COMST-00621-2022.R2

A Tutorial on the Characterisation and Modelling of Low Layer Functional Splits for Flexible Radio Access Networks in 5G and Beyond

Jordi Pérez-Romero, Oriol Sallent, Antoni Gelonch, Xavier Gelabert, Bleron Klaiqi, Marcus Kahn, David Campoy

***Abstract*— The centralization of baseband (BB) functions in a radio access network (RAN) towards data processing centres is receiving increasing interest as it enables the exploitation of resource pooling and statistical multiplexing gains among multiple cells, facilitates the introduction of collaborative techniques for different functions (e.g., interference coordination), and more efficiently handles the complex requirements of advanced features of the fifth generation (5G) new radio (NR) physical layer, such as the use of massive multiple input multiple output (MIMO). However, deciding the functional split (i.e., which BB functions are kept close to the radio units and which BB functions are centralized) embraces a trade-off between the centralization benefits and the fronthaul costs for carrying data between distributed antennas and data processing centres. Substantial research efforts have been made in standardization fora, research projects and studies to resolve this trade-off, which becomes more complicated when the choice of functional splits is dynamically achieved depending on the current conditions in the RAN. This paper presents a comprehensive tutorial on the characterisation, modelling and assessment of functional splits in a flexible RAN to establish a solid basis for the future development of algorithmic solutions of dynamic functional split optimisation in 5G and beyond systems. First, the paper explores the functional split approaches considered by different industrial fora, analysing their equivalences and differences in terminology. Second, the paper presents a harmonized analysis of the different BB functions at the physical layer and associated algorithmic solutions presented in the literature, assessing both the computational complexity and the associated performance. Based on this analysis, the paper presents a model for assessing the computational requirements and fronthaul bandwidth requirements of different functional splits. Last, the model is used to derive illustrative results that identify the major trade-offs that arise when selecting a functional split and the key elements that impact the requirements.**

***Index Terms*—5G, baseband (BB) functions, cloud radio access network (C-RAN), computational complexity, fronthaul, functional split, massive multiple input multiple output (MIMO).**

I. Introduction

To address the rapid increase in mobile data traffic, mobile network operators (MNOs) have extensively deployed radio access networks (RANs) with numerous base stations. Since the roll out of an RAN is costly, the evolution of mobile networks along successive generations 2G/3G/4G/5G has been driven by not only impressive technological advances and vast improvements in the achieved spectral efficiency over the air interface but also new RAN architectures.

A base station consists of a remote radio head (RRH), which performs all RF processing functionality (e.g., filtering and power amplification), and a baseband unit (BBU), which provides the remaining necessary signal processing functions (e.g., orthogonal frequency division multiple access (OFDMA) signal processing, channel coding, digital modulation, etc.) and upper layers of the radio interface protocol stack (e.g., medium access control, radio link control, etc.). In early generations, an RRH and a BBU were jointly placed at a cell site. Subsequently, the BBU is shifted from the cell site to a centralized location, which is often referred to as a BBU hotel. A BBU hotel pools the BBUs of multiple base stations and is typically located far (usually up to 20 km) from the RRHs. The RRHs are connected to the BBUs via the fronthaul (FH) links. Centralization has several benefits for MNOs, such as reduced space requirements at cell sites, reduced expenditure for cooling solutions at cell sites, easier test access and faster deployments [1].

Nevertheless, a centralized RAN offers more opportunities

Manuscript received December 22, 2022; revised April 28, 2023 and June 21, 2023. Accepted July 10, 2023. This work has been partially funded by Huawei Technologies. Work by X. Gelabert and B. Klaiqi is partially funded by the European Union's Horizon Europe research and innovation programme (HORIZON-MSCA-2021-DN-0) under the Marie Skłodowska-Curie grant agreement No 101073265. Work by J. Perez-Romero and O. Sallent is also partially funded by the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreements No. 101096034 (VERGE project) and No. 101097083 (BeGREEN project) and by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 under ARTIST project (ref. PID2020-115104RB-I00). This last project has also funded the work by D. Campoy. The expressed views and opinions are, however, those of the authors only and may not reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

J. Pérez-Romero, O. Sallent, A. Gelonch and D. Campoy are with the Dept. of Signal Theory and Communications at Universitat Politècnica de Catalunya (UPC), c/ Jordi Girona 1-3, Campus Nord, Barcelona, 08034, Spain. (e-mail: jordi.perez-romero@upc.edu, sallent@tsc.upc.edu, antoni.gelonch@upc.edu, david.campoy.garcia@estudiantat.upc.edu).

X. Gelabert, B. Klaiqi and M. Kahn are with Huawei Technologies Sweden AB, Kista, Sweden. (e-mail: xavier.gelabert@huawei.com, bleron.klaiqi@huawei.com, marcus.kahn@huawei.com)

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org

than simply jointly stacking the BBUs of different base stations. While a decentralized BBU at the cell site requires dimensioning the computing resources at the BBU according to the individual traffic peak load, thus wasting processing resources and power at idle times, the centralized approach exploits the resource pooling and statistical multiplexing gain among multiple cell sites, thus being much more efficient in both energy and cost. On the other hand, the centralization of BBUs also enables network virtualization, in which the BBU hotel can be replaced by a server built on open hardware such as x86/ARM CPU, leading to the cloud RAN (C-RAN) [2]. Thus, C-RAN is considerably different from traditional base stations that are built on proprietary hardware, where the software and hardware are close-sourced and provided by single vendors. In this way, C-RAN enables agility, flexibility and time-to-market acceleration of future network features, which can be implemented via software upgrades without implying hardware changes. If a process at any BBU can communicate with another process at any other BBU within the BBU pool, as they can be interconnected with very high bandwidth and low latency, collaborative radio technologies can be leveraged. For example, physical layer processing techniques such as coordinated multipoint (CoMP), in which multiple base stations jointly coordinate their transmissions to a given user equipment (UE), can be employed to mitigate interference [3], resulting in improved performance, especially for cell edge users. Centralization can more efficiently handle the advanced baseband computation needs to meet the complex requirements of new signal processing functions in the 5G new radio (NR) physical (PHY) layer, such as those related to the use of massive multiple input multiple output (MIMO).

There is, however, a trade-off between the centralization benefits and the fronthaul cost for carrying the radio data between distributed antennas and data processing centres. The required fronthaul bandwidth could be as high as 10 Gbps per radio cell of 20 MHz for fully centralized baseband processing [4]. On the other hand, the distance between RRHs and their controlling BBUs is constrained by the standardized round trip time (RTT) budget, which enables specifying response times and retransmission periods. Thus, this distance directly depends on the computing platform and the degree of software optimization of the RAN functions. This dependence calls for multiprocessor-based computing architectures, including multicore and multithreading architectures, to accelerate the processing time of radio frames and thus increase the distance between the BBU and the RRH.

Fig. 1 shows a flexible RAN architecture that provides the freedom to resolve the above trade-offs at any desired operation point. The approach consists of allocating certain computing capabilities close to the cell site, where the baseband low (BBL) functions are executed at a BBL platform, and certain computing capabilities at a centralized location, where the baseband high (BBH) functions are executed at a BBH platform. As more functions are moved towards the BBH platform, higher computing efficiency and higher coordination gain in radio resources can be achieved at the expense of higher

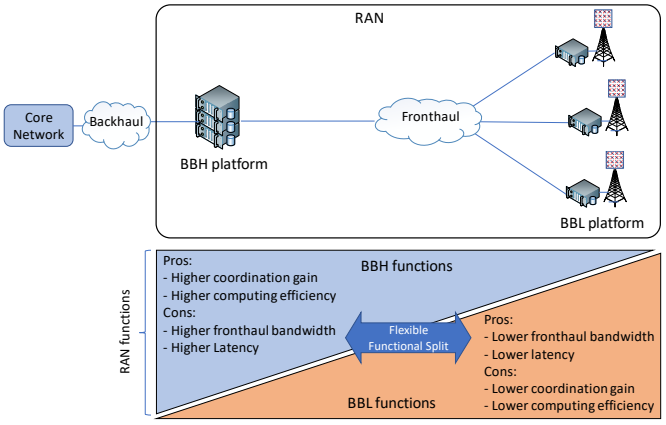


Fig. 1. Flexible functional split concept.

fronthaul bandwidth requirements and increased latency. Moving to functions towards the BBL platform leads to the opposite behaviour, i.e., lower computing efficiency and coordination gains but also lower fronthaul bandwidth and latency. This move results in a trade-off. In this context, fixing the functional split refers to deciding which radio functions are to be assigned to the BBL platform and which radio functions are to be executed at the BBH platform. Enabling the split of functions between BBL and BBH to be changed over time is referred to as a dynamic functional split. A dynamic functional split may therefore exploit the most appropriate split to satisfy a certain quality of service (QoS) for the offered services, such as a high data rate or low latency. Due to varying mobile traffic demand, the ability to dynamically select the optimal functional split is crucial for efficient usage of the fronthaul bandwidth and baseband processing resources.

Much research has been conducted in recent years to highlight the functional split problem, as reflected by some existing surveys that have summarized the key findings of different papers, research projects and industrial fora. A thorough description and analysis of different functional split options is presented in the survey [5], which uses as a reference the options investigated by the Third Generation Partnership Project (3GPP) in [6]. For each split option, this survey identifies references covering it, classified among theoretical surveys, references from simulations and references from practical experiments. Moreover, the split options are assessed in terms of their advantages and disadvantages and the implications on the bit rates and latency at the fronthaul. The survey [7] analysed different system architectures proposed by both industry and academia. These architectures were analysed based on the functional splits and the trade-off between implementation complexity and performance gains. Similarly, a comprehensive overview of C-RAN with optical fronthaul is presented in [8], discussing the split options considered by the 3GPP and providing an overview of some references for each case. Recently, a survey [9] presented a literature review of the functional splits proposed by the 3GPP and by O-RAN Alliance and overviewed the fronthaul requirements and implementation solutions.

The idea of a flexible functional split was introduced by the iJOIN project in [10][11][12] via the RAN as a Service (RANaaS) concept, which consists of partially centralizing functionalities of the RAN depending on the actual needs and network characteristics. The central element of RANaaS is the capability of providing a flexible and possibly dynamic functional split of the radio protocol stack between the central RANaaS platform (i.e., BBH in the context of this document) and the local radio access points (i.e., BBL). This capability introduces more degrees of freedom in processing design and flexibility in the actual execution of functions to adapt to the actual backhaul and access network characteristics by choosing an optimal operating point between full centralization and local execution. The flexible RAN functional split is also identified in [13] as one of the elements for developing scalable and flexible architectures for 5G. The survey [5], in addition to discussing this initial concept, also overviews some references that have proposed strategies for dynamical selection among a set of functional splits, considering both simulation-based approaches and practical implementations. A similar type of analysis for flexible functional splits is conducted in [8]. The paper identifies the trade-off between system performance (i.e., capacity, availability, and reliability) and performance of the fronthaul link (bit error rate, latency, etc.) that influences the hardware choices and cost. As a result, the paper concludes that the optimal balance is intricate to establish and dependent on both technical factors and commercial factors that are market- and operator-specific, and therefore, multiple models will probably be adopted and globally deployed in practice. In [14], the problem of dynamically selecting the appropriate functional split among three options is formulated as an integer lineal programming problem and solved by means of a heuristic algorithm. Similarly, [15] also investigated the optimization of the functional split by using a pure integer nonlinear programming model that chooses among three split options. In [16], a joint resource allocation problem that considers the selection of a functional split, the BBU server allocation and the scheduling policy to minimize the delay is formulated and solved. In [17], an adaptive RAN that can switch between two different centralization options at runtime without service interruption was presented, including some results from a specific implementation. Recently, the work in [18] proposed the dynamic adaptation of the functional split in accordance with the interference experienced by the user equipment, while in [19], the authors formulated the optimization of split selection by considering performance and operating cost and analysed different adaptation strategies. The impact of a flexible functional split on the fronthaul delay was investigated in [20] using queuing theory, while the fronthaul and backhaul requirements for four different functional splits were studied in [21] both in qualitative terms and in terms of data rate.

With the above, this paper attempts to fill several gaps identified in the open literature, namely,

1) The strong impact that C-RAN architectures and functional splits have on practical radio network deployments and MNOs' business has motivated the appearance of multiple standardization and industry initiatives. This impact has created a plethora of options, often following different terminologies, and there is a lack of compilation of such efforts in a comprehensive and homogeneous way. In Section II, the functional split approaches considered by different industrial fora are summarized and compared to identify the equivalences and harmonize the terminologies.
2) Considering that the functional split spans different layers of the radio interface protocol stack with particular relevance of the physical layer functions, there is a lack of a comprehensive analysis, that is, multiple papers propose specific algorithms for specific physical layer functions, sometimes accompanied by the corresponding complexity analysis. However, to the best of the authors' knowledge, there is a lack of papers that homogeneously provides a complete description of each of the involved PHY layer functions with the different algorithmic solutions and their complexity, which is fundamental for properly assessing a given functional split. To fill this research gap, a description of the different PHY layer functions in the transmission and reception chain of a 5G NR base station are presented in Section III, followed by an analysis of the computational complexity for the most demanding PHY layer functions in Section IV. This analysis compiles and harmonizes the results of multiple algorithmic solutions presented in the literature for each analysed PHY layer function.
3) To provide a comprehensive perspective on the functional split problem, the computational requirements at the BBL and BBH and the bit rate requirements at the fronthaul for different functional splits are presented in Section V. Moreover, the relevant system-level parameters that impact these requirements are identified.

By addressing the abovementioned research gaps, this tutorial paper contributes to the characterization and modelling of functional splits for C-RAN. For this purpose, the tutorial is organized in accordance with the skeleton and the objectives presented in Fig. 2 and includes four different parts. Part A, which is covered in Section II, provides an overview of the functional split options explored by different standardization and industrial fora, introducing the concepts of high layer splits and low layer splits, analysing the equivalences among existing options and comparing the terminologies of different fora. Focusing on the low layer split options, which involve splitting at different positions of the PHY layer processing, Part B of the tutorial, which is covered in Section III, presents the BB functions for the downlink (DL) transmission and uplink (UL) reception processing chains, establishing the basis for the possible splits between two functions and characterizing the inputs, outputs and processing conducted by each function. Part C, which is covered in Section IV, provides an overview of relevant state-of-the-art solutions for each BB function and presents a model of the computational complexity depending on the selected algorithm. This section also discusses the performance of different solutions by compiling and presenting

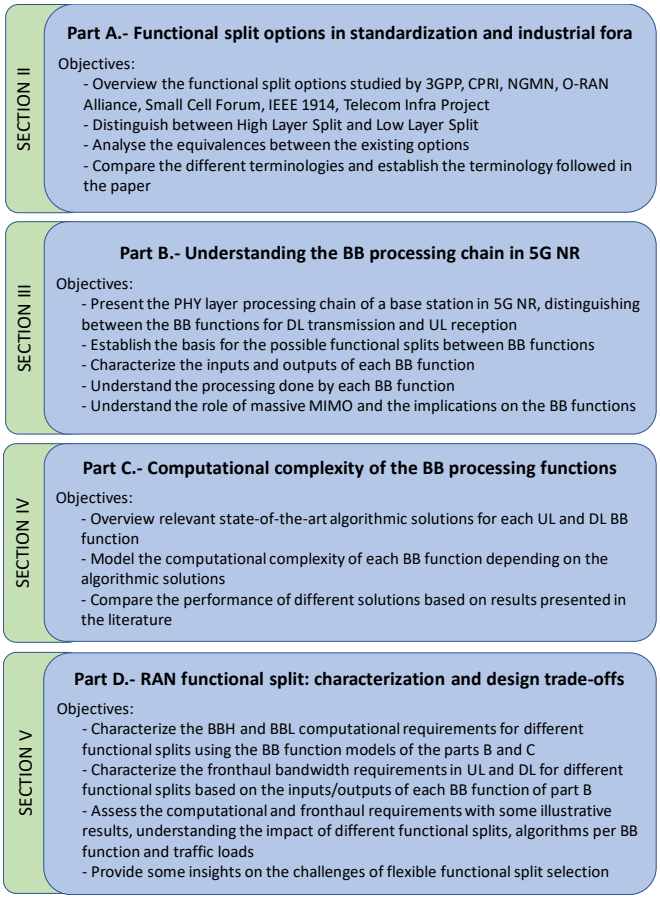


Fig. 2. Organization of the tutorial.

in a compact way results from various papers. Part D presented in Section V characterizes different functional splits in terms of the BBH and BBL computational requirements and the UL and DL fronthaul bandwidth requirements, utilizing the models that have been presented in Parts B and C. Furthermore, to establish a solid basis for the subsequent development of algorithmic solutions that are aimed at dynamic functional split optimization, an assessment of the model is also included in Section V. By introducing "best-case" and "worst-case" reference configurations and quantitatively assessing their performance for various functional split options, the relevance of the functional split choice is highlighted, thus stimulating the need for further research in this area. Section V also elaborates on research challenges related to RAN functional split optimization and discusses some forward visions in this area. Section VI concludes the paper by summarizing the main lessons learned.

In relation to previous surveys [5][8][9] that have also considered the functional splits, their focus and their survey nature make them fundamentally different from this tutorial paper, which targets a characterization, modelling and assessment exercise of functional splits rather than a comprehensive survey. In this respect, although the analysis of the functional splits proposed by industrial initiatives is also covered by these papers, none of them presents the analysis of the BB PHY layer functionalities and the associated complexity/performance for different algorithms. Similarly, they do not evaluate the impact of the BB functions on the BBH/BBL computational complexity requirements for different functional splits.

Table I presents the list of acronyms that are used throughout the paper.

TABLE I
LIST OF ACRONYMS USED IN THE PAPER

3GPP	3rd Generation Partnership Project
5G NR	Fifth Generation New Radio
A2B	Antenna-to-Beamspace
AAT	Antenna Array Theory
ADMA	Angle Division Multiple Access
ADMM	Alternating Direction Method of Multipliers
ALT	Approximate Lower Triangular
ANM	Atomic Norm Minimization
AS	Angular Spread
AU	Antenna Unit
A/D	Analogue-to-Digital
B2A	Beamspace-To-Antenna
BB	Baseband
BBH	Baseband High
BBL	Baseband Low
BBU	Baseband Unit
BEACHES	Beamspace Channel Estimators
BER	Bit Error Rate
BG	Base Graph
BLER	Block Error Rate
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
BS	Base Station
BSCE	Beam Space Channel Estimation
CCM	Channel Covariance Matrices
CD	Cholesky Decomposition
CG	Conjugate Gradient
CI	Chebyshev Iteration
CoMP	Coordinated MultiPoint
CP	Cyclic Prefix
CPRI	Common Public Radio Interface
C-RAN	Cloud RAN
CRC	Cyclic Redundancy Check
C-RNTI	Cell Radio Network Temporary Identity
CS	Compressive Sensing
CSI	Channel State Information
CU	Central Unit
D/A	Digital-to-Analogue
DFT	Discrete Fourier Transform
DL	Downlink
DM-RS	Demodulation Reference Signal
DOA	Direction of Arrival
DU	Distributed Unit
eCPRI	enhanced CPRI
eRE	enhanced Radio Equipment

eREC	enhanced Radio Equipment Control
EVD	EigenValue Decomposition
FAPI	Functional Application Protocol Interface
FDD	Frequency Division Duplex
FFT	Fast Fourier Transform
FH	Fronthaul
GS	Gauss–Seidel
HARQ	Hybrid Automatic Repeat reQuest
HDT	Hard Decision Threshold
HLS	High Layer Split
HS	Horizontal Shuffle
ICIC	InterCell Interference Coordination
IDFT	Inverse Discrete Fourier Transform
IDS	Informed Dynamic Scheduling
IFFT	Inverse Fast Fourier Transform
IP	Internet Protocol
IQ	Inphase and Quadrature
JI	Jacobi Iteration
LBP	Layered Belief Propagation
LDPC	Low Density Parity Check
LLR	Log-Likelihood Ratio
LLS	Low Layer Split
LoS	Line of Sight
LS	Least-square
LTE	Long Term Evolution
LWNS	Layered vicinal variable Node Scheduling
MAC	Medium Access Control
MF	Matched Filter
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
mMIMO	massive MIMO
MMSE	Minimum Mean Square Error
MNO	Mobile Network Operator
M-QAM	M-Quadrature Amplitude Modulation
MRT	Maximum Ratio Transmission
MSE	Mean Square Error
MOPS	Millions of Operations Per Second
MU-MIMO	Multi-User MIMO
nFAPI	network FAPI
NGFI	Next Generation Fronthaul Interface
NGMN	Next Generation Mobile Networks
NI	Newton Iteration
NSA	Neumann Series Approximation
O-CU-CP	O-RAN Central Unit-Control Plane
O-CU-UP	O-RAN Central Unit - User Plane
O-DU	O-RAN Distributed Unit
OFDMA	Orthogonal Frequency Division Multiple Access
OMP	Orthogonal Matching Pursuit
O-RU	O-RAN Radio Unit
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
PDCCH	Physical Downlink Control Channel
PDCP	Packet Data Convergence Protocol
PDMA	Path-Division Multiple Access
PDSCH	Physical Downlink Shared Channel
PHY	Physical
PRB	Physical Resource Block
PSK	Phase Shift Keying
PUCCH	Physical Uplink Control Channel
PUSCH	Physical Uplink Shared Channel
RAN	Radio Access Network
RANaaS	RAN as a Service
RAP	Radio Access Point
RB	Resource Block
RB-LBP	Residual-Based Layered Belief Propagation
RBP	Residual Belief Propagation
RE	Resource Element
RE	Radio Equipment
REC	Radio Equipment Control
RF	Radio Frequency
RI	Richardson
RLC	Radio Link Control
RRC	Radio Resource Control
RU	Radio Unit
RV	Redundancy Version
SBEM	Spatial Basis Expansion Model
SCF	Small Cell Forum
SD	Steepest Descent
SDBI	Soft Decision Bit Information
SDJC	Steepest Descent Jacobi
SNR	Signal to Noise Ratio
SOR	Successive Over-Relaxation
SRS	Sounding Reference Signal
SURE	Stein's Unbiased Risk Estimator
SVD	Singular Value Decomposition
TB	Transport Block
TDD	Time Division Duplex
TIP	Telecom Infra Project
TMA	Tridiagonal Matrix inversion Approximation
TPE	Truncated Polynomial Expansion
TRS	Tracking Reference Signal
TTI	Transmission Time Interval
UDP	User Datagram Protocol
UE	User Equipment
UL	UpLink
VLSI	Very Large Scale Integration
WG	Working Group
WeJi	Weighted Jacobi
ZF	Zero Forcing

II. Functional Split Options In Standardization and Industrial Fora

Different efforts have been conducted in the industry during recent years towards establishing standards for the disaggregation of cellular base stations, which involve the

functional split of the radio interface protocol stack. This section outlines some of the main actors' views on functional disaggregation, considered split options, and related literature towards the considered C-RAN implementation. Arguably, the most notable organizations addressing the standardization and specification of architectures supporting functional splits are, in no particular order, the Third Generation Partnership Project (3GPP) [22], Common Public Radio Interface (CPRI) cooperation [23], Next Generation Mobile Networks (NGMN) Alliance [24], Open RAN (O-RAN) Alliance [25], Small Cell Forum (SCF) [26], IEEE 1914 next generation fronthaul interface (NGFI) working group (WG) [27], and Telecom Infra Project (TIP) initiative [28]. The reader will soon realize that each organization introduces its own terminology and nomenclature in describing the problem at hand. We therefore conclude this section with a summary and harmonization of the main technical notions and terminology described below, which will enable a more cohesive and understandable delivery of concepts in subsequent sections. The main initiatives are summarized in this section.

A. Third Generation Partnership Project (3GPP)

As part of the study item intended to address the evolution of the radio interface and radio network architecture towards the 5G system, in [6], the 3GPP explored different options for carrying out the functional split of the radio interface between a gNodeB (gNB) central unit (CU) and a gNB distributed unit (DU). The involved layers of the radio interface protocol stack are the PHY layer, medium access control (MAC) layer, radio link control (RLC) layer, packet data convergence protocol (PDCP) layer and the radio resource control (RRC) layer. For details on the specific functionalities hosted at each of these layers, the reader is referred, for example, to Chapter 6 of [29].

Fig. 3 depicts the different 3GPP functional split options, which are specified by split points of the radio interface protocol stack (identified by dashed red lines) so that layers above these split points are hosted at the central unit and layers below this point are hosted at the distributed unit. The figure distinguishes the low layer split (LLS) options, which consider splits happening within the PHY layer and below the MAC layer, and the high layer split (HLS) options, in which the splits are defined above the MAC layer.

Among the LLS options, in option 8 (RF/PHY), the DU only hosts the RF processing and analogue-to-digital (A/D) or digital-to-analogue (D/A) conversions. In this case, raw inphase and quadrature (IQ) samples are transmitted over the fronthaul. The LLS options 7-1, 7-2, and 7-3 correspond to different split points in the PHY layer transmission/reception chain. In particular, with option 7-1, the distributed unit includes the OFDMA processing functions, i.e., fast Fourier transform (FFT) in the UL, inverse FFT (IFFT) in the DL and cyclic prefix (CP) insertion/removal, while the remainder of the PHY functions are centralized at the CU. In split option 7-2, the resource mapping/demapping and precoding functions are moved down to the DU, while in option 7-3, the DU hosts most of the PHY functions with the exception of downlink channel coding, which is centralized. In option 6 (MAC/PHY), the DU hosts the PHY and RF functionalities, while the MAC layers and above are at the CU.

Concerning the HLS options, option 5 (intra-MAC) considers that the MAC layer is split between a low-MAC sublayer that runs at the DU and includes the time critical functions (e.g., Hybrid Automatic Repeat reQuest (HARQ)) and a high-MAC sublayer at the CU handles functions such as centralized scheduling or intercell interference coordination (ICIC). In split option 4 (RLC/MAC), the distributed unit hosts the MAC/PHY/RF functions, while the central unit hosts the RLC/PDPC/RRC layers. In option 3 (intra RLC), the split is performed between a low RLC sublayer composed of segmentation functions and a high RLC that includes, among others, the retransmission functionality of the RLC layer. In option 2 (RLC/PDCP), the CU only hosts the PDCP and RRC functions. In option 1 (PDCP/RRC), the CU only includes the RRC layer, so it only contains control plane functionality, while all the processing of the user plane is performed at the DU.

During the technical specification phase of 5G NR in the 3GPP Rel. 15, the decision was made to standardize the high layer functional split between the PDCP and the RLC layer, which corresponds to split option 2 [30]. This split is supported by the F1 interface between the gNB-CU and the gNB-DU and remains the only split that has been standardized. The F1 interface supports the separation between the user plane function and control plane function within the CU through the split of the interface between the F1-U interface and F1-C interface.

The possibility of a low layer split between CU and DU was considered in a Rel. 15 study item whose outcomes are collected in [31]. The study considered the intra-PHY options 7-1, 7-2 and 7-3 with the MAC-PHY split option 6 and assessed them in terms of the required fronthaul bandwidth (some results are presented in [32]). The study item concluded that all identified low layer split options were technically feasible, but it could not converge on the selection of a single option. Then, it was decided that the preference for the 3GPP was to be open to all the identified low layer split options and even further to the variants thereof.

The survey [5] analysed the fronthaul bit rate requirements of the different split options. The highest requirements correspond to split option 8, in which the required bit rate is constant (i.e., independent to the actually occupied subcarriers that vary with the actual traffic) and scales with the number of antennas. Therefore, it is not very scalable for massive multiple-input multiple-output (mMIMO) scenarios [5]. A similar behaviour occurs with split option 7-1, in which the bit rate is also constant and scales with the number of antennas. However, the bit rate decreases with respect to split option 8 as now the IFFT/FFT and cyclic prefix insertion/removal are executed at the DU, and thus, the fronthaul only needs to transmit the symbols for each subcarrier instead of the time domain samples of the OFDMA symbols. Then, starting from split option 7-2 and in all the other splits, the resource element mapping is executed at the DU. Therefore, the fronthaul link only transports subframe symbols for the occupied subcarriers,

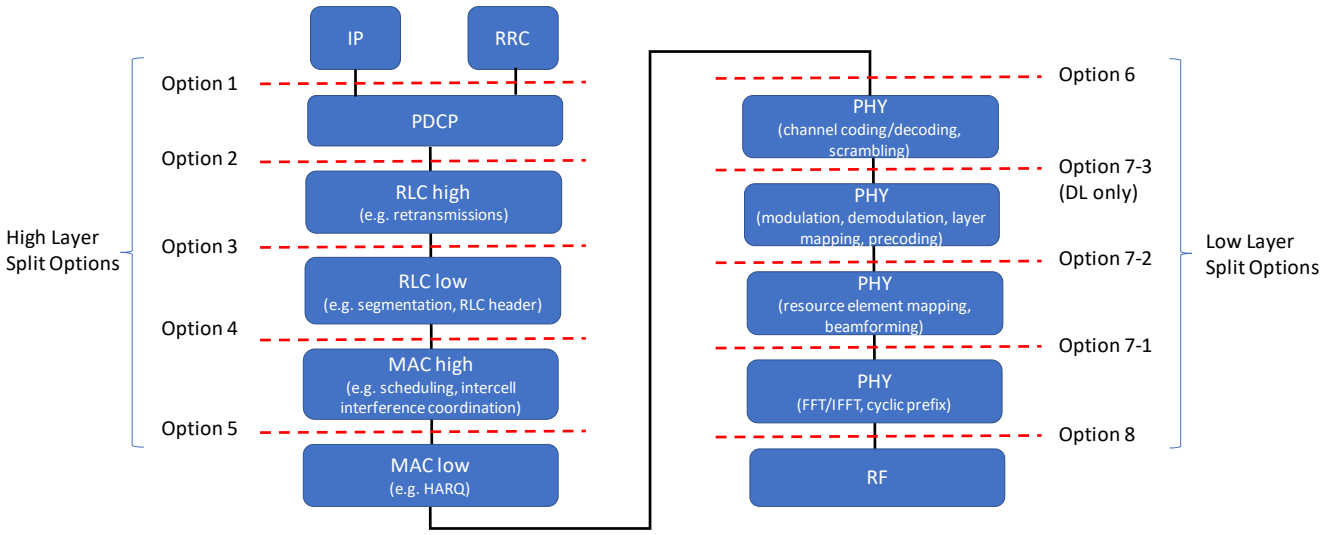


Fig. 3. Detailed functions of the split options proposed by the 3GPP. Split points are indicated by dashed red lines.

leading to a variable bit rate that depends on the actual traffic supported by the cell. Correspondingly, the required bit rate decreases with respect to previous options at the expense of allocating more complexity in the DU. The fronthaul bit rate progressively decreases when considering options that place more functionalities in the DU.

B. Common Public Radio Interface (CPRI)

The CPRI is an industry cooperation among Ericsson AB, Huawei Technologies Co. Ltd., NEC Corporation, Alcatel Lucent and Nokia Networks towards defining a publicly available specification for the internal interface of a radio base station between the radio equipment control (REC) and the radio equipment (RE), enabling independent technology evolution for the two and flexible and efficient product differentiation [33]. The RE hosts RF functions such as filtering, frequency conversion and amplification, so it is typically located near the antenna and corresponds to a RRH. The REC hosts the functions of the digital baseband domain. Therefore, the CPRI specification is a widely employed interface that can support the 3GPP functional split option 8 discussed in Section II.A enabling the transmission of IQ samples via the fronthaul. However, despite the benefits offered by the fully centralized C-RAN with split option 8, it is not cost-effective for 5G and beyond as it requires very high FH bandwidth (scales linearly with the number of antennas, which is very large in mMIMO), and no statistical multiplexing gains can be exploited. The CPRI specification defines the layer 1 and layer 2 protocols for the transfer of user plane, configuration and management and synchronization information between REC and RE, as well as between two REs. The user plane information is sent in the form of IQ data multiplexed by a time division multiplexing scheme onto an electrical or optical transmission line. The CPRI specification considers its use in GSM, UMTS, LTE and WiMAX standards, although the possibility of using it with other standards is not precluded. Some parts of the CPRI specification are left vendor proprietary, which renders the interoperability of equipment from different vendors challenging.

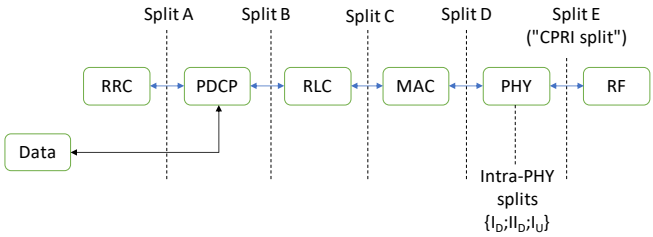


Fig. 4. Functional splits between eRE and eREC considered in eCPRI.

The CPRI specification evolved towards the enhanced CPRI (eCPRI), which is catered towards LTE and 5G NR standards by considering different possible functional splits between the enhanced radio equipment (eRE) and the enhanced radio equipment control (eREC) [34]. In this way, eCPRI provides more flexibility and cost-efficiency than CPRI, following a packet-oriented Ethernet-based design, and decreases the data rate demands in the fronthaul interface depending on the selected split option. Similar to the 3GPP, the functional split options identified by eCPRI are depicted and labelled in Fig. 4. The highest split is referred to as split A and is defined above the PDCP layer, so that the eREC hosts the layer 3 functions, i.e., the RRC and data (e.g., IP), while the eRE hosts the PDCP layers and below; thus, it is equivalent to split 1 of the 3GPP. Then, in split B, which is equivalent to split option 2 of the 3GPP, the PDCP layer is moved up to the eREC, while the RLC layers and below are at the eRE. Similarly, in split C, the eREC hosts the RLC and above functions, while the eRE hosts the MAC and below functions; thus, it is equivalent to split option 4 of the 3GPP. In split D, the eREC hosts the MAC layers and above, while the eRE includes the PHY and RF parts, which is thus equivalent to split option 6 of the 3GPP. Then, the main emphasis of eCPRI is placed on the intra-PHY split options, while Split E between the PHY and the RF is commonly referred to as the “CPRI” split. Concerning the intra-PHY split

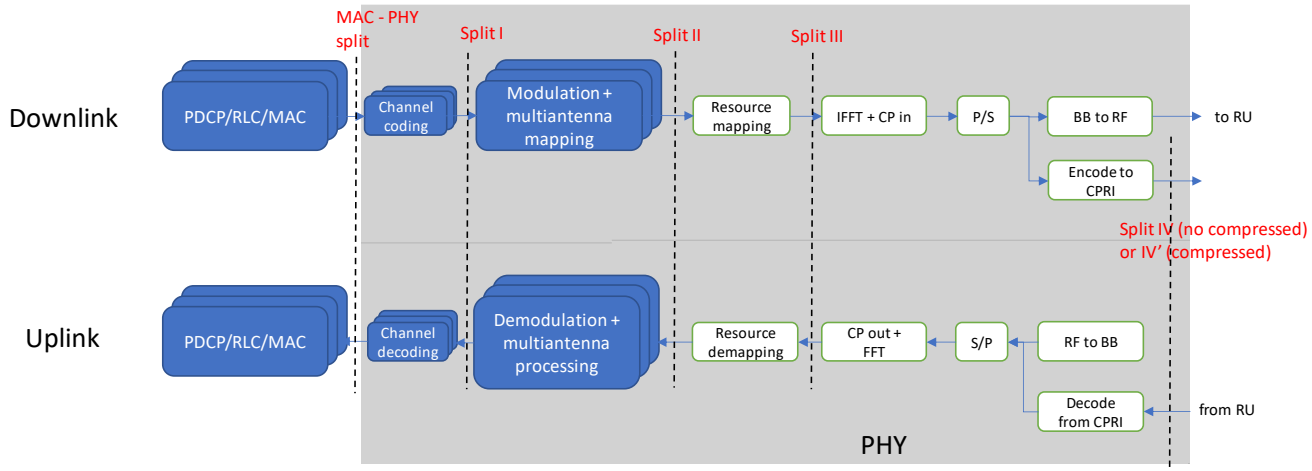


Fig. 5. Functional split options considered by NGMN.

options, the split I_D is defined only for the DL transmission in a way that the channel coding and scrambling functions are executed at the eREC while the modulation, layer mapping, precoding, resource element mapping and IFFT are executed at the eRE. The split option II_D moves the split down to the precoding so that the eRE hosts the resource element mapping and IFFT. This split option is for the DL, but there is an equivalent split for the UL reception, which is the I_U. While the I_D split is equivalent to option 7-3 of the3GPP, splits II_D and I_U correspond to option 7-2 of the 3GPP.

C. Next Generation Mobile Networks (NGMN)

In [35], the Next Generation Mobile Networks (NGMN) Alliance developed a document to guide the industry on the design of solutions for key technologies for C-RAN realization, namely, the functional split solutions for the fronthaul design, the efficient DU pool and the implementation of virtualization for C-RAN. In relation to the fronthaul design, different split options for the LTE MAC and PHY layer were analysed, as shown in Fig. 5.

The study of NGMN considers a fronthaul between a central DU and a distributed radio unit (RU), targeting the reduction of fronthaul bandwidth while keeping advanced C-RAN features such as the support of CoMP. Then, the considered split options include the MAC-PHY split, which is equivalent to split option 6 of the 3GPP, in which the MAC and above layers are centralized while the PHY layer is entirely distributed. Splits I, II and III are intra-PHY splits, while splits IV and IV' are two variants of the CPRI split PHY-RF without compression and with compression, respectively. Concerning the intra-PHY splits, in split I, the channel coding is centralized, while the modulation and multiantenna mapping and below functions are distributed. Split II also centralizes the modulation and multiantenna mapping functions, and split III centralizes all the functions above the IFFT/FFT.

The analysis of the different options in the NGMN study [35] was performed under high-latency and low-latency fronthaul scenarios. For the high-latency fronthaul scenario, no function split solutions are recommended, while for the low-latency case, the solution with CPRI compression (i.e., Split IV) is recommended from the data rate perspective. Split II is recommended from the perspective of enabling packet-based fronthaul networks. With this option, data can be encapsulated in the form of packets and transmitted using a packet switching protocol.

The NGMN "RAN Functional Split and X-Haul" project has also explored the 5G RAN functional decomposition in documents [36][37] with the target of understanding the various RAN functional splits and the transport requirements to support the different deployment options. In [36], the main focus was the HLS options, considering the 3GPP option 2 (PDCP/RLC) and analysing different deployment options in terms of the separation of control and user plane functions at the CU. Later, document [37] presented some updates regarding the HLS, including an analysis of the transport dimensioning and a discussion on security options. The transport dimensioning analysis suggests that to satisfy the requirements of 5G services, a typical transport interface may have to support 10 Gbit/s and possibly even 25 Gb/s per site. The document also provided an overview of LLS activities in different industry fora, acknowledging the existence of multiple options for LLS being developed in parallel and encouraging industry groups to ensure that fragmentation is avoided wherever possible. In relation to transport dimensioning for the LLS, the document in [37] analyses the specification work conducted by the xRAN/O-RAN Alliance and assesses the improvements with respect to the use of compressed CPRI in terms of throughput requirements.

D. Open RAN (O-RAN) Alliance

The O-RAN Alliance was formed in 2018 as a merger between xRAN Alliance and C-RAN Alliance and is aimed at evolving towards a more open and intelligent virtualized RAN with embedded artificial intelligence (AI)-powered radio control. The O-RAN architecture [38] considers the disaggregation of the RAN by splitting the gNB into an O-RAN Central Unit - Control Plane (O-CU-CP), an O-RAN Central Unit - User Plane (O-CU-UP), an O-RAN Distributed Unit (O-DU) and an O-RAN Radio Unit (O-RU). The O-CU-CP and O-CU-UP interact with the O-DU through the F1-C and F1-U

interfaces standardized by the 3GPP according to functional split option 2, which corresponds to a high layer split. The O-DU and O-RU interact through the Open FrontHaul (FH) interface that is specified by the O-RAN WG4 and corresponds to a low layer split of the radio interface protocol stack.

The Open FH interface encompasses the control, user and synchronization (CUS) plane, and its specifications are given in [39], being applicable for both 5G NR and LTE standards. The selected split point for this interface is referred to as option 7-2x and is a combination between option 7-1 and option 7-2 of the 3GPP. In comparison with split option 7-2, split 7-2x has a simplified interface and an open interface protocol specifically designed to enable interoperability between O-RUs and O-DUs from different vendors, and there is no complex timing for the O-RU and O-CU/O-DU link [40]. In particular, by placing resource element mapping at the O-DU as in option 7-2x, data will be transmitted after user multiplexing, thereby simplifying control signals on the fronthaul so that it becomes simpler to achieve multiprovider RAN [40][41][42].

According to the split option 7-2x, the O-RU hosts the A/D and D/A conversion, the IFFT/FFT processing, and CP addition/removal. Moreover, analogue and digital beamforming functions are also included. The O-DU hosts the channel coding, modulation and resource element mapping functions, in addition to the functions of the MAC and RLC layers. Regarding the MIMO precoding function, the split option 7-2x allows a variation in the position of this function to support two categories of O-RU equipment, namely, category A, which has less complexity and does not support precoding, and category B, which supports precoding. Then, for O-RU category A, precoding is carried out at the O-DU, while for O-RU category B, precoding is carried out at the O-RU. The Open FH interface supports the use of eCPRI or IEEE 1914.3 messages. They are encapsulated in Ethernet frames and optionally using IP/UDP.

E. Small Cell Forum (SCF)

The small cell forum (SCF) investigated in [43] split a small cell into two components, a central small cell where functions are virtualized and a remote small cell with nonvirtualized functions. Different functional splits, as depicted in Fig. 6 were analysed in terms of the bandwidth and latency requirements of the fronthaul link. The DL bandwidth requirements could range from approximately 150 Mb/s with the RRC-PDPC split option to approximately 2.4 Gb/s with PHY split IV, while the UL requirements ranged from approximately 50 Mb/s to 2.4 Gb/s. The study concluded that due to the wide range of possible small-cell deployment scenarios, a one-size-fits-all solution is very unlikely and that the fronthaul transport options available in a small-cell deployment are a major factor driving which option to select.

Among the different split options, SCF has paid special attention to the MAC/PHY split, for which it has defined open interfaces referred to as the functional application protocol interface (FAPI) and network functional application protocol interface (nFAPI). While FAPI assumes that MAC/PHY functions are at the same location so that FAPI becomes an

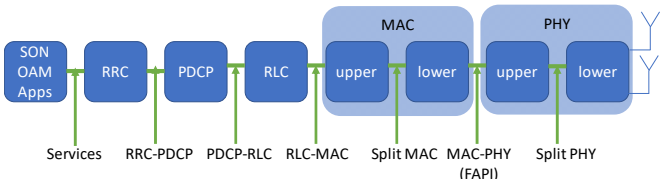


Fig. 6. Functional splits between central small cells and remote small cells in SCF.

internal interface within the eNB or gNB, nFAPI assumes that a packet switched network is used to support communication between the MAC and the PHY, located at different components. FAPI and nFAPI were initially defined for LTE and have been recently defined for 5G NR as well in documents [44] and [45], respectively.

F. IEEE 1914 - Next Generation Fronthaul Interface (NGFI)

The IEEE 1914 next generation fronthaul interface (NGFI) working group (WG) was created in February 2016 to explore NGFI-related key technologies, develop relevant standards and accelerate a mature NGFI ecosystem. Currently, there are two projects under this WG. The 1914.1 project describes the NGFI use case and develops the NGFI transport architecture as well as the requirements. The 1914.3 project is specifying the encapsulation format of the radio signal into Ethernet packets. The work of these two projects has resulted in the creation of standards [46][47].

The IEEE 1914.1 standard defines reference architectures for fronthaul, possible deployment scenarios covering both high- and low-layer functional splits and fronthaul requirements. Note that the standard complies with 3GPP-defined partitioning schemes but is not aimed at defining them. IEEE 1914.1 defines a two-level fronthaul architecture that considers two interfaces, namely, NGFI-I and NGFI-II. NGFI-I satisfies the low-layer functional split requirements, and NGFI-II satisfies the high-layer functional split requirements.

The IEEE 1914.3 standard defines the encapsulation and mapping of radio protocols for transport over Ethernet frames using radio over Ethernet, enabling the transfer of IQ user-plane data, vendor-specific data and control and management information. This standard provides structure-agnostic definitions for any digitized radio data and structure-aware definitions for the CPRI.

G. Telecom Infra Project (TIP)

The Telecom Infra Project (TIP) is an engineering-focused initiative driven by operators, suppliers, developers, integrators, and start-ups to disaggregate the traditional network deployment approach. A relevant TIP project in relation to functional splits is the "vRAN Fronthaul" project, which is aimed at developing an ecosystem for multivendor vRAN solutions with a focus on nonideal transport/fronthaul links [48][49]. The project is focused on the low layer split options related to splits 7-x in 3GPP (Section II.A), as they offer the most support for advanced RAN coordination features (e.g., all CoMP variants, ICIC, etc.) while maximizing the total cost of ownership gains from a reduction in radio complexity and

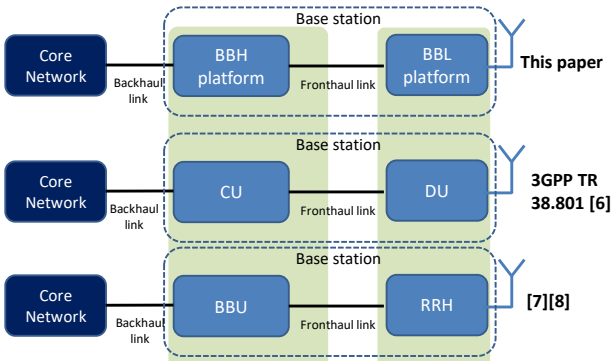
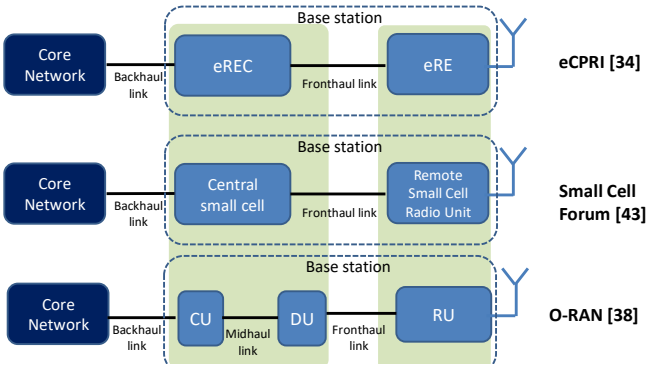


Fig. 7. Terminology for splitting the base station functions.



increased ability for resource pooling and load balancing.

Document [49] presents some trials obtained from the TIP community labs for option 7-2, with the objective of understanding how radio performance is impacted by fronthaul impairments.

Another relevant TIP project is the "OpenRAN" initiative to define and build 2G/3G/4G/5G RAN solutions based on general purpose vendor-neutral hardware, open interfaces and software. Document [50] from this project describes the technical specifications for a white Box 5G NR base station regarded as an open and disaggregated platform based on commercial off-the-shelf components and disaggregated software that can replace traditional proprietary RAN solutions. The platform considers the 3GPP compliant F1 interface for the high layer split between CU and DU and a low layer split between DU and RU based on split option 7-2x with eCPRI.

H. Summary of Terminologies and Equivalences

The analysis of the different initiatives and works related to functional splits for C-RAN reflects that different terminologies are used to refer to the entities that host the distributed/centralized base station functions. The lack of uniformity in terminology can be a source of misunderstandings and confusion and can sometimes raise concerns about whether specific terms may have different connotations when applied to similar contexts. Thus, researchers in this domain can be affected by the notion that when a given term has different meanings for different people, communication disintegrates and problems multiply. In an attempt to overcome these hurdles, a harmonized view is provided in this section.

Fig. 7 shows the terminology adopted in this paper (i.e., distributed functions run at the BBL hardware platform placed close to the antenna units and centralized functions run at the BBH hardware platform that is part of a central BBU execution environment) and compares it with other terminologies used in other references and fora. In particular, the BBL hardware platform of this paper corresponds in other references to the RRH (e.g., [7][8]), remote small cell radio unit (e.g., [43]), radio access point (RAP) (e.g., [10][11]), DU (e.g., [6]), RU (e.g., [38]) or eRE (e.g., [34]). The BBH hardware platform corresponds in other references to the BBU (e.g., [5][7][8][51]), CU (e.g., [6]), central small cell (e.g., [43]) or eREC (e.g., [34]). In addition, in some cases, the BBH platform can be further split into two parts interconnected via a midhaul link, such as the RU and DU of the O-RAN Alliance architecture [38].

TABLE II
CORRESPONDENCE BETWEEN FUNCTIONAL SPLIT OPTIONS DEFINED BY DIFFERENT FORA

3GPP [6]	eCPRI [34]	Small Cell Forum [43]	NGMN [35]
8	E	III_b	IV
7-1	-	III	III
7-2	II_D (DL), I_U (UL)	-	II
7-3	I_D	~ II	I
6	D	MAC/PHY	MAC-PHY
5	-	split MAC	-
4	C	RLC/MAC	-
3	-	-	-
2	B	PDCP/RLC	-
1	A	RRC/PDCP	-

Another element of differentiation among the studies carried out by the different industrial initiatives in relation to the C-RAN functional split is the terminology used to refer to the split options. Table II presents the equivalences among the options considered in different fora.

I. Summary of Lessons Learned

The main lessons learned in this section of the tutorial are summarized as follows:

- Different standardization bodies and industrial fora have analysed the functional split of the radio interface protocol stack for the disaggregation of cellular base stations. This tutorial has discussed the most relevant fora, namely, 3GPP, CPRI, NGMN, O-RAN Alliance, SCF, IEEE 1914 NGFI WG and TIP.
- In general, the different evaluated options fall into two main groups, namely, the low layer split options, which consider splits involving the PHY layer and up to the MAC layer, and the high layer split options, which consider splits defined above the MAC layer. While some fora, such as 3GPP, SCF or IEEE 1914 NGFI, have considered both high layer splits and low layer splits, other fora, such as O-RAN, NGMN or TIP, have focused on low layer splits.

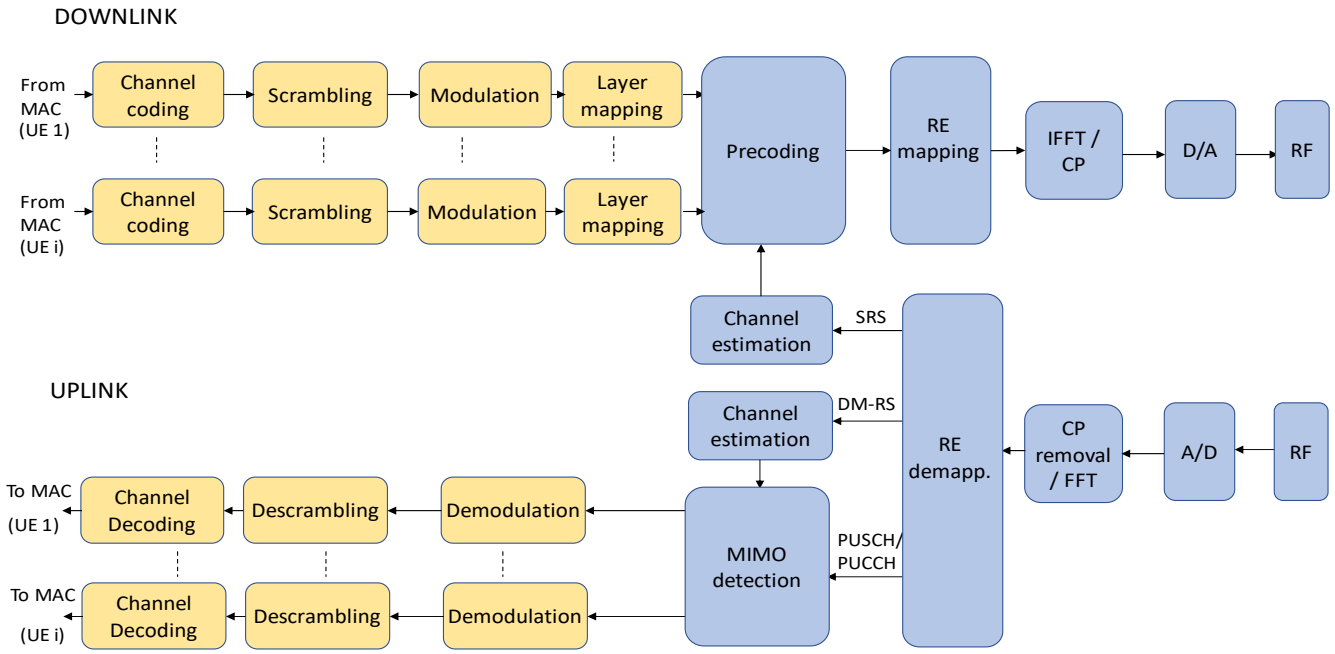


Fig. 8. Physical layer processing functions of the DL transmitter and the UL receiver of a gNB.

- The analysed fora have utilized different terminologies for the names of the split options and the entities hosting the centralized and distributed functions. The correspondences among the different options have been presented to provide a harmonized view.

III. Understanding the BB Processing Chain in 5G NR

This section presents an overview of the different baseband (BB) processing functions corresponding to the PHY layer of a base station (BS), i.e., a gNB in 5G NR, with the support of massive MIMO (mMIMO), which assumes B antennas at the base station and multiple UE devices that in total have U antennas, usually with $B>>U$ [53]. This scenario typically corresponds to the case of having U UE devices, each with a single antenna but can also represent other situations, e.g., having $U/2$ UE devices in the scenario, each with two antennas.

Fig. 8 plots the sequence of physical layer functions involved in DL transmission and UL reception for processing the user plane transport blocks delivered by the MAC layer for the physical downlink shared channel (PDSCH) and physical uplink shared channel (PUSCH) of many UE devices. The PDSCH transmits DL UE data and is reconfigurable via downlink control information and radio resource control (RRC). The PUSCH, on the other hand, is used to transmit UE data in the opposite direction, i.e., in the UL, and is accompanied by a demodulation–reference signal (DM-RS) that enables coherent demodulation. The rate at which transport blocks are delivered to the PHY layer is typically one per transmission time interval (TTI), i.e., one slot of 14 symbols or a mini-slot that includes a few symbols of a slot. Fig. assumes that OFDMA is used in both DL and UL. Note that in the case that discrete Fourier transform (DFT)-precoded OFDMA was employed in the UL, which is a technique that mainly targets coverage-challenged scenarios, the main difference would be the inclusion of an inverse discrete Fourier transform (IDFT) process block in the UL reception chain prior to demodulation.

This paper uses as a reference for the modelling of the BB functions the use of the time division duplex (TDD) technique, in which the UL and DL operate on the same channel at different times, as utilized in some of the most significant bands for 5G NR deployments (e.g., 3.5 GHz or mmWave bands). With the TDD, it can be assumed that there exists UL-DL channel reciprocity, so that the DL channel can be estimated using the reference signals sent in the UL (i.e., sounding reference signals (SRSs), as shown in Fig. 8). SRSs are periodically transmitted in UL over a large bandwidth and are used to support the precoding and scheduling of UE devices. Note that, in the case of the frequency division duplex (FDD), a similar modelling could be conducted. In this case, the main difference would be that the DL channel estimation would be done at the UE, which would provide channel state information (CSI) reports to be used when selecting the DL precoder. Unlike the TDD mode, FDD channel reciprocity cannot be applied, leading to very high uplink feedback overhead to obtain DL CSI. This overhead per UE linearly grows with the number of antennas.

Fig. 8 reflects that some of the PHY layer functions, which are depicted in yellow, are executed on a per UE basis. Other functions, e.g., RE mapping, consider the symbols obtained after the processing of transport blocks from numerous UE.

The next two subsections describe in a summarized tutorial style the processing functions of DL transmission and UL reception. An enlightening description of the different functions is provided, emphasizing the main concepts associated with certain complex signal processing principles embedded in 5G NR. For the interested reader, detailed descriptions of the

various BB functions are provided in more specialized literature, such as [29][54], the 3GPP specifications of the TS 38.21X series [55][56][57] or the O-RAN specifications [39]. In this respect, each subsection below includes detailed references for the corresponding BB function.

A. DL Processing

1) Channel coding

This set of processes is in charge of generating the redundancy bits for each transport block to facilitate the detection and correction of errors at the receiver side. In the case of the PDSCH, low-density parity-check (LDPC) codes are employed. LDPC coding is separately applied to different segments of a transport block, which are denoted as "code blocks" (CBs). In this way, in the case of errors, it is not necessary to retransmit the whole transport block but only the erroneous code blocks[1]. Then, the overall channel coding process of a transport block involves different steps, as illustrated in Fig. 9 based on [29]. At the output of the process, a coded transport block, which is also referred to as codeword, is obtained.

The first step in the channel coding is a cyclic redundancy check (CRC) attachment, which consists of calculating and adding a CRC for each transport block. This step is performed to detect errors in the decoding process. The size of the CRC depends on the transport block size; it is 24 bits for transport blocks larger than 3824 bits and 16 bits otherwise. Details of the CRC attachment are given in the TS 38.212 specification; refer to Section 5.1 of [56].

When the transport block is larger than the maximum code block size of the LDPC encoder, it is segmented into multiple equal-sized code blocks, as shown in the second step of Fig. 9. The maximum code block size of the encoder is 8424 bits for base graph 1 and 3840 bits for base graph 2, where the base graph (BG) represents the parity-check matrix employed by the LDPC encoder. During this process, an additional CRC of 24 bits is also appended to each code block (third step of Fig. 9). This CRC is used to detect errors at the code block level. Details of this segmentation process and CRC attachment per code block are specified in Section 5.2 of [56].

The fourth step is LDPC coding, which is applied to each code block after the segmentation process. The basis for LDPC is a sparse parity-check matrix as sparseness simplifies the decoding process. Quasicyclic LDPC codes with a dual-diagonal structure of the kernel part of the parity check matrix are utilized in 5G NR, which gives a decoding complexity that is linear with the number of coded bits. Moreover, the LDPC codes used in 5G NR are systematic, meaning that, after coding, a code block will consist of the original bits followed by parity or redundancy bits.

The parity-check matrix of the LDPC code, which determines how the redundancy bits are generated, is represented by a graph. Two base matrices are defined and referred to as base graphs BG1 and BG2, which enables the

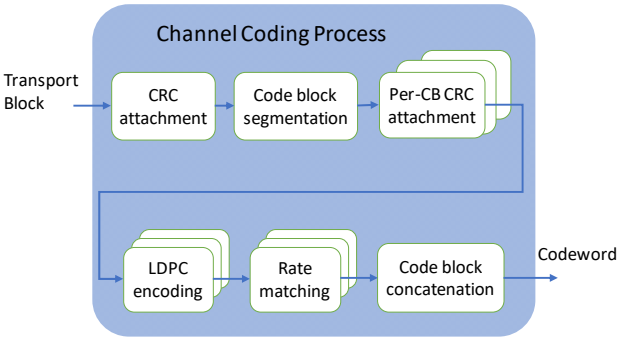


Fig. 9. Channel coding processes (based on [29]).

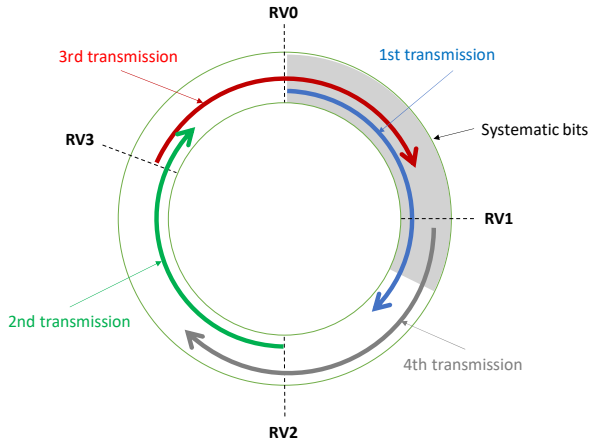


Fig. 10. Example of how the bits of each RV of a code block are determined.

efficient handling of a wide range of payload sizes and code rates. BG1 is designed for code rates ranging from 1/3 to 22/24, and BG2 ranges from 1/5 to 5/6. The choice between BG1 and BG2 depends on the transport block size and code rate targeted for the first transmission. The specific parity-check matrix to be used for coding a code block with a given size is obtained from the selected base graph matrix after applying a lifting process dependent on the size of the code block. All the details of the channel coding process are provided in Section 5.3 of [56].

The rate matching process is applied to each code block after the coding process to extract a suitable number of coded bits to match the resources assigned for transmission (fifth step of Fig. 9). The different redundancy versions (RVs) of the code block are generated. These RVs will be employed in the subsequent retransmissions of the code block performed by the HARQ process in the case of errors.

The rate matching starts by puncturing a fixed number of the systematic bits of a code block. The number of punctured bits can be relatively high, up to 1/3 of the systematic bits. Then, the remaining bits are written into a circular buffer, starting with nonpunctured systematic bits and continuing with parity bits. The selection of bits to transmit is based on reading the required number of bits from the circular buffer where the exact set of

[1] The retransmissions are performed at the "code block group" level, i.e., when there are some erroneous code blocks inside a group of code blocks, the whole group is retransmitted. Thus, the HARQ feedback does not have to be provided individually per code block and instead it is sent per group.

bits to transmit is dependent on the RV, which corresponds to different starting positions in the circular buffer. This point is illustrated in Fig. 10 based on [29].

The bits of an RV are interleaved. For this purpose, the bits obtained from the circular buffer are written row-by-row into the block interleaver and read column-by column. The number of rows is the modulation order, so that each column includes the bits of a modulation symbol. The details of the rate matching and interleaving process are provided in Section 5.4 of [56].

After interleaving, the bits of each code block are sequentially concatenated to form the sequence of bits that represent the coded transport block, which is referred to as the codeword. This step is detailed in Section 5.5 of [56].

2) Scrambling

This function uses as input each codeword resulting from the channel coding processes and multiplies it by a bit-level scrambling sequence that depends on the identity of the UE, that is, the C-RNTI (Cell Radio Network Temporary Identity), and a data scrambling identity configured in each UE. This process is needed at the receiver side to properly distinguish the useful signal from an interfering signal at the same frequency. Without scrambling, the channel decoder at the receiver could be equally matched to an interfering signal as to the useful signal, thus being unable to properly suppress the interference. By applying different scrambling sequences for the useful and interfering transmissions, the interfering signals after descrambling are randomized, ensuring full utilization of the processing gain provided by the channel code [29]. The details of the scrambling process for the PDSCH are provided in Section 7.3.1.1 of [55].

3) Modulation

This function maps the bits of the scrambled codeword into complex modulation symbols in accordance with the modulation scheme that has been selected by scheduling at the MAC layer. Supported modulation schemes in the DL for the current release 17 of 5G NR are QPSK, 16QAM, 64QAM and 256QAM; these schemes correspond to 2, 4, 6 and 8 bits per symbol, respectively. 1024QAM, which corresponds to 10 bits per symbol, has been recently included as part of release 18, whose standardization is ongoing. Higher-order modulations such as 256QAM or 1024QAM enable increased spectral efficiency whenever the channel conditions are good, while low-order modulations such as QPSK enable operation with poorer channel conditions at the expense of lower spectral efficiency. The selection of the modulation is dynamically performed by a link adaptation algorithm that jointly chooses the modulation and channel coding rate in accordance with the experienced channel conditions. This selection is indicated in the modulation and coding scheme (MCS), which specifies both the modulation and channel coding rate. The details of the modulation process for the PDSCH are provided in Section 7.3.1.2 of [55], while the list of possible MCSs is given in Section 5.1.3 of [57].

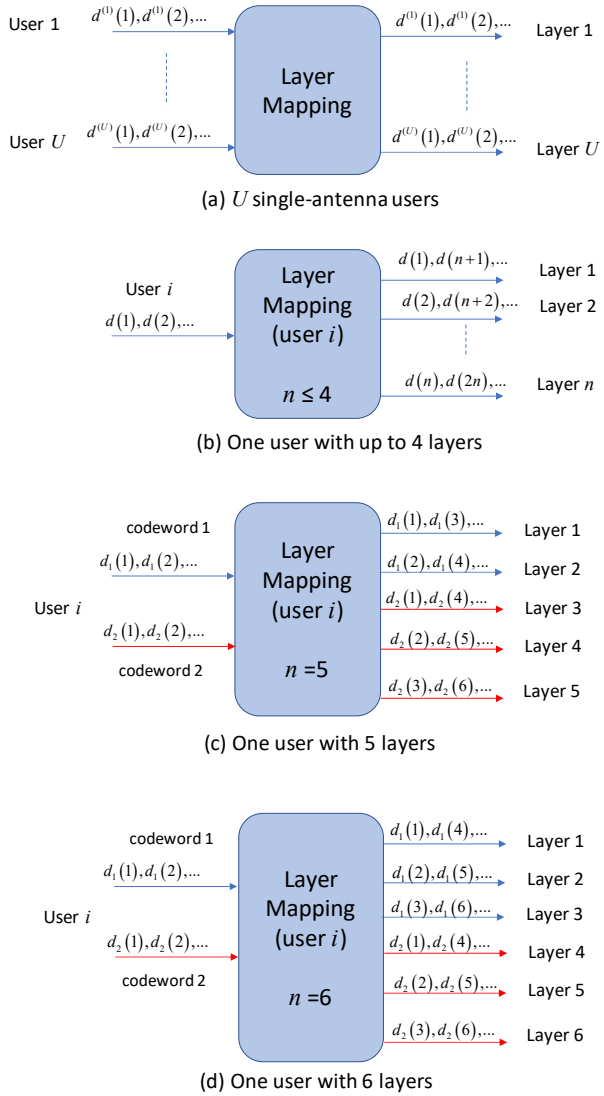


Fig. 11. Layer mapping.

4) Layer Mapping

This function is executed in the case of spatial multiplexing and consists of determining which modulation symbols are sent through each spatial layer. In the case of mMIMO with single antenna UE, each spatial layer corresponds to a different UE, so the layer mapping is straightforward and simply maps the symbols of the *i*-th user to the *i*-th layer (Fig. 11a). In contrast, when there are UE with more than one antenna, they can use multiple layers. The number of layers for each UE is decided by the MAC scheduler, taking into account the channel conditions experienced by the UE, the amount of information to transmit, and the UE capabilities. In this case, the way to map the symbols to the spatial layers up to a maximum of 8 layers is standardized in 3GPP TS 38.211. The specific details of this mapping are provided in Section 7.3.1.3 of [55] and are summarized in this section.

When the number of layers for a UE is lower than or equal to 4, only one transport block is processed per TTI, and the layer mapping consists of mapping every *n*-th symbol to the *n*-th layer. This process is illustrated in Fig. 11b for a given *i*-th UE.

For example, if there are 3 layers, the sequence of symbols $\{d(1), d(2), d(3), d(4), d(5), d(6)\}$ is mapped so that $\{d(1), d(4)\}$ are sent to layer 1, $\{d(2), d(5)\}$ to layer 2 and $\{d(3), d(6)\}$ to layer 3.

When the number of layers is higher than 4, there will be two transport blocks per TTI, and thus, the modulation symbols will belong to two codewords. In this case, the layer mapping distributes the symbols of both codewords across the different layers. For example, Fig. 11c illustrates the case of 5 layers. In this case, the symbols of the first codeword are sent through layers 1 and 2, and the symbols of the second codeword are sent through layers 3, 4 and 5. Similarly, the case of 6 layers is illustrated in Fig. 11d. The symbols of the first codeword are sent through layers 1 to 3, and the symbols of the second codeword are sent through layers 4 to 6. Using 7 layers is similar to the use of 5 layers, so that the first codeword is sent through layers 1, 2, and 3 and the second codeword is sent through the remaining four layers. Using 8 layers, which is the maximum number, is similar to the use of 6 layers, but with each codeword sent through four layers.

5) Multiantenna Precoding

This process maps the symbols of each layer to the different antennas (i.e., antenna ports following 3GPP terminology[2]). It is assumed that, from the baseband perspective, each antenna is associated with one transmitter/receiver (TRX), where a TRX refers to the processing chain associated with a D/A or A/D converter [39].

We assume an mMIMO system with B antennas at the BS and a total of U ($<<B$) antennas among all the UE devices so that there are a total of U layers to be transmitted. The vector with the symbols of each layer is a $U\times1$ vector denoted as $\mathbf{x}$. Then, the precoding consists of multiplying vector $\mathbf{x}$ with a precoding matrix $\mathbf{W}$ containing B rows and U columns, as illustrated in Fig. 12. The result is the $B\times1$ vector $\mathbf{y}$ with the symbols that are sent through each antenna. The precoding matrix $\mathbf{W}$ depends on the channel matrix $\mathbf{H}$ that includes the channel coefficients between each BS transmit antenna and each UE antenna (i.e., $B\times U$ matrix). Therefore, the precoding process involves a specific processing function for computing the appropriate precoding matrix $\mathbf{W}$. Assuming UL-DL channel reciprocity, this function is fed by the DL channel estimation performed using the UL SRS signals, which will obtain an estimate of the matrix $\mathbf{H}$ denoted as $\hat{\mathbf{H}}$.

Note that in the downlink, the DM-RS, utilized by the receiver at the UE for channel estimation, is subject to the same precoding matrix $\mathbf{W}$ as the PDSCH. In this way, when the receiver at the UE estimates the channel based on the received DM-RS, the estimation will consider the joint effect of the channel (e.g., propagation effects) and the precoding done at the transmitter. The precoding is not explicitly visible to the receiver but is considered part of the overall channel effects [29]. Additionally, as a result, the receiver at the UE does not need to explicitly know what precoding matrix is being employed at the base station as it will be able to estimate the symbols sent in each layer based on the channel estimation conducted using the DM-RS. For this reason, precoding is not fully specified in the standard, although it is fully expected to be present in practice [59].

The precoding process can be employed for different purposes, such as beamforming or spatial multiplexing. Fig. 13 illustrates an example of precoding applied for digital

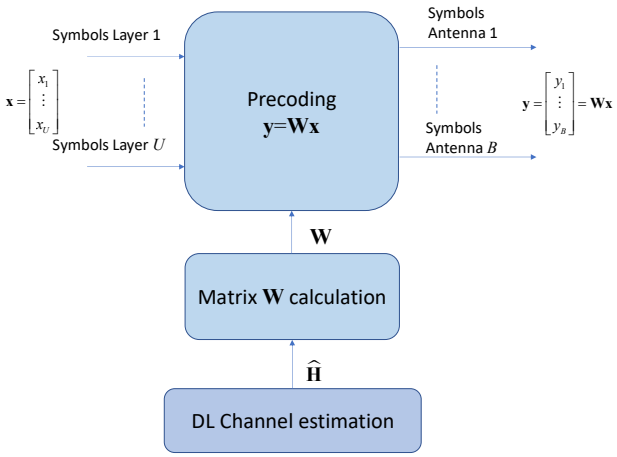


Fig. 12. Multiantenna precoding.

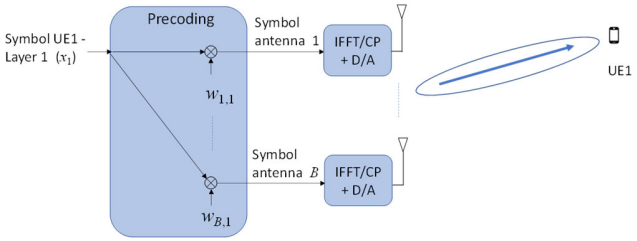


Fig. 13. Example of precoding applied for digital beamforming.

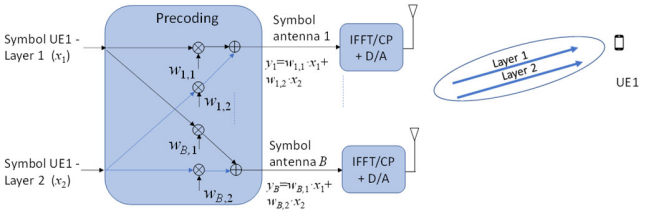


Fig. 14. Example of precoding applied for spatial multiplexing for a single UE.

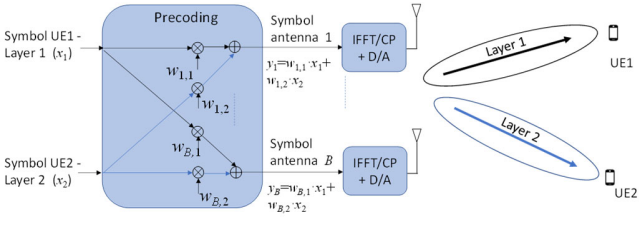


Fig. 15. Example of precoding for spatial multiplexing with MU-MIMO.

[2] In 3GPP terminology, the antenna port is a logical concept defined such that the channel over which a symbol of the antenna port is conveyed can be inferred from the channel over which another symbol on the same antenna port is conveyed. Each individual downlink transmission is carried out from a specific antenna port, the identity of which is known to the device [29].

beamforming by mapping one symbol of one layer to B antennas by means of vector $\mathbf{W}=[w_{1,1},..., w_{B,1}]^T$. As a result of the process, a beam is created to transmit the signal to UE1, which corresponds to a single layer transmission using beamforming.

Fig. 14 presents an example of spatial multiplexing for a single UE. In this case, UE1 has two layers transmitted by B antennas after the precoding process. To support this example, the receiver at UE1 must have at least 2 antennas; otherwise, it would not be possible to send more than one layer to this UE. The precoding matrix $\mathbf{W}$ is a $B\times2$ matrix, and the symbol transmitted by each antenna is a linear combination of the symbols of each layer.

Spatial multiplexing can also be achieved for multiple UE devices using multiuser MIMO (MU-MIMO). An example is shown in Fig. 15. Formally, the precoding process is similar to the previous example, but now the symbols of each layer are addressed to different UE devices. As a result of the process, two different beams are created: one beam for UE1 and another beam for UE2.

6) Resource Element Mapping

This process uses as input the modulation symbols to be transmitted on each antenna and maps them to the set of available resource elements (REs) of the resource blocks (RBs) assigned by the MAC scheduler for the transmission of the considered transport block. One RE corresponds to one subcarrier in the frequency domain and one symbol in the time domain, while one RB corresponds to 12 contiguous subcarriers in the frequency domain.

The mapping process takes into account that some of the REs are employed for reference signals (i.e., DM-RS, channel state information reference signals, tracking reference signals and phase-tracking reference signals), synchronization signals or DL reserved resources.

The inputs of the RE mapping are obtained from all the transport blocks and physical channels (e.g., PDSCH, Physical Downlink Control Channel (PDCCH), etc.), although this does not prevent the mapping from being separately conducted for each transport channel, as the RBs available to each channel are defined by the MAC scheduling. As an output of the process, the symbols per subcarrier to be transmitted over each antenna are obtained.

The time-frequency resources to be utilized for transmission are signalled by the scheduler as a set of virtual RBs and a set of OFDMA symbols. The RE mapping maps the symbols to these resources in a frequency-first, time-second manner. Then, the virtual RBs are mapped to physical resource blocks (PRBs) in the bandwidth part used for transmission. This mapping can be achieved in either a noninterleaved way or an interleaved way [29]. Noninterleaved mapping means that a virtual RB directly maps to a PRB which is useful when the scheduler allocates transmissions to physical RBs with good channel conditions. Interleaved mapping means that virtual RBs are mapped to physical RBs using an interleaver spanning the whole bandwidth part and operating on pairs or quadruplets of RBs. The benefit of interleaved mapping is to achieve frequency diversity. Chapter 9.9 of [29] provides further details on the RE mapping process.

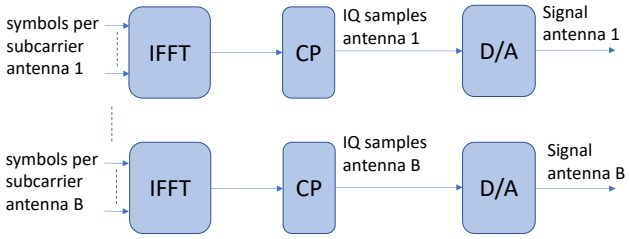


Fig. 16. IFFT, cyclic prefix insertion and D/A conversion.

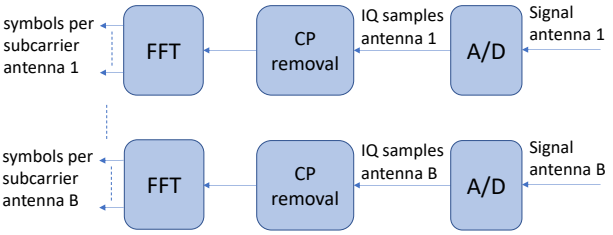


Fig. 17. A/D conversion, cyclic prefix removal and FFT.

7) IFFT, Cyclic Prefix Insertion and Digital-to-Analogue Conversion

This processing is applied for each antenna. The symbols to be transmitted in each subcarrier are used as input, and the IFFT is applied to determine the time domain IQ samples of the OFDMA symbol that is obtained as a combination of all the individual symbols per subcarrier. After this process, the cyclic prefix is inserted by adding at the beginning of the symbol the last N_{CP} time samples of the OFDMA symbol resulting from the IFFT. The digital IQ samples are converted to the analogue signal by means of a D/A converter. The overall process is shown in Fig. 16.

B. UL Processing

1) Analogue-to-Digital Conversion, Cyclic Prefix Removal and FFT

These functions are executed for the signal received in each antenna of the base station. After executing the A/D conversion to obtain the time-domain digital IQ samples of each received OFDMA symbol, the time samples corresponding to the cyclic prefix are removed, and an FFT is executed over the remaining samples of the OFDMA symbol. The outputs of this FFT will be the complex symbols received in each of the subcarriers (i.e., resource elements) for each considered antenna. The overall process is shown in Fig. 17.

2) RE Demapping

This function determines what symbol is received in each physical channel based on the REs utilized by this channel and the output of the FFT process. This process allows separation of the different physical channels of the different users, e.g., PUSCH, Physical Uplink Control CHannel (PUCCH), etc., and

the corresponding reference signals to be used by the channel estimation (i.e., DM-RS and SRS).

3) Massive MIMO Detection

This function estimates the symbols sent through the U uplink antennas of the UE devices. The case of U single antenna UE devices is illustrated in Fig. 18. To characterize the mMIMO detection process at the base station, we denote as $\mathbf{s}$ the $U\times1$ vector with the symbols $s_1,...,s_U$ transmitted in each one of the U transmit antennas in a given subcarrier and denote by $\mathbf{H}$ the $B\times U$ channel matrix with the complex channel coefficients for each pair of transmit and receive antennas, where $h_{b,u}$ denotes the channel between the u-th transmit antenna and the b-th receive antenna. Then, the $B\times1$ vector $\mathbf{y}$ with the symbols $y_1,...,y_U$ received in each of the B antennas at the BS is given by

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

where $\mathbf{n}$ is the $B\times1$ vector with the noise power in each receive antenna. As illustrated in Fig. 18, mMIMO detection consists of obtaining estimation $\tilde{\mathbf{s}}$ of the transmitted symbols $\mathbf{s}$ by utilizing the received symbols $\mathbf{y}$ and an estimation of the channel matrix, denoted as $\hat{\mathbf{H}}$, provided by the channel estimation function. Different algorithms exist for mMIMO detection. Section IV.A.1 elaborates on some of these algorithms, emphasizing the perspective of computational complexity and performance.

Note that the mMIMO detection process determines the symbols sent by the different users in a given subcarrier. Therefore, the same process has to be executed for all the involved subcarriers, each with its corresponding channel estimation. In this case, a given channel estimation can be valid for multiple subcarriers within the channel coherence bandwidth, which may be used to decrease the overall computational complexity of the channel estimation block.

4) Channel Estimation

This function is in charge of estimating the uplink channel matrix $\mathbf{H}$, which is needed to support the mMIMO detection process, as discussed in the previous Section III.B.3. On the other hand, assuming a TDD system with UL-DL channel reciprocity, channel estimation using UL signals can also be employed to support the DL precoding process, as discussed in Section III.A.5.

The most widely used channel estimation techniques rely on the use of known training sequences of pilot symbols, i.e., the reference signals transmitted by the UE in the UL [60]. These are the techniques that will be considered in this paper. However, there are other techniques, such as blind channel estimation, which only uses the received data symbols, or semiblind techniques, which use both the received data symbols and pilot symbols, thus facilitating a reduction in the number of pilots to be sent [61].

For modelling purposes, we denote as $\mathbf{P}$ the matrix with the known symbols of the training sequence for the different UE devices. Matrix $\mathbf{P}$ is a $U\times L$ matrix, where each row corresponds

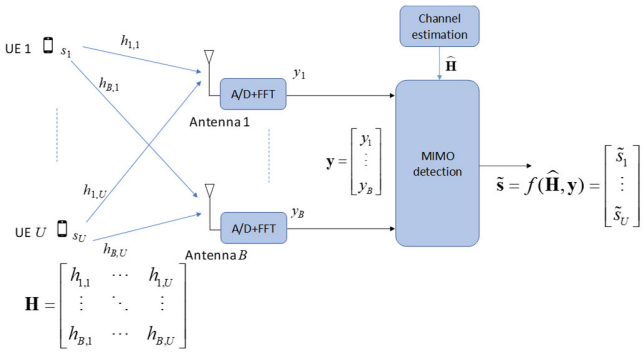


Fig. 18. Massive MIMO detection.

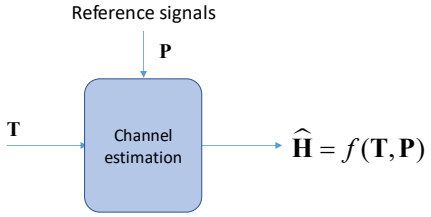


Fig. 19. Channel estimation for massive MIMO.

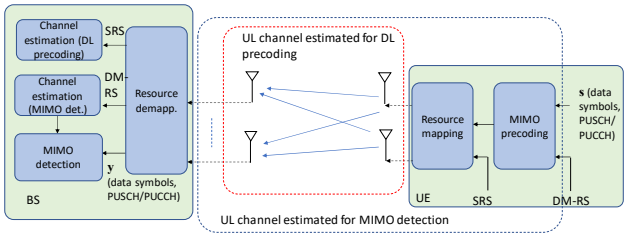


Fig. 20. Uplink channel estimation.

to the training sequence of one UE and L is the length of this sequence. Assuming a $B\times U$ channel matrix $\mathbf{H}$, the $B\times L$ matrix $\mathbf{T}$ with the received symbols of each training sequence in the different antennas is given by

$$\mathbf{T}=\mathbf{H}\mathbf{P}+\mathbf{N}, \tag{2}$$

where $\mathbf{N}$ is another $B\times L$ matrix that represents noise. Then, the channel estimation process consists of determining the estimation $\hat{\mathbf{H}}$ of the channel matrix using the received symbols in matrix $\mathbf{T}$ and the known transmitted symbols of matrix $\mathbf{P}$, as shown in Fig. 19.

Concerning the reference signals used for the estimation in a TDD system with UL-DL reciprocity, there is a difference between the channel estimation used for MIMO detection and the estimation performed for downlink precoding. As shown in Fig. 8, it is assumed that in the channel estimation for MIMO detection, matrix $\mathbf{P}$ contains the uplink DM-RS, while in the channel estimation for downlink precoding, matrix $\mathbf{P}$ contains the uplink SRS. The reason for this selection is that the uplink DM-RS is affected by the uplink precoding matrix, similar to the remainder of the data symbols $\mathbf{s}$ sent by the UE, as depicted in the example of Fig. 20, which shows the case of a UE with two antennas sending two layers in the uplink. The uplink MIMO precoding is considered part of the uplink channel, and by estimating the channel matrix $\mathbf{H}$ using the received DM-RS symbols, this estimation will correspond to the channel

conditions experienced by the data symbols. Then, MIMO detection can operate without explicitly considering the uplink precoding matrix.

When the UL channel matrix is estimated to support DL precoding, the target is to estimate the channel conditions that will be experienced by the DL data, which are not affected by the UL precoding matrix. Therefore, it is not appropriate to make this estimation using the DM-RS signals. Instead, the SRS signals become more adequate as they are not affected by the UL precoding, as shown in Fig. 20.

a) Configuration of Uplink DM-RSs

Uplink DM-RSs are only transmitted in the RBs used for PUSCH transmission. Up to 12 orthogonal DM-RSs are specified for MU-MIMO transmission purposes.

The values of the different symbols that constitute the DM-RSs are obtained from a Gold sequence of length 2^{31}-1 [55]. This sequence determines a symbol value per subcarrier for all the subcarriers in a channel. Then, to orthogonalize the DM-RSs sent through the different antennas, these symbols are multiplied by +1 or -1 depending on the specific subcarrier, time symbol and antenna where a DM-RS has to be transmitted. These multiplicative values +1 or -1 form a length 2 orthogonal code sequence. In this way, the orthogonalization of the DM-RSs sent through the different antenna ports is performed in the time, frequency and code domains.

The positions of the DM-RSs in the resource grid are dependent on the *mapping type* (A or B), on the use of single symbol or double-symbol DM-RSs, and on the *DM-RS type* (1 or 2).

The *mapping type* defines the time-domain structure. Specifically, in mapping type A, the first DM-RS is located in symbol 2 or 3 counted from the start of the slot, and the number of DM-RS transmissions per slot can range from 1 to 4. Mapping type A is useful for data transmissions that occupy most of a slot. In contrast, in mapping type B, the DM-RS is located in the first symbol of the data allocation, i.e., the DM-RS location is not given relative to the start of the slot but to the start of the data. Therefore, this mapping is convenient for transmissions using mini-slots.

The DM-RS in the time domain can be a single transmission or a double symbol transmission. Double symbol transmission is used to provide a larger number of orthogonal DM-RSs when using multiple antenna ports, so it becomes particularly adequate when there are multiple layers in MU-MIMO.

The *DM-RS type* essentially determines the density of DM-RSs in the frequency domain. Specifically, with type 1, the DM-RSs are transmitted with a separation of one subcarrier, so in one PRB of 12 subcarriers, there are 6 DM-RS signals. In contrast, type 2 DM-RSs exhibit a lower density. In this case, a PRB includes 4 DM-RSs (transmitted in two groups of two adjacent subcarriers). This lower density is exploited to generate a larger number of orthogonal DM-RSs in different antenna ports (i.e., by shifting in frequency the positions of the DM-RS for different ports). As a result, with DM-RS type 2, up to 12 orthogonal DM-RSs can be generated, while with DM-RS type 1, only up to 8 orthogonal DM-RS can be generated.

Sections 5.9 and 9.11 of [29] and Section 6.4.1.1 of [55] provide further details on the uplink DM-RSs.

b) Configuration of Uplink SRSs

Uplink SRSs used in 5G NR are extended Zadoff-Chu sequences [55]. The mapping of the uplink SRSs in the time/frequency/antenna domains for a UE is based on the following configuration:

- In the time domain, an SRS spans 1, 2 or 4 consecutive symbols and is located within the last 6 symbols of a slot based on the initial 3GPP Release 15 specifications for 5G NR. This span was subsequently extended to consider durations of 8 and 12 symbols, mainly related to the use of SRS for positioning purposes.
- In the frequency domain, an SRS has a "comb" structure, meaning that an SRS is transmitted on every 2 or 4 subcarriers based on the initial Release 15 specifications, which were subsequently extended in Release 16 to consider 8 subcarriers. The SRSs are configured to span a certain bandwidth. It is possible to configure frequency hopping patterns, in which the SRS of a UE is transmitted at different PRBs in different symbols.
- In the antenna domain, the SRS transmitted through different antenna ports uses the same SRS sequence but applies different phase rotations to separate the different ports. This sequence can be configured with 1, 2 or 4 antenna ports.

Combining the time domain and frequency domain configuration, as well as the parameters of the sequence (e.g., root sequence and phase shifts), it is possible to create different sets of orthogonal SRS transmissions [62].

Section 8.3 of [29] and Section 6.4.1.4 of [55] provide further details on the uplink SRS signals.

5) Demodulation

This function performs the mapping between the detected symbols at the output of the MIMO detection process in each subcarrier/user antenna and the corresponding bits depending on the modulation used. In addition to the same M-Quadrature Amplitude Modulation (M-QAM) modulations used in DL (QPSK, 16QAM, 64QAM, and 256QAM), the UL can also use π/2-BPSK in the case of operating with DFT-precoded OFDMA. In M-QAM modulation, there are M constellation symbols, each with $m=\log_2 M$ constituent bits. Then, the demodulation process consists of determining which of the M

constellation symbols fits better with the detected symbol at the output of the MIMO detection and obtaining its constituent bits. Soft decision demodulation refers to the process of extracting soft decision bit information (SDBI) for the constituent bits in an M-ary modulation symbol. Having accurate soft decision information is very useful for codes that apply iterative soft-input-soft-output decoding, such as turbo-codes or LDPC codes, and achieves excellent performance close to the Shannon limit.

6) Descrambling

This process consists of multiplying bit-to-bit the received codewords by the scrambling sequence that was utilized by the UL transmitter at the UE. Since the scrambling sequence is composed of 1 or -1 values, in practice, the descrambling process involves keeping or inverting the sign of each soft bit obtained at the output of the demodulator. The employed UL scrambling sequence depends on the identity of the UE, on a scrambling configuration, and on the index of the random-access preamble that was transmitted by the UE. The details of the UL scrambling process for the PUSCH are provided in Section 6.3.1.1 of [55].

7) Channel Decoding

The process uses as input the received soft bits obtained from the demodulator and descrambling and performs the LDPC decoding of each code block exploiting the redundancy introduced in the channel coding at the UL transmitter to correct possible errors introduced by the channel. The CRC included in each code block is used to detect if there are still residual errors after the decoding process. Following the opposite processes explained in Section III.A.1 from the transmitter side, the decoded code blocks are then concatenated to form the received transport block.

C. Summary of Lessons Learned

The main lessons learned in this part of the tutorial are summarized as follows:

- The PHY layer offers multiple possibilities for defining a low-layer functional split option depending on the sequence of BB processing functions executed in a base station. This section details each of these BB functions for both DL transmission and UL reception, considering a base station that supports massive MIMO.
- The DL processing functions generate the transmitted signals and consist of the following functions: (i) the channel coding that generates the redundancy bits for enabling detection and correction of errors at the receiver side for each transmitted transport block. In 5G NR, the channel coding is based on the use of LDPC codes. (ii) The scrambling that multiplies the bits of each codeword by a sequence that enables the distinction between the useful signals and the interfering signals at the same frequency. (iii) The modulation that maps the scrambled bits into complex modulation symbols following an M-QAM modulation scheme. (iv) Layer mapping determines which modulation symbols are sent through each spatial layer. (v) The multiantenna precoding that maps the symbols of each spatial layer to the transmit antennas using a multiplication by a precoding matrix that is determined based on the DL channel estimation. (vi) The resource element mapping that maps the symbols of each antenna onto the available subcarriers/symbols in that antenna. (vii) The IFFT, cyclic prefix insertion and D/A conversion generate the time samples and, based on these, the analogue signal transmitted by each antenna.
- The UL processing functions obtain the signals received by each antenna and extract the received transport blocks. These functions are described as follows: (i) A/D conversion, cyclic prefix removal and FFT obtain the signal received by each antenna and generate the complex symbols received in each subcarrier of that antenna. (ii) Resource element demapping determines the symbols sent in each physical channel. (iii) Massive MIMO detection estimates the symbols sent by each UE (spatial layer) based on the symbols received in each antenna and the estimated channel matrix. (iv) Channel estimation is employed to estimate the uplink channel matrix utilized in the MIMO detection process and, assuming a TDD system with UL-DL channel reciprocity, also involves DL multiantenna precoding. (v) Demodulation determines the bits received in each spatial layer from the modulated symbols. (vi) Descrambling multiplies bit-to-bit the received codewords by the same scrambling sequence applied at the UL transmitter. (vii) Channel decoding estimates the received bits of each transport block and exploits the redundancy introduced at the channel coding of the transmitter side.
- Each BB function has been presented to highlight its inputs, outputs and operation, which establishes the basis for defining the computational complexity of each function.
- The use of massive MIMO with multiple antennas at the base station implies that all the DL functions after the precoding and the UL functions before the MIMO detection, i.e., resource mapping/demapping and IFFT/FFT functions, have to be executed per antenna. Therefore, functional splits defined at these functions will require the information sent through the fronthaul scales with the number of antennas at the base station.

IV. Computational Complexity of the BB Processing Functions

This section presents an analysis of the computational complexity of the relevant BB processing functions explained in Section III based on different state-of-the-art techniques. In addition, this section analyses the performance that can be obtained with different techniques for each BB function.

A. Massive MIMO Baseband Processing Functions

1) Massive MIMO Detection

As explained in Section III.B.3, massive MIMO detection estimates $\tilde{\mathbf{s}}$ of the vector of transmitted symbols $\mathbf{s}$ by the U users

with the estimated channel matrix $\widehat{\mathbf{H}}$ and the vector of received symbols $\mathbf{y}$ at each BS antenna. The optimum detector for a MIMO system with U transmit antennas and B receive antennas is the maximum likelihood (ML) detector, which minimizes the Euclidean distance between the received vector $\mathbf{y}$ and all possible combinations of transmitted symbol vectors [52]. However, since this approach involves brute force search for all possible combinations, its complexity exponentially grows with the number of transmit antennas, i.e., for a constellation with M symbols, the ML algorithm requires M^U calculations of the Euclidean distance, which becomes impractical (e.g., for U=16 and 64QAM, it yields $64^{16}=7.9\cdot10^{28}$ calculations). For this reason, suboptimal detection techniques have been considered. In this respect, multiple algorithmic solutions have been presented in the literature for massive MIMO detection, covering both linear and nonlinear schemes [53]. In this section, we focus on linear MIMO detectors, which are some of the most widely utilized detectors. Linear MIMO detectors obtain estimation $\tilde{\mathbf{s}}$ of the transmitted symbols $\mathbf{s}$ by the multiplication of vector $\mathbf{y}$, which contains the symbols received in each antenna, by a $U\times B$ matrix that depends on the estimated channel matrix $\widehat{\mathbf{H}}$.

The three basic schemes for linear MIMO detection are [53]:

- Matched Filter (MF) detection: This technique simply multiplies vector $\mathbf{y}$ by the conjugate transpose of the estimated channel matrix $\widehat{\mathbf{H}}^H$, that is,

$$\tilde{\mathbf{s}} = \widehat{\mathbf{H}}^H \mathbf{y}. \tag{3}$$

  This technique is aimed at maximizing the received signal-to-noise ratio for the signal of each user by disregarding the effect of multiuser interference. This technique works properly only when U is much smaller than B and provides a worse performance than more complex detectors.
- Zero Forcing (ZF) detection: This technique is based on inverting the channel matrix $\mathbf{H}$ and thus removing the effect of the channel, that is,

$$\tilde{\mathbf{s}} = \left(\widehat{\mathbf{H}}^H \widehat{\mathbf{H}}\right)^{-1} \widehat{\mathbf{H}}^H \mathbf{y}. \tag{4}$$

  This technique disregards the effect of noise and works properly in interference-limited scenarios. However, this technique may amplify the noise in the case of small-valued channel coefficients.
- Minimum Mean-Square Error (MMSE) detection: This technique minimizes the mean-square error between the transmitted signal and the estimated signal and is given by

$$\tilde{\mathbf{s}} = \left(\widehat{\mathbf{H}}^H \widehat{\mathbf{H}} + a\mathbf{I}\right)^{-1} \widehat{\mathbf{H}}^H \mathbf{y}, \tag{5}$$

  where $a=U/SNR$, SNR is the signal-to-noise ratio and $\mathbf{I}$ the $U\times U$ identity matrix. MMSE detection is capable of achieving significantly better performance than the ZF detector when the noise power is large, and indeed, it is widely utilized by many detectors in the literature.

The main problem for the practical implementation of these techniques, in particular ZF and MMSE, is that they embrace a matrix inversion. Specifically, for MMSE, the problem is how to determine the inverse of matrix $\widehat{\mathbf{H}}^H\mathbf{H} + a\mathbf{I}$ to achieve an efficient trade-off between complexity and detection performance. There exist different techniques for linear MMSE detection proposed in the literature that mainly differ on how to implement this inverse. From the wide range of references available in the open literature, we focus on those that include a computational complexity analysis.

In [63], three different techniques for linear MMSE detection are investigated and compared. The first technique is an exact MMSE via LDL decomposition, which decomposes the matrix $\widehat{\mathbf{H}}^H\mathbf{H} + a\mathbf{I}$ into a lower triangular matrix $\mathbf{L}$ and a diagonal matrix $\mathbf{D}$ that yield $\mathbf{L}\mathbf{D}\mathbf{L}^T = \widehat{\mathbf{H}}^H\mathbf{H} + a\mathbf{I}$ (note that the reference [63] assumes that channel matrix $\widehat{\mathbf{H}}^H$ is real, so that $\widehat{\mathbf{H}}^H = \widehat{\mathbf{H}}^T$). The advantage of this method is that the inverse of triangular and diagonal matrices is trivial and obtained only with sums and products. The second technique is an approximate MMSE via Neumann series approximation (NSA), which allows the estimation to be computed without the need for explicit matrix inversion or decomposition. The third technique is an approximate MMSE via conjugate gradient (CG) methods that provide very efficient alternatives to solve linear equation systems that are positive definite. The paper compares the computational complexity and performance of the three techniques and concludes that it is not obvious whether the approximate approaches have better total complexity versus performance trade-off than the exact approach but that it highly depends on the particular channel setting and properties.

Similarly, [64] proposed a low-complexity near-optimal algorithm using the CG method, which iteratively achieves MMSE performance without matrix inversion. The paper compares the CG method against a conventional NSA algorithm, reducing the computational complexity from $O(U^3)$ to $O(U^2)$ while achieving the near-optimal performance of classical MMSE with matrix inversion by using only a small number of iterations. Regarding the NSA algorithm, paper [65] proposed an enhanced method in which only the numerically dominant elements of Gram matrix $\widehat{\mathbf{H}}^H\mathbf{H}$ are employed for a low-complexity matrix inversion.

[66] presented different matrix decomposition algorithms for MMSE and compared them in terms of computational complexity. The compared algorithms include QR decomposition using the Gram–Schmidt process, Cholesky decomposition and LDL decomposition. A comparison is also performed against approximate inversion-based detectors, including the NSA, Gauss–Seidel (GS) and conjugate gradient (CG) algorithms.

The Richardson (RI) method, which avoids matrix inversion, is utilized with different variations as noted in [67][68][69][70] to reduce the complexity from $O(U^3)$ to $O(U^2)$ with respect to other matrix decomposition methods.

While all the above methods directly apply MMSE detection on vector $\mathbf{y}$, which contains the signals received in the B antennas at the BS, there exist methods that perform a transformation of the signals received in the different antennas

TABLE III
COMPUTATIONAL COMPLEXITY IN TERMS OF THE NUMBER OF REAL MULTIPLICATIONS FOR DIFFERENT LINEAR MMSE DETECTION ALGORITHMS

Algorithm	**Number of real multiplications**	**Reference**	**Number of multiplications for B=64, U=16, and i=3 iterations**
CG	$(i\text{-}1)(8U^2+28U)$	[64] (note 1)	4992
	$(i+1)(4U^2+20U)$	[66]	5376
QR decomposition (using Gram–Schmidt)	$4U^3+2U^2$	[66]	16896
Cholesky decomposition	$(1/3)(2U^3+3U^2-5U)$	[66]	2960
LDL decomposition	$(1/3)(2U^3+12U^2-14U)$	[66]	3680
RI	$4iU^2+2(i+1)$U	[67]	3200
	$4iU^2+2iU$	[68]	3168
	$(4B+4i)U^2+2BU$	[69]	70656
	$4BU+2U+(8BU-6U)i$	[70]	28416
NSA	$(2B+1)U^2+(4B-1)U+4U(7U-6)$	[65]	43888
	$(i\text{-}1)(2U^3+2U^2-2U)$	[66]	17344
	$2BU^2+4BU+12U^2+4(i\text{-}2)U^3$, for $i\geq3$	[72] (expression from [70])	56320
GS	$6iU^2$	[66]	4608
	$2BU^2+4BU+4(i+2)U^2$	[73] (expression from [70])	41984
TMA	$2BU^2+4BU+28U^2-16U+4(i\text{-}2)U^3$, for $i\geq3$	[74] (expression from [70])	60160
SDJC	$2BU^2+4BU+4U^2+6U+(4U^2-2U)i$	[75]	40960
WeJi	$2BU^2+4BU+4U^2+4U+(4U^2-4U)i$	[76] (expression from [70])	40832
Beamspace Local LMMSE	$B\cdot\log_2B + U\cdot M$, where M is the number of selected beams	[71] (note 2)	640 (assuming M=U)

note 1: The computation presented in [64] is given in complex multiplications, so the expression is multiplied here by 4 to transform it into real multiplications.
note 2: The expressions of [71] are given in orders of magnitude. Here, we have made the approximation $O(x)\approx x$.

to the "beamspace domain". In this way, the signal of each UE is concentrated only around a few beams in this domain, with the number of beams being much less than the number of B antennas. Thus, the linear MMSE detection process is performed considering a smaller matrix. An example of these methods is the beamspace local linear MMSE (LMMSE) approach from [71].

Table III presents a summary comparison of the computational complexity achieved by different MMSE detection algorithms investigated in the literature. In addition to the previously mentioned approaches, we also include tridiagonal matrix inversion approximation (TMA) [74], steepest descent Jacobi (SDJC) [75], and weighted Jacobi (WeJi) [76]. Computational complexity is measured in terms of the number of required multiplications. For methods that consider an iterative approach, i refers to the number of iterations. For comparison purposes, the last column of the table also presents the particularization for an example configuration of B=64 antennas at the BS and U=16 antennas at the UE devices and considers a typical case for many methods of i=3 iterations.

Table III reveals substantial differences among the considered algorithms, which can be of one order of magnitude. These significant differences are also observed among different variations of the same family of methods, e.g., for the different RI methods or the different NSA methods. It is also remarkable that, according to the analysed references, the computational complexity of some algorithms depends on U but not on B, while for other references, it depends on both B and U, which yields substantial variations in the values obtained in the right-most column in Table III.

In addition to the computational complexity of a MIMO detection technique, the achieved performance, e.g., in terms of the bit error rate (BER), is also a relevant aspect to consider. In this respect, different illustrative results on the performance for some of the previously discussed algorithms are presented. Table IV presents a comparative analysis based on performance results collected from several sources in the literature. As the basis for the comparison, we use the BER vs. signal-to-noise ratio (SNR) performance of the ideal MMSE detector with exact matrix inversion. Then, as a relevant metric for the comparison, Table IV presents, for each algorithm, the increase in the required SNR that is needed to achieve a bit error rate of 1E-3 with respect to the ideal MMSE detector. Values near 0 dB reflect that the achieved performance is very similar to that of the ideal MMSE, while larger values of this metric reflect a worse performance. Table IV also indicates, for each reference, the conditions under which the corresponding results have been obtained in terms of the number of antennas at the BS and UE devices, utilized modulation, channel conditions and number of iterations i.

TABLE IV
PERFORMANCE OF DIFFERENT MASSIVE MIMO DETECTION ALGORITHMS MEASURED IN TERMS OF THE INCREASE IN SNR WITH RESPECT TO THE IDEAL MMSE DETECTOR NEEDED TO ACHIEVE A BIT ERROR RATE OF 1E-3

Algorithm	**Conditions and values**			**Source**
	B=128, *U*=16, 64 QAM, Rayleigh			Fig. 3 from [64]
	i=3	*i*=4	*i*=5	
CG	1 dB	0 dB	0 dB	
NSA	N/A (>> 4 dB)	N/A (> 4 dB)	2 dB	
	B=128, *U*=16, 64 QAM, Rayleigh, code rate 1/2			Fig. 3 from [67]
	i=2	*i*=3	*i*=5	
NSA	N/A (>>2 dB)	0.7 dB	0 dB	
RI	0.7 dB	0.3 dB	0 dB	
	64 QAM, Rayleigh, code rate 1/2			Fig. 3 from [70]
	B=128, *U*=16, *i*=2	*B*=128, *U*=32, *i*=3		
RI	0.1 dB	0.2 dB		
TMA	N/A (> 3 dB)	N/A (>> 3 dB)		
SDJC	0.5 dB	1.2 dB		
WeJi	0.5 dB	2.4 dB		
	B=64, *U*=16, 64 QAM, Rayleigh			Fig. 4 from [66]
NSA	N/A (>> 8 dB)			
GS	1 dB			
CG	N/A (>>8 dB)			

The comparison between the CG algorithm and the NSA algorithm from [64] reflects that the CG algorithm significantly outperforms the NSA algorithm. With 3 iterations, the CG algorithm requires an SNR that is only approximately 1 dB greater than that of the ideal MMSE, while it achieves almost the same performance as the ideal MMSE with 4 or 5 iterations. In contrast, the NSA algorithm always requires a larger SNR than that of the CG algorithm for the same number of iterations. Note that some cases of NSA are indicated in the table as N/A as in the results of [64], the BER never reaches the value of 1E-3 in the range of analysed SNR values.

Concerning the comparison from [67] between the Richardson (RI) method and the conventional NSA algorithm, the RI method with just 2 iterations outperforms the conventional NSA algorithm, and with 3 and 5 iterations, it achieves a performance similar to that of the exact MMSE method. Moreover, although the results from [64] and [67] have not been obtained under the same conditions as [67] assumes a convolutional code with a rate of 1/2, the table seems to suggest that the RI method also outperforms the CG method as RI with *i*=3 iterations is closer to MMSE (0.3 dB) than the CG algorithm with 3 iterations (1 dB). The performance of the Richardson method is also superior to other methods, as shown in the results of Table IV obtained from reference [70], which compares the RI method against the TMA, SDJC and WeJi methods.

The results from reference [66] compare the NSA, GS and CG methods with 3 iterations. The results reflect that GS outperforms both the NSA and CG algorithms, as it requires an SNR that is 1 dB higher than that of the exact MMSE case, while both NSA and CG approaches are not able to reach the BER of 1E-3 in the range of the analysed values, meaning that they will need much more than an SNR of 8 dB compared to that of the exact MMSE case.

From the results of Table IV, it can be concluded that, from the BER perspective, the best performance is generally achieved by the Richardson algorithm, followed closely by the GS algorithm. Moreover, these two techniques are better than the CG and NSA algorithms. In general, NSA offers the worst performance among the considered algorithms. In addition, the Richardson algorithm also outperforms other methods, such as the TMA, SDJC or WeJi methods.

2) Channel Estimation for Massive MIMO

Following the notation of the general model of Section III.B.4, the classical training-based estimation methods are [60]:

- Least-square (LS) channel estimation: This technique estimates the channel matrix as the matrix that minimizes the square of the difference between the received matrix $\mathbf{T}$ and matrix $\widehat{\mathbf{H}}\mathbf{P}$. This result is achieved by multiplying matrix $\mathbf{T}$ by the pseudoinverse $\mathbf{P}^{\dagger}$ of matrix $\mathbf{P}$, which contains the transmitted training sequences of reference signals (i.e., pilots). This step leads to

$$\widehat{\mathbf{H}} = \mathbf{T}\,\mathbf{P}^{\dagger} = \mathbf{T}\,\mathbf{P}^{H}\left(\mathbf{P}\,\mathbf{P}^{H}\right)^{-1}. \tag{6}$$

  Assuming that matrix $\mathbf{P}$ is known in advance, its pseudoinverse $\mathbf{P}^{\dagger}$ can be precomputed. Therefore, the computational complexity in this case is given by the product of the $B\times L$ matrix $\mathbf{T}$ by the $L\times U$ matrix $\mathbf{P}^{\dagger}$, where L is the length of a training sequence. This step leads to $B\cdot U\cdot L$ multiplications of complex values. Correspondingly, the complexity in terms of the number of real multiplications is

$$c_{LS} = 4\cdot B\cdot U\cdot L. \tag{7}$$

- MMSE estimation: This technique intends to minimize the mean square error between the actual matrix $\mathbf{H}$ and the estimated matrix $\widehat{\mathbf{H}}$. From [60], this estimation is given by

$$\widehat{\mathbf{H}} = \mathbf{T}\left(\mathbf{P}^{H}\mathbf{R}_{\mathbf{H}}\mathbf{P} + \sigma^{2}B\mathbf{I}\right)^{-1}\mathbf{P}^{H}\mathbf{R}_{\mathbf{H}}, \tag{8}$$

  where $\mathbf{R}_{\mathbf{H}} = E\left[\mathbf{H}^{H}\mathbf{H}\right]$ is the channel correlation matrix.

A practical limitation of the MMSE estimation is that it requires a priori knowledge of the channel correlation matrix $\mathbf{R}_{\mathbf{H}}$, which may be unrealistic in practical applications. Then, one option is to estimate $\mathbf{R}_{\mathbf{H}}$ as the correlation of the channel matrix resulting from an LS estimation [77]. Similarly, another option presented in [60] is to relax the MMSE estimation by approximating $\mathbf{R}_{\mathbf{H}}$ with an identity matrix $\alpha\mathbf{I}$ scaled with parameter α, which is adjusted to minimize the mean square error. This step leads to the following estimation, which is referred to as relaxed MMSE (RMMSE),

$$\widehat{\mathbf{H}} = \mathbf{T}\left(\mathbf{P}^H\mathbf{P} + \frac{\sigma^2 B{\cdot}U}{tr\{\mathbf{R_H}\}}\mathbf{I} \right)^{-1} \mathbf{P}^H . \quad (9)$$

The trace $tr\{\mathbf{R_H}\}$ of matrix $\mathbf{R_H}$ can be estimated as the trace of matrix $\widehat{\mathbf{H}}_{LS}^H\widehat{\mathbf{H}}_{LS}$, where $\widehat{\mathbf{H}}_{LS}$ is the LS channel estimation. To assess the computational complexity of the MMSE estimation, the following operations are considered according to (9):

1) The computation of $tr\{\mathbf{R_H}\}$ using $\widehat{\mathbf{H}}_{LS}$ requires $4{\cdot}B{\cdot}U{\cdot}L$ real-valued multiplications for making the LS channel estimation and $4{\cdot}B{\cdot}U$ real-valued multiplications for determining the U elements of the main diagonal of matrix $\widehat{\mathbf{H}}_{LS}^H\widehat{\mathbf{H}}_{LS}$ (note that each element requires B complex multiplications). Then, the total is $4{\cdot}B{\cdot}U{\cdot}(L{+}1)$.
2) The number of multiplications for inverting the $L{\times}L$ matrix $\left(\mathbf{P}^H\mathbf{P} + \sigma^2 BU\mathbf{I}/tr\{\mathbf{R_H}\}\right)$ using the Gauss–Jordan elimination method is $L^3/3{+}L^2/2{-}5L/6$ complex-valued multiplications according to [78][79], thus, a total of $4L^3/3{+}2L^2{-}10L/3$ real-valued multiplications. It is assumed that matrix $\mathbf{P}^H\mathbf{P}$ is precomputed in advance, and we disregard the complexity for determining the scaling value ($\sigma^2\, BU/tr\{\mathbf{R_H}\}$) of the identity matrix.
3) The product of the $L{\times}L$ matrix resulting from the inversion and the $L{\times}U$ matrix $\mathbf{P}^H$ requires $4{\cdot}L^2{\cdot}U$ real-valued multiplications.
4) The product of the $B{\times}L$ matrix $\mathbf{T}$ by the matrix $L{\times}U$ matrix resulting from the previous product requires $4{\cdot}B{\cdot}U{\cdot}L$ real-valued multiplications.

Then, the total computational complexity is estimated as

$$c_{MMSE} = 4L^3/3 + 2L^2 - 10L/3 + 4L^2U + 4B{\cdot}U{\cdot}(2L+1). \quad (10)$$

Further variations of the MMSE and RMMSE channel estimations are presented in different works, such as [80] or [81], which considers an estimation of the combined interference plus noise power from users with the same training sequence and includes it in the MMSE estimator.

Surpassing the classical techniques, there exist other channel estimation techniques that exploit the low rank (sparse) properties of channel environments, which are strongly manifested in mmWave frequencies due to their predominantly directional communication, with only a small number of strong propagation paths, such as the line-of-sight (LoS) component and a few first-order reflections. These techniques are the low-rank channel estimation techniques [82]. The use of these techniques is beneficial from different perspectives, such as reducing the computational complexity or improving the quality of the estimated channel by reducing the channel estimation errors due to aspects such as the pilot contamination that appears when multiple users reuse the same training sequence [83]. Low-rank channel estimation techniques fall under different categories:

a) Channel Covariance Matrices (CCM) Techniques

These techniques are based on exploiting the low-rank properties inside the channel covariance matrices of different users or the sparsity inside the instantaneous channels. If the rank r of the CCM of a user is much lower than the number of antennas at the BS, i.e., $r{<<}B$, then the channel can be represented by r eigenvectors, which would reduce the channel dimension from B to r. This approach devises a finite scattering environment for massive MIMO systems and suggests that the angular spread (AS) of each user is restricted within a narrow region. Examples of works that have employed these techniques are [83][84][85]. In [83], it is demonstrated that the exploitation of covariance information under certain subspace conditions on the covariance matrices can lead to a complete removal of pilot contamination effects for numerous transmit antennas, and then the authors develop a Bayesian channel estimation method explicitly using covariance information and exploiting the notion that desired user signals and interfering user signals are received with approximately finite-rank covariance matrices. The work in [84] presented a joint spatial division multiplexing (JSDM) scheme based on a multiuser precoder to restrict the beamforming vector of each user within the orthogonal complement of the channel subspaces of the other users. The approach includes channel estimation based on the channel covariance matrix of each user in conjunction with linear MMSE estimation. The authors of [85] present a method for estimating the covariance matrices and apply it to MMSE estimation, formulating the optimization of how to assign users to the available pilot sequences.

b) Compressive Sensing (CS) Techniques

These rank minimization methods directly exploit the low-rank properties of channel matrices with the aid of CS theory without the need for any additional knowledge about the statistical distribution or physical parameters of the propagation channels. CS is a signal processing technique for estimation based on obtaining solutions to underdetermined linear systems. CS relies on the principle that, by optimization, the sparsity of the function to be estimated can be exploited to recover it from a more reduced set of samples. In [86], an open-loop channel estimator is presented for a hybrid MIMO system in mmWaves consisting of RF beamformers with large antenna arrays followed by a baseband MIMO processor. The exploitation of channel sparsity is formulated as a CS problem that estimates the angle of departure/angle of arrival (AoD/AoA) and the corresponding gain of each significant path. This problem is based on the parametric channel model with quantized angle grids and is solved by the orthogonal matching pursuit (OMP) method that employs a redundant dictionary consisting of array response vectors with finely quantized grids. A different approach is presented in [87] by exploiting the notion that the degrees of freedom of the physical channel matrix are smaller than the number of free parameters. Then, channel

estimation is formulated as an atomic norm minimization (ANM) problem and is efficiently solved via the alternating direction method of multipliers (ADMM). In [88], a channel estimation approach is presented based on the CS technique with nonlinear recursive optimization. In [89], compressive training signals are transmitted over multiple pilot tones and compressive measurements are sent back to the BS, which recovers the subchannel vectors for each user with CS technology.

c) *Antenna Array Theory (AAT) Techniques*

These techniques are based on applying array signal processing to determine the angular spread information of incident signals from different users to BS antennas. In this respect, a widely utilized signal processing technique is to apply the DFT over the received signals in the different antennas, which are regarded as spatial sample points. Then, this transformation allows passing from the antenna domain to the angular domain (equivalently, the beamspace domain). In massive MIMO, the existence of many antennas at the BS greatly enhances the resolution of this DFT transformation and allows accurate identification of the angular spread of the incoming signal. The nonzero points at the output of the DFT reflect the beamspace subchannels that concentrate around the central direction of arrival (DOA) of the incident signals, while the width of these nonzero points corresponds to the angular spread of the incident signals. Such a strategy is known as the spatial basis expansion model (SBEM) [90]. With narrow angular spreads, the signals of the UE devices become concentrated only around a few angles or beamspace subchannels instead of over all the antennas. Thus, the original channel becomes sparse in the beamspace domain.

Paper [91] presents a very large scale integration (VLSI) hardware architecture for channel estimation based on the SBEM, with the capacity to fully implement an angle division multiple access (ADMA). The authors define all required building blocks, adjust the algorithm to a real implementation and identify the required quantization schemes. The transmission strategy has been slightly adjusted to ease implementation. Similarly, in [92], a channel estimator that relies on Stein's unbiased risk estimator (SURE) and is referred to as BEAmspace Channel EStimator (BEACHES) is proposed. BEACHES exploits the sparsity of mmWave channels in the beamspace domain and adaptively denoises the channel vector. The proposed VLSI architecture is built around three modules: a) antenna-to-beamspace (A2B) conversion module, b) SURE-based denoiser module, and c) beamspace-to-antenna (B2A) conversion module. An architecture for channel estimation based on path-division multiple access (PDMA) that takes into account the dual-wideband effect, which becomes relevant in massive MIMO for mmWaves, is presented in [93]. The paper employs a method based on successive dichotomy to reduce the searching complexity instead of carrying out a selection for quantization.

Other AAT-based channel estimation techniques include the methodology of the beamspace local LMMSE approach from [71], which has been mentioned in the previous subsection or the beamspace channel estimation (BSCE) of [94]. In BSCE, channel estimation errors can be reduced on the condition that the directions of beams are near those of dominant paths. Nevertheless, the number of beams is limited to the number of antennas when using a DFT matrix for the space transformation. Then, to increase the number of beams, [94] proposed using multiple DFT matrices that form beams in mutually different directions. BSCE is organized in four stages: (1) LS channel estimation, which estimates the channel frequency response for each antenna using the LS technique. (2) Antenna-to-beamspace transformation, which transforms channel responses from the antenna space into the beamspace. Assuming that the BS has a two-dimensional antenna array, this transformation is achieved with a 2D DFT transformation. (3) Beam selection, which selects the nonzero beams as those whose channel estimates in the beam domain are higher than a threshold and sets the channel estimates of unselected beams to zero. (4) Beam-to-antenna space transformation, which calculates the eigenvectors in beamspace by the singular value decomposition (SVD) or eigenvalue decomposition (EVD) of the matrix whose elements are the estimated channel of selected beams. It is assumed that the selected beams are less than B (dimension in antenna space); therefore, the SVD or EVD in beamspace will require lower computational complexity than in the antenna space.

To illustrate the reader of this tutorial on an example for computing the computational complexity of one of these algorithms, the Appendix presents the details of this computation for the BSCE method.

The computational complexity of some of the channel estimation techniques mentioned above is presented in Table V in terms of the number of real-valued multiplications. The corresponding reference for the estimation is also included. For the iterative methods, i represents the number of iterations. For some of the included references, the complexity is given in terms of the order of magnitude. The table also includes the number of multiplications for the reference configuration B=64 antennas at the BS and U=16 antennas at the UE. Moreover, the length of the training sequence has been set to L=U=16 following [60].

Focusing on the performance obtained by the different techniques, Table VI presents the results in terms of the mean

TABLE V
COMPUTATIONAL COMPLEXITY IN TERMS OF THE NUMBER OF REAL-VALUED MULTIPLICATIONS FOR DIFFERENT CHANNEL ESTIMATION STRATEGIES

Algorithm	Number of real Multiplications	Reference for the computation	Number of multiplications for B=64, U=16, and L=16
LS	$4BUL$	Eq. (7)	65536
MMSE	$4L^3/3+2L^2-10L/3+4\cdot L^2\cdot U+4\cdot B\cdot U\cdot(2L+1)$	Eq. (10)	157472
SBEM	$4(L+\log_2(B)+V)$ V: Number of orthogonal training sequences	[91]	104 (considering V=4 as in [91])
BEACHES	Order is $O(B\log_2 B)$	[92]	Order: 384
OMP	Order is $O(i\beta B\log_2(\beta B)+i^2B+i^3B+i^4)$ i ranging from 2 to 45 β: oversampling factor, typically β=4	[92]	Order: 4880 for i=2, 1E7 for i=45
ANM	Order is $O(iB^3)$ i ranging from 130 to 360	[92]	Order: 34E6 for i=130
PDMA	$(4B(F+\log_2\max(B,F)+2)+8)U$ F: number of subcarriers	[93] (note 1)	561280 (assuming F=128 as in [93])
Beamspace local LLMSE	$L\cdot B\cdot\log_2 B+M\cdot B\cdot L+B\cdot M^2+B\cdot L\cdot U+U\cdot B\cdot M^2$ M: number of selected beams	[71] (note 2)	317440 (assuming M=U=16)
BSCE	$4(3UM^2+(2C+2)M^3)+(8B^2+6B+4BL+2)U$ M: number of selected beams	See Appendix	743456 (assuming M=U=16, C=2)

note 1: The computation presented in [93] is per user, so the expression is multiplied here by U.
note 2: The expressions of [71] are given in orders of magnitude. Here, we have made the approximation $O(x)\approx x$.

TABLE VI
PERFORMANCE OF DIFFERENT CHANNEL ESTIMATION ALGORITHMS IN TERMS OF MSE

Algorithms		Conditions and Values		Source
		B=30, U=10 (but results do not change when increasing up to B=140 antennas)		Figs. 3 and 5 from [81]
		SNR=5 dB	SNR=10 dB	
LS		0.3	0.3	
MMSE		0.23	0.23	
		B=128, user mobility, maximum rank r=16		Fig. 6 from [82]
		SNR=5 dB	SNR=10 dB	
CS		0.06	0.03	
SBEM		0.04	0.02	
JSDM	AS=4°	0.035	0.012	
	AS=14°	0.035	0.015	
	AS=16°	0.1	0.06	
		B=128, U=8, non-LoS channel		Fig. 5c from [92]
		SNR=5 dB	SNR=10 dB	
ANM		0.2	0.06	
NOMP		0.15	0.05	
BEACHES		0.15	0.06	

square error (MSE) for SNR=5 dB and SNR=10 dB for some of the abovementioned techniques extracted from different references. For the LS and MMSE techniques, the results from [81] reflect that both techniques exhibit an MSE floor when increasing the SNR, and thus, no significant variations are observed between the two SNR values of Table VI. The MSE floor is larger with the LS technique than with the MMSE technique.

Concerning the low-rank channel estimation techniques, Table VI presents a comparison from [82] among the JSDM technique of [84] as an example of the CCM-based techniques, the CS-based technique of [89] and the SBEM technique of [90] as an example of the AAT-based techniques. The results are presented for different values of the statistical angular spread (AS) covered by the users while moving, although the obtained MSE only significantly varies with the AS for the JSDM technique. The results show that the JSDM performs the best among all methods when the statistical AS is 4º or 14º but that it strongly degrades as the statistical AS increases and performs worse than CS and the SBEM. Moreover, it is observed that the SBEM slightly performs better than CS. The results from Table VI also reflect that the MSE values obtained with the LS and MMSE techniques are higher than those obtained with the low-rank channel estimation techniques. Although the comparison may not be accurate as the results from [81] and [82] are not obtained under the same conditions, the notion that the differences are approximately one order of magnitude suggests that the exploitation of channel sparsity leads to performance improvements compared with LS and MMSE. Table VI also presents a comparison from [92] between the BEACHES method that uses antenna array theory and the ANM and Newtonized OMP (NOMP) methods that are based on CS. The differences among all these methods are quite small, particularly for SNR=10 dB, while for SNR=5 dB, both NOMP and BEACHES perform slightly better than ANM.

The above results concluded that, in general, array antenna theory-based methods for low-rank channel estimation offer better performance than CS-based or CCM-based methods. Moreover, all these methods outperform the MMSE and LS techniques.

3) MIMO Precoding

Following the notation presented in Section III.A.5, the precoding obtains the $B\times 1$ vector $\mathbf{y}$, which includes the symbols

to be sent through each antenna, based on the $U\times1$ vector $\mathbf{x}$, which includes the symbols of each layer. Assuming a linear precoding scheme, which is the most usual approach, this step is performed by the multiplication $\mathbf{y}=\mathbf{W}\mathbf{x}$, where $\mathbf{W}$ is a precoding matrix of dimensions $B\times U$ obtained from the estimated channel matrix $\widehat{\mathbf{H}}$ of dimensions $B\times U$. Then, the different precoding methods mainly differ in how the precoding matrix $\mathbf{W}$ is computed.

The basic linear precoding schemes are listed as follows [95][96]:

- Maximum ratio transmission (MRT) or matched filter (MF): This approach is aimed at maximizing the gain of the signal at the receiving terminal. Then, the precoding matrix is just the conjugate of the channel matrix, i.e.

$$\mathbf{W} = \sqrt{\beta}\,\widehat{\mathbf{H}}^{*}, \tag{11}$$

  where β is a scaling power factor. The MRT algorithm achieves the sum capacity of the massive MIMO system when the number of antennas at the BS is much larger than the number of antennas at the users, which means that the interuser interference (IUI) is low compared to the noise.
- ZF: ZF mitigates the interference caused to other users by pointing the signal beam into the intended user while nulling the other directions where other users are located. The ZF precoding matrix is

$$\mathbf{W} = \sqrt{\beta}\,\widehat{\mathbf{H}}^{*}\left(\widehat{\mathbf{H}}^{T}\widehat{\mathbf{H}}^{*}\right)^{-1}. \tag{12}$$

  The ZF algorithm performance is close to optimal when the noise is negligible compared to the IUI.
- MMSE: This method exploits the benefits of the MRT and ZF algorithms. The method has acceptable performance with moderate noise and interference. The MMSE precoding matrix is calculated as

$$\mathbf{W} = \sqrt{\beta}\,\widehat{\mathbf{H}}^{*}\left(\widehat{\mathbf{H}}^{T}\widehat{\mathbf{H}}^{*} + \mathbf{V} + \alpha\mathbf{I}_{U}\right)^{-1}, \tag{13}$$

  where α is a positive regularizing factor and $\mathbf{V}$ is a $U\times U$ deterministic Hermitian nonnegative definite matrix. Matrix $\mathbf{V}$ can be set to 0 to achieve a balance between increasing the channel gain towards intended receive terminals (such as in MRT) and eliminating the IUI (such as in ZF).

The computation of ZF and MMSE precoding matrices according to the above formulae comprises the inversion of a very large matrix, particularly for large values of B and U, which can lead to very high computational complexities. For this reason, different methods have been proposed to reduce the complexity of the basic precoding algorithms. Following the classification presented in [95], these methods fall under the following categories:

a) Linear Precoder Based on Matrix Inversion Approximation

Methods under this category intend to approximate the inversion of the matrix rather than computing it. Examples of these methods are the truncated polynomial expansion (TPE) [97], the Neumann series approximation (NSA) [98][99], the Newton iteration (NI) algorithm [100] and the Chebyshev iteration (CI) algorithm [101].

The TPE algorithm approximates the matrix inversion of the MMSE algorithm by a polynomial of J terms of the Gram matrix $\mathbf{G} = \widehat{\mathbf{H}}^{T}\widehat{\mathbf{H}}^{*}$ with scalar coefficients. By properly adjusting the number of terms J, it is possible to adjust the complexity depending on the experienced SNR. The NSA algorithm expands the inverse matrix of the MMSE algorithm by a series of matrix vector multiplications that have a simple flow of data and can be highly parallelized. Similarly, the NI and CI methods also approximate the matrix inversion process by means of iterative procedures that differ in the specific expressions utilized in each iteration. One of the main challenges in these iterative algorithms is obtaining the initial value to start the iterations, which involves extra calculations apart from those of the iterative process itself. Optimization of these initial values can be performed so that they become easier to acquire [101]. Moreover, combinations among methods have also been presented, such as the joint NI-NSA algorithm in [102] or the joint CI-NSA algorithm in [103].

b) Fixed-Point Iteration-Based Algorithms

The algorithms under this category determine the vector $\mathbf{y}=\mathbf{W}\mathbf{x}$ that is the output of the precoding process, but without explicitly computing the precoding matrix $\mathbf{W}$. Instead, these algorithms determine $\mathbf{y}$ by iteratively solving a linear equation system that depends on the Gram matrix $\mathbf{G}$ and the input vector $\mathbf{x}$. Examples of these algorithms include the Gauss–Seidel (GS) algorithm [104], Successive Over-Relaxation (SOR) algorithm [105], Conjugate Gradient (CG) algorithm [106], Steepest Descent (SD) [106], Richardson (RI) algorithm [107] and Jacobi Iteration (JI) algorithm [108]. These approaches differ in how the iterations are being defined. For example, the GS algorithm relies on a factorization of the Gram matrix into a diagonal matrix, a lower triangular matrix and an upper triangular matrix, and the iterations involve inverting triangular matrices, as they are easy to invert. Then, the SOR algorithm relies on a similar principle but uses a variable relaxation factor. The CG iterations use a conjugate direction matrix related to the Gram matrix, and the JI algorithm decomposes this matrix into a diagonal matrix and an off-diagonal matrix.

Based on the discussions in [95], the GS algorithm converges slightly faster than the NSA algorithm and achieves better BER performance and lower complexity. Similarly, the CG algorithm also has lower complexity than the NSA algorithm and achieves the same performance as MMSE but with one order of magnitude less computational complexity. The JI algorithm shows lower performance than the

TABLE VII
COMPUTATIONAL COMPLEXITY IN TERMS OF NUMBER OF MULTIPLICATIONS FOR DIFFERENT ALGORITHMS OF COMPUTING THE PRECODING MATRIX

Algorithm		**Number of real multiplications**	**Reference for the computation**	**Number of multiplications for B=64, U=16, and i=3 iterations**
Basic algorithms	ZF	$4(U^3+2BU^2+UB+B)$	[103] (note 1)	151808
	MMSE	$4(3U^3+2U^2+UB^2+BU^2+B)$	[111] (note 2)	379136
Matrix inversion approximation algorithms	NSA	$(i\text{-}2)U^3+BU^2+U^2+2BU+B$, $O(U^2)$ for $i\leq 2$	[95]	22848
	NI	$2iU^3+U^2+UB+B$	[95]	25920
	CI	$2U^3+8iU^2+2U^2+2BU^2+2UB+2$	[95][112]	49666
Fixed point iteration algorithms	GS	$4iU^2+UB+B$	[95]	4160
	SOR	$i(4U^2+4U)+UB+B$	[95]	4352
	CG	$i(4U^2+10U)+UB+B$	[95]	4640
	JI	$i(4U^2-2U)+UB+B$	[95]	4064
Matrix decomposition algorithms	CSM	$16U^2-12U-4$	[110] (note 1)	3900

note 1: The expression in this reference is given in complex multiplications, so it is multiplied by 4 in the table.
note 2: The MMSE algorithm in this reference is referred to as regularized zero forcing (RZF). Additionally, the expression in the reference is given in complex multiplications, so it is multiplied by 4 in the table.

GS algorithm but benefits from parallelism and allows efficient hardware implementation.

c) Precoding Based on Matrix Decomposition

These algorithms determine the precoding matrix using decomposition techniques applied over the matrix to be inverted in the ZF or MMSE methods to more efficiently perform the inversion. Typical methods belonging to this category are QR decomposition [109] and Cholesky decomposition (CD) with Sherman Morrison (CSM) [110]. The precoding matrix based on QR decomposition is expressed as a function of a unitary matrix and the inverse of an upper triangular matrix, which is easy to compute. Similarly, the precoding matrix using Cholesky decomposition also involves the inverse of two triangular matrices.

Table VII presents the computational complexity in terms of the number of real-valued multiplications for some of the linear precoding methods discussed above. As in the previous sections, for the iterative methods, i denotes the number of iterations.

In relation to performance, different comparisons among algorithms are provided in the different papers. For example, Table VIII collects the results in terms of the required SNR for achieving a BER of 1E-3 with different precoding algorithms extracted from different references. The results of Table VIII extracted from [103] present a comparison focused on the matrix-inversion algorithms, including the NSA algorithm, the combination of the NI and NSA algorithms and the combination of the CI and NSA algorithms, as well as the ZF precoding with exact matrix inversion included as a reference. The results reflect that with i=3 iterations, the joint CI+NSA algorithm achieves a performance similar to the exact ZF. Additionally, it outperforms the joint NI+NSA algorithm. The NSA algorithm offers a similar performance to the joint algorithms but with a larger number of iterations.

TABLE VIII
PERFORMANCE OF DIFFERENT PRECODING ALGORITHMS IN TERMS OF THE REQUIRED SNR FOR A TARGET BER OF 1E-3

Algorithms		**Conditions and values**		**Source**
		B=128, U=16, 64 QAM, Rayleigh channel		Fig. 2 from [103]
Basic	ZF	23 dB		
Matrix inversion approximation algorithms	NSA	i=4	N/A (>>30 dB)	
		i=5	26 dB	
		i=6	24.5 dB	
	NI+NSA	i=2	N/A (>> 30 dB)	
		i=3	24 dB	
		i=4	23	
	CI+NSA	i=2	24 dB	
		i=3	23 dB	
		i=4	23 dB	
		B=128, U=16, 64 QAM, Rayleigh channel		Fig. 2 from [106]
Basic	MMSE	15 dB		
Fixed point iteration algorithms	SD	i=4	17 dB	
	RI	i=4	17.1 dB	
	CG	i=3	16 dB	
	JI	i=4	N/A (>>18 dB)	
	CG+JI	i=2	15.5 dB	
	SD+JI	i=4	15.7 dB	
		B=256, U=16		Fig. 2 from [110]
Basic	ZF	14 dB		
Matrix decomposition algorithms	CSM	14 dB		

Regarding the fixed-point, iteration-based algorithms, Table VIII presents a performance comparison of some of them based on the results of [106], including the SD, RI, CG, JI, CG+JI and SD+JI algorithms, as well as the ideal MMSE algorithm. The combined algorithms offer better performance than their individual counterparts, the CG, JC and SD algorithms. The best performance, in best-to-worst order, is achieved by the CG,

TABLE IX
NUMBER OF OPERATIONS FOR DIFFERENT M-QAM DEMODULATION STRATEGIES

Operation	ML	Max	HDT [114]	Modified Max from [113]	Modified Max from [115]
exp(·)	$O(2^m)$	0	0	0	0
log(·)	$O(m)$	0	0	0	0
comparisons	0	$O(m\cdot 2^m)$	0	0	$O(m)$
additions	$O(m\cdot 2^m)$	$O(m\cdot 2^m)$	$O(m)$	$O(m)$	$O(m^2+m)$
multiplications	$O(m\cdot 2^m)$	$O(m\cdot 2^m)$	$O(m)$	$O(m)$	$O(m^2+m)$
TOTAL	$O((2m+1)\ 2^m+m)$	$O(3m\cdot 2^m)$	$O(2m)$	$O(2m)$	$O(2m^2+3m)$

SD, RI and JI algorithms. Moreover, although not included in the table, the results from [106] also reflect that all these fixed-point, iteration-based algorithms outperform the NSA matrix inversion approximation algorithm, which is not able to achieve the target BER of 1E-3 in the SNR margins analysed in the paper.

In relation to the matrix decomposition-based algorithms, the results of Table VIII coming from [110] compare the BER obtained by the CSM against the exact ZF algorithm, reflecting that both methods achieve a very close performance. Moreover, although not included in the table, the results of [110] also show that these algorithms clearly outperform the NSA algorithm based on matrix inversion approximation and the SOR fixed-point, iteration-based algorithm, as none of these algorithms is able to achieve the value of BER=1E-3 with the SNR ranges analysed in the paper.

B. M-QAM Demodulation

As discussed in Section III.B.5, the use of soft decision information from the demodulation process is useful for the subsequent channel decoding processes. In this respect, soft demodulation techniques are widely utilized by 5G NR receivers. A summary of different categories of soft demodulation techniques based on [113] is presented in this section:

a) Maximum Likelihood (ML) Soft Demodulation
The ML soft demodulation scheme exhaustively searches the constellation symbols with the highest probability to estimate the SDBI. The typical implementation of the ML soft demodulator is to compute maximum a posteriori log-likelihood ratios (LLRs) for all information bits of the received symbol by assuming the baseband equivalent signal model. This ML method yields the maximum performance, but it involves logarithmic and exponential operations, and its computational complexity exponentially grows with the number of bits per symbol m. The complexity of the ML scheme can be reduced by eliminating the logarithmic and exponential operations, resulting in a soft demodulation approach referred to as the Max method. However, it still involves $\min|\cdot|^2$ operations, and the complexity still exponentially grows with m.

b) Hard Decision Threshold (HDT)-Based Demodulation
These demodulation schemes compute the SDBI as a weighted distance between the detected symbol and the HDT line (i.e., the line that defines the separation between the regions of the constellation in which one or another symbol is decided). For conventional QAM and PSK modulation, HDT-based demodulation requires only one distance calculation per bit. Nevertheless, if the modulation is of high order or the HDT lines are not simple and continuous, nonnegligible performance degradation will occur. An example of HDT-based schemes is presented in [114].

c) Modified Max Method Demodulation
These schemes eliminate the search process to identify the symbol with the minimum distance in the constellation. These methods determine the location of the detected symbol by comparison with the HDT lines, thereby requiring only m comparisons to identify the nearest symbol, i.e., the symbol with the minimum distance in the constellation [115]. Afterwards, another equation is applied to each bit to identify the symbol with the opposite bit value to that of the nearest symbol so that a Max equation can be applied. In [113], a universal soft demodulator that provides a performance equivalent to that of the Max-log scheme while having only linear-order complexity is presented. This Modified Max method does not require any searching process to identify the minima, and instead, it entails the mapping of the detected symbol to a specific region.

The computational complexity of the abovementioned soft M-QAM demodulation methods are listed in Table IX from [113].

In relation to the performance, Table X obtained from some results presented in [113] shows the required energy of bit to the noise power spectral density (Eb/No) to achieve a target bit error rate of 1E-3 with the different algorithms and for different modulation schemes. The results assume an additive white Gaussian noise (AWGN) channel and the use of 5G LDPC codes with a code rate of 1/2. The HDT-based method provides the worst behaviour, requiring the largest Eb/No to achieve the target BER. On the other hand, the modified Max method produces the same performance as the Max method but with

TABLE X
PERFORMANCE OF DIFFERENT SOFT DEMODULATION ALGORITHMS IN TERMS OF REQUIRED *EB/NO* FOR A TARGET BER OF 1E-3 (SOURCE: FIG. 9 FROM [113])

Algorithms	Modulation	*Eb/No*
ML	16QAM	2.6 dB
	64QAM	4.8 dB
	256QAM	7.3 dB
	1024QAM	10.2 dB
Max and Modified Max from [113]	16QAM	2.6 dB
	64QAM	4.9 dB
	256QAM	7.5 dB
	1024QAM	10.4 dB
HDT	16QAM	2.9 dB
	64QAM	5.4 dB
	256QAM	8.3 dB
	1024QAM	11.2 dB

only linear-order computational complexity, regardless of the modulation schemes. Moreover, both Max methods perform very closely to the ML scheme with differences only up to 0.2 dB.

C. Channel Coding/Decoding

5G LDPC codes adopt the structure of quasicyclic (QC) LDPC codes, which naturally enables parallelism and facilitates encoding and decoding processes. Let us consider a code with parameters (n, k), meaning that it obtains as an input a code block of k information bits and delivers as an output a codeword of n bits. LDPC codes are represented by a parity-check matrix **H**, which is a binary matrix that satisfies the condition $\mathbf{Hx}^{\mathbf{T}}=0$, where **x** is an output codeword. The dimension of the parity-check matrix is $m \times n$, where $m=n-k$ is the number of parity-check equations. In the case of 5G NR, the specific parity-check matrix to be used for coding a code block with size k is obtained from a base graph matrix of size $m_b \times n_b$ with $k_b=n_b-m_b$ and after applying a lifting process that uses a permutation matrix of size Z. Specific details of the channel coding process are provided in Section 5.3 of [56].

1) LDPC Encoder

Various approaches have been suggested to improve the hardware complexity of LDPC encoders. One of the most conventional approaches is systematic encoding, in which the generator matrix is derived from the parity check matrix by exploiting Gaussian elimination. The main drawback related to this method is that the storage overhead is dramatically increased for large block sizes, which limits its practical applicability.

The Richardson–Urbanke (RU) algorithm is a widely utilized LDPC encoding scheme developed by Richardson and Urbanke [116]. The underlying principle of the method is the transformation of the parity-check matrix into an approximate lower triangular (ALT) form by using only row and column permutations, which preserves the sparseness of the matrix. The structure of the parity-check matrix in lower triangular form is shown in Fig. 21. This matrix is divided into sub matrices **A**, **B**, **C**, **D**, **T**, and **E**, where **T** is a lower triangular matrix. With this structure, the output codeword **x** is a vector with the structure [**s**, $\mathbf{p}_a$, $\mathbf{p}_c$], where **s** is the systematic portion with size $k=n-m$ bits, and $\mathbf{p}_a$ and $\mathbf{p}_c$ are parity bits, with sizes g and $m-g$, respectively. Parameter g is referred to as the gap, as it measures in some way the "distance" of a given parity-check matrix to a lower triangular matrix.

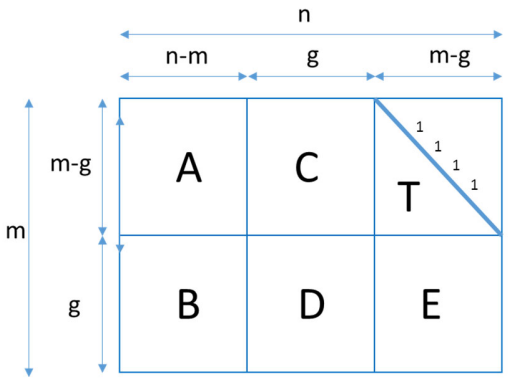


Fig. 21. Structure of the parity-check matrix in approximate lower triangular form.

TABLE XI
COMPUTATIONAL COMPLEXITY OF LDPC CODING ALGORITHMS

Algorithm	Computational complexity	Reference
Gaussian elimination	$O(n^2)$	[117]
RU method	$O(n)$	[116]

The computational complexity of the Gaussian elimination and RU methods is given in Table XI, based on the results presented in [116][117]. As shown in the table, the complexity of the Gaussian elimination method is quadratic in the block length n. Instead, for the RU method, it was first proven in [116] that its complexity was upper bounded by $n+g^2$. Moreover, it was also proven that for all known "optimized" codes, gap g is less than $\sqrt{n}$, thus resulting in a linear encoding complexity with n.

Note that other alternative approaches have been proposed in the literature. For example, a low-complexity, high-throughput LDPC encoder design was proposed in [117] to overcome the limitation that the RU method suffers from a long critical path, which could make the LDPC encoder unsuitable for high throughput applications. Although [117] shows that this alternative approach requires significantly less area and memory storage while maintaining a high throughput, no comparison in terms of computational complexity is provided. Therefore, it is not considered in the forthcoming analysis.

2) LDPC Decoding

LDPC codes have attracted considerable attention because of their superior error correction capability using belief propagation (BP) decoding. BP decoding is conventionally performed by the repetition of the flood schedule, where all

TABLE XII
COMPUTATIONAL COMPLEXITY OF DIFFERENT LDPC DECODING ALGORITHMS

Algorithm	Complexity	Reference
Flooding	$I_{max}\ 2E$	[118]
Horizontal Shuffle (HS)	$I_{max}\ E(d_v+1)$	[118]
IVRBP	$I_{max}\ [m(m\cdot d_c-1)+E[(d_v-1)+(d_c-1)]+E(d_v-1)(d_c-1)]$	[118]
OVRBP	$I_{max}\ [n(n-1)+E(d_c+1)]$	[118]
LWNS	$I_{max}\ 2E\cdot d_c$	[118]
Min-Sum (MSA)	$I_{max}\ [(2\cdot n\cdot d_v + 2\cdot m)+ 2\cdot(2\cdot d_c -1)\cdot m]= I_{max}\ 6E$	[122]
Sum-Product (SPA)	$I_{max}\ [(2\cdot n\cdot d_v + m\cdot(2\cdot d_c-1))+6\ m\cdot d_c] =I_{max}\ (10E-m)$	[122]
RB-LBP	$I_{max}\ [3E+2(m(m-3)/2+E)]$	[120]
LPHD LBP	$I_{max}\ 2E$	[120]
RBP	$I_{max}\ [E(d_v-1)+E+E(d_v-1)(d_c-1)+2(E(E-1))]$	[120]
NW-RBP	$I_{max}\ [E(d_v-1)+E+E(d_v-1)(d_c-1)+2(m(E-1))]$	[120]
Conventional LBP	$I_{max}\ 2E$	[120]

variable-to-check and all check-to-variable messages are successively updated in parallel [118]. However, the convergence process is reduced as the latest updated information is not available until the next iteration. To accelerate the convergence and improve error correction performance, sequential scheduling methods were proposed, as presented in [119], with both predetermined and fixed sequences of updates. This sequential scheduling strategy is different from flooding in that the latest information is available in the current iteration. A typical sequential scheduling is layered BP (LBP), in which new information obtained from the upper layer can be immediately utilized by the lower layer. This approach randomly determines the order of the layers.

To achieve faster convergence performance, informed dynamic scheduling (IDS) strategies have also been introduced. One such strategy is residual belief propagation (RBP) decoding [120]. RBP decoding consists of a dynamically adjusted order of message updates based on the residual value defined as the difference between the current value and the old message value. In the node-wise RBP algorithm (NW RBP), the residual is calculated from the difference in the check-to-variable message values before and after an update. Based on different message selection and update strategies, several dynamic decoding algorithms have been reported [118]. These algorithms include the Informed Variable-to-Check Residual Belief Propagation (IVC RBP) algorithm, where the priority of message update is given to the most unstable variable node[3]; the Oscillating Variable nodes based Residual Belief Propagation (OVRBP), where stability metrics are employed based on the number of unsatisfied parity check equations of each variable node; the Horizontal Shuffle (HS), which uses a belief propagation method that updates the variable nodes in descending order of their column-weight [121]; or the layered vicinal variable node scheduling (LWNS) algorithm. In the LWNS algorithm, preprocessing is applied to each variable node to identify the subgraph to which it is attached. Based on this preprocessing, when a variable node is updated, it exchanges information with all its connected variable nodes before moving to the next update [118].

The decoding complexity of the 5G LDPC codes is reduced since they puncture the first two block columns of the parity-check matrix; however, this decreases the convergence rate. In [120], a fixed schedule is proposed that decodes the layers with the least-punctured edges and those with the highest-degree (LPHD algorithm), which has a much faster convergence speed than conventional schemes. As fixed scheduling cannot take full advantage of the dynamic changes in decoding messages, [120] also proposed using residual-based layered belief propagation (RB-LBP) to dynamically rearrange the layers among different iterations.

The computational complexity of different LDPC decoding algorithms is presented in Table XII in terms of the total number of operations. The number of additions, subtractions, multiplication, division, comparison or max (min) procedures and table look-up operations are considered. In general, most of these operations correspond to one equivalent addition, with the exception of the comparison operation or max(min) operation that in most cases corresponds to two equivalent additions and the look-up operation, which corresponds to six equivalent additions. The expressions of Table XII depend on the following parameters: n is the code length, $m=n-k$ is the number of parity bits, d_v is the average variable degree of the LDPC parity check matrix, d_c is the average check degree of the LDPC parity check matrix, $E=d_c.m=d_v.n$ is the total number of edges in the entire Tanner graph, and I_{max} is the maximum number of iterations. The lower complexity candidates are flooding, conventional LBP, LPHD LBP and LWNS.

Table XIII presents the performance of different LDPC decoding algorithms measured in terms of the minimum *Eb*/*No* to achieve a BER of 1E-3, based on the results presented in different references. Similarly, Table XIV presents the required number of iterations to achieve this BER value. The results from [118] show that the LWNS algorithm requires a lower *Eb*/*No* than OVRBP, IVRBP, HS and Flooding and the lowest number of iterations. The results from reference [120] reveal that NW-RBP and LPHD LBP achieve similar performance in terms of the required *Eb*/*No*, outperforming the SPA, RBP and

[3] A node is unstable if its sign before and after an update is reversed.

TABLE XIII
PERFORMANCE OF DIFFERENT LDPC DECODING ALGORITHMS IN TERMS OF REQUIRED EB/NO FOR A TARGET BER OF 1E-3

Algorithms	Conditions and values	Source
	code rate=0.75 (n=576, m=144), BPSK, AWGN channel	Fig. 7 from [118]
Flooding	3.5 dB	
HS	3.2 dB	
IVRBP	3 dB	
OVRBP	3 dB	
LWNS	2.7 dB	
	code rate=0.468, QAM, AWGN channel, I_{max}=30	Fig. 2 from [120]
SPA	2.2 dB	
LPHD LBP	1.95 dB	
RBP	2.45 dB	
NW-RBP	1.9 dB	
Conventional LBP	2.05 dB	

TABLE XIV
PERFORMANCE OF DIFFERENT LDPC DECODING ALGORITHMS IN TERMS OF THE REQUIRED NUMBER OF ITERATIONS TO ACHIEVE A TARGET BER OF 1E-3

Algorithms	Conditions and values	Source
	code rate=0.75 (n=576, m=144), BPSK, AWGN channel, *Eb/No*=3.5 dB	Fig. 10 from [118]
Flooding	5	
HS	3	
OVRBP	1	
LWNS	1	
	code rate=0.324, QAM, AWGN channel, *Eb/No*=2 dB	Fig. 4 from [120]
SPA	18	
LPHD LBP	8	
RBP	4	
NW-RBP	5	
Conventional LBP	9	

conventional LBP techniques. Moreover, NW-RBP is also able to work with a reduced number of iterations. Note that the differences in the simulation parameters used in the respective works of [120] and [118] do not allow a direct comparison between the two.

D. FFT/IFFT Functions

The FFT and IFFT are the most efficient implementations of the DFT and IDFT functions needed by OFDMA reception and transmission, respectively.

The implementation complexity of the FFT (and equivalently IFFT) function is defined in the literature as the sum of addition and multiplication counts. This sum is denoted as the flop count. In 1968, Yavne [123] presented what became known as the "split-radix" FFT algorithm, which exhibits an improvement of 20% over the classic "radix-2" algorithm that was previously presented by Cooley and Tukey in 1965 [124]. A modified version of the split-radix proposed in [125] lowers the flop count by ~5.6% without sacrificing numerical accuracy. In [126], by scaling the Twiddle Factor[4], the authors decrease the number of multiplication counts without affecting the number of additions. In addition, this modification also improves the signal-to-quantization noise ratio (SQNR) by more than 1.6 dB. In [127], the authors apply, first, a slight modification of Rader & Brenner's 'real-factor' FFT for Radix-4 and, second, a scaling operation to the Twiddle Factors so that the net computational complexity is reduced to the Standard Split Radix FFT. Although the number of arithmetic operations is not the sole factor in determining the time required to compute a DFT on a computer, the question of the minimum possible count is of longstanding theoretical interest. Currently, the lowest flops count is achieved by Johnson and Frigo [125]. Table XV summarizes the computational complexity achieved with the abovementioned methods, where N_{FFT} is the number of samples of the FFT.

E. Summary of Lessons Learned

The main lessons learned in this section are summarized as follows:

- This section has overviewed relevant state-of-the-art solutions for the most demanding BB functions, which include massive MIMO BB processing (detection, channel estimation, and precoding), demodulation, channel coding/decoding and FFT/IFFT. Expressions of the computational complexity for each BB function and algorithmic solutions have been presented based on information extracted from the literature. Similarly, comparative performance assessments have been discussed based on the compilation and compact presentation of results from different publications.
- The computational complexity of the BB processing functions in UL and DL is highly dependent on the specific algorithmic solutions selected by each function. Differences of more than one order of magnitude are observed with different algorithms. Since the selection of one or another algorithmic solution is implementation-dependent, it is possible to substantially reduce the complexity of a given functional split by properly choosing a convenient solution that offers a good trade-off between complexity and performance.

V. RAN FUNCTIONAL SPLIT: CHARACTERISATION AND DESIGN TRADE-OFFS

This section presents a system model to characterise the computational and fronthaul requirements of the different PHY layer processing functions that run on the BB resources depending on the considered functional split. A 5G RAN composed of different sites is assumed. One site includes

[4] In FFT Algorithms a "twiddle factor" is any of the trigonometric constant coefficients that are multiplied by the data in the course of the algorithm.

TABLE XV
COMPUTATIONAL COMPLEXITY OF DIFFERENT FFT IMPLEMENTATION STRATEGIES

Algorithm	Flop count	Reference
Standard Radix-2	$5\,N_{FFT}\log_2 N_{FFT}$	[124]
Proposed Radix-2	$(8/3)\,N_{FFT}\log_2 N_{FFT}$	[126]
Radix-4	$N_{FFT}\log_2 N_{FFT}$	[127]
Proposed Radix-4	$4\,N_{FFT}\log_2 N_{FFT}$	[127]
Split Radix	$4\,N_{FFT}\log_2 N_{FFT}-6\,N_{FFT}+8$	[123]
Modified Split-Radix	$\frac{34}{9}N_{FFT}\log_2 N_{FFT}-\frac{124}{27}N_{FFT}-2\log_2 N_{FFT}-\frac{2}{9}(-1)^{\log_2 N_{FFT}}\log_2 N_{FFT}+\frac{16}{27}(-1)^{\log_2 N_{FFT}}+8$	[125]

different cells that can belong to one or multiple sectors. Each cell operates with 5G NR over a certain frequency and bandwidth.

The BB resources are split between the BBH resources residing at a central location and the BBL resources residing near the radio units, as illustrated in Fig. 1. The physical equipment for each cell includes one antenna unit (AU) connected to an RU whose functions run at a BBL multiprocessor system. The BBH resources at the central location are provided by a certain number of boards offering a certain total computational capacity.

The FH network interconnects the different sites with the central location. Different topologies of the FH network can be considered depending on the scenario, with links of different capacities and different levels of aggregation of the data derived from multiple cells towards the BBH.

Based on the above considerations, a model of computational and bandwidth requirements for one cell is presented in this section. The system model assumes that a cell operates a channel with nominal bandwidth B_c and subcarrier spacing Δf, resulting in N_{PRB} PRBs [58]. The symbol duration is denoted as T_s and includes the useful symbol part, of duration $1/\Delta f$, and the cyclic prefix duration.

Multiuser massive MIMO is considered in both UL and DL, with a total of B antennas at the base station (sector) for this cell and U antennas in total for all the UE devices. Thus, it is assumed that U is the total number of layers that can be spatially multiplexed in UL or DL. The TTI duration is assumed to be 1 slot.

TDD duplexing is considered using a fixed configuration with a number of UL, DL and special slots. The slot duration is denoted as T_{slot}. Each UL and DL slot includes 14 symbols used entirely for UL or DL. The special slot includes DL symbols, guard symbols where nothing is transmitted and UL symbols. It is assumed that the UL symbols of the special slot are employed to transmit the SRS signals used for DL channel estimation. An example of a TDD configuration that follows the recommended structure by GSMA for the 3.5 GHz band in [128] is illustrated in Fig. 22. The configuration has three DL slots, one UL slot and one special slot in the middle. The special slot includes 10 DL symbols, two guard symbols and two UL symbols for transmitting the SRS per the assumption of this paper. This TDD structure is periodically repeated over time. To characterize this structure, we denote the ratio of UL slots as r_{UL} (r_{UL}=1/5 in the example), which equals the ratio of UL symbols

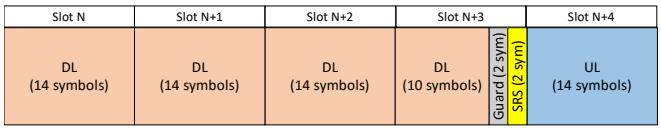


Fig. 22. Example of TDD configuration.

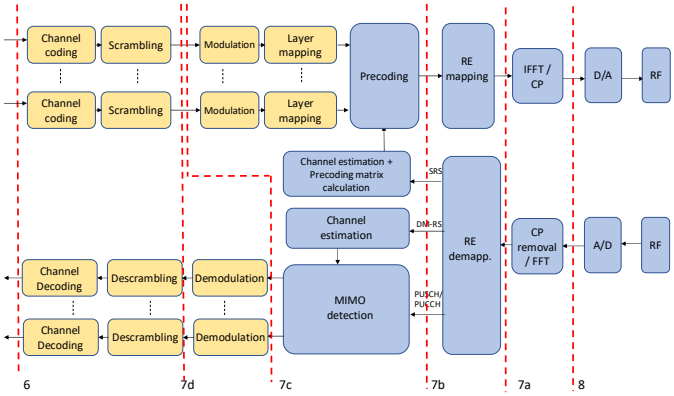


Fig. 23. Example functional splits considered in this paper.

excluding SRS, r_{DL} as the ratio of DL slots (r_{DL}=3/5 in the example), $r_{s,UL}$ as the ratio of UL symbols including the SRS ($r_{s,UL}$=16/70 in the example), $r_{s,DL}$ as the ratio of DL symbols ($r_{s,DL}$=52/70 in the example) and r_{SRS} as the ratio of SRS symbols (r_{SRS}=2/70 in the example). Moreover, we let T_{TDD} denote the number of slots that define the TDD structure repetition (T_{TDD}=5 slots in the example).

The traffic in the cell is characterized in terms of a certain TB generation rate for UL and DL, denoted λ_{UL} and λ_{DL} (packets/s), and a certain average TB size denoted L_{UL} and L_{DL} (bits). The total UL and DL average bit rates (b/s) are given by $R_{b,UL}=\lambda_{UL}\cdot L_{UL}$ and $R_{b,DL}=\lambda_{DL}\cdot L_{DL}$.

With all these considerations, this section presents a general model of the computational and bandwidth requirements of a cell based on the study of the individual BB functions presented in the previous section. The model intends to assess the computational requirements at the BBL and BBH and the fronthaul bandwidth requirements that characterize the operation of the system on average terms.

The functions hosted at BBH and BBL as well as the amount of data to be sent through the fronthaul depend on the selected functional split. Fig. 23 depicts the example functional splits that are considered in this paper based on the PHY layer

TABLE XVI
BB PROCESSING FUNCTIONS IN EACH FUNCTIONAL SPLIT AND COMPUTATIONAL REQUIREMENTS

Function		Required computations and periodicity	8	7a	7b	7c	7d	6
DL	IFFT	$c_{fft}(N_{FFT})$ operations (Table XV) for each DL OFDMA symbol and BS antenna. $C_{IFFT}= c_{fft}(N_{FFT})\cdot B\cdot r_{s,DL}/T_s$ (operations/s)	H	L	L	L	L	L
UL	FFT	$c_{fft}(N_{FFT})$ operations (Table XV) for each UL OFDMA symbol and BS antenna. $C_{FFT}= c_{fft}(N_{FFT})\cdot B\cdot r_{s,UL}/T_s$ (operations/s)	H	L	L	L	L	L
DL	Precoding (matrix multiplication)	$c_{prec}(B,U)=4\cdot B\cdot U$ operations for each subcarrier in each occupied PRB in a DL OFDMA symbol. $C_{PREC}= c_{prec}(B,U)\cdot 12\cdot N_{DL\_PRB}\cdot r_{s,DL}/T_s$ (operations/s)	H	H	H	L	L	L
DL	DL channel estimation and precoding matrix computation	$c_{dl\_ch\_est}(B,U,L_{SRS})$ operations for the channel estimation (Table V) and $c_{pr\_mat}(B,U)$ operations for the precoding matrix computation (Table VII). They are executed once per special slot and for each occupied PRB in DL. L_{SRS} is the length of the sequence of SRS symbols. $C_{DL\_CH\_EST}=[c_{dl\_ch\_est}(B,U,L_{SRS})+c_{pr\_mat}(B,U)]\ N_{DL\_PRB}/(T_{TDD}T_{slot})$ (operations/s)	H	H	H	L	L	L
UL	UL channel estimation	$c_{ul\_ch\_est}(B,U,L_{DM\text{-}RS})$ operations (Table V) executed once per UL slot and for each occupied PRB in UL. $L_{DM\text{-}RS}$ is the length of the sequence of DM-RS symbols. $C_{UL\_CH\_EST}=c_{ul\_ch\_est}(B,U,L_{DM\text{-}RS})N_{UL\_PRB}\cdot r_{UL}/T_{slot}$ (operations/s)	H	H	H	L	L	L
UL	MIMO detection	$c_{det}(B,U)$ operations (Table III) for each subcarrier of the PUSCH in each occupied PRB in an UL OFDMA symbol. $C_{DET}\approx c_{det}(B,U)\cdot 12\cdot N_{UL\_PRB}\cdot r_{UL}/T_s$ (operations/s) (the approximation assumes that the number of DM-RS symbols in the PRB is negligible with respect to the number of PUSCH symbols)	H	H	H	L	L	L
UL	Demodulation	$c_{dem}(m)$ operations (Table IX) for each UL layer of each subcarrier in each occupied PRB of an UL OFDMA symbol. m is the number of bits per symbol of the modulation. $C_{DEM}=c_{dem}(m)\cdot 12\cdot N_{UL\_PRB}\cdot U\cdot r_{UL}/T_s$ (operations/s)	H	H	H	H	L	L
DL	Scrambling	m multiplications (one per bit) per modulated symbol (i.e., for each DL layer of each subcarrier in each occupied PRB of a DL OFDMA symbol). $C_{SCR}=m\cdot 12\cdot N_{DL\_PRB}\cdot U\cdot r_{s,DL}/T_s$ (operations/s)	H	H	H	H	H	L
UL	Descrambling	m multiplications (one per bit) per demodulated symbol (i.e., for each UL layer of each subcarrier in each occupied PRB of an UL OFDMA symbol). $C_{DSC}= m\cdot 12\cdot N_{UL\_PRB}\cdot U\cdot r_{UL}/T_s$ (operations/s)	H	H	H	H	H	L
DL	Channel coding	c_{cod} operations (Table XI) for each code block segment in a DL transport block. $C_{COD}=c_{cod}\cdot\lambda_{DL}\lceil L_{DL}/8424\rceil\approx c_{cod}\cdot R_{b,DL}/8424$ (operations/s)	H	H	H	H	H	L
UL	Channel decoding	c_{dec} operations (Table XII) for each code block segment in an UL transport block when this code segment is erroneously received in the first HARQ transmission according to the BLock Error Rate (*BLER*). $C_{DEC}=c_{dec}\cdot BLER\cdot\lambda_{UL}\cdot\lceil L_{UL}/8424\rceil\approx c_{dec}\cdot BLER\cdot R_{b,UL}/8424$ (operations/s)	H	H	H	H	H	L

processing functions presented in Section III. The nomenclature of the splits is based on that considered by 3GPP in [6], which denotes the split between MAC and PHY layers as split 6, the intra-PHY splits as split 7 and the RF/PHY split as split 8. Moreover, since the examples here consider more intra-PHY splits than those considered by 3GPP, the terminology of these splits, 7a, 7b, 7c, and 7d, is specific to this paper.

A. Characterization of Computational Requirements

Table XVI summarizes the required computations, associated execution periodicity and computational requirements in operations/s for each of the BB processing functions in the UL and DL transmission chain corresponding to a single cell. This table is given in reference to the computational complexity expressions listed in the tables of Section IV. Only the computationally relevant functions are included here as the computational complexity associated with RE mapping/demapping, layer mapping or modulation can be disregarded (e.g., the modulation process can be performed with a look-up table to convert from groups of bits to I/Q components). Moreover, Table XVI also presents the location of each BB processing function either at BBH (H) or BBL (L) for each functional split. Using the information of the table, it is possible to obtain the required computational requirement (operations/s) at the BBH and BBL for each split by aggregating the functions located at the BBH or BBL.

Table XVI shows that some functions require a fixed number of operations (e.g., IFFT or FFT), which only depend on the

TABLE XVII
DATA TRANSMITTED THROUGH THE FH IN EACH FUNCTIONAL SPLIT AND BANDWIDTH REQUIREMENT

Split	UL transmitted data and periodicity	DL transmitted data and periodicity
8	$N_{FFT}+N_{CP}$ time domain IQ samples per UL OFDMA symbol and per BS antenna, where N_{FFT} is the FFT size and N_{CP} is the number of samples of the cyclic prefix. $F_{UL,8}(\text{b/s})=n_{IQ}\cdot(N_{FFT}+N_{CP})\cdot B\cdot r_{s,UL}/T_s$	$N_{FFT}+N_{CP}$ time domain IQ samples per DL OFDMA symbol and per BS antenna. $F_{DL,8}(\text{b/s})=n_{IQ}\cdot(N_{FFT}+N_{CP})\cdot B\cdot r_{s,DL}/T_s$
7a	One IQ sample per subcarrier and per BS antenna transmitted every UL OFDMA symbol. $F_{UL,7a}(\text{b/s})=n_{IQ}\cdot 12\cdot N_{PRB}\cdot B\cdot r_{s,UL}/T_s$	One IQ sample per subcarrier and per BS antenna transmitted every DL OFDMA symbol. $F_{DL,7a}(\text{b/s})=n_{IQ}\cdot 12\cdot N_{PRB}\cdot B\cdot r_{s,DL}/T_s$
7b	One IQ sample per subcarrier and BS antenna in each occupied UL PRB transmitted for each UL symbol of the UL slots. In addition, one IQ sample per subcarrier and BS antenna in each PRB for each UL symbol in the special slot carrying the SRS signals. $F_{UL,7b}(\text{b/s})=n_{IQ}\cdot 12\cdot N_{UL\_PRB}\cdot B\cdot r_{UL}/T_s+ n_{IQ}\cdot 12\cdot N_{PRB}\cdot B\cdot r_{SRS}/T_s$	One IQ sample per subcarrier and BS antenna in each occupied DL PRB transmitted for each DL symbol. $F_{DL,7b}(\text{b/s})=n_{IQ}\cdot 12\cdot N_{DL\_PRB}\cdot B\cdot r_{S,DL}/T_s$
7c	One IQ sample per subcarrier and per layer in each occupied UL PRB. Transmitted every UL OFDMA symbol in the UL slots. $F_{UL,7c}(\text{b/s})=n_{IQ}\cdot 12\cdot N_{UL\_PRB}\cdot U\cdot r_{UL}/T_s$	m bits per subcarrier and per layer in each occupied DL PRB. Transmitted every DL OFDMA symbol. $F_{DL,7c}(\text{b/s})=m\cdot 12\cdot N_{DL\_PRB}\cdot U\cdot r_{s,DL}/T_s$
7d	m soft bits obtained from the demodulator per subcarrier and per layer in each occupied PRB. Transmitted every UL OFDMA symbol in the UL slots. $F_{UL,7d}(\text{b/s})=n_{soft}\cdot m\cdot 12\cdot N_{UL\_PRB}\cdot U\cdot r_{UL}/T_s$	m bits per subcarrier and per layer in each occupied DL PRB. Transmitted every DL OFDMA symbol. $F_{DL,7d}(\text{b/s})=m\cdot 12\cdot N_{DL\_PRB}\cdot U\cdot r_{s,DL}/T_s$
6	UL transport blocks $F_{UL,6}(\text{b/s})=R_{b,UL}$	DL transport blocks $F_{DL,6}(\text{b/s})=R_{b,DL}$

FFT size (N_{FFT}), number of antennas and TDD configuration. In contrast, the remaining functions depend on the amount of traffic in the cell, which is reflected by the number of occupied PRBs (N_{DL_PRB} and N_{UL_PRB} in DL and UL, respectively). In this way, these functions offer some degree of freedom to dynamically select the most appropriate functional split depending on the traffic conditions in the different cells connected to one central location.

B. Characterization of Fronthaul Bandwidth Requirements

The FH bandwidth requirements depend on the amount of information that is being transmitted through the FH in each functional split and on the periodicity when this information has to be transmitted. Table XVII presents this information in UL and DL for each of the considered functional splits and the resulting bandwidth requirements in b/s. The table assumes that the different I/Q samples are sent through the FH using n_{IQ} bits per sample, while the demodulated soft bits are sent using n_{soft} bits per demodulated bit.

C. Performance Assessment

This section presents some results to illustrate the computational and fronthaul bandwidth requirements corresponding to the different functional splits. The considered scenario assumes a single cell characterized by the parameters shown in Table XVIII. The evaluation is performed for different values of the DL and UL bit rates $R_{b,DL}$ and $R_{b,UL}$, which correspond to the aggregate of all the UE devices connected to the cell. It is assumed that UL and DL bit rates are related as $R_{b,UL}=R_{b,DL}\cdot r_{UL}/r_{s,DL}$ in accordance with the number of UL and DL data symbols that are sent in the considered TDD structure. The number of occupied PRBs is computed as the PRBs needed to support the bit rate with the spectral efficiency associated with the selected modulation and coding scheme and the TDD structure, as indicated in the table.

To assess the variability in terms of computational requirements that can be obtained by using different algorithms for each of the BB functions, a comparison is performed between the best-case configuration, which consists of selecting for each BB function the algorithm of Section IV with the lowest computational complexity, and the worst-case configuration, which consists of selecting the algorithm with the largest complexity. The selected algorithms in each configuration are indicated in Table XIX. Fig. 24 illustrates the complexity in operations required by each BB function with the best- and worst-case configurations. Significant differences of several orders of magnitude are observed for some of the BB functions, such as for the channel coding and decoding or the DL channel estimation, which also includes the precoding matrix computation (note that the operations of the multiplication by the precoding matrix are excluded from Fig. 24 as they are the same in the best- and worst-case configurations). These algorithms are also the most demanding algorithms in terms of the total number of operations.

When assessing the computational requirements of the BB functions, in addition to the number of operations required by each function, we need to consider the periodicity when each BB function has to be executed in accordance with the characterization given in Table XVI. Fig. 25 and Fig. 26 depict the computational requirements in millions of operations per second (MOPS) for each BB function with the best-case configuration and worst-case configuration, respectively. The

TABLE XVIII
PARAMETERS CONSIDERED IN THE EVALUATION

Parameter	Value
Subcarrier spacing	Δf=30 kHz
Channel bandwidth and number of PRBs	100 MHz, N_{PRB}=273 PRBs
Slot duration	T_{slot}=0.5 ms
Symbol duration	$T_s=T_{slot}/14$=35.7 μs
IFFT/FFT size	N_{FFT}=4096
Number of time samples of cyclic prefix	$N_{CP}= N_{FFT}/14$=292
Number of antennas at the BS	B=64
Number of antennas at the UE devices (equivalently number of layers)	U=16
TDD structure configuration (Fig. 22)	T_{TDD}=5, r_{UL}=1/5, r_{DL}=3/5, $r_{s,UL}$=16/70, $r_{s,DL}$=52/70, r_{SRS}=2/70
Length of training sequences for channel estimation	L_{SRS}=12, $L_{DM\text{-}RS}$=8
Modulation and coding scheme	64 QAM (m=6 bits/symbol), coding rate r=666/1024
Spectral efficiency in DL and UL	$S_{DL}=m\cdot r\cdot U\cdot(14/15)\cdot r_{s,DL}$=43.29 b/s/Hz $S_{UL}=m\cdot r\cdot U\cdot(14/15)\cdot r_{UL}$=11.65 b/s/Hz
Number of occupied PRBs in DL and UL	$N_{DL\_PRB}=R_{b,DL}/(12\cdot\Delta f\cdot S_{DL})$ $N_{UL\_PRB}=R_{b,UL}/(12\cdot\Delta f\cdot S_{UL})$
Number of bits to encode an IQ sample	n_{IQ}=32
Number of bits to encode a softbit at the output of the demodulator	n_{soft}=8
Channel decoding parameters (Section IV.C.2)	$BLER$=0.1, I_{max}=10, d_c=2, n=8424/r=12952, d_v=0.699, E=4528

TABLE XIX
ALGORITHMS CONSIDERED IN THE EVALUATION

BB function	Best-case configuration	Worst-case configuration
FFT/IFFT	Radix-4 [127]	Standard Radix-2 [124]
UL Channel Estimation	Beamspace Local LMMSE [71]	ANM [92]
MIMO detection	Beamspace Local LMMSE [71]	Richardson [69]
DL channel estimation	SBEM [91]	ANM [92]
Precoding matrix computation	CSM [110]	MMSE
Demodulation	HDT [114]	ML [113]
Channel coding	RU [116]	Gaussian elimination [117]
Channel decoding	Flooding [118]	RBP [120]

results are presented for a total DL bit rate aggregated for all the users connected to the cell of 100 Mb/s and 1 Gb/s. The corresponding UL bit rates considering the abovementioned

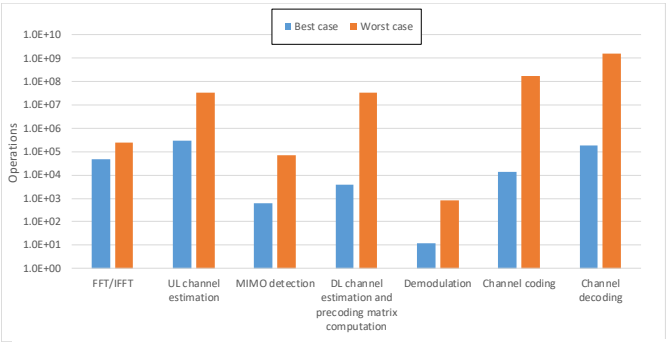


Fig. 24. Required operations by each BB function with the best- and the worst-case configurations.

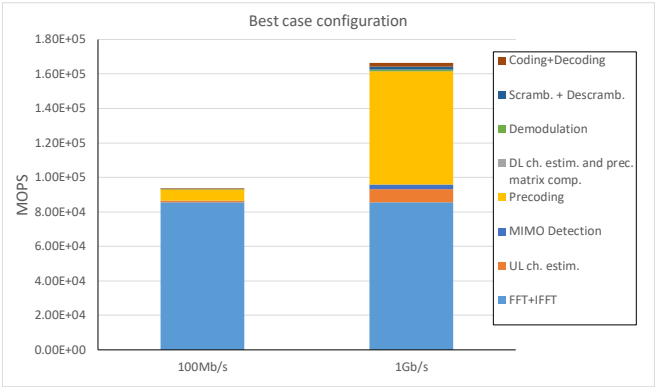


Fig. 25. Computational complexity in MOPS for each BB function with the best-case configuration for total DL bit rates of 100 Mb/s and 1 Gb/s.

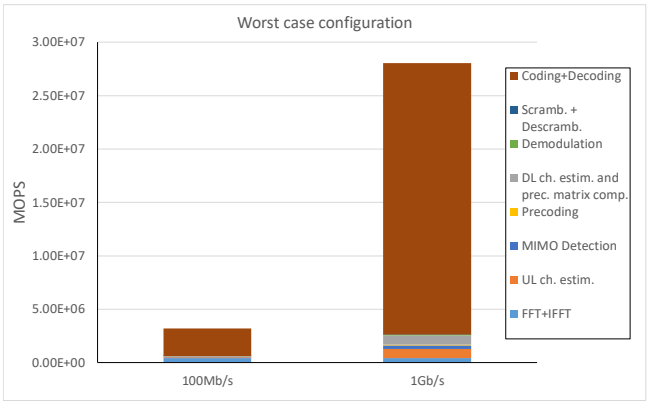
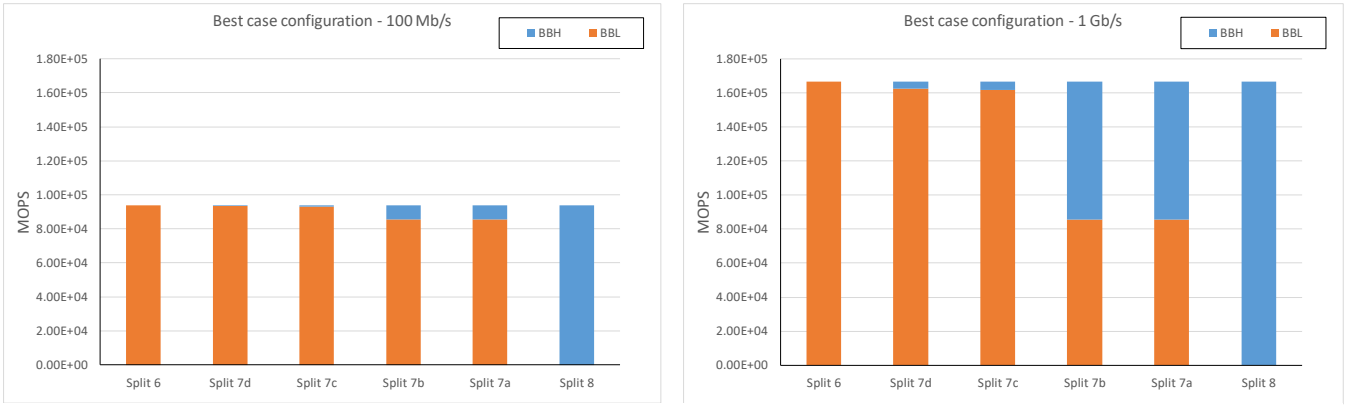


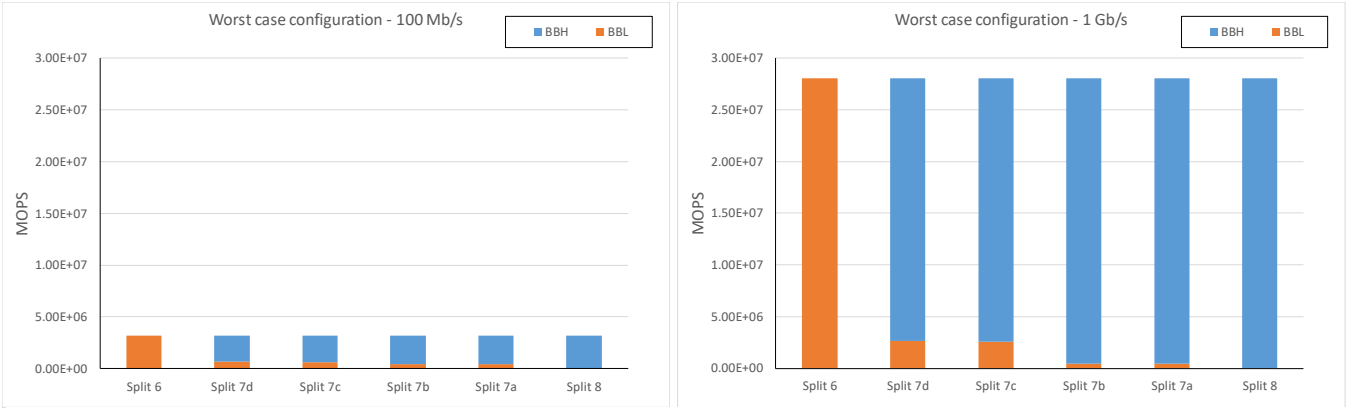
Fig. 26. Computational complexity in MOPS for each BB function with the worst-case configuration for total DL bit rates of 100 Mb/s and 1 Gb/s.

relationship based on the TDD configuration are approximately 27 Mb/s and 270 Mb/s. Overall, it is observed that the aggregate computational requirements of the worst-case configuration are approximately 170 times larger than those of the best-case configuration for a bit rate of 1 Gb/s and 34 times larger for a bit rate of 100 Mb/s. Examining the worst-case configuration in Fig. 26, it is observed that the most demanding BB function corresponds to channel coding and decoding, and its requirements are much larger than those of the other BB functions. This behaviour is attributed to the large number of operations performed by the RBP algorithm, which depends on E^2 or equivalently on the square of the codeword length, as



(a) (b)

Fig. 27. BBH and BBL computational requirements for the different functional splits with the best-case configuration for total DL bit rates of (a) 100 Mb/s and (b) 1 Gb/s.

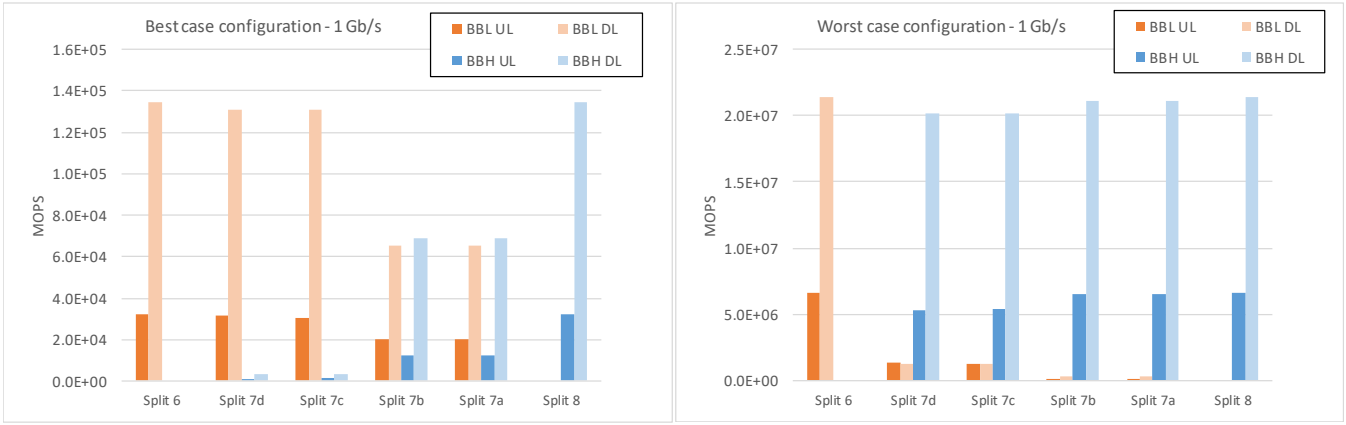


(a) (b)

Fig. 28. BBH and BBL computational requirements for the different functional splits with the worst-case configuration for total DL bit rates of (a) 100 Mb/s and (b) 1 Gb/s.

shown in Table XII. In contrast, according to the best-case configuration in Fig. 25, by choosing a more convenient decoding algorithm such as Flooding, the computational requirements are drastically reduced by a factor of approximately 12000, and the channel coding/decoding functions are no longer the most demanding functions. In the best-case configuration, the functions requiring the largest complexity are the FFT/IFFT and the precoding matrix multiplication, while all the other functions represent a small fraction of the total computational requirements. These functions have to be executed every symbol, while others, such as the channel estimation, are less frequently executed, i.e., on a time slot basis, and hence, fewer operations are needed. Note that the FFT/IFFT operations are executed considering all the subcarriers of the cell regardless of whether they are actually occupied, while the remaining operations, such as the precoding matrix multiplication, MIMO detection or demodulation, are executed only for the occupied subcarriers. As a result, the computational requirements are dependent on the total bit rate. Fig. 25 shows that for the 100 Mb/s case, the computational requirements are mainly driven by the FFT/IFFT functions. When the total bit rate is increased to 1 Gb/s, the remaining functions, particularly the precoding matrix multiplication, constitute a significant part of the total computational complexity.

The results shown in Fig. 25 and Fig. 26 also determine the BBL and BBH computational requirements depending on where each BB function is executed in a given functional split, as detailed in Table XVI. Fig. 27 and Fig. 28 plot the corresponding BBL and BBH computational requirements for each of the functional splits from Table XVI and consider the best-case configuration and worst-case configuration, respectively. In the best-case configuration, with a DL data rate of 100 Mb/s, as shown in Fig. 27a, most of the computational burden remains at the BBL for most of the splits. For this data rate, the highest computationally demanding BB function is the FFT/IFFT, which is executed at the BBL in all the splits, with the exception of split 8. In contrast, when considering a much higher DL data rate of 1 Gb/s, the higher PRB and subcarrier occupation increases the computational requirement of the remaining processing functions, such as precoding or channel estimation. Correspondingly, there is more room to balance the computational load between the BBH and BBL. For example, this occurs with splits 7a and 7b, in which the BBL



(a) (b)

Fig. 29. BBH and BBL computational requirements in UL and DL for the different functional splits for a total DL bit rate of 1 Gb/s with the (a) best-case configuration and (b) worst-case configuration.

computational requirements are associated with the FFT/IFFT, while the BBH computational requirements are mainly driven by the precoding matrix multiplication function.

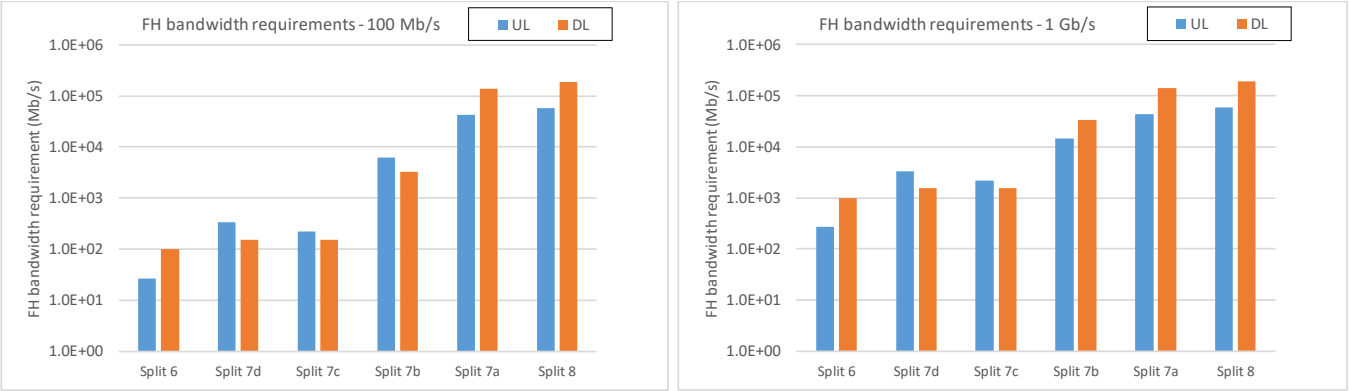
In the worst-case configuration, most of the computational complexity burden corresponds to the channel coding and decoding processes. As shown in Fig. 28, this configuration leads to the opposite behaviour of the best-case configuration, as now the highest computational requirements are at the BBH for most of the functional splits, with the exception of split 6, in which the channel coding/decoding is hosted at the BBL.

To gain insight into the different contributions of the UL and DL BB functions to the computational requirements, Fig. 29 plots the BBH and BBL computational requirements for each split distinguishing between the UL and the DL. The results include the best- and worst-case configurations for a DL bit rate of 1 Gb/s. The specific BB functions considered for determining the UL and DL requirements are indicated in the first column of Table XVI.

According to Fig. 29, the DL functions represent the most important contribution to the BBH/BBL requirements in most of the splits. The main reason is that the considered TDD structure has more DL symbols than UL symbols, meaning that DL functions have to be executed more frequently than UL functions. However, depending on the split and best/worst-case configuration, some differences are noticed. Starting with extreme splits 6 (all functions at the BBL) and 8 (all functions at the BBH), the best-case configuration in Fig. 29a reflects that the computational requirements of UL functions are approximately 24% of the DL function requirements. However, in the worst-case configuration of Fig. 29b, this percentage increases to approximately 31%, mainly due to the highest contribution of the UL channel decoding algorithm. For the intermediate splits 7a and 7b, the BBL just hosts the FFT/IFFT functions. In this case, the percentage of BBL UL requirements with respect to the BBL DL requirements is directly given by the rate of UL symbols to the DL OFDMA symbols of the TDD structure (i.e., Eq. $r_{s,UL}/r_{s,DL}$=30.7%). Thus, this percentage is the same in the best- and worst-case configurations. However, when considering splits 7a and 7b in the BBH, with the best-case configuration, the DL contribution increases mainly due to the matrix precoding multiplication, resulting in the BBH UL requirements being just 17% of the BBH DL requirements. In contrast, with the worst-case configuration, this percentage increases to 31% due to the higher contribution of the UL channel decoding. For splits 7c and 7d with the best-case configuration, the BBL hosts most of the computational burden, and in this case, the UL BBL requirements are approximately 24% of the DL BBL requirements. With the worst-case configuration, the BBH assumes most of the requirements, and the UL BBH is approximately 26% of the DL BBH. For this configuration, the UL BBL and DL BBL requirements are quite similar, mainly as in these splits, the requirements are driven by the UL and DL channel estimation functions, which require a similar amount of MOPS as this worst-case configuration. In any case, the total BBL requirements are much smaller than those of the BBH.

Fig. 30 plots the FH bandwidth requirements in the UL and DL for the considered functional splits and for DL data rates of 100 Mb/s and 1 Gb/s. The largest requirements correspond to splits 8 and 7a, with 187 Gb/s and 139 Gb/s, respectively, in the DL. The difference between both splits is attributed to the notion that in split 8, the FH carries the samples of the signals in the time domain, thus including the cyclic prefix samples and a total of N_{FFT} samples per symbol. In contrast, in split 7a, the FH carries the samples in the frequency domain, so it only accounts for the actual number of subcarriers that fall inside the cell bandwidth, which is smaller than N_{FFT}, and does not include the overhead associated with the cyclic prefix. In the UL, the FH requirements with splits 8 and 7a are 57 Gb/s and 43 Gb/s, respectively. These values are lower than the DL requirements because of the lower number of UL slots in the TDD frame structure.

Note that the requirement in splits 8 and 7a is independent of the actual data rate as no distinction is made between occupied subcarriers and unused subcarriers (i.e., in split 8, the number of samples is given by the FFT size and the number of samples of the cyclic prefix, and in split 7a, it is given by the number of subcarriers in the cell regardless of whether they are occupied



(a) (b)

Fig. 30. FH bandwidth requirements for the different functional splits for total DL bit rates of (a) 100 Mb/s and (b) 1 Gb/s.

or unused). In contrast, when moving to splits 7b, 7c, etc., the FH only transmits information corresponding to occupied subcarriers, so the results differ when the DL bit rate is 100 Mb/s (Fig. 30a) and 1 Gb/s (Fig. 30b).

The FH bandwidth requirement with split 7b is higher than with splits 7c, 7d or 6. The reason is that in split 7b, the fronthaul separately carries symbols for each BS antenna, while in splits 7c, 7d or 6, the information corresponds to the different layers U with $U<<B$. Note that in split 7b with a DL rate of 100 Mb/s, the UL FH requirement is slightly higher than the DL FH requirement (Fig. a), while it is the opposite in the case of 1 Gb/s (Fig. b). The reason for this difference is that the DL of split 7b only carries the symbols of the occupied DL subcarriers, while in the UL, it carries the symbols of the occupied UL subcarriers and the SRS signals of the special slot that span across the whole channel even if no UL data are transmitted on the corresponding subcarrier of the UL slot (see expressions in Table). For the case of 1 Gb/s, the impact of the occupied subcarriers predominates over the SRS signals, while for the lower rate case of 100 Mb/s, there are fewer occupied subcarriers. Thus, the term of the SRS signals has more relevance, leading to an opposite behaviour.

An examination of splits 7c and 7d reveals that the DL FH requirement is the same in both cases as the FH transmits the bits of the different layers in both splits. In contrast, the UL FH requirement is slightly larger with split 7d than with split 7c. With split 7c, the UL FH carries the symbols prior to demodulation, with n_{IQ}=32 bits per symbol, while with split 7d, the UL FH carries the soft bits at the output of the demodulator, resulting in $n_{soft} \cdot m$=8·6=48 bits per symbol.

The lowest FH requirement is obtained with split 6 as in this case, only the bits of the different TBs are transmitted through the FH, while in the remaining splits, the bits are associated with TB bits and coding redundancy.

Overall, the illustrative results presented in this section reflect that substantial variations may occur in the computational and FH bandwidth requirements depending on the specific algorithm selected for each BB function, the functional split and the existing data rate in a cell. Therefore, the dynamic optimization of the functional split requires smart optimization mechanisms that are able to properly select and configure the splits for each cell in accordance with the existing conditions. It is thus envisaged that this topic will constitute a research area of interest for the forthcoming years.

Note that the computational complexity analysis presented in this paper is hardware-independent, that is, we make no assumptions about the characteristics of the underlying hardware that will execute each of the aforementioned BB functions. As a result, the analysis implicitly assumes a homogeneous hardware platform, i.e., all BB functions are mapped to processing elements of the same type, e.g., digital signal processors (DSPs) or field programmable gate arrays (FPGAs). This assumption allows a direct comparison between the computational complexity of one function against another, which exhibits substantial variations, as reflected in the paper. However, if a heterogeneous hardware platform should be considered, the mapping of each function onto different processor types (DSPs, accelerators, etc.) introduces a hardware-dependent weighting of the computational complexity, that is, reflecting the notion that the same operation (e.g., a multiplication) would induce a different cost when mapped to different processor types. For example, some processor types may be more energy-efficient, whereas other processor types might support more programmability. Of course, this condition is highly dependent on the adopted hardware implementation, which is beyond the scope of this work and therefore omitted from our analysis for the sake of generality.

D. Summary of Lessons Learned and Challenges in RAN Functional Split Optimization

The main lessons learned in this last part of the tutorial are summarized in the following section:

- In general, for a given functional split, the computational requirements at the BBL and BBH depend on the computational complexity of the algorithmic solution applied in each BB function and on the number of executions needed for each BB function. This number of executions depends on not only the time periodicity of the BB function in accordance with a given TDD frame structure but also the number of occupied resources (subcarriers) on which the BB function has to be executed

in accordance with a certain load of the cell. Based on these considerations, this section has presented a model for numerically assessing the computational requirements of each BB function. This model is then employed to obtain the total requirements at the BBL and BBH for several low layer splits.

- The fronthaul bandwidth requirements depend on the amount of data that has to be exchanged between the BBH and the BBL in accordance with the two BB functions that define the splitting point in a functional split. Depending on the split, the amount of data can significantly vary, as the lower splits (e.g., splits 8, 7a or 7b) require an exchange of data per base station antenna, while the upper splits (e.g., 7c, 7d or 6) require an exchange of data per spatial layer/user, which is much lower than the number of antennas. This section presents a model for numerically estimating the UL and DL fronthaul bandwidth requirements for different splits.
- The assessment of the presented model in a specific scenario reveals that important variations in computational requirements arise depending on the selected algorithms for each BB function. In particular, for a total downlink bit rate of 1 Gb/s, it was observed that the worst-case configuration of algorithms leads to 170 times higher computational requirements than the best-case configuration. This configuration is particularly critical for some functions, such as channel decoding, in which the difference in computational complexity between the best-case algorithm and the worst-case algorithm can be almost four orders of magnitude.
- The selection of the BB algorithms significantly impacts the workload distribution between the BBH and the BBL since for the worst-case configuration, most of the computational requirements are derived from the channel coding and decoding, which are hosted at the BBH for most of the considered splits. For the best-case configuration, most of the requirements are derived from the IFFT/FFT functions, which are primarily hosted at the BBL.
- The current traffic or load conditions in a cell substantially impact the BBH/BBL computational requirements of a functional split. The reason is that some functions, such as the IFFT/FFT, require a fixed number of operations regardless of the amount of traffic in the cell, while for most of the other functions, such as MIMO detection, precoding or channel estimation, the number of operations depends on the PRB occupation and associated traffic in the cell. For low loads, whose requirements are limited by the IFFT/FFT, there are few variations between the BBH and the BBL when changing among the intra-PHY functional splits, and differences only arise when moving from an intra-PHY split to a PHY-MAC split 6. In contrast, when increasing the cell load, the higher contribution of channel estimation or precoding functions to the computational requirements provides more room to balance the load between the BBH and the BBL by changing the intra-PHY functional split.
- The fronthaul bandwidth requirements experience significant variations depending on the functional split. The most demanding splits are the PHY-RF split 8 and the lowest intra-PHY split 7a, whose requirements are independent of the cell load. Moving the split upper in the PHY layer tends to decrease the bandwidth requirement and make it dependent on the load conditions.

The above findings reflect that there are different dimensions that impact the adequate selection of the functional split for a cell. These dimensions are the load level in the cell, the BBH/BBL computational requirements in relation to the actual capacity of BBH/BBL platforms, the fronthaul bandwidth requirements in relation to the fronthaul topology and the capacity of each link, and the algorithms used for each BB function. Therefore, the dynamic optimization of the RAN functional split requires smart algorithms that are able to properly trade-off all these components to dynamically configure the splits of each cell and even the selected BB algorithms, depending on the existing conditions and in response to the time-varying nature of factors such as the load demand affected by the user density and service mix.

A relevant aspect to consider when performing dynamic functional split optimization is the target of such optimization. For example, optimizing the workload distributions across the different BBH/BBL platforms would enable the switch-off of certain processors for energy saving purposes during certain periods of the day. Therefore, the functional split could be optimized to minimize the total energy consumption across the BBH and BBL. Similarly, other targets could consider the maximization of the degree of centralization, under the rationality that centralization facilitates coordination among cells, e.g., for CoMP. In this case, the dynamic functional split should intend to move as many BB functions as possible to the BBH subject to fulfilling some bounds in the fronthaul requirements.

Fronthaul topologies normally include links with different levels of aggregation. For example, some fronthaul links will carry aggregated data from multiple cells of the same site, other links will aggregate multiple sites, etc. In this context, the statistical multiplexing gain resulting from nonhomogeneous traffic loads at different cells can be exploited for optimizing the fronthaul network by properly adapting the fronthaul bandwidth requirements via cell-dependent functional split selection. For example, in a fronthaul link that aggregates several cells, the less loaded cells could use different splits than the high loaded cells, so that the aggregate fronthaul data rate requirement is minimized. This optimization would reduce the fronthaul deployment and maintenance costs for MNOs.

The development of a dynamic functional split optimization solution should also analyse the most convenient algorithmic tools for a specific problem. The many dimensions that impact decisions suggest the use of artificial intelligence/machine learning-based solutions, e.g., in the form of supervised learning or reinforcement learning. However, the complexity

associated with this type of solution and the need to properly train the solutions needs to be balanced in relation to the actual achievable performance. Depending on this trade-off, the use of simpler heuristic algorithms could become more attractive even if in some cases they may lead to suboptimal configurations.

The time scale of operation of the RAN functional split optimization process also deserves attention. Modifying the functional split in the short term (e.g., seconds) would facilitate adaptation to highly varying traffic demands. However, this modification would be at the cost of frequent movements of BB functions between the BBH and the BBL, thus incurring delays associated with the instantiation or termination of the BB functions (e.g., implemented as virtualised network functions) in one or another platform. Therefore, these constraints have to be traded off with respect to the potential benefits of a fast adaptation capability to identify an adequate operation time scale.

Another challenge is related to the architectural implementation to support this dynamic RAN functional split optimization. This implementation could be part of a self-organizing network (SON) function in charge of automatically modifying the per-cell functional split so that SON frameworks such as those considered in the 3GPP or SCF in [129][130] could be considered. Similarly, the possibility of integrating the functional split optimization as part of the RAN Intelligent Controller (RIC) of the O-RAN architecture [38] can also be envisaged following the trends of O-RAN and SON integration discussed in works such as [131][132]. In this case, the dynamic RAN functional split optimization algorithm could be implemented as an rApp of the nonreal-time RIC or as an xApp of the near-real-time RIC depending on the time scale of operation.

The complexity associated with how to incorporate different trade-offs when developing practical solutions, combined with the potential benefits for MNOs in terms of more energy-efficient RANs and less fronthaul costs, foresees that dynamic RAN functional split optimization will constitute a promising research area for the forthcoming years. This will be important for The consolidation of 5G network deployments and their evolution beyond 5G and 6G.

VI. Conclusions

This paper has presented a tutorial on the characterization of functional splits for flexible RANs, in which part of the base station functions run on the BBL platform near the cell sites while other functions are centralized at a BBH platform. After summarizing the efforts conducted by different industrial fora and standardization in relation to functional splits and comparing them in terms of equivalences and terminologies, the paper has focused on the low-layer functional splits that involve the BB processing functions at the PHY layer. The paper has presented a detailed overview of these PHY layer functions in the transmission and reception chain of a 5G NR base station supporting massive MIMO and using as a reference a TDD duplexing mode. In downlink transmission, these processes include channel coding, scrambling, modulation, layer mapping, MIMO precoding, resource element mapping, IFFT, cyclic prefix insertion and D/A conversion. In the uplink reception, the processes are essentially the counterparts of the downlink processes, namely, A/D conversion, cyclic prefix extraction, FFT, resource element demapping, channel estimation, MIMO detection, demodulation, descrambling and channel decoding. Each BB process has been described by presenting their inputs, the operations that they conduct and the resulting outputs.

Following the general description of the BB processes, the paper has presented a comprehensive and harmonized analysis of the computational requirements and performance evaluation of different solutions in the literature for the most demanding processes, namely, massive MIMO processing functions (detection, channel estimation, precoding), M-QAM demodulation, channel coding and decoding and FFT/IFFT processes. In particular, the paper has categorized the different algorithmic solutions of each BB function, gathering expressions of the computational complexity of each algorithm and collecting illustrative performance metrics that enable their comparison. This type of analysis is fundamental for properly assessing the implications of one or another functional split in terms of computational complexity requirements for the BBH and BBL.

Based on the analysis of each BB function, the paper has provided a system model to characterize the computational complexity and fronthaul bandwidth requirements of different functional splits. This model accounts for the periodicity of execution of each BB function and for the amount of data that has to be delivered via the fronthaul in each functional split. The presented system model has been used to derive some illustrative results of the computational complexity and fronthaul bandwidth requirements for different functional splits and different configurations of the algorithms selected for each BB function.

Overall, the obtained results reflect that the adequate selection of the functional split for a cell needs to trade-off the cell load conditions, BBH/BBL computational requirements, fronthaul bandwidth requirements and the algorithms used for each BB function. Therefore, the dynamic optimization of the functional split requires smart algorithms that are able to properly balance all these components to configure the splits of each cell and even the selected BB algorithms depending on the existing conditions. In this way, it will be possible to optimize the workload distributions across the different BBH/BBL platforms, enabling, for example, the switch-off of certain processors for energy saving purposes or the optimization of the fronthaul network, thus leading to benefits for MNOs in terms of cost reduction. These aspects foresee that dynamic functional split optimization will constitute a promising research area for the forthcoming years.

Appendix: Computational Complexity of the BSCE method

The estimation of the computational complexity of the BSCE channel estimation method from [94] is presented here,

considering the different steps of the algorithm. The notation described in Section III.B.4 is applied.

Step A) LS Channel Estimation

This step estimates the channel frequency responses for each antenna. The estimation for the u-th UE antenna and b-th BS antenna is given by

$$\hat{h}_{u,b}^{LS} = \frac{p_u^*}{|p_u|^2} t_b, \quad (14)$$

where p_u is the pilot sent in the u-th UE antenna, t_b is the pilot received in the b-th BS antenna and * represents the complex conjugation.

Assuming U users currently in the system and that the ratio $p_u^*/|p_u|^2$ is a priori known so that it can be obtained from tables, the complexity of step A (number of real multiplications here) to compute all channel estimates for all the B BS antennas and U UE antennas and considering that the training sequence has length L is $C_A=4\cdot B\cdot U\cdot L$.

Step B) Antenna-to-Beam Space Transformation

The channel estimator transforms the $B\times 1$ vector of the channel estimates of the previous step $\hat{\mathbf{h}}_u^{LS}=\left[\hat{h}_{u,1}^{LS},\ldots,\hat{h}_{u,B}^{LS}\right]^T$ in antenna space for each antenna user u to the beamspace vector $\tilde{\mathbf{h}}_u^{bs}=\left[\tilde{h}_{u,1}^{bs},\ldots,\tilde{h}_{u,N_B}^{bs}\right]^T$ of size $N_B\times 1$, where N_B is the number of beams. This transformation is conducted by the following expression

$$\tilde{\mathbf{h}}_u^{bs} = \mathbf{A}^H \hat{\mathbf{h}}_u^{LS}. \quad (15)$$

$\mathbf{A}$ is the $B\times N_B$ space transform matrix, where a DFT matrix is often used, in which $N_B=B$. When the base station has two-dimensional array antennas, the space transform matrix is given by the Kronecker product of DFT matrices in the horizontal and vertical directions with $B = B_{T\text{-}x}\cdot B_{T\text{-}z}$, where $B_{T\text{-}x}$ and $B_{T\text{-}z}$ are the number of antennas in the horizontal direction and vertical direction, respectively. Note that matrix $\mathbf{A}$ can be precalculated as it is only dependent on the antenna array distribution.

Equivalently, in matrix notation considering the channel estimates for all the users, the transformation (15) is expressed as

$$\tilde{\mathbf{H}}^{bs} = \mathbf{A}^H \hat{\mathbf{H}}^{LS}, \quad (16)$$

where $\hat{\mathbf{H}}^{LS}$ is the $B\times U$ matrix where the u-th column is vector $\hat{\mathbf{h}}_u^{LS}$ and $\tilde{\mathbf{H}}^{bs}$ is the $N_B\times U$ matrix in the beamspace where the u-th column is vector $\tilde{\mathbf{h}}_u^{bs}$.

To carry out the transformation (16), it is required to perform the product of the $N_B\times B$ matrix $\mathbf{A^H}$ by the $B\times U$ matrix $\hat{\mathbf{H}}^{LS}$ Assuming complex samples, the complexity of step B becomes $C_B=4\cdot B\cdot N_B\ U$ real multiplications.

Step C) Beam Selection

The channel estimator selects the nonzero beams from the B beams and sets the channel estimates of unselected beams to zero. This approach assumes that the number of dominant beams is small and that setting the channel responses of nondominant beams to zero can reduce the channel estimation errors. The paper [94] assumes, for simplicity, a selection process based on selecting beams whose magnitudes are higher than a threshold. In this case, the estimate of the channel frequency response between the u-th antenna of the user and the b-th beam of the BS after beam selection is given by

$$\hat{h}_{u,b}^{bs} = \begin{cases} \tilde{h}_{u,b}^{bs} & \text{if } \left|\tilde{h}_{u,b}^{bs}\right|^2 > \eta\sigma^2 / P_{P,u} \\ 0 & \text{otherwise} \end{cases}, \quad (17)$$

where η is a coefficient with a positive number, σ^2 is the noise power at the gNB, and $P_{P,u}$ is the power of the reference signal from the u-th user antenna, which is a known value according to specifications. In addition, η is set to a value optimized for maximizing throughput.

The complexity in terms of multiplications is given by the computation of $\left|\tilde{h}_{u,b}^{bs}\right|^2$ for each of the U user antennas and N_B beams. The computation of a modulus requires two real multiplications (i.e., multiplication of the two real parts and two imaginary parts). Then, the complexity of step C becomes $C_C=2N_B\cdot U$.

Step D) Beam-to-Antenna Space Transformation

The channel estimator calculates the eigenvectors in beamspace by means of the singular value decomposition (SVD) or eigenvalue decomposition (EVD) of the matrix whose elements are the channel estimates of the selected beams (i.e., beams b with elements $\hat{h}_{u,b}^{bs}$ different from 0). If the number of selected beams M is less than the dimension in antenna space B, the SVD or EVD in beamspace will require lower computational complexity than in antenna space. Moreover, the number of selected beams M should be at least the number of user antennas, i.e., $M \geq U$.

We define $\hat{\mathbf{H}}_{bs}$ as the $U\times M$ matrix whose elements are the channel estimates $\hat{h}_{u,b}^{bs}$ of the selected beams. The channel estimator calculates the M-dimensional eigenvectors by performing SVD of this matrix or EVD of $\hat{\mathbf{H}}_{bs}^H\hat{\mathbf{H}}_{bs}$. We denote $\hat{\mathbf{v}}_u^{bs}$ as the u-th eigenvector of dimension M. There are a total of U eigenvectors, $1\leq u\leq U$. From each eigenvector $\hat{\mathbf{v}}_u^{bs}$, the N_B-dimension eigenvector $\mathbf{v}_u^{bs}$ is obtained by keeping the same value of $\hat{\mathbf{v}}_u^{bs}$ for the components of $\mathbf{v}_u^{bs}$ associated with the selected beams and by inserting zeros for the components of $\mathbf{v}_u^{bs}$ associated with unselected beams. The channel estimator transforms the beamspace eigenvector $\mathbf{v}_u^{bs}$ to the antenna space eigenvector $\mathbf{v}_u$ using the following transformation

$$\mathbf{v}_u = \mathbf{A}^*\mathbf{v}_u^{bs} / \| \mathbf{A}^*\mathbf{v}_u^{bs} \|, \quad (18)$$

where normalization is performed to set the norm of each B-dimensional eigenvector to one. The result of this transformation is a total of U eigenvectors with dimension B.

According to [133], the required number of real multiplications for carrying out the SVD of the $U\times M$ complex matrix is $4(3UM^2+(2C+2)M^3)$ with C between 2 and 4 for extended precision. We multiplied by four the complexity of the real matrix defined at [133] to consider complex value operations. Regarding transformation using the $B\times N_B$ matrix $\mathbf{A}^*$, for each of the U eigenvectors $\mathbf{v}_u^{bs}$ of dimension $N_B\times 1$, we

need to carry out $4 \cdot B \cdot N_B$ real multiplications. Additionally, complex vector normalization requires: (a) calculating the norm: assuming that the $B \times 1$ vector $\mathbf{A}^* \mathbf{v}_u^{bs}$ has already been computed in the previous multiplication, the norm requires $2B$ multiplications and a square root (comparable to one multiplication), (b) inverting the norm requires one division (comparable to one multiplication) and (c) normalizing all complex vector components requires $2B$ multiplications by the inverse of the norm. Therefore, normalization requires $2B + 1 + 1 + 2B = 4B + 2$ operations. Then, the total number of real multiplications for each eigenvector is $4 \cdot B \cdot N_B + 4B + 2$, and considering that there are U eigenvectors, this yields $(4 \cdot B \cdot N_B + 4B + 2)U$. Then, the total complexity of step D becomes $C_D = 4(3UM^2 + (2C+2)M^3) + (4BN_B + 4B + 2)U$. This calculation considers that the SVD is executed only once for all the U user antennas.

The total complexity of the BSCE estimator is $C_A + C_B + C_C + C_D$, which yields $4BUL + 4BN_BU + 2N_BU + 4(3UM^2 + (2C+2)M^3) + (4BN_B + 4B + 2)U = 4(3UM^2 + (2C+2)M^3) + (8BN_B + 4B + 4BL + 2N_B + 2)U$.

For the usual case of $N_B = B$, the expression becomes $4(3UM^2 + (2C+2)M^3) + (8B^2 + 6B + 4BL + 2)U$.

REFERENCES

[1] "C-RAN: the road towards green RAN. Version 2.5", China Mobile, White Paper, Oct. 2011.

[2] C-L. I, J. Huang, R. Duan, C. Cui, J. Jiang, and L. Li, "Recent Progress on C-RAN Centralization and Cloudification", *IEEE Access*, Aug. 2014, pp. 1030-1039.

[3] M. S. J. Solaija, H. Salman, A. B. Kihero, M. I. Saglam, and H. Arslan, "Generalized Coordinated Multipoint Framework for 5G and Beyond", *IEEE Access*, Vol. 9, 2021, pp. 72499-72515.

[4] V. Quintuna Rodriguez, F. Guillemin, A. Ferrieux, and L. Thomas, "Cloud-RAN functional split for an efficient fronthaul network" in *Proc. International Wireless Communications and Mobile Computing (IWCMC)*, 2020.

[5] L. M. P. Larsen, A. Checko, and H. L. Christiansen, "A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, 2019, pp. 146–72.

[6] *Study on new radio access technology: Radio access architecture and interfaces (Release 14), v14.0.0*, 3GPP Technical Report TR 38.801, Mar. 2017.

[7] M. Peng, Y. Sun, X. Li, Z. Mao, and C. Wang, "Recent Advances in Cloud Radio Access Networks: System Architectures, Key Techniques and Open Issues", *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, 3rd Quarter 2016, pp. 2282-2304.

[8] I. A. Alimi, A. L. Teixeira, and P. Pereira Monteiro, "Toward an Efficient C-RAN Optical Fronthaul for the Future Networks: A Tutorial on Technologies, Requirements, Challenges, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, 2018, pp. 708–769.

[9] B. Khan, N. Nidhi, H. OdetAlla, A. Flizikowski, A. Mihovska, J-F. Wagen, and F.J. Velez, "Survey on 5G Second Phase RAN Architectures and Functional splits", *TechRxiv*, Oct. 2022. Available: https://doi.org/10.36227/techrxiv.21280473.v1

[10] D. Sabella *et al.*, "RAN as a Service: Challenges of Designing a Flexible RAN Architecture in a Cloud-based Heterogeneous Mobile Network", in *Proc. Future Network & Mobile Summit*, 2013.

[11] P. Rost *et al.*, "Cloud Technologies for Flexible 5G Radio Access Networks", *IEEE Communications Magazine*, May 2014, pp.68-76.

[12] A. Maeder *et al.*, "Towards a Flexible Functional Split for Cloud-RAN Networks," in *Proc. IEEE European Conference in Networks and Communications (EuCNC)*, 2014.

[13] A. Maeder *et al.*, "A Scalable and Flexible Radio Access Network Architecture for Fifth Generation Mobile Networks", IEEE Communications Magazine, Nov. 2016, pp. 16-23.

[14] D. Harutyunyan, and R. Riggio, "Flex5G: Flexible Functional Split in 5G Networks", *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, Sep. 2018, pp. 961-975.

[15] H. Mei, and L. Peng, "Flexible functional split for cost-efficient C-RAN", *Computer Communications*, 161, 2020, pp. 368-374

[16] I. Koutsopoulos, "The Impact of Baseband Functional Splits on Resource Allocation in 5G Radio Access Networks", in *Proc. IEEE Conference on Computer Communications (IEEE INFOCOM)*, 2021.

[17] A. Martínez Alba, J. H. Gómez Velásquez, and W. Kellerer, "An adaptive functional split in 5G networks", in *Proc. IEEE INFOCOM Workshops - 3rd Workshop on Flexible and Agile Networks: 5G and Beyond (FlexNets)*, 2019.

[18] A. Martínez Alba, S. Janardhanan, and W. Kellerer, "Enabling Dynamically Centralized RAN Architectures in 5G and Beyond", *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, Sep. 2021, pp. 3509-3526.

[19] A. Martínez Alba, and W. Kellerer, "Dynamic Functional Split Adaptation in Next-Generation Radio Access Networks ", *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, Sep. 2022, pp. 3239-3263.

[20] L. Díez, A. Martínez Alba, W. Kellerer, and R. Agüero, "Flexible Functional Split and Fronthaul Delay: A Queuing-Based Model", *IEEE Access*, Vol. 9, 2021, pp. 151049-151066.

[21] J. Bartelt *et al.*, "Fronthaul and Backhaul Requirements for Flexibly Centralized Radio Access Networks", *IEEE Wireless Communications*, Oct. 2015, pp. 105-111.

[22] *3rd Generation Partnership Project (3GPP)*, Accessed: April 6, 2023. [Online]. Available: https://www.3gpp.org/

[23] *Common Public Radio Interface (CPRI)*, Accessed: April 6, 2023. [Online]. Available: http://www.cpri.info/

[24] *Next Generation Mobile Networks (NGMN) Alliance*, Accessed: April 6, 2023. [Online]. Available: https://www.ngmn.org/

[25] *Open-RAN (O-RAN) Alliance*, Accessed: April 6, 2023. [Online]. Available: https://www.o-ran.org/

[26] *Small Cell Forum (SCF)*, Accessed: April 6, 2023. [Online]. Available: https://www.smallcellforum.org/

[27] *IEEE Next Generation Fronthaul Interface (1914) Working Group*, Accessed: Apr. 6, 2023. [Online]. Available: https://sagroups.ieee.org/1914/

[28] *Telecom Infra Project (TIP)*, Accessed: Apr. 6, 2023. [Online]. Available: https://telecominfraproject.com/

[29] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology*, Academic Press (Elsevier), 2018.

[30] *NG-RAN; Architecture description (Release 17), v17.4.0*, 3GPP Standard TS 38.401, Mar. 2023.

[31] *Study on CU-DU lower layer split for NR; (Release 15), v15.0.0*, 3GPP Technical Report TR 38.816, Dec. 2017.

[32] *CU-DU split: Refinement for Annex A (Transport network and RAN internal functional split)*, document TSG-RAN WG3 Meeting #93-bis R3-162102, 3GPP, Oct. 2016.

[33] *CPRI Specification v7.0, Common Public Radio Interface (CPRI); Interface Specification*, CPRI, Oct. 2015.

[34] *eCPRI Specification v2.0, Common Public Radio Interface (CPRI): eCPRI Interface Specification*, CPRI, May 2019.

[35] "Further study on critical C-RAN Technologies", NGMN Alliance, White Paper, Mar. 2015.

[36] "NGMN Overview on 5G RAN Functional Decomposition", NGMN Alliance, White Paper, Feb. 2018.

[37] "5G RAN CU-DU network architecture, transport options and dimensioning", NGMN Alliance, White Paper, Apr. 2019

[38] *O-RAN Architecture Description*, document O-RAN.WG1.O-RAN-Architecture-Description-v07.00, O-RAN Alliance, Oct. 2022

[39] *O-RAN Working Group 4 (Open Fronthaul Interfaces WG). Control, User and Synchronization Plane Specification*, document O-RAN.WG4.CUS-0-v10.00, O-RAN Alliance, Oct. 2022

[40] M. Mohsin *et al.*, "On Analyzing Beamforming Implementation in O-RAN 5G", *MDPI Electronics*, Vol. 10, 2021.

[41] A. Umesh, T. Yajima, T. Uchino, and S. Okuyama, "Overview of O-RAN Fronthaul Specifications", *NTT DOCOMO Technical Journal*, Vol. 21, No. 1, Jul, 2019.

[42] "An Introduction to O-RAN", National Instruments, White Paper, 2020, Accessed: Apr. 6, 2023. Available: https://www.ni.com/gate/gb/GB_infointrooran/US

[43]*Small Cell Virtualization Functional Splits and Use Cases, Release 7.0, document SCF 159.07.02*, Small Cell Forum, Jan. 2016.
[44]*5G FAPI: PHY API Specification. Q4 update, document SCF 222.10.05*, Small Cell Forum, Jul, 2022
[45]*5G nFAPI specifications, document SCF 225.2.0*, Small Cell Forum, May 2021.
[46]*IEEE Standard for Packet-based Fronthaul Transport Network*, IEEE Std 1914.1™, 2019.
[47]*IEEE Standard for Radio over Ethernet Encapsulations and Mappings*, IEEE Std 1914.3™, 2018.
[48]"Creating an ecosystem for vRANs supporting non-ideal fronthaul", Telecom Infra Project, White Paper, 2019.
[49]"Learnings from virtualized RAN technology trials over non-ideal fronthaul", Telecom Infra Project, White Paper, 2019.
[50]"OpenRAN 5G NR Base Station Platform Requirements Document", Telecom Infra Project, White Paper, 2020.
[51]A. Tukmanov *et al.*, "Fronthauling for 5G and beyond," in *Access, Fronthaul and Backhaul Networks for 5G & Beyond*. Stevenage, U.K.: Inst. Eng. Technol., 2017, pp. 139–168.
[52]R. S. Kshetrimayum, "Introduction to MIMO detection", in *Fundamentals of MIMO Wireless Communications*, Cambridge University Press, 2017, pp. 184-212.
[53]M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO Detection Techniques: A Survey", *IEEE Communications. Surveys & Tutorials*, vol. 21, no. 4, 4th Quarter, 2019.
[54]A. Zaidi, F. Athley, J. Medbo, U. Gustavsson, and G. Darisi, *5G Physical Layer. Principles, Models and Technology Components*, Academic Press (Elsevier), 2018.
[55]*NR; Physical channels and modulation (Release 17), v17.3.0,* 3GPP Standard TS 38.211, Sep. 2022.
[56]*NR; Multiplexing and channel coding (Release 17), v17.3.0,* 3GPP Standard TS 38.212, Sep. 2022.
[57]*NR; Physical layer procedures for data (Release 17), v.17.3.0*, 3GPP Standard TS 38.214, Sep.2022.
[58]*NR; Base Station (BS) radio transmission and reception (Release 17), v17.7.0*, 3GPP Standard TS 38.104 Sep. 2022.
[59]*5G Explained: Downlink Data in 5G NR*, Mathworks, Accessed: Apr. 6, 2023. [Online]. Available: https://es.mathworks.com/videos/5g-explained-downlink-data-in-5g-nr-1558600809645.html
[60]M. Biguesh, and A. B. Gershman, "Training-Based MIMO Channel Estimation: A Study of Estimator Tradeoffs and Optimal Training Signals", *IEEE Transactions on Signal Processing*, vol. 54, no. 3, March 2006, pp. 884-893.
[61]E. Nayebi, and B. D. Rao, "Semi-blind Channel Estimation for Multiuser Massive MIMO Systems", *IEEE Transactions on Signal Processing*, vol. 66, no. 2, 2018, pp. 540-553.
[62]*NR SRS Configuration*, Mathworks, Accessed: Apr. 6, 2023. [Online]. Available: https://www.mathworks.com/help/5g/ug/nr-sounding-reference-signals.html
[63]M.Cirkic, and E. G. Larsson, "On the Complexity of Very Large Multi-User MIMO detection", in *Proc. IEEE 15th Int. Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2014.
[64]Y. Hu, Z. Wang, X. Gao, and J. Ning, "Low-complexity signal detection using CG method for uplink large-scale MIMO systems", in *Proc. IEEE International Conference in Communication Systems (ICCS)*, Nov. 2014, pp. 477–481.
[65]B. Kang, J-H. Yoon, and J. Park, "Low complexity massive MIMO detection architecture based on Neumann method", in *Proc. International SoC Design Conference (ISOCC)*, 2015
[66]S. Shahabuddin, M. H. Islam, M. S. Shahabuddin, M. A. Albreem and M. Juntti, "Matrix Decomposition for Massive MIMO Detection", in *Proc. IEEE Nordic Circuits and Systems Conference (NorCAS)*, 2020.
[67]X. Gao, L. Dai, C. Yuen and Y. Zhang, "Low-Complexity MMSE Signal Detection Based on Richardson Method for Large-Scale MIMO Systems", in *Proc. IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, 2014.
[68]X. Gao, L. Dai, Y. Ma, and Z. Wang, "Low-Complexity Near-Optimal Signal Detection for Uplink Large-Scale MIMO Systems," *IET Electronics Letters*, vol. 50, no. 18, Aug. 2014, pp. 1326–1328.
[69]B. Kang, J-H. Yoon, and J. Park, "Low-Complexity Massive MIMO Detectors Based on Richardson Method", *ETRI Journal*, 39, 2017, pp. 326-335.
[70]J. Tu, M. Lou, J. Jiang, D. Shu, and G. He, "An Efficient Massive MIMO Detector Based on Second-Order Richardson Iteration: From Algorithm to Flexible Architecture", *IEEE Transactions on Circuits and Systems-I, Regular Papers*, Vol. 67, No. 11, Nov. 2020, pp. 4015-4028.
[71]M. Abdelghany, U. Madhow, and A. Tölli, "Beamspace Local LMMSE: An Efficient Digital Backend for mmWave Massive MIMO", in *Proc. IEEE 20th Int. Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019.
[72]M. Wu, B. Yin, G. Wang, C. Dick, J.R. Cavallaro, and C. Studer, "Large-Scale MIMO Detection for 3GPP LTE: Algorithms and FPGA Implementations", *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, Oct. 2014, pp. 916-929.
[73]C. Zhang, Z. Wu, C. Studer, Z. Zhang, and X. You, "Efficient soft-output Gauss-Seidel data detector for massive MIMO systems", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 68, No. 12, Dec. 2021, pp. 5049-5060.
[74]C. Zhang, X. Lian, Z. Wu, F. Wang, S. Zhang, Z. Zhang, and X. You, "On the low-complexity, hardware-friendly tridiagonal matrix inversion for correlated massive MIMO systems", *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, Jul. 2019, pp. 6272–6285.
[75]X. Qin, Z. Yan, and G. He, "A near-optimal detection scheme based on joint steepest descent and Jacobi method for uplink massive MIMO systems", *IEEE Communications Letters*, vol. 20, no. 2, Feb. 2016, pp. 276–279.
[76]G. Peng, L. Liu, S. Zhou, S. Yin, and S. Wei, "A 1.58 Gbps/W 0.40 Gbps/mm2 ASIC implementation of MMSE detection for 128 × 8 64 - QAM massive MIMO in 65 nm CMOS", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 5, May 2018, pp. 1717–1730.
[77]Y. Soo Cho, J. Kim, W. Y. Yang, and C-G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*, John Wiley & Sons, 2010.
[78]*Computational Complexity of Mathematical Operations*, Accessed: Apr. 6, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations
[79]*Gaussian elimination,* Accessed: Apr. 6, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Gaussian_elimination
[80]S. Thallapalli, and R. Pandey, "Performance Evaluation of Channel Estimation in Multicell Multiuser Massive MIMO Systems", in *Proc. International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, 2019.
[81]F. A. P. de Figueiredo, F. A. C. M. Cardoso, I. Moerman, and G. Fraidenraich, "Channel Estimation for Massive MIMO TDD Systems Assuming Pilot Contamination and Frequency Selective Fading", *IEEE Access*, Sep. 2017, pp. 17733 - 17741.
[82]H. Xie, F. Gao and S. Jin, "An Overview of Low-Rank Channel Estimation for Massive MIMO Systems", *IEEE Access*, vol. 4, Nov. 2016, pp. 7313-7321.
[83]H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A Coordinated Approach to Channel Estimation in Large-Scale Multiple-Antenna Systems", *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, Feb. 2013, pp. 264-273.
[84]A. Adhikary, J. Nam, J-Y. Ahn, and G. Caire, "Joint Spatial Division and Multiplexing-The Large-Scale Array Regime", *IEEE Transactions on Information Theory*, vol. 59, no. 10, Oct. 2013, pp. 6441-6462.
[85]D. Neumann, K. Shibli, M. Joham, and W. Utschick, "Joint Covariance Matrix Estimation and Pilot Allocation in Massive MIMO System", in *Proc. IEEE International Conference in Communications (ICC)*, 2017.
[86]J. Lee, G-T. Gil, and Y. H. Lee, "Channel Estimation via Orthogonal Matching Pursuit for Hybrid MIMO Systems in Millimeter Wave Communications", *IEEE Transactions on Communications*, vol. 64, no. 6, Jun. 2016, pp. 2370-2386.
[87]P. Zhang, L. Gan, S. Sun, and C. Ling, "Atomic norm denoising-based channel estimation for massive multiuser MIMO systems", in *Proc. IEEE International Conference on Communications (ICC)*, 2015, pp. 4564-4569.
[88]X. Rao, and V. K. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems", *IEEE Transactions on Signal Processing*, vol. 62, no. 12, Jun. 2014, pp. 3261-3271.
[89]Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO", *IEEE Transactions on Signal Processing*, vol. 63, no. 23, Dec. 2015, pp. 6169-6183.
[90]H. Xie, F. Gao, S. Zhang, and S. Jin, "A Unified Transmission Strategy for TDD/FDD Massive MIMO Systems with Spatial Basis Expansion Model", *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, Apr. 2017, pp. 3170-3184.

[91]X. Liu *et al.*, "Efficient Channel Estimator with Angle-Division Multiple Access", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, Feb. 2019, pp. 708-718.

[92]S. H. Mirfarshbafan, A. Gallyas-Sanhueza, R. Ghods, and C. Studer, "Beamspace Channel Estimation for Massive MIMO mmWave Systems: Algorithm and VLSI Design", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, Dec. 2020, pp. 5482-5495.

[93]Z. Ji, Y. Ji, B. Wang, F. Gao, H. Wang, and C. Zhang, "A New Uplink Channel Estimation Architecture for Massive MIMO Systems with PDMA", in *Proc. IEEE 13th International Conference on ASIC (ASICON)*, 2019.

[94]J. Shikida, K. Muraoka, and N. Ishii, "Sparse Channel Estimation Using Multiple DFT Matrices for Massive MIMO Systems", in *Proc. IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018.

[95]M. A. Albreem, A. H. Al Habbash, A. M. Abu-Hundrouss, and S. S. Ikki, "Overview of Precoding Techniques for Massive MIMO", *IEEE Access*, Apr. 2021, pp. 60764 - 60801.

[96]N. Fatema, G. Hua, Y. Xiang, D. Peng, and I. Natgunanathan, "Massive MIMO Linear Precoding: A Survey", *IEEE Systems Journal*, Vol. 12, No. 4, Dec. 2018, pp. 3920-3931.

[97]A. Müller, A. Kammoun, E. Björnson, and M. Debbah, "Efficient Linear Precoding for Massive MIMO Systems using Truncated Polynomial Expansion", in *Proc. IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2014.

[98]H. Prabhu, O. Edfors, J. Rodrigues, L. Liu, and F. Rusek, "Hardware Efficient Approximative Matrix Inversion for Linear Pre-coding in Massive MIMO", in *Proc. IEEE Int. Symposium on Circuits and Systems (ICAS)*, 2014.

[99]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*, Jan, 2013, pp. 40-60.

[100] C. Tang, C. Liu, L. Yan, and Z. Xing, "High Precision Low Complexity Matrix Inversion Based on Newton Iteration for Data Detection in the Massive MIMO", *IEEE Communications Letters*, vol. 20, no. 3, Mar. 2016, pp. 490-493.

[101] C. Zhang, Z. Li, L. Shen, F. Yan, M. Wu, and X. Wang, "A Low-Complexity Massive MIMO Precoding Algorithm Based on Chebyshev Iteration", *IEEE Access*, Oct. 2017, pp. 22545 - 22551.

[102] L. Shao, and Y. Zu, "Joint Newton iteration and Neumann series method of convergence-accelerating matrix inversion approximation in linear precoding for massive MIMO systems", *Math. Problems Eng.*, vol. 2016, May 2016, pp. 1-5,

[103] X. Qiang, Y. Liu, Q. Feng, J. Liu, X. Ren, and M. Jin, "Approximative matrix inversion based linear precoding for massive MIMO systems", in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2020.

[104] M. A. M. Albreem, A. A. El-Saleh, and M. Juntti, "Linear massive MIMO uplink detector based on joint Jacobi and Gauss-Seidel methods", in *Proc. 16th Int. Conf. Design Reliable Communication Networks (DRCN)*, 2020.

[105] Y. Bai, Z. Liang, C. Zhai, Y. Xin, and W. Li, "Joint precoding using successive over-relaxation matrix inversion and Newton iteration for massive MIMO systems", in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019.

[106] W. Song, X. Chen, L. Wang, and X. Lu, "Joint conjugate gradient and Jacobi iteration based low complexity precoding for massive MIMO systems", in *Proc. IEEE/CIC International Conference on Communications in China (ICCC)*, 2016

[107] X. Gao, L. Dai, Y. Ma, and Z. Wang, "Low-complexity near-optimal signal detection for uplink large-scale MIMO systems", *Electronics Letters*, vol. 50, no. 18, Aug. 2014, pp. 1326–1328.

[108] X. Qin, Z. Yan, and G. He, "A near-optimal detection scheme based on joint steepest descent and Jacobi method for uplink massive MIMO systems", *IEEE Communications Letters*, vol. 20, no. 2, Feb. 2016, pp. 276-279.

[109] J. Wu, S. Fang, L. Li, and Y. Yang, "QR decomposition and gram Schmidt orthogonalization based low-complexity multi-user MIMO precoding", in *Proc. 10th Int. Conf. Wireless Commun. Netw. Mobile Comput. (WiCOM)*, 2014, pp. 61-64

[110] Y. Xu, W. Zou, and L. Du, "A fast and low-complexity matrix inversion scheme based on CSM method for massive MIMO systems", *EURASIP J. Wireless Commun. Netw.*, Dec. 2016.

[111] Y. Bai, Z. Liang, C. Zhai, Y. Xin, and W. Li, "Joint precoding using successive over-relaxation matrix inversion and Newton iteration for massive MIMO systems," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019.

[112] K. K.-C. Lee, Y.-H. Yang, and J.-W. Li, "A low-complexity AEPDF-assisted precoding scheme for massive MIMO systems with transmit antenna correlation", *Journal of Signal Processing Systems*, vol. 92, Jan. 2020, pp. 529-539.

[113] M. Zhang, and S. Kim, "Universal soft demodulation schemes for M-ary phase shift keying and quadrature amplitude modulation", *IET Communications*, vol. 10, no. 3, 2016, pp. 316-326.

[114] M. Zhang, S. Kim, and Y. Kim, "Universal soft decision demodulator for M-ary adaptive modulation systems", in *Proc. of IEEE Asia Pacific Conference in Communications (APCC 2012)*, Jeju, Korea, Oct. 2012, pp. 574–578

[115] Q. Wang, Q. Xie, Z. Wang, S. Chen, and L. Hanzo, "A universal low-complexity symbol-to-bit soft demapper", *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, 2014, pp. 119–130

[116] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes", *IEEE Transactions on Information Theory*, vol. 47, no. 2, Feb. 2001, pp. 638–656.

[117] T. T. B. Nguyen, T.N. Tan, and H. Lee, "Efficient QC-LDPC Encoder for 5G New Radio", *Electronics*, 2019.

[118] M. Benhayoun, M. Razi, A. Mansouri, and A. Ahaitouf, "Low-Complexity LDPC Decoding Algorithm Based on Layered Vicinal Variable Node Scheduling", *Modelling and Simulation in Engineering*, 2022.

[119] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. IEEE Workshop Signal Process. Syst. (SIPS)*, Austin, TX, USA, Oct. 2004, pp. 107–112.

[120] B. Wang, Y. Zhu and J. Kang, "Two Effective Scheduling Schemes for Layered Belief Propagation of 5G LDPC Codes", *IEEE Communications Letters*, vol. 24, no. 8, Aug. 2020, pp. 1683-1686.

[121] C. A. Aslam, Y. L. Guan, and K. Cai, "Improving the Belief-Propagation Convergence of Irregular LDPC Codes Using Column-Weight Based Scheduling", *IEEE Communications Letters*, vol. 19, no. 8, Aug. 2015, pp. 1283-1286.

[122] M. Sybis, K. Wesolowski, K. Jayasinghe, V. Venkatasubramanian, and V. Vukadinovic, "Channel Coding for Ultra-Reliable Low-Latency Communication in 5G Systems," in *Proc. IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016

[123] R. Yavne, "An economical method for calculating the discrete Fourier transform," in *Proc. AFIPS Fall Joint Comput. Conf.*, 1968, vol. 33, pp. 115–125.

[124] J. W. Cooley and J. W. Tukey, "An algorithm for the machine computation of the complex Fourier series", *Math. Comp.*, vol. 19, Apr. 1965, pp. 297–301.

[125] S. G. Johnson and M. Frigo, "A Modified Split-Radix FFT With Fewer Arithmetic Operations", *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 111-119, Jan. 2007.

[126] S. Qadeer, M. Z. A. Khan, S. A. Sattar and Ahmed, "A Radix-2 DIT FFT with reduced arithmetic complexity", in *Proc. International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2014, pp. 1892-1896.

[127] M. Z. A. Khan and S. Qadeer, "A new variant of Radix-4 FFT", in *Proc. 13th International Conference on Wireless and Optical Communications Networks (WOCN)*, 2016.

[128] "5G TDD Synchronisation. Guidelines and Recommendations for the Coexistence of TDD Networks in the 3.5 GHz Range", GSMA, White Paper, Apr. 2020, Accessed: Apr. 6, 2023. Available: https://www.gsma.com/spectrum/wp-content/uploads/2020/04/3.5-GHz-5G-TDD-Synchronisation.pdf

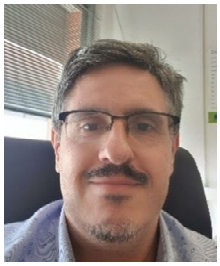
[129] *Management and orchestration; Self-Organizing Networks (SON) for 5G networks (Release 17), v17.7.0*, 3GPP Standard TS 28.313, Dec. 2022.

[130] *Small Cell SON and Orchestration from 4G to 5G, document SCF 233.10.01*, Small Cell Forum, Jul. 2020.

[131] C-L. I, S. Kuklinski, T. Chen, and L. Ladid, "A Perspective of O-RAN Integration with MEC, SON, and Network Slicing in the 5G Era", *IEEE Network*, Nov./Dec., 2020, pp. 3-4

[132] S. Kuklinski, L. Tomaszewski, and R. Kolakowski, "On O-RAN, MEC, SON and Network Slicing integration", in *Proc. IEEE Globecom Workshops*, 2020.

[133] T. F. Chan, "An Improved Algorithm for Computing the Singular Value Decomposition", *ACM Transactions on Mathematical Software*, vol. 8, no. 1, Mar. 1982.



Jordi Pérez-Romero (Member, IEEE) received a degree in telecommunications engineering and a Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1997 and 2001, respectively.
He is currently a Professor with the Department of Signal Theory and Communications of UPC. He is working in the field of wireless communication systems, with a particular focus on 5G and beyond cellular systems, including radio resource management and network optimization. He has been involved in different European projects with different responsibilities, such as researcher, work package leader, and project responsible, has participated in different projects for private companies and has contributed to the 3GPP and ETSI standardization bodies. He has coauthored more than 300 papers in international journals and conferences. He has also coauthored three books and has contributed to seven book chapters. He is an Associate Editor for the IEEE Vehicular Technology Magazine.

Oriol Sallent is a Professor at the Universitat Politècnica de Catalunya, Barcelona, Spain. He has participated in a wide range of European and national projects, with diverse responsibilities as a principal investigator, coordinator, and work package leader. He regularly serves as a consultant for a number of private companies. He has contributed to standardization bodies such as 3GPP, IEEE, and ETSI. He is the coauthor of 13 books and has authored or coauthored 300+ papers, mostly in high-impact IEEE journals and renowned international conferences. His research interests include 5G RAN (Radio Access Network) planning and management, artificial intelligence-based radio resource management, virtualisation of wireless networks, cognitive management cognitive radio networks and dynamic spectrum access and management, among others.

Antoni Gelonch is associate professor at Dept. of Signal Theory and Communications at the Universitat Politècnica de Catalunya (UPC). He received the Ph.D. in Telecommunications degree from UPC in 1997. His research interest has moved along years from the development of suitable hardware platforms for implementing wireless systems, combining FPGAs, DSPs and GPPs, and attending real-time processing constraints, to the application of Software Radio concept and the development of appropriate frameworks to accelerate its development and deployment. Currently, he is focused on addressing the implementation, virtualization, and resource management issues of 5G.

Xavier Gelabert (Member, IEEE) received the MS degree in electrical engineering from the Royal Institute of Technology (KTH) in 2003, and the MS degree in telecommunication engineering from the Technical University of Catalonia (UPC) in 2004. He also holds a PhD degree from UPC, 2010. Since 2012, he has been a researcher with Huawei's Stockholm Research Centre. He has more than 15 years of research experience across academia (UPC, GATech, and KCL), a non-profit research institute (iTEAM), a Telco Operator (Orange Labs), and an equipment vendor (Huawei). He actively contributed to 3GPP NR in RAN1 and RAN2 and holds a number of related patents. His research interests include radio, spectrum and compute resource management, self-organized networks, and more recently, baseband systems implementation and design.

Bleron Klaiqi (Member, IEEE) received the Dipl.Ing. (M.Sc.) degree in electrical engineering and information technology from RWTH Aachen University, Germany, and the Ph.D. degree in electronic and electrical engineering from The University of Sheffield, U.K. From 2006 to 2012, he was a Senior Engineer for UMTS/HSPA Layer 1 and interworking between multiple RATs at Intel Corporation, Nuremberg, Germany. During his Ph.D. studies, he was a Marie-Curie Experienced Researcher for EU H2020 Decade and is3DMIMO projects. He is currently with Huawei, Kista, Sweden, working as 5G/6G baseband research engineer. His main research interests are in massive MIMO and beyond, application of AI/ML in baseband (L1/L2), reconfigurable intelligent surfaces (RIS), and joint communication and sensing. He served as a reviewer for several major IEEE journals and conferences.

Marcus Kahn is a Senior Baseband Expert with Huawei Technologies Sweden AB, Stockholm Research Centre, with joining date 2002. He holds a MSc degree in telecommunication and radio systems from the Royal Institute of Technology (KTH), 1997. He has a background in FPGA and ASIC design, his latest focus being in baseband systems.

David Campoy received the degree in telecommunications engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2022. He is currently pursuing the M.S. degree in telecommunications engineering with specialization in wireless communications at UPC. His research interests include signal processing and cellular communications systems, with special focus in 5G and beyond networks.