and $M = 1024$, and with the same computational complexity. However, their architecture may differ substantially, as the smaller LIS requires higher output dimensionality per panel and lower reduction than the larger LIS, where aggressive dimensionality reduction can be used. In summary, the smaller LIS ($M = 1024$) is harvesting a significant fraction of the available channel capacity, while the larger LIS is only exploiting a very small fraction of it. This presents a very interesting design trade-off.

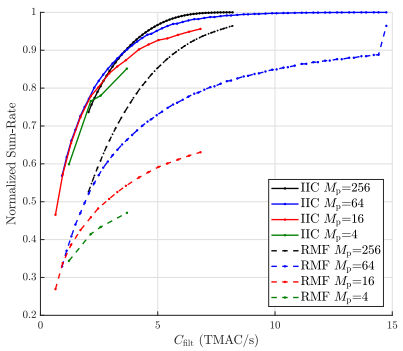
6.4 Interconnection bandwidth

In this section we analyze the interconnection bandwidth during the filtering phase, covering panel-node and node-node links. This bandwidth (in bps) is given by

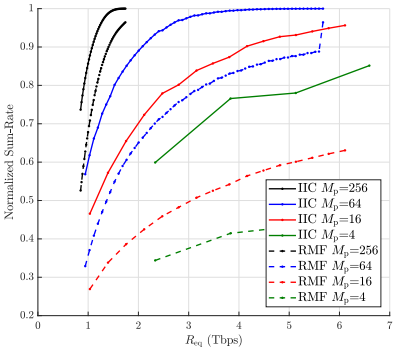
$$R_{\text{inter}} = \underbrace{2wf_{\text{B}}PN_{\text{p}}}_{\text{front-end}} + \underbrace{2wf_{\text{B}}\sum_{n=1}^{L} N_{\text{SPU}}^{(n)}N_{\text{b}}^{(n)}}_{\text{backplane}}, \tag{17}$$

where w is the bit-width of the SPU input/output (real and imaginary parts). Our analysis includes the movement of data happening internally at panels/nodes level, which covers the data transfer between the inputs ports to the SPU for processing, and from it to the output ports. We name this transfer data-rate as *intra-connection data-rate* or R_{intra}, and is given by

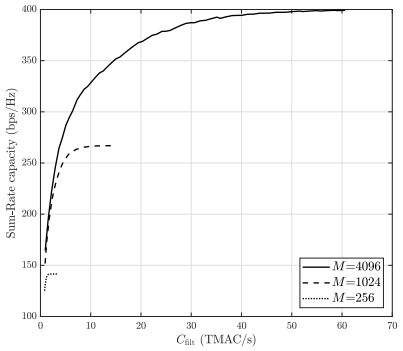
$$R_{\text{intra}} = \underbrace{PR_{\text{intra,FE}}}_{\text{front-end}} + \underbrace{\sum_{n=1}^{L} N_{\text{SPU}}^{(n)}R_{\text{intra,BP}}}_{\text{backplane}},$$



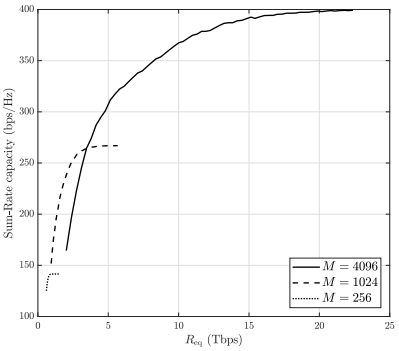
(a) Sum-rate vs C_{filt}



(b) Sum-rate vs R_{eq}



(c) Sum-rate vs C_{filt} for different M



(d) Sum-rate vs R_{eq} for different M

Figure 9: Sum-rate capacity normalized by channel capacity at CDSP interface versus computational complexity (9a) and interconnection data-rate (9b). In all cases, results for different panel sizes are shown, together with both algorithms. For both cases: $\beta_{\text{b1}} = \beta_{\text{b2}}$. Simulated points represent different N_{p} values. Sum-rate capacity versus computational complexity (9c), and versus interconnection data-rate for different LIS size (9d). $M_{\text{p}} = 64$, IIC method, $\alpha = \frac{1}{10}$, and $\rho = 10$. $K = 64$ in all cases.

where $R_{\text{intra,FE}} = 2wf_{\text{B}}(M_{\text{p}} + N_{\text{p}})$, and $R_{\text{intra,BP}} = 2wf_{\text{B}}(4N_{\text{b}}^{(n-1)} + N_{\text{b}}^{(n)})$ correspond to a panel and backplane node respectively. We are aware that R_{intra} does not include all internal data-rate in a real system, as this is highly depen-

dent on the specific implementation, internal topology, and type of processing unit employed in the panel. However, the spirit of this work is to provide a general analysis and first order approximation of the complexity required, applicable to all possible implementations, instead of being attached to a specific hardware implementation, and provide exact analysis numbers.

In order to take both magnitudes into consideration in the analysis, we define the relative cost α, as $\alpha \triangleq \text{cost}(R_{\text{intra}})/\text{cost}(R_{\text{inter}})$, and the cost equivalent interconnection data-rate R_{eq} as: $R_{\text{eq}} \triangleq R_{\text{inter}} + \alpha R_{\text{intra}}$. In this analysis, the ratio power/data-rate is considered as cost magnitude. If serial link (serdes) technology is assumed for intra-connection, and Ethernet for interconnection, then a power consumption of $1.29 - 24.8$mW/Gbps, and 40mW/Gbps are obtained respectively according to different sources [34–37]. The serdes power range is very wide, so we take 4mW/Gbps as reference, that gives $\alpha \sim \frac{1}{10}$[15].

Fig. 9b shows normalized sum-rate capacity versus equivalent interconnection bandwidth during filtering for both algorithms and different panel sizes. According to the results, IIC achieves better performance than RMF for same panel size, and large panels are capable of harvesting most of the channel capacity in our simulations. It is relevant to point out that small panels require more total interconnection data-rate than large panels, however this is more distributed among panels and nodes, reducing the bottlenecks considerably. Fig. 9d shows the sum-rate capacity versus interconnection bandwidth during filtering for different LIS sizes (IIC assumed). Similar conclusions can be drawn compared to Fig. 9c.

6.5 Processing latency

The processing latency represents the time between when the estimated channel of a subcarrier is available at panels and when the data of that subcarrier is filtered and available at the CDSP input for detection. The latency can be expressed as $L_{\text{tot}} = L_{\text{form}} + L_{\text{filt}}$, where L_{form} is the formulation latency, and L_{filt} is the latency for data filtering. More specifically, $L_{\text{form}} = L_{\text{form}}^{\text{proc}} + (n_{\text{P}} - 1)L_{\text{local}}^{\text{com}} + (L+1)L_{\text{global}}^{\text{com}}$, where n_{P} is the number of panels involved ($n_{\text{P}} = 1$ in RMF and P in IIC for the worst case)[16], $L_{\text{form}}^{\text{proc}}$ is the time needed to calculate the filter coefficients, $L_{\text{local}}^{\text{com}}$ refers to panel-to-panel communication latency (only in IIC), and $L_{\text{global}}^{\text{com}}$ refers to panel-to-node, and node-to-node link communication latency. For filtering latency: $L_{\text{filt}} = L_{\text{filt}}^{\text{proc}} + (L+1)L_{\text{global}}^{\text{com}}$, which accounts for filtering in panels and nodes, and communication latency.

[15] These numbers are dependent on the technology used, however, the method still holds.
[16] Depending on the users distribution there may not be a need to go through all panels ($n_{\text{P}} < P$) with the subsequent benefits. We leave this for future work.

We assume the IIC formulation is performed sequentially along all panels (worst case) using local connections, and then across nodes in the tree.

The latency for processing greatly depends on the hardware architecture used to implement the algorithms. Here we assume highly optimized accelerators (e.g., application-specific integrated circuit) are used such that the available data parallelism (N_{paral}) can be explored using N_{proc} processing units ($N_{\text{proc}} < N_{\text{paral}}$), i.e., the N_{proc} units will take $N_{\text{paral}}/N_{\text{proc}}$ clock cycles to iteratively process N_{paral} parallel operations. Moreover, the channel matrix (of the subcarrier that is being processed) is cached in register files (the latency for memory access is hidden). The main component of $L_{\text{form}}^{\text{proc}}$ is the time needed to perform SVD which is implemented by Householder bidiagonalization followed by QR method based on Givens rotations. The processing of each column and row can be done in parallel, while sequential processing is needed between columns and rows due to the data dependency.

With these assumptions, the total processing latency in formulation phase is $L_{\text{form}}^{\text{proc}} = \frac{\widetilde{C}_{\text{form}} T_{\text{CLK}}}{N_{\text{proc}}}$, where $\widetilde{C}_{\text{form}} = n_{\text{P}} C_{\text{form}}^{(0)} + \sum_{n=1}^{L} C_{\text{form}}^{(n)}$. The first term in $\widetilde{C}_{\text{form}}$ represents the serial processing in the front-end, and the second term represents the computational complexity of one branch of the tree. $C_{\text{form}}^{(0)}$ and $C_{\text{form}}^{(n)}$ are defined after (16). T_{CLK} is the clock period, and it is assumed that one MAC can be done within one clock cycle. In case of filtering, processing latency is given by $L_{\text{filt}}^{\text{proc}} = \frac{\widetilde{C}_{\text{filt}} T_{\text{CLK}}}{N_{\text{proc}}}$, where $\widetilde{C}_{\text{filt}} = \sum_{n=0}^{L} C_{\text{filt}}^{(n)}$ is the computational complexity corresponding to a path between a panel and the CDSP, and $C_{\text{filt}}^{(n)}$ is defined after (15).

6.6 Case study and discussion

The performance has been analyzed, together with the computational complexity, interconnection data-rate, and processing latency. General expressions for these different metrics have been presented based on general system parameters, such as the number of users, number of antennas, number of panels, and signal bandwidth, among others; what makes it easy to particularize for concrete implementations. Nevertheless, based on the trade-off analysis shown in Fig. 9a and Fig. 9b, we can see $M_{\text{p}} = 64$ as an attractive option, as it provides higher capacity than $M_{\text{p}} = 16$ for the same computational complexity and interconnection data-rate, while it is able to reach channel capacity in our analysis scenario. On top of that, its physical dimensions ($30cm \times 30cm$ at 4GHz) make it easy to handle and mount. Numerical values of the analyzed complexity for this panel size are presented in Table 1. The following parameter values are assumed: $n_{\text{P}} = P = 16$, $T_{\text{CLK}} = 1ns$, $N_{\text{paral}} = 100$, $L_{\text{local}}^{\text{com}} = 100ns$ (serdes

Table 1: $M = 1024$, $M_{\mathrm{p}} = 64$, $N_{\mathrm{p}} = 13$, $N_{\mathrm{b}} = 25$, $K = 50$, $w = 16$ bits, $f_{\mathrm{B}} = 100$MHz. Units are as follows: C_{form} [GMAC], C_{filt} [TMAC/s], R_{inter} [Tb/s], R_{intra} [Tb/s], L [μs]

Method	C_{form}	C_{filt}	R_{inter}	R_{intra}	L_{form}	L_{filt}
IIC	3.1	2.4	1.1	5.4	111.4	1.0
RMF	0.02	2.4	1.1	5.4	1.0	1.0

technology assumed [34,35]), and $L^{\mathrm{com}}_{\mathrm{global}} = 300ns$ (Ethernet assumed [36–38]). Assuming 12 subcarriers per PRB, the subcarrier spacing in our example is: $\frac{f_{\mathrm{B}}}{12N_{\mathrm{PRB}}} = 30$KHz, and the OFDM symbol duration is therefore $\approx 33\mu s$.

The benefits of the distributed architecture are evident in terms of interconnection data-rate reduction. Looking at the CDSP input interface, the reduction is easily obtained as: $\frac{M}{K} \sim 20$x. Of course, this comes at the cost of a performance loss due to dimensionality reduction, but as it was explained before, the system is fully configurable, offering a rich performance-complexity trade-off. It is important to consider that even though computational complexity and interconnection data rates numbers may seem large, they are distributed among all processing units in the LIS. This LIS contains 21 SPUs (panels + backplane nodes).

Regarding latency, $L_{\mathrm{form,RMF}}$ and L_{filt} values seem reasonable for the NR frame structure. We observe $L_{\mathrm{form,IIC}}$ shows much higher value due to the higher computational complexity required in this method (equivalent to 3 OFDM symbols in this example). For a certain LIS system, this latency is sensitive to the β used in panels and nodes (which translates into complexity cost). Therefore, we can see a trade-off between these system parameters and how often filters are updated in panels and tree nodes. It is important to remark that latency is analyzed from a worst case point of view, where all panels in the LIS are serially connected and jointly contribute to the formulation. In reality this may not be the best approach as this may only be helpful in cases with very high density of users with dominant interference over noise. We envision groups of panels performing formulation in parallel, where those panels belonging to the same group perform serial processing, reducing the formulation latency considerably. We are aware that depending on the implementation, latency may be different (selection of memory system, hardware, interconnection, etc), and here we provide high level analysis assuming the use of dedicated accelerators without any overhead.

In the present work, we also compare our proposed scheme to existing ones in literature, in particular we select one scheme based on daisy-chain topology, which can be found in recent works, such as [25] and [26] for Massive MIMO,

Table 2: Description of three different plans for dimensionality reduction used for comparison, and corresponding system parameters values.

	M_{p}	N_{p}	N_{b}	K
Case A	64	50	50	50
Case B	64	32	50	50
Case C	64	16	16	50

and in cell-free massive MIMO networks with serial fronthaul (also known as radio stripe) with examples in [39] and [40] among others. [39] performs exact L-MMSE (linear minimum mean-square error) detection using daisy-chain topology with distributed processing. For the comparison, IIC is used as the method for dimensionality reduction, and L-MMSE for detection in the CDSP. The filtering matrix used for linear detection is based on the equivalent channel formed by the combination of the wireless channel and the filtering matrices implemented in the front-end and backplane, which is available at the CDSP following the procedure described in Section 5.1. We establish 3 different configurations with different levels of dimensionality reduction: a scenario with low dimensionality reduction (case A), medium reduction (case B), and high reduction (case C). System parameters can be seen in Table 2. The other system parameters are chosen according to Table I. BER (bit error rate) has been simulated for 16 QAM transmission and the SNR (ρ) that achieves BER=10^{-3} for uncoded transmission is included. For this experiment all users are assumed to transmit with the same mean power, and BER accounts for errors in all users. For the sum-rate capacity we provide an approximate value of $\mathbb{E}\left\{\sum_k^K (\log_2(1+\mathrm{SINR}_k))\right\}$, where SINR_k represents the instantaneous signal-to-interference-plus-noise ratio (SINR) for user k, and the expectation is with regard to the wireless channel. We employ 100 channel realizations for this approximation. The ergodic wireless channel capacity (EC) has also been included for reference. The computational complexity, interconnection data-rate and latency have also been included, and only take data filtering phase into account (no formulation). The result of the analysis is summarized in Table 3. In case of [39], and for comparison purposes only, we assume each AP (access point) acts as one LIS panel.

Table 3: Summary of the comparison between proposed IIC method and L-MMSE detection with daisy-chain serial processing architecture with L-MMSE. Ergodic wireless channel capacity (EC) is added as performance reference. Three different dimensionality reduction plans for IIC are included for comparison. System parameters are according to Table 2. Other values are: $M = 1024$, $M_{\mathrm{p}} = 64$, $K = 50$, $w = 16$ bits, and $f_{\mathrm{B}} = 100$MHz.

Metric/Method	EC	Alg.1 [39]	Case A	Case B	Case C
SNR @ BER=10^{-3} (dB)	-	25.6	25.6	25.6	43.0
Sum-rate capacity @ SNR=26 dB (bps/Hz)	470.9	461.6	461.6	461.6	323.7
C_{filt} (TMAC/s)	-	10.24	10.37	7.09	2.62
R_{inter} (Tbps)	-	2.56	3.52	2.37	1.24
$L_{\mathrm{filt}}(\mu s)$	-	2.01	1.13	1.08	0.95

For the computational complexity, C_{filt} in (15) has been used to account for panels and backplane, while an extra 0.25 TMAC/s = $f_{\text{B}}K^2$ has been added to take linear detection (filtering) in CDSP into account. The computational complexity in case of [39] has been calculated as $2f_{\text{B}}MK$. Regarding interconnection data-rate, R_{inter} in (17) has been used for the three cases, while $2wf_{\text{B}}PK$ is used to calculate the corresponding value for [39].

Our analysis indicates that the proposed method allows to exploit locality of users in the panels by taking advantage of the dimensionality reduction that the IIC method provides. As discussed before, it offers the same performance in terms of BER and sum-rate compared to an exact L-MMSE solution, while requiring almost half the computational complexity, slightly less interconnection data-rate, and half latency (in case B) compared to the daisy-chain topology.

7 Conclusions

In this article we have presented distributed uplink processing algorithms and the corresponding hardware architecture for efficient implementation of large intelligent surface (LIS). The proposed processing structure consists of local processing units near antennas to reduce incoming data dimensionality without losing much information, and hierarchical backplane network with distributed processing-combining units to support flexible and efficient data aggregation. We have systematically analyzed the system capacity and implementation cost with different design parameters, and provided design guidelines for the implementation of LIS.

Appendix

A. Proof of Proposition 1

Proof: According to (9), the sum-rate capacity with the multi-panel architecture is given by

$$C = \log_2 |\mathbf{I}_K + \rho \mathbf{A}|, \tag{18}$$

where $\mathbf{A} = \sum_{i=1}^{P} \mathbf{H}_i^H \mathbf{Q}_i \mathbf{Q}_i^H \mathbf{H}_i$. For a certain channel realization, the maximum capacity is achieved if all eigenvalues of $\mathbf{A}$ are equal, that is: $\lambda_n = \overline{\lambda}, 1 \leq n \leq K$[17]. In that case, the capacity would be: $C_{\text{ub1}} = K \log_2(1 + \rho\overline{\lambda})$. Now, let us find the maximum value for $\overline{\lambda}$ as follows

$$\begin{aligned} \overline{\lambda} &= \max_{\{\mathbf{Q}_i\}} \frac{1}{K} \operatorname{Tr}\{\mathbf{A}\} = \frac{1}{K} \sum_{i=1}^{P} \max_{\mathbf{Q}_i} \operatorname{Tr}\{\mathbf{H}_i^H \mathbf{Q}_i \mathbf{Q}_i^H \mathbf{H}_i\} \\ &= \frac{1}{K} \sum_{i=1}^{P} \sum_{n=1}^{N_{\text{p}}} \lambda_n^{(i)}, \end{aligned} \tag{19}$$

and then $C \leq C_{\text{ub1}}$, proving the proposition[18]. ■

B. Proof of solution to local optimization in Algorithm 2

Proof: We drop the panel index for simplicity. The objective function to maximize is

$$\begin{aligned} |\rho \mathbf{H}^H \mathbf{Q}\mathbf{Q}^H \mathbf{H} + \mathbf{Z}| &= |\mathbf{Z}||\mathbf{I}_K + \rho \mathbf{Z}^{-1/2} \mathbf{H}^H \mathbf{Q}\mathbf{Q}^H \mathbf{H}\mathbf{Z}^{-1/2}| \\ &= |\mathbf{Z}||\mathbf{I}_{N_{\text{p}}} + \rho \mathbf{Q}^H \mathbf{H}\mathbf{Z}^{-1}\mathbf{H}^H \mathbf{Q}| \\ &= |\mathbf{Z}||\mathbf{Q}^H(\rho \mathbf{H}\mathbf{Z}^{-1}\mathbf{H}^H + \mathbf{I}_{M_{\text{p}}})\mathbf{Q}|. \end{aligned}$$

$|\mathbf{Z}|$ does not depend on $\mathbf{Q}$, therefore the solution to our problem is the same as the solution of the maximization of the second determinant, which consists of the ordered eigenvectors (in descent order of corresponding eigenvalue) of the matrix: $\mathbf{H}\mathbf{Z}^{-1}\mathbf{H}^H$. ■

C. Proof of white filtered noise

As an example, the filtered noise due to the first four panels and the connected nodes is denoted by $\mathbf{n}_1^{(1)}$ and obtained as: $\mathbf{n}_1^{(1)} = \mathbf{W}_{\text{B},1}^{(1)H} \mathbf{W}_{\text{P},1-4}^H \mathbf{n}_{1-4}$, where

[17] Note that we assume $rank(\mathbf{A}) = K$, and $PN_{\text{p}} \geq K$. [18] We remark that this bound may not be attained in practice, as it needs a favorable set of $\{\mathbf{H}_i\}$, such that it provides uniform eigenvalues in $\mathbf{A}$ with the proper selection of $\{\mathbf{Q}_i\}$.

$\mathbf{W}_{\mathrm{P},1-4}$ is the combined filtering matrix of the first four panels and it is defined as $\mathbf{W}_{\mathrm{P},1-4} = \mathrm{diag}(\mathbf{W}_{\mathrm{P},1}, \mathbf{W}_{\mathrm{P},2}, \mathbf{W}_{\mathrm{P},3}, \cdots, \mathbf{W}_{\mathrm{P},4})$, and $\mathbf{n}_{1-4}$ is the aggregated input noise vector corresponding to the first four panels and it is defined as $\mathbf{n}_{1-4} = [\mathbf{n}_1^T, \mathbf{n}_2^T, \mathbf{n}_3^T, \mathbf{n}_4]^T$. The covariance is given by

$$\begin{aligned}
\mathbb{E}\left\{\mathbf{n}_1^{(1)}\mathbf{n}_1^{(1)H}\right\} &= \mathbf{W}_{\mathrm{B},1}^{(1)H}\mathbf{W}_{\mathrm{P},1-4}^{H}\mathbb{E}\{\mathbf{n}_{1-4}\mathbf{n}_{1-4}^{H}\}\mathbf{W}_{\mathrm{P},1-4}\mathbf{W}_{\mathrm{B},1}^{(1)} \\
&= \mathbf{W}_{\mathrm{B},1}^{(1)H}\mathbf{W}_{\mathrm{P},1-4}^{H}\mathbf{I}_{4M_{\mathrm{p}}}\mathbf{W}_{\mathrm{P},1-4}\mathbf{W}_{\mathrm{B},1}^{(1)} \\
&= \mathbf{W}_{\mathrm{B},1}^{(1)H}\mathbf{I}_{4N_{\mathrm{p}}}\mathbf{W}_{\mathrm{B},1}^{(1)} = \mathbf{I}_{N_{\mathrm{b}}^{(1)}}
\end{aligned}$$

Bibliography

[1] J. R. Sanchez, O. Edfors, F. Rusek, and L. Liu, Processing Distribution and Architecture Tradeoff for Large Intelligent Surface Implementation, in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1-6.

[2] J. R. Sanchez, F. Rusek, O. Edfors, and L. Liu, An Iterative Interference Cancellation Algorithm for Large Intelligent Surfaces, *arXiv e-prints*, p. arXiv:1911.10804, Nov. 2019.

[3] S. Hu, F. Rusek, and O. Edfors, Beyond Massive MIMO: The Potential of Data Transmission With Large Intelligent Surfaces, *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2746-2758, May 2018.

[4] S. Hu, F. Rusek, and O. Edfors, The Potential of Using Large Antenna Arrays on Intelligent Surfaces, in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, June 2017, pp. 1-6.

[5] S. Hu, K. Chitti, F. Rusek, and O. Edfors, User Assignment with Distributed Large Intelligent Surface (LIS) Systems, in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2018, pp. 1-6.

[6] S. Hu, F. Rusek, and O. Edfors, Beyond Massive MIMO: The Potential of Positioning With Large Intelligent Surfaces, *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1761-1774, April 2018.

[7] M. Di Renzo *et al.*, Smart Radio Environments Empowered by Reconfigurable Intelligent Surfaces: How It Works, State of Research, and The Road Ahead, *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2450-2525, 2020.

[8] Liu *et al.*, Reconfigurable Intelligent Surfaces: Principles and Opportunities, *IEEE Communications Surveys Tutorials*, pp. 1-1, 2021.

[9] Björnson *et al.*, "Reconfigurable Intelligent Surfaces: A signal processing perspective with wireless applications," in IEEE Signal Processing Magazine, vol. 39, no. 2, pp. 135-158, March 2022.

[10] H. Wymeersch, J. He, B. Denis, A. Clemente, and M. Juntti, Radio Localization and Mapping With Reconfigurable Intelligent Surfaces: Challenges, Opportunities, and Research Directions, *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 52-61, 2020.

[11] W. Tang *et al.*, Wireless Communications With Reconfigurable Intelligent Surface: Path Loss Modeling and Experimental Measurement, *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 421- 439, 2021.

[12] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, Intelligent Reflecting Surface-Aided Wireless Communications: A Tutorial, *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3313-3351, 2021.

[13] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M. Alouini, and R. Zhang, Wireless Communications Through Reconfigurable Intelligent Surfaces, *IEEE Access*, vol. 7, pp. 116 753-116 773, 2019.

[14] C. Huang *et al.*, Holographic MIMO Surfaces for 6G Wireless Networks: Opportunities, Challenges, and Trends, IEEE Wireless Commun., vol. 27, no. 5, pp. 118-125, Oct. 2020.

[15] A. Taha, M. Alrabeiah, and A. Alkhateeb, Enabling Large Intelligent Surfaces With Compressive Sensing and Deep Learning, *IEEE Access*, vol. 9, pp. 44 304-44 321, 2021.

[16] Y. Han, W. Tang, S. Jin, C. Wen, and X. Ma, Large Intelligent Surface-Assisted Wireless Communication Exploiting Statistical CSI, *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8238-8242, Aug 2019.

[17] C. Huang, G. C. Alexandropoulos, A. Zappone, M. Debbah, and C. Yuen, Energy Efficient Multi-User MISO Communication Using Low Resolution Large Intelligent Surfaces, in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec 2018, pp. 1-6.

[18] M. Jung, W. Saad, Y. Jang, G. Kong, and S. Choi, Performance Analysis of Large Intelligent Surfaces (LISs): Asymptotic Data Rate and Channel Hardening Effects, *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2052-2065, 2020.

[19] J. V. Alegría and F. Rusek, Achievable Rate with Correlated Hardware Impairments in Large Intelligent Surfaces, in *2019 IEEE 8th Interna- tional Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2019, pp. 559-563.

[20] D. Dardari, Communicating with Large Intelligent Surfaces: Fundamental Limits and Models, *IEEE Journal on Selected Areas in Communications*, pp. 1-1, 2020.

[21] R. J. Williams, E. de Carvalho, and T. L. Marzetta, A Communication Model for Large Intelligent Surfaces, in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1-6.

[22] S. Malkowsky *et al.*, The World's First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation, *IEEE Access*, vol. 5, pp. 9073-9088, 2017.

[23] 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, Dec 2017.

[24] A. Puglielli *et al.*, Design of Energy- and Cost-Efficient Massive MIMO Arrays, *Proceedings of the IEEE*, vol. 104, no. 3, pp. 586-606, March 2016.

[25] J. R. Sanchez, F. Rusek, O. Edfors, M. Sarajli'c, 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.

[26] J. R. Sanchez, F. Rusek, M. Sarajlic, O. Edfors, and L. Liu, Fully Decentralized Massive MIMO Detection Based on Recursive Methods, in *2018 IEEE International Workshop on Signal Processing Systems (SiPS)*, 2018, pp. 53-58.

[27] J. V. Alegria, J. R. Sanchez, F. Rusek, L. Liu, and O. Edfors, Decentralized Equalizer Construction for Large Intelligent Surfaces, in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, Sep. 2019, pp. 1-6.

[28] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays, *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40-60, Jan 2013.

[29] J. V. Alegria, F. Rusek, J. R. Sanchez, and O. Edfors, Trade-Offs in Quasi-Decentralized Massive MIMO, in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1-6.

[30] J. Vidal Alegria, F. Rusek, and O. Edfors, Trade-Offs in Decentralized Multi-Antenna Architectures: The WAX Decomposition, *IEEE Transactions on Signal Processing*, vol. 69, pp. 3627-3641, 2021.

[31] C. Jeon, K. Li, J. R. Cavallaro, and C. Studer, On the Achievable Rates of Decentralized Equalization in Massive MU-MIMO Systems, in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 1102-1106.

[32] Wei Yu, Wonjong Rhee, S. Boyd, and J. M. Cioffi, Iterative Water- Filling for Gaussian Vector Multiple-Access Channels, *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 145-152, Jan 2004.

[33] 3GPP specification series: 38 series, 2022. [Online]. Available: https://www.3gpp.org/DynaReport/38-series.htm

[34] S. Safwat *et al.*, A 12Gbps All Digital Low Power SerDes Transceiver for On-chip Networking, in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, 2011, pp. 1419-1422.

[35] R. Navid *et al.*, A 40 Gb/s Serial Link Transceiver in 28 nm CMOS Technology, *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 814-827, 2015.

[36] J. L. Wei *et al.*, 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.

[37] J. K. Lee and Y. Jang, Compact 4 x 25 Gb/s Optical Receiver and Transceiver for 100G Ethernet Interface, in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, 2015, pp. 758-760.

[38] Texas Instruments, DP83867IR/CR Robust, High Immunity 10/100/1000 Ethernet Physical Layer Transceiver. [Online]. Available: https://www.ti.com/lit/ds/symlink/dp83867ir.pdf

[39] 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.

[40] Z. H. Shaik, E. Björnson, and E. G. Larsson, Cell-Free Massive MIMO with Radio Stripes and Sequential Uplink Processing, in *2020 IEEE International Conference on Com- munications Workshops (ICC Workshops)*, 2020, pp. 1-6.

Paper VI

Positioning for Distributed Large Intelligent Surfaces using Neural Network with Probabilistic Layer

Wireless-based positioning with large antenna arrays is a promising enabler of the high accuracy positioning services envisioned for 6G. These systems provide high spatial resolution due to the large number of antennas, while enjoying the benefit of sharing a common infrastructure between communication and positioning. Among the available techniques for wireless-based positioning, channel state information (CSI)-based fingerprinting via neural networks (NNs) offers high accuracy under challenging propagation conditions, without the need of storing and accessing large amounts of measurement data during inference. On the other hand, large antenna systems, such as Large Intelligent Surfaces (LIS), benefits from a distributed architecture and local processing of wireless signals received from nearby antennas, producing intermediate results that can be aggregated, and therefore considerably reducing the demand on interconnection bandwidth. In this work, we propose a method to perform positioning of users based on estimated CSI in a LIS built from panels. Following this method, panels provide a parameterized probability density function for the location of each user, which can be shared conveniently and fused in different panels or a central processing unit (CPU), providing high positioning accuracy using very low interconnection bandwidth.

Jesús Rodríguez Sánchez, Ove Edfors and Liang Liu
"Positioning for Distributed Large Intelligent Surfaces using Neural Network with Probabilistic Layer,"
in *Proceedings of the 2021 IEEE Global Communications Conference Workshops (Globecom Workshops)*, 2021.

1 Introduction

6G is envisioned to enable services based on high positioning accuracy in indoor and outdoor venues [1]. It is well known that large antenna arrays with very high spatial resolution are expected to be part of 6G radio-access systems. In addition to enabling the promised high communication data-rates, they can also provide high positioning accuracy. This facilitates a common infrastructure for communications and positioning, which is quite beneficial from cost and maintenance points of view. Large Intelligent Surfaces (LIS) is a technology providing these two capabilities [2, 3].

However, such large antenna systems, despite their benefits, present formidable implementation challenges, where computational and interconnection resources face critical bottlenecks that need to be overcome in order to realize these systems. In order to alleviate these limitations, a panelized LIS with tree topology has already been proposed for communication purposes [4], where panels contain processing capabilities to perform local baseband processing, with very little or limited cooperation among them. Individual panel results are aggregated before reaching the central processing unit (CPU), reducing considerably the interconnection bandwidth compared to a centralized LIS architecture, where raw baseband samples are shared with the CPU during the uplink detection phase, with the additional high computational burden.

In order to ensure LIS is able to support positioning applications (apart from communication ones), we explore efficient algorithms that can be mapped onto the tree-based panelized topology with distributed processing proposed in [4] without much hardware overhead. Following this idea, in this work we propose a method to perform positioning of users that suits such architecture. In this method, each panel estimates channel state information (CSI) (functionality available already in communication), which is further processed by a local neural network (NN) in order to map CSI to positioning information.

Neural networks have recently been applied to wireless positioning [5–15], mainly in the Massive MIMO arena, but also for indoor applications, for example based on WiFi. Most previous work is based on centralized processing providing point estimates of the user location. Recently, [15] proposed a distributed scheme for indoor positioning with probabilistic description and support for fusion of position information from several access points. Models based on probabilistic descriptions are far superior to the ones based on point estimates for one fundamental reason: probabilistic results contain a measure of the uncertainty in the estimate (an estimate with very high uncertainty does not provide much information), which is of great importance as it allows the model to express its uncertainty in the result based on the observations; in addition, uncertainty is the base for fusion of different estimates, which al-

lows them to be properly weighted. Following this reasoning, in our proposed method panels provide a parameterized probability density function for the location of a certain user, which can be conveniently shared and fused in different panels, tree processing nodes, or in the CPU, providing high accuracy using much lower interconnection bandwidth than the centralized architecture, where panels would share their estimated CSI with the CPU, and one NN would process all incoming CSI to deliver a position estimate. As we will see in Section 5, this decentralized approach can achieve few hundred times reduction in the interconnection data-rate.

2 System model

The system under consideration is graphically represented in Fig. 1. We consider a single-antenna user u [1] whose position, denoted by $\mathbf{p}^u = (x_u, y_u, z_u) \in \mathbb{R}^3$, is to be estimated. The user is transmitting a signal which is received by multiple panels forming a LIS. Each panel contains M_p antenna elements, together with radio-frequency, analog and baseband (BB) processing capabilities in order to perform down conversion of the received signal and obtain CSI. Once CSI is available locally at a panel, a machine learning algorithm based on neural networks produces a probabilistic description of the user position, denoted by $p_i(\mathbf{p}^u)$ for panel i. In other words, $p_i(\mathbf{p}^u)$ is the probability density of the user being in position $\mathbf{p}^u$. Multiple probability functions, from different panels, can be fused into a single probability density function, which can be used for further fusion down the pipeline with other panels or sensors, or to obtain a point estimate of the user location.

2.1 Signal model

We consider a LIS containing a total of M active antenna elements, and divided into P square-shaped panels, each with M_p elements, such that $M_\mathrm{p} \cdot P = M$. We assume an OFDM-based transmission system, centered at carrier frequency f_c, with a bandwidth BW across which N_sc equally spaced subcarriers contain pilots for channel estimation.

The $M \times 1$ received vector at the LIS for a certain subcarrier is given by

$$\mathbf{y} = \mathbf{h}x + \mathbf{n}, \tag{1}$$

where x is the transmitted pilot signal for which we, without loss of generality, assume $x = 1$, $\mathbf{h}$ is the channel response vector, and $\mathbf{n} \sim \mathcal{CN}(0, \sigma_n^2\mathbf{I})$ is an $M \times 1$ i.i.d. noise vector.

[1] Extension to multiple users is straightforward under the assumption that the channel responses are independently measured.

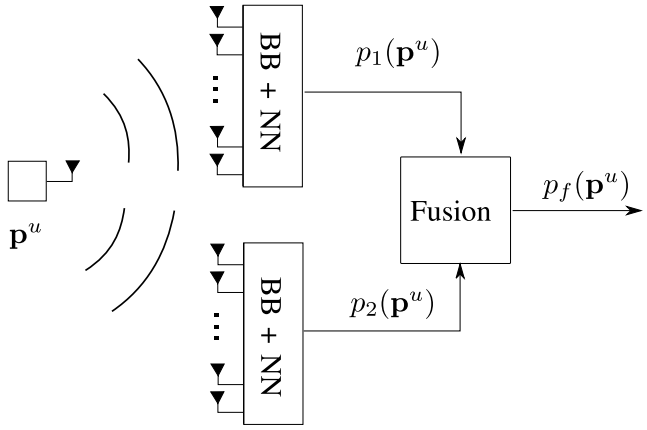


Figure 1: System model with a single-antenna user transmitting. Two distant panels provide an individual probabilistic description of the user's location, which is fused into a single probability function.

We model the line-of-sight (LoS) channel between the user at location $\mathbf{p}^u$ and a LIS antenna element at location $\mathbf{p}$ by the complex value [3]

$$h(\mathbf{p}, \mathbf{p}^u) = \frac{1}{d_u} \sqrt{\cos \phi(\mathbf{p}, \mathbf{p}^u)} \exp\left(-j \frac{2\pi d_u}{\lambda}\right), \tag{2}$$

where $\phi(\mathbf{p}, \mathbf{p}^u)$ is the relative orientation angle between the user antenna and the LIS antenna element at $\mathbf{p}$. When $\phi(\mathbf{p}, \mathbf{p}^u) = 0$ the LIS antenna is facing perpendicularly to the incoming wavefront. $d_u = \|\mathbf{p} - \mathbf{p}^u\|$ is the distance between the user and the antenna. λ is the wavelength at the corresponding subcarrier frequency. For our analysis we will consider more realistic channel models, concretely based on multipath propagation caused from specular reflection in walls, where a certain reflection coefficient is assumed, being denoted as α, and $0 \leq \alpha \leq 1$. The channel in this case is modeled as linear combination of individual components with respective reflection coefficients, this is $\sum \alpha_i h_i$, where α_i and h_i are the reflection coefficient and channel associated to the i-th multipath component respectively.

2.2 User position density model

In this work panels output the inferred probability distribution of the user position, which we model as a multivariate Gaussian distribution, this is $p_i(\mathbf{p}^u) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ for the i-th panel, where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the mean, and $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ the

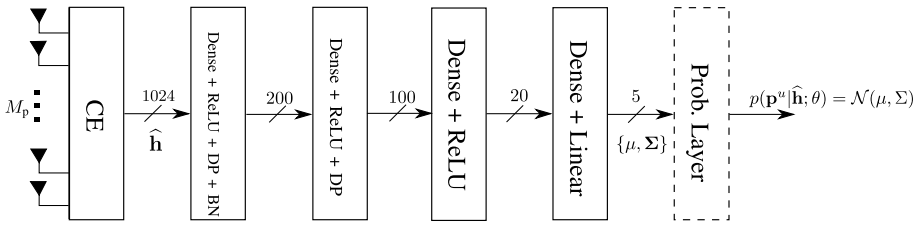


Figure 2: Processing pipeline for positioning inference in a panel, including channel estimation block and neural network. Acronyms: CE = channel estimation, DP = dropout, BN = batch normalization. Probabilistic layer is only used during training (dashed line). Numerical values represent dimensionality (real numbers) of the data exchanged assuming $M_{\mathrm{p}} = 64$ and $N_{\mathrm{sc}} = 8$. 2D positioning assumed.

covariance. The reason to select the distribution as Gaussian is twofold: the distribution is represented exclusively by the tuple $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$, requiring only a reduced number of values to be shared with the fusion module, and the fusion process becomes relatively simple, as further described in Section 4.

3 Positioning via Neural Networks

As presented in the Introduction, NN have recently been used for user positioning in wireless systems. It provides a low complexity approach for inference, as an alternative to CSI-based fingerprinting stored in a data-base.

3.1 Feature extraction

In our analysis we model the CSI estimate as $\mathcal{CN}(\mathbf{h}, \sigma_n^2\mathbf{I})$, and it is used during training and inference [2]. These user-specific complex-valued CSI obtained at pilot subcarriers are separated into real and imaginary parts and stacked together in a feature vector $\widehat{\mathbf{h}}$.

3.2 Neural Network with probability

The NN architecture is illustrated in Fig. 2. After the channel is estimated and the feature vector ($\widehat{\mathbf{h}}$) formed, four dense layers are used (three with ReLU activation functions and one with linear outputs). The output of the last

[2] We remark that we consider a noisy channel estimate in our analysis during training and inference.

dense layer is the probability distribution parameters $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. These parameters completely represent the distribution, and can be used during inference for point estimate (i.e. selecting the mean μ) and for fusion (see Section 4).

The last layer (dashed line in the figure) is the probabilistic layer and provides the probability density function using the input parameters. This is only used during training, and is described in more detail in next subsection.

The numerical values depicted in Fig. 2 correspond to the case of users positioned on a plane (used in our analysis in Section 5), therefore a 2D multivariate distribution is generated. In this particular case, the mean μ as a 2×1 vector, and the covariance $\boldsymbol{\Sigma}$ as a 2×2 matrix which, due to symmetry, only has three quantifying values.

3.3 Training Neural Networks with probability layers

The goal of training is to learn the NN parameters, here named θ, comprising weights and biases. In this section, and for clarity, we note that the NN outputs are a function of the input channel estimate $\widehat{\mathbf{h}}$ and θ, this is $\mu \equiv \mu(\widehat{\mathbf{h}}, \theta)$, and $\boldsymbol{\Sigma} \equiv \boldsymbol{\Sigma}(\widehat{\mathbf{h}}, \theta)$. As commented before, the probabilistic layer is only used during training. Its mission is to provide the corresponding probability density functions, which is required for computing the loss function.

The loss function used in this work is the Negative Log-Likelihood function, defined as

$$\mathrm{NLL}(\theta) = -\sum_{n} \log\{p(\mathbf{p}_n^u | \widehat{\mathbf{h}}_n; \theta)\}, \tag{3}$$

where $p(\mathbf{p}^u | \widehat{\mathbf{h}}_n; \theta) = \mathcal{N}(\boldsymbol{\mu}(\widehat{\mathbf{h}}_n, \theta), \boldsymbol{\Sigma}(\widehat{\mathbf{h}}_n, \theta))$, and $\{\widehat{\mathbf{h}}_n, \mathbf{p}_n^u\}$ is the training set made with different locations covering the area of service. For simplicity, we assume no error in the estimate of these locations. For a certain training location $\mathbf{p}_n^u$, the channel estimate is obtained $\widehat{\mathbf{h}}_n$, and the corresponding NN output for the current θ is calculated: $\{\boldsymbol{\mu}(\widehat{\mathbf{h}}_n, \theta), \boldsymbol{\Sigma}(\widehat{\mathbf{h}}_n, \theta)\}$. Given these input parameters, the probabilistic layer provides the full probability density, which is used to compute the likelihood, by evaluating the probability density at training location $\mathbf{p}_n^u$, as $p(\mathbf{p}_n^u | \widehat{\mathbf{h}}_n; \theta)$, as shown in (3). High values of likelihood (or equivalently lower values of NLL) indicate a good fit between the parameterized distribution and the ground truth location. The sum covers usually a subset of the training set (minibatch), and gives a cost value. This cost function is minimized using the Adam optimizer [16], whose outcome is the maximum likelihood (ML) solution (which minimizes the NLL), this is $\theta_{\mathrm{ML}} = \arg\min_\theta \mathrm{NLL}(\theta)$, which is used for inference.

Additionally during training, and to take the effect of noise distribution into account, we take N_{rep} samples per training location in the training set.

This means that for each location we sample the random variable $\widehat{\mathbf{h}}$ multiple times. This also imply to augment the size of the training set (even though the number of physical locations remains the same).

4 Probability fusion

The goal of probability fusion is to consolidate a finite number of probability distributions into a single one. In our case, we are interested in the fusion of probability densities provided by the panels as shown in Fig 1. Given that the individual density functions are Gaussian, Gaussian conflation represents a convenient fusion method, as it ensures the resulting distribution is also Gaussian, and leads to the classical weighted least squares method, providing the best unbiased and maximum likelihood estimators [17]. Following this method, the fused distribution is proportional to the product of the individual ones.

As mentioned before, the Gaussian conflation of P individual Gaussian distributions is also Gaussian, with covariance and mean represented respectively as

$$\boldsymbol{\Sigma}_f = \left(\sum_{i=1}^{P} \boldsymbol{\Sigma}_i^{-1} \right)^{-1}, \tag{4}$$

and

$$\boldsymbol{\mu}_f = \boldsymbol{\Sigma}_f \left(\sum_{i=1}^{P} \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i \right). \tag{5}$$

5 Simulation results and analysis

The scenario considered for our analysis is a volume of size (width×depth×height) $= (10m \times 10m \times 0.4m)$. The volume has four solid reflecting walls covering the sides (we do not consider reflection in floor and roof). Four panels of size $(0.4m \times 0.4m)$ are installed on the walls, occupying the center of each, as shown in Fig. 3a with red lines. The users are located on the plane crossing the panels by half, this is $(x, y, 0)$, assuming panels are within $(x, y, -0.2m)$ and $(x, y, 0.2m)$. The panels are trained individually according to the method described in Subsection 3.3. The system parameters are: $M_{\mathrm{p}} = 64$, $f_{\mathrm{c}} = 3$GHz ($\lambda = 10$cm), BW $= 100$MHz, $N_{\mathrm{sc}} = 8$, $N_{\mathrm{rep}} = 10$, and noise variance $\sigma_n^2 = 0.002$. For simplicity we only consider one specular reflection in the walls with $\alpha = 0.1$ [3]. We first consider 9 locations for inference (red dots). In

[3] Extension to more advanced channel models is left for future work.

Table 1: Mean and std error for panel 0 inference and fusion results in experiment where user is in nine locations, depicted in Fig. 3 . Units in cm.

Panels	0	$0,1$	$0,1,2$	$0,1,2,3$
mean error	31	4.2	2.7	3.0
std error	21	4.1	2.1	2.3

Table 2: Mean and std error for panel 0 inference and fusion results in experiment with square-shaped trajectory, depicted in Fig.4. Units in cm.

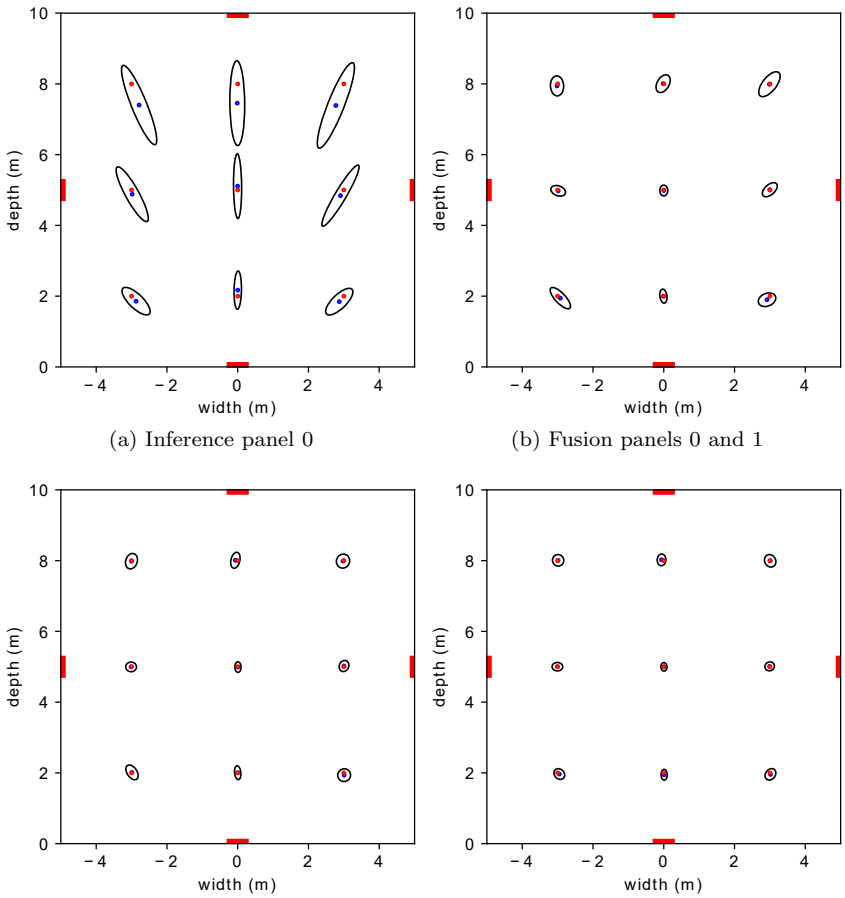
Panels	0	$0,1$	$0,1,2$	$0,1,2,3$
mean error	15	5.7	4.3	3.5
std error	11	3.0	2.3	1.7

Fig. 3a it is shown the result of inference of panel 0 (bottom) as 2 std ellipses (black) and the respective means (blue dots). We observe that the angular accuracy is quite good for all positions, while the distance accuracy gets worse for points further away from the panel, as they lay outside of the near-field region and less information about the distance is contained in observed, the increasingly planar, wave front. Error values are shown in Table 1, where error is measured between the distribution mean (μ) and the ground truth. Figure 3b shows the result of the fusion between panel 0 (bottom) and 1 (left), with an important improvement in accuracy as both panels complement each other [4]. Results of extended fusion process is shown in Fig. 3c and 3d, with incremental improvements in the accuracy.

Fig. 4 shows the result of another experiment, where we analyze 100 locations for a square-shaped trajectory. For some of them, 2 std and mean results of inference from panel 0 are shown in Fig. 4a, while 4b shows only the mean values of all locations. Error results are shown in Table 2. Result of fusion between panel 0 and 1 is shown in Fig. 4c and 4d for 2 std and mean, where we observe an important improvement in accuracy, similar to the observed in previous experiment. Results of the fusion of panels 0, 1 and 2 are shown in Fig. 4e and Fig. 4f. The result of fusion of all panels is shown in Fig. 4g and 4h.

From interconnection bandwidth point of view, there is a significant reduction in exchange of data compared to the centralized approach. Each panel shares 5 values (in case of 2D), instead of 1024 required by $\widehat{\mathbf{h}}$, which is a 200x reduction.

[4] The uncertainty in the distance shown by panel 0 is compensated by the high accuracy in the angle from panel 1 and vice versa.



(a) Inference panel 0

(b) Fusion panels 0 and 1

(c) Fusion panels 0, 1, and 2

(d) Fusion panels 0, 1, 2, and 3

Figure 3: Results of inference on 9 user locations. Top view of the scenario. Red lines represent panels, and red points denote ground truth locations. Blue points denote mean of distribution, and black 2 std ellipse. Fig. 3a represents results from panel 0. Fig. 3b represents the result of the fusion of panels 0 and 1. Fig. 3c represents the result of the fusion of panels 0, 1 and 2, and Fig. 3d shows the result of the fusion of all four panels.

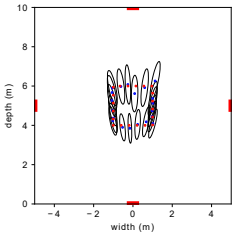
6 Conclusions

In the preset work we have introduced a novel method for wireless positioning in distributed Large Intelligent Surfaces using neural network with probabilistic

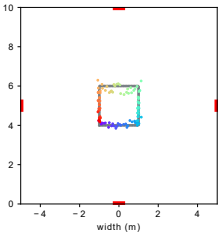
layer. Each panel forming the LIS provides a probabilistic description of the user location based on the local channel estimate, that can later be fused to a single probability distribution comprising information from more/all panels. By choosing a parameterized probability distribution, as the Gaussian, only the parameters need to be inferred, considerably reducing the interconnection bandwidth with the fusion module or CPU. Our analysis show that by fusion of two panels is enough to achieve fraction of wavelength accuracy level in a scenario with users distributed over a $100\lambda \times 100\lambda$ area.

7 Acknowledgment

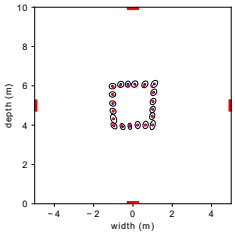
This work was supported by ELLIIT, the Excellence Center at Linköping-Lund in Information Technology.



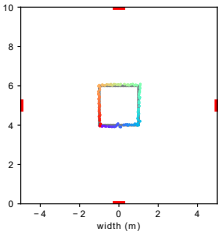
(a) Inference panel 0



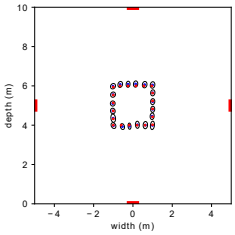
(b) Mean ($\boldsymbol{\mu}$) panel 0



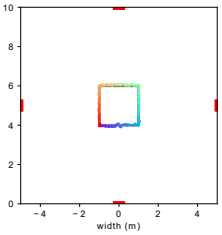
(c) Fusion panels 0 and 1



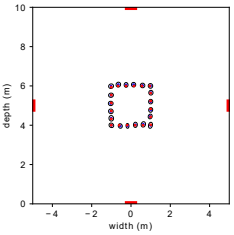
(d) Mean ($\boldsymbol{\mu}$) fusion panels 0 and 1



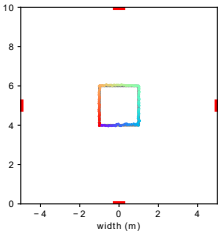
(e) Fusion panels 0, 1 and 2



(f) Mean ($\boldsymbol{\mu}$) fusion panels 0, 1 and 2



(g) Fusion panels 0, 1, 2 and 3



(h) Mean ($\boldsymbol{\mu}$) fusion panels 0, 1, 2 and 3

Figure 4: Results of inference on 100 user locations forming a square-shaped trajectory. Top view of the scenario. Left column: Red points denote ground truth. Blue points denote mean of inferred distribution, and black 2 std ellipse. Fig. 4a represents results from panel 0 (only 20 out of 100 locations are shown for convenience). Figures 4c, 4e, and 4g represent different results of fusion with different panels. Right column: Figures 4b, 4d, 4f, and 4h show the mean. Colors used are only intended to ease the visual association between estimated and true locations, given the high number of points.

Bibliography

[1] C. De Lima, D. Belot, R. Berkvens, A. Bourdoux, D. Dardari, M. Guillaud, M. Isomursu, E.-S. Lohan, Y. Miao, A. N. Barreto, M. R. K. Aziz, J. Saloranta, T. Sanguanpuak, H. Sarieddeen, G. Seco-Granados, J. Suutala, T. Svensson, M. Valkama, B. Van Liempd, and H. Wymeersch, Convergent communication, sensing and localization in 6G systems: An overview of technologies, opportunities and challenges, *IEEE Access*, vol. 9, pp. 26 902-26 925, 2021.

[2] S. Hu, F. Rusek, and O. Edfors, Beyond Massive MIMO: The Potential of Data Transmission With Large Intelligent Surfaces, *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2746-2758, May 2018.

[3] S. Hu, F. Rusek, and O. Edfors, Beyond Massive MIMO: The Potential of Positioning With Large Intelligent Surfaces, *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1761-1774, April 2018.

[4] J. Rodriguez Sanchez, F. Rusek, O. Edfors, and L. Liu, Distributed and Scalable Uplink Processing for LIS: Algorithm, Architecture, and Design Trade-offs, *arXiv e-prints*, Dec. 2020. [Online]. Available: http://arxiv.org/abs/2012.05296

[5] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, Deep convolutional neural networks for massive MIMO fingerprint-based positioning, in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1-6.

[6] S. D. Bast, A. P. Guevara, and S. Pollin, CSI-based positioning in massive mimo systems using convolutional neural networks, in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1-5.

[7] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, Machine learning methods for RSS-based user positioning in distributed massive MIMO,

IEEE Transactions on Wireless Communications, vol. 17, no. 12, pp. 8402-8417, 2018.

[8] J. Fan, S. Chen, X. Luo, Y. Zhang, and G. Y. Li, A machine learning approach for hierarchical localization based on multipath MIMO fingerprints, *IEEE Communications Letters*, vol. 23, no. 10, pp. 1765-1768, 2019.

[9] S. De Bast and S. Pollin, MaMIMO CSI-based positioning using CNNs: Peeking inside the black box, in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1-6.

[10] M. Arnold, S. Dorner, S. Cammerer, and S. Ten Brink, On deep learning-based massive MIMO indoor user localization, in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1-5.

[11] G. Cerar, A. vigelj, M. Mohori, C. Fortuna, and T. Javornik, Improving CSI-based massive MIMO indoor positioning using convolutional neural network, in *2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, 2021, pp. 276-281.

[12] B. Berruet, O. Baala, A. Caminada, and V. Guillet, Delfin: A deep learning based CSI fingerprinting indoor localization in IoT context, in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1-8.

[13] P. Ferrand, A. Decurninge, and M. Guillaud, DNN-based localization from channel estimates: Feature design and experimental results, in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1-6.

[14] C. Geng, H. Huang, and J. Langerman, Multipoint channel charting with multiple-input multiple-output convolutional autoencoder, in *2020 IEEE/ION Position, Location and Navigation Symposium*, PLANS 2020, 2020, pp. 1022-1028.

[15] E. Gönültaş, E. Lei, J. Langerman, H. Huang, and C. Studer, CSI-based multi-antenna and multi-point indoor positioning using probability fusion, *IEEE Transactions on Wireless Communications*, pp. 1-1, 2021.

[16] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[17] T. P. Hill, Conflations of Probability Distributions, *Transactions of the American Mathematical Society*, vol. 363, no. 06, pp. 3351-3351, aug 2008. [Online]. Available: http://arxiv.org/abs/0808.1808