Tampere University

Aleksi Virta

OPTIMIZATION OF CSI PRECODING MATRIX GENERATION IN 5G NEW RADIO

Faculty of Information Technology and Communication Sciences
Master of Science Thesis
May 2025

Abstract

Aleksi Virta: Optimization of CSI Precoding Matrix Generation in 5G New Radio
Master of Science Thesis
Tampere University
Faculty of Information Technology and Communication Sciences
May 2025

Precoding is an integral part of the transmitter signal processing chain in wireless communication, allowing to improve signal quality and capacity in both modern and future generations of telecommunication. The 3rd Generation Partnership Project (3GPP) technical specification TS 38.214 [1] provides definitions for Type I single panel and multi-panel codebooks. The codebooks consist of precoding matrices, which are used for precoding the vector representing a cyclic prefix orthogonal frequency-division multiplexed waveform linearly using the matrix product. The choice of a precoding matrix is carried out according to precoding matrix indicator, which is a part of channel state information (CSI) reporting.

This thesis aims to propose an efficient technique to implement 5G Type I CSI single panel precoding matrices using 4 or 8 antenna ports. This proposal is of interest for a practical codebook generation, simplifying it in terms of the memory needed to store the precoding matrices. The proposed algorithm has been validated and compared against a so called direct implementation, where the precoding matrices are stored as they are defined in the standard. While 4 and 8 antenna ports cases have been studied more thoroughly, the proposed algorithm can be extended into cases of more antenna ports. The definition of NR codebooks presented in the 3GPP standard is revised and analyzed without loss of generality from signal processing point of view. Optional mathematical expressions for the codebooks are presented alongside with proofs for their derivation. The proposed schemes demonstrate that the required memory can be reduced by approximately 95 % compared to the direct storing of each precoder. Our proposals thus bring fundamental benefit to precoder generation systems. Furthermore, with respect to direct precoding using a single matrix multiplication, the analysis carried out in this thesis indicates that precoding in parts shows a reduction of up to 87.5 % in number of needed real additions, and a reduction of up to 37.5 % of needed real multiplications associated with complex arithmetics. Lastly, further development ideas are presented to gain

prospective of the impact and potential ideas for extending the algorithm to more complex cases.

Key words: 5G, Precoding, CSI, Precoder generation, optimization, algorithm

Foreword

This thesis and its research was undertaken in collaboration with Nokia Solutions and Networks Oy, with the guidance of Alaaeddin Loulou, Ruifeng Duan from the Nokia side, and Mikko Valkama from Tampere University.

Writing this thesis was more than anything a learning process for me. For sure, it was occasionally a difficult process as well, where by ”occasionally” I mean a near-constant experience. Jokes aside, it is important to highlight the fun and instructive moments I had while investigating and studying this field. I have indeed learned a lot I'm glad I undertook the journey.

I would like to start by thinking my instructors, Alaaeddin Loulou, Ruifeng Duan, and Mikko Valkama for their guidance, expertise, and for repeatedly and tirelessly helping me out with whatever question I was able to come up with. I would also like to thank my team manager Mika, for his support and understanding towards me and the process.

Completing this journey would have been like crawling through shards of glass without my friends. They have played a key role in building myself higher and making progress with the thesis. Both supporting and challenging my ideas has made a huge difference in my writing and self-development in several walks of life. What you have done for me is something irreplaceable.

None of this wouldn't have been possible without the support and encouragement my family has given me. The faith they've had in whatever crazy endeavor I've ever undertaken, is something I deeply value and cherish, you all have my endless gratitude for that.

My final thanks go to Essi, who has been there for me unconditionally. Thank you for believing in me, and pushing me forward at all costs. You are an inexhaustive source of joy and energy to me, helping me grow and become better. You have my uttermost love and respect. Kiitos.

Now, may I yield me unto repose, it seemeth, for my labour here is done.

Tampere 22.05.2025

Aleksi Virta

Contents

List of Symbols and Abbreviations

$\cdot$ Arithmetic multiplication operator

$\mathbb{C}$ Set of complex numbers

$\mapsto$ Element-wise mapping

$\odot$ Hadamard product

ω_n The nth root of unity $e^{\frac{-2\pi j}{n}}$

$\otimes$ Kronecker product

$\mathbf{I}_n$ The $n \times n$ identity matrix

$\mathbf{M}(:, i)$ The ith column of matrix $\mathbf{M}$

$\mathbf{M}(:, v)$ The matrix formed of columns of matrix $\mathbf{M}$ as indicated by integer vector v

$\mathbf{M}(i, :)$ The ith row of matrix $\mathbf{M}$

$\mathbf{M}(v, :)$ The matrix formed of rows of matrix $\mathbf{M}$ as indicated by integer vector v

$\mathbf{M}^{\mathrm{T}}$ Transpose of matrix $\mathbf{M}$

$\mathrm{Arg}(z)$ Argument of a complex number z

$\mathrm{card}(S)$ Cardinality of set S

$\mathrm{ran}(x)$ Range of variable x

$\mathbb{Z}$ Set of integers

$c_{n \times m}$ A matrix in $\mathbb{C}^{n \times m}$ with all entries equal to c

j Imaginary unit

v_i^k kth element of vector v_l

3GPP 3rd generation partnership project

4G 4th Generation

5G 5th Generation

CC Channel coding

CP Cyclic prefix

CQI Channel quality indicator

CRI CSI-RS resource indicator

CSI Channel state information

CSI-RS CSI reference signal

DFT Discrete Fourier transform

diag(v) The diagonal matrix where the diagonal is v

FFT Fast Fourier transform

gNB gNodeB

HW Hardware

L1 Layer 1

lcm(a,b) Least common multiplier of integers a and b

LI Layer indicator

LTE Long term evolution

MIMO Multiple-Input Multiple-Output

mMIMO Massive MIMO

MU-MIMO Multi-user MIMO

NR New Radio

OFDM Orthogonal frequency-division multiplexing

OFDMA Orhtogonal frequency-division multiple access

PMI Precoding matrix indicator

QPSK Quadrature phase shift keying

RE Resource element

RI Rank indicator

RSRP Reference signal received power

SS Synchronization signal

SW Software

UE User equipment

1 Introduction

The fifth generation (5G) of the cellular network technology was introduced to meet new and more demanding requirements such as higher transmission diversity and more reliable transmission. With respect to the preceding communication technologies, studying 5G and standard development has identified key technologies presented in Figure 1.1 to help meeting these new requirements [6]. From consumer point of view, 5G has further enabled entertainment

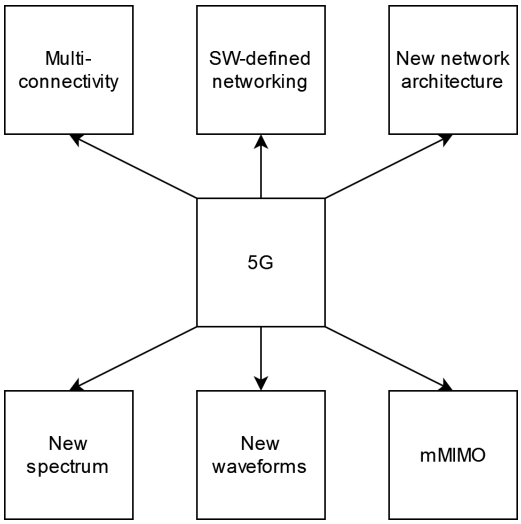

Figure 1.1. Key technologies in 5G [6].

applications such as enhanced video streaming and virtual reality [26, 27]. Revolutionary applications including self-driving cars and remote controlled operations have become a possibility as well [6]. To support higher transmission diversity in comparison to the fourth generation (4G), 5G utilizes the massive Multiple-Input Multiple-Ouput (mMIMO) systems [4, 5]. The use of multi-user MIMO (MU-MIMO) is enabled by efficient beam management carried out by the gNodeB (gNB) to mitigate interference between the users. To support these applications, the technology used in 4G needs to be updated. For example, more base stations are needed to form a larger and more reliable grid to secure more reliable connectivity and accessing larger radio frequencies. This requires integration in circuits and semiconductor technologies [6]. Moreover, efficient signal transmission is an important part of ensuring connectivity and higher data rates. In the heart of the signal transmission is the physical layer, which interacts between both the radio frontends and physical interface.

The physical layer is an inseparable part of signal transmission, as it is responsible of, for example, the choice of modulation, coding schemes, and multi-

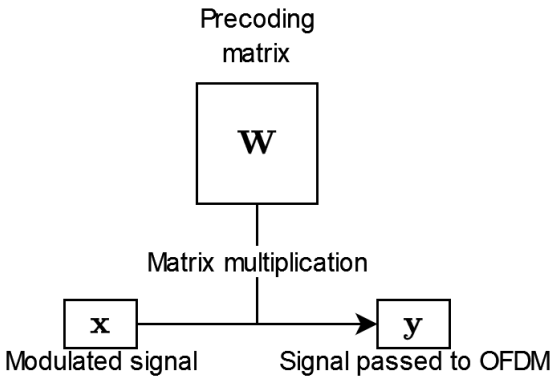

Figure 1.2. Precoding modulated signal $\mathbf{x}$ with precoder $\mathbf{W}$ resulting in signal $\mathbf{y}$.

antenna transmission modes to ensure the desired coverage and throughput [6]. This can be achieved by processing orthogonal frequency-division multiplexing (OFDM), scrambling, layer mapping, and precoding. To mitigate memory overhead and computational complexity, it is essential to look into possibilities of optimizing the physical layer signal processing. This thesis proposes an implementation to reduce the required memory to store the precoding matrices with the trade-off of computational complexity needed to retrieve the matrices. The proposed schemes demonstrate that the memory required can be reduced by approximately 95 % compared to the direct storing of each precoder. Additionally, with respect to direct precoding using a single matrix multiplication, the analysis carried out in this thesis indicates that precoding in parts shows a reduction of up to 87.5 % in number of needed real additions, and a reduction of up to 37.5 % of needed real multiplications. This implies that the schemes bring fundamental benefit to precoder generation systems.

Precoding refers to the linear processing performed prior the OFDM, in which the signal is mapped with the aim of increasing spectral efficiency in different single-user (SU) and MU-MIMO schemes [10]. Mathematically, the signal is processed using linear maps, which further translates into the use of matrices. These matrices are referred to as *precoding matrices* or for short *precoders*, and they are defined over varying numbers of antenna ports and transmission layers. An illustration of the mathematical meaning of the precoding is presented in Figure 1.2. A key requirement of precoder design is the channel state information (CSI) reporting and the use of precoding matrix indicator (PMI). The user equipment (UE), such as a smartphone, uses various kinds of data. This data may consist of, for example, time and frequency information, which the UE uses to report CSI. Through this approach, CSI can be used to, for example, time and frequency tracking [5]. The CSI report covers

several factors, one of which is the PMI. To acquire CSI in a frequency division duplexing system the CSI reporting is employed [22]. Conversely, in time domain duplexing networks it is possible exploit the channel reciprocity when transmission and reception using same number of antennas is possible for the UE [22]. The PMI corresponds to a codebook index given by table 5.2.2.2.1-1 in [1], indicating the appropriate precoding matrix to be utilized. Based on the PMI, the gNB can determine a suitable precoder and communicate the information back to UE.

The precoders are defined for combinations of numbers of antenna ports and layers, which translate directly into the dimensions of the precoding matrix. For example, in the case of 4 and 8 antenna ports, the maximum numbers of elements in a single precoder are achieved by 4 and 8 layers which result in 16 and 64 elements, respectively. As the numbers of used ports and layers grow, the effort to generate a single matrix grows proportionally to them. However, as can be learned from other studies, the precoding matrices admit a mathematical structure allowing representation of the matrices in a more concise manner compared to one given by the 3GPP standards [7, 10, 11]. Since precoding can be viewed as a matrix multiplication in the scope of this thesis, the structural conveniences discussed for example in [7, 10, 11] may allow to simplify the precoding itself by decomposing to smaller parts and removing redundant calculations. In contrast to theoretical results, a discussion on practical implementation and effective codebook generation method has been carried out in [24, 25], but not in the specific case discussed in this thesis, thus motivating further studies.

From implementation and hardware (HW) point of view, according to the best knowledge of the author, the efficiency of NR Type I CSI precoding matrix generation has not been widely studied, even though it remains an integral part of the precoding process. The number of coefficients in the precoding matrices is directly proportional to the used antenna ports and layers, a fact that leads to essentiality of efficient storing and generation of the codebooks. Another aspect to consider, is that the number of needed calculations increases inevitably the more there are signals to be precoded. Because of these factors, even the slightest of reductions in complexity and memory consumption can add significant value in the long term.

The aim of this thesis is to propose and analyze an efficient implementation for retrieving the NR Type I CSI codebooks for the number of ports $P_{CSI-RS} \in \{4, 8\}$ as an option to a direct storing and fetching of the codebooks. The underlying mathematical background will be presented to demonstrate the structure and connection the codebooks share to establish a more concise way

of presenting and retrieving the codebooks. The mathematical ideas presented especially in, [7, 10, 11] lay a foundation for the study conducted and the ideas proposed in this thesis. An algorithm to retrieve the codebooks is presented based on these ideas. Finally, an implementation based on the presented algorithm is proposed and followed by the analysis of the complexity and the benefits with respect to the direct storing of the precoding matrices.

The proposed implementation and algorithm will cover the retrieval of all NR Type I CSI precoding matrices for number of ports $P_{CSI-RS} \in \{4, 8\}$ and rank $\nu \in \{1, \ldots, P_{CSI-RS}\}$. The efficiency of the implementation is then analyzed from the point of view studying the impact on memory use and computational complexity. The results are compared against the direct retrieval, where each precoder matrix has been stored in the memory and then retrieved as is. Alongside with providing proofs and derivations for some of the key characteristic of NR Type I CSI precoders from the point of view of this thesis, it will be shown that the proposed implementation enables a precoder retrieval scheme allowing a reduction of 95.2 % of the needed bits for the direct implementation.

The remainder of this thesis is organized as follows. Chapter 2 covers a literature review on the related works and their importance to the thesis. The current state of art is analyzed and earlier proposals are discussed. In Chapter 3, the theoretical background focusing on the necessary mathematics and NR precoding is presented. The purpose of this is to provide sufficient preliminary foundation to understand the topic of the thesis. A mathematical overview of the codebook generation is presented also in the third chapter. Chapter 4 proposes a retrieval algorithm which has been implemented using Matlab and in Chapter 5 the proposed implementation is compared against direct precoder retrieval from the point of view of memory and complexity. Lastly, Chapter 6 discusses further development ideas and potential extensions of the existing implementation to cover more port-layer combinations.

2 Related Works

Studies on optimization of NR precoding are focusing on precoder and PMI selection or search [7, 8, 9, 10, 11, 12, 13, 14], and codebook design [15, 16, 17, 18, 19, 20, 21]. Additionally, patents regarding more efficient codebook retrieval have been also been published [24, 25]. However, they do not cover exactly the NR CSI Type I single panel codebook retrieval, nor the problem of memory overhead related to the corresponding codebook storing.

A solution for the problem of overhead reduction related to precoding has been presented from several points of view in [7, 8, 9, 10, 11, 12, 13, 14]. The aim of these studies is to reduce the computational complexity of precoder search using various methods. The works of [7, 11, 12, 13] focus on Type I precoder, [8] on Type II, and [9, 10, 14] suggest solutions covering both cases. These studies suggest, that exploiting the precoder structure proves useful in optimizing the complexity of the precoder search. The focus is on deriving the expression $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2$ for the precoder. Matrix $\mathbf{W}_1$ can be factorized using for example the Kronecker product [7, 9, 11], allowing to represent the matrix in terms of simpler matrices. Matrix $\mathbf{W}_2$ can be represented using Kathri-Rao product [7, 11], as a quadrature phase shift keying (QPSK) co-phasing matrix [9, 10, 13, 14]. The factorization $\mathbf{W}_1\mathbf{W}_2$ is assumed in [7, 8, 9, 11, 13, 14] whereas [10] is presenting a derivation up to 4 layers in case of 4 antenna ports. This leaves room for providing a more rigorous proof to gain a better understanding of the Type I codebook structure. This will be the aim of this thesis, alongside with the derivation to represent the non-full rank precoders using the full-rank one is also presented.

Solutions aiming to optimize the precoding scheme by redesigning the precoder have been suggested by [15, 16, 17, 18, 19, 20, 21]. The focus is on enhancing precoding in NR Type II CSI precoding [15, 16, 17, 21], Type I [18], and LTE-A [19]. Diverse approaches have been utilized, such as ones based on artificial intelligence [16], and algorithmic solutions [17, 20, 21]. Furthermore, some of the solutions are based on more mathematical framework, such as Helmholtz equations [15], singular value decomposition [18], and weighing the precoder using phase adjustment functions [19]. The integral observation from these studies is, that the state of the art precoding scheme has room for improvement based on the amount of studies. Moreover, as the covered studies examine novel structures and architectures for codebooks and their derivation, a gap for studying the generation of conventional NR Type I CSI codebooks remains. Alongside with studying the mathematical derivation of the conven-

tional NR Type I CSI codebooks, this thesis seeks to address the problem of efficient storing and generation of these codebooks posed by direct implementation of the codebooks.

The authors of [24, 25] describe an implementation for their proposals in their respective patent applications. These implementations provide a method and apparatus for multi-polarized MIMO system codebook generation using a single-polarized precoder [24], and for codebook generation to provide elements for a reconfigurable intelligent surface precoding [25]. A detailed description of the proposed implementation is carried out in both applications. However, these applications are restricted in specific focal points, namely, multi-polarization MIMO systems [24] and reconfigurable intelligent surface systems [25]. This leaves room for studying and implementing the methods and apparatus for NR Type I CSI codebook generation at a more general level, which is the aim of this thesis.

Pertinent to this thesis is to study and implement an algorithm for Type I CSI single panel precoder generation and retrieval. The contribution of this thesis crystallizes in the detailed description of the proposed algorithm and derivation of the mathematical structural properties, such as the formula $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2$, of the NR CSI Type I precoders for numbers of ports 4 and 8, which form and underlying basis of the algorithm proposed in this thesis. The proposed algorithm alongside with implementation for codebook generation and retrieval shares some of the mathematical aspects presented particularly in [7, 10, 11, 24]. The approaches exploiting the NR codebook structure serve as a foundation for the contributions of this thesis, highlighting the implementation of an algorithm and mathematical proofs. The emphasis in the proposed implementation is on reducing the memory overhead in comparison to the direct implementation of precoder generation.

3 Background

In this chapter, the underlying aspects of 5G physical layer processing and precoding are discussed. We will also cover the core elements of NR Type I CSI reporting and single panel PMI in Section 3.2, following mostly the work of [6]. The key structural attributes of the NR Type I single panel codebooks are described from the perspective of this thesis. The underlying mathematical results are derived and presented, covering all the cases discussed in this thesis. Lastly, we will describe an NR Type I single panel CSI codebook generation algorithm.

The lowest layer interacting with the digital and analog frontends and the physical media is referred to as the physical layer and it is responsible of multiple processing steps of the transmitted signal [6]. In transmission, the physical layer input data consists of transport blocks, whereas the output is an OFDM signal [6]. On the high level, NR downlink physical layer processing includes

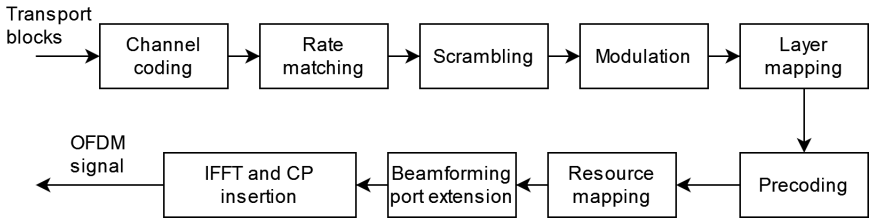

Figure 3.1. An overview of the downlink physical layer processing steps adapted from [23]

channel coding, rate matching, scrambling, modulation, layer mapping, precoding, resource mapping, beamforming port extension, and OFDM modulation and cyclic prefix (CP) addition [23]. In the scope of this thesis, we are interested in the precoding stage of the downlink physical layer. An illustration of the simplified overview of the physical layer processing is presented in Figure 3.1.

3.1 Precoding in Physical Layer Signal Processing Chain

Precoding enables reduction of feedback complexity and overhead in 5G systems [7]. By spatial filtering in MIMO systems, precoding enables sending multiple data streams [10]. In order to understand the data processing of precoding, we will first discuss the data flow in downlink physical layer processing.

A point of interest in our case is to look into the downlink broadcast channel, where the data is coded in multiple steps. These steps include payload

generation, scrambling, transport block cyclic redundancy check attachment to enable detecting errors, channel coding (CC), and rate matching [2]. Delving into CC, the input data is encoded using polar code and the codeword data is then rate matched according to technical specification TS 38.212 Sections 7.1.4-7.1.5 [2]. The encoded and rate matched data is then scrambled, modulated and layer mapped according to technical specification TS 38.211 Sections 7.3.3.1-7.3.3.3 [3]. At this stage, the data can be viewed as a complex vector $\mathbf{x} \in \mathbb{C}^{\nu}$, where ν is the number of spatial layers. This data is the precoder input.

Let N_1 and N_2 be the numbers of MIMO transmission antenna ports in horizontal and vertical dimensions, respectively, and let $P_{CSI-RS} = 2N_1N_2$ [10]. Assuming a UE has N_R receiver antennas, the signal $\mathbf{y} \in \mathbb{C}^{N_R}$ observed by the UE can be expressed as

$$\mathbf{y} = \mathbf{HPx} + \mathbf{n}. \tag{3.1}$$

Here $\mathbf{H} \in \mathbb{C}^{N_R \times P_{CSI-RS}}$ is the downlink channel estimation matrix, $\mathbf{P} \in \mathbb{C}^{P_{CSI-RS} \times \nu}$ is the precoder, $\mathbf{n} \in \mathbb{C}^{N_R}$ is white noise, and $\mathbf{x} \in \mathbb{C}^{\nu}$ is the input to the precoder [10]. In terms of codebook based MIMO linear precoding, it can be modelled as the matrix multiplication $\mathbf{Px}$ of (3.1). In order to satisfy (3.1), the channel conditions experienced by the UE need to be estimated with $\mathbf{H}$, and $\mathbf{P}$ needs to be chosen carefully based on the estimation. The matrix $\mathbf{P}$ can be chosen according to the CSI, which is based on the report on channel state by the UE.

3.2 CSI

To estimate the prevalent channel conditions, the base station uses CSI. It contains information on the wireless telecommunication channel such as signal scattering, fading, and attenuation. CSI enables adaptation of the transmitted waveform to the surrounding conditions. In many cases, CSI is estimated at the receiver and reported to the transmitter. In summary, it allows the UE to adapt into varying channel conditions while maintaining good user experience.

On a high level, CSI can be divided into two sub-categories based on the use-case, the instantaneous and the statistical CSI. The instantaneous CSI is useful in situations where the channel conditions are more stable and known whereas the statistical CSI fits more in rapidly varying conditions [6]. In a general case, a suitable combination of both CSI types is used [6]. More detailed information on CSI configuration is determined by NR CSI framework [1].

The CSI framework enables selecting CSI reporting setting, CSI reference signal (CSI-RS) resource setting, and measurement setting based on the CSI requirements [22]. Fine-tuning the CSI configuration is fundamental as the

suitable use of statistical and instantaneous CSI is crucial to be taken into account in the reporting. CSI-RS can be used for beam management by using CSI-RS to indicate the resourcing and allocation of transmitted beams in a multi-UE situation [22]. More specifically, the gNB decides based on the UE measurements on received beams and beam quality reports, the beams assignation to the UE. This assignment can be conducted adaptively under dynamic conditions [22]. Furthermore, CSI-RS is used to carry the information of CSI reporting between the gNB and the UE.

The gNB controls the setting for CSI reporting, which is then conveyed by CSI-RS to the UE. The UE then interprets the report setting to report the desired CSI to the gNB. A high level description of the process of CSI acquisition is depicted in Figure 3.3. The CSI may carry information on channel quality indicator (CQI), PMI, CSI-RS resource indicator (CRI), synchronization signal (SS) block resource indicator, layer indicator (LI), rank indicator (RI) and/or layer 1 reference signal received power (L1-RSRP) measurements [1]. The components that the CSI may contain are illustrated in Figure 3.2. The UE is configured with more than one reporting settings *CSI-ReportConfig* and resource settings *CSI-ResourceConfig* for reporting and measuring all components but SS-block resource indicator [6]. Based on the CSI, the gNB is able to approximate the channel conditions. This allows gNB to, for example, efficiently control channel dependent scheduling and precoding matrix assignation [6].

The most suitable codebook and further the precoder are selected based on the reported PMI value. The UE calculates the value of PMI based on the reported values of RI and CRI [1]. In NR, high-resolution spatial channel information can be provided using precoding involving selection of subbands and channel eigenvector approximations. Through this approach, the UE will know which transmission scheme to use for data transmission. This knowledge allows choosing the optimal precoding matrix, which is instrumental in securing an efficient transmission [6]. An important factor to take into account in the transmission is the number of users. As NR supports both SU-MIMO and MU-MIMO, the CSI is divided into Type I and Type II CSI to meet respective needs [1, 13].

The configuration parameters for either Type I or Type II CSI are contained in *CodebookConfig*, which can be *CSI-ReportConfig* [1]. Type I, or the standarad resolution, CSI can be applied for use-cases where the number of received layers is smaller than the number of receiver antennas. Conversely, the high-resolution, i.e., Type II CSI is needed in for example MU-MIMO cases. In these use-cases, Type II CSI enables beamforming or precoding used for

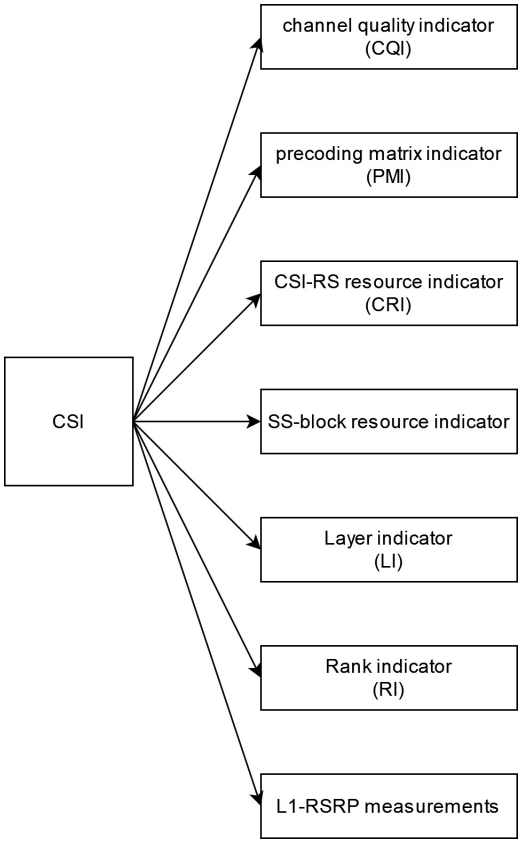

Figure 3.2. Possible components of CSI [1].

inter-UE interference suppression [13].

3.3 NR Precoding Matrix

In conditions for which the standard resolution CSI is sufficient, Type I CSI codebook is applicable. The codebook consists of precoders of form $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2$, where $\mathbf{W}_1$ is a wideband matrix common for all subbands and $\mathbf{W}_2$ is a subband matrix representing the fast fading properties. This structure allows viewing precoding as a two stage procedure. An illustration of the effect of $\mathbf{W}_1$ and $\mathbf{W}_2$ in beam selection is presented in Figure 3.4. Type I supports up to 8 layers and 32 antenna ports implying that the largest precoding matrices can reach up to 8×32-matrices. This highlights the importance of efficient matrix generation and retrieval in HW and software (SW). Based on the ideas in [7, 11, 10], the DFT structure of the precoders offers a basis for optimizing the generation.

Matrix $\mathbf{W}_1$ is constructed using DFT matrices and reported for the complete bandwidth. In particular, $\mathbf{W}_1$ can be presented in form $\mathbf{I}_2 \otimes \mathbf{D}$, where $\mathbf{D}$ is a diagonal matrix with a DFT vector as the main diagonal [7]. Contrarily, matrix

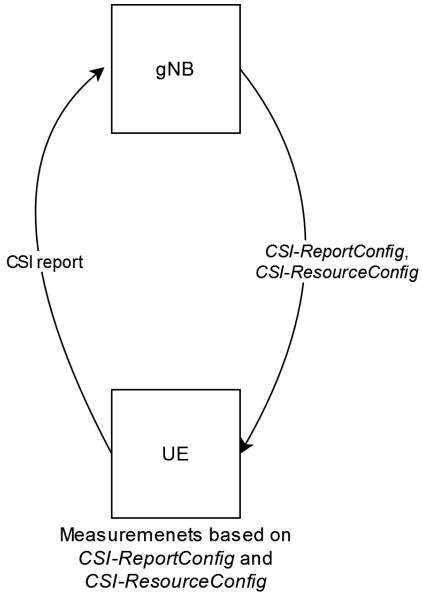

Figure 3.3. Overview of the CSI reporting: The gNB controls the resourcing and reporting configurations which the UE uses to measure the channel conditions. Based on the measurements, the UE reports the CSI back to the gNB.

$\mathbf{W}_2$ acts as a selector for the DFT vectors within $\mathbf{W}_1$ enabling representation of the short-term frequency-dependent characteristics [15]. In practice, $\mathbf{W}_1$ indicates the beam direction with the structure corresponding to two possible polarizations and $\mathbf{W}_2$ handles the choice of beam. Finally, the PMI carries the necessary info for $\mathbf{W}_1$ and $\mathbf{W}_2$ aiming to approximate the CSI. In Type I single panel precoding, $\mathbf{W}_2$ performs quadrature phase-shift key (QPSK) cophasing on two polarizations [9]. It is also possible, that a device is subsequently selecting CQI with the assumption of randomly selected $\mathbf{W}_2$ on a physical resource block basis [6]. In this case, it is not necessary to report $\mathbf{W}_2$.

As defined in Section 3.1, N_1 and N_2 correspond, respectively, to the hor-

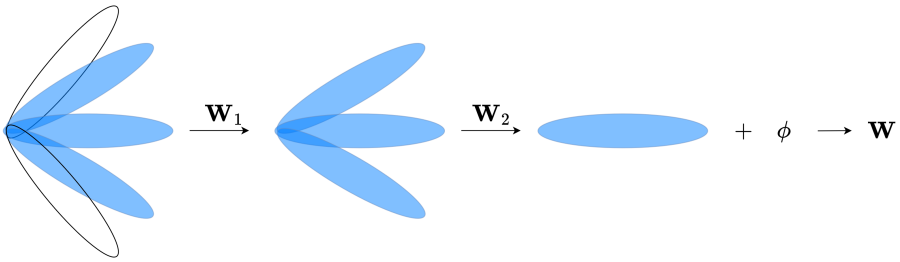

Figure 3.4. Illustration of Type I CSI precoder effect adapted from [6]. Matrix $\mathbf{W}_1$ selects a subset of subbands. Matrix $\mathbf{W}_2$ acts as a beam selector and adds cophasing resulting in $\mathbf{W}$.

izontal and vertical dimensions of the antenna panel [10]. The number of antenna ports is defined as $P_{CSI-RS} = 2N_1N_2$. Additionally, indices O_1 and O_2 have been defined in [1]. They correspond respectively to the horizontal and vertical dimension over-sampling factors [7]. In the scope of thesis, cases $P_{CSI-RS} \in \{4, 8\}$ will be studied. The configurations for (N_1, N_2) and (N_1, N_2) defined in these cases are listed in Table 3.1. They are used to determine the ranges for indices that relate to PMI.

Technical specification TS 38.214 defines precoders for two codebook modes, 1 and 2, but in the scope of this thesis, codebook mode 2 is not covered separately. This is due to the definitions indicating that codebook mode 2 differentiates from codebook mode 1 by the indexing order of the precoders [1]. Therefore, the proposed algorithm covers both codebook modes regardless of which one is used to its derivation.

Table 3.1. Values for N_1, N_2, O_1, O_2 when $P_{CSI-RS} \in \{4, 8\}$ defined in [1].

P_{CSI-RS}	(N_1, N_2)	(O_1, O_2)
4	$(2, 1)$	$(4, 1)$
8	$(2, 2)$	$(4, 4)$
8	$(4, 1)$	$(4, 1)$

In Type I single panel precoding, each PMI value corresponds to an index triplet $(i_{1,1}, i_{1,2}, i_2)$ for rank $\nu \notin \{2, 3, 4\}$ [1]. Inversely, PMI is associated with an index tuple $(i_{1,1}, i_{1,2}, i_{1,3}, i_2)$, where index $i_{1,3}$ is further mapped into a pair (k_1, k_2) [1]. The mapping of $i_{1,3}$ into (k_1, k_2) is presented in Tables 3.2-3.3. Index $i_{1,2}$ is mapped into l, l', l'', l''', $i_{1,2}$ into m, m', m'', m''', and i_2 into n. For $\nu \in \{2, 3, 4\}$, $i_{1,3}$ will contribute to l' and m'.

We begin by defining the mappings and index ranges for full rank precoders. The ranges for indices $i_{1,1}, i_{1,2}, i_2$ in the full rank case are defined in the Table 3.4. Later on, we will prove that the non-full rank precoders can be extracted from the full rank one. For $P_{CSI-RS} = 8$ and $\nu = 8$, indices $i_{1,1}, i_{1,2}, i_2$ are mapped according to N_1 into index tuples I_1 and I_2 as follows

$$\begin{cases} (i_{1,1}, i_{1,2}, i_2) \mapsto I_1, & N_2 = 1 \\ (i_{1,1}, i_{1,2}, i_2) \mapsto I_2, & N_2 > 1, \end{cases} \tag{3.2}$$

where $\mapsto$ denotes element-wise mapping, and I_1 and I_2 have been defined as

$$\begin{aligned} I_1 &= (i_{1,1}, i_{1,1} + O_1, i_{1,1} + 2O_1, i_{1,1} + 3O_1, 0, 0, 0, 0, i_2) \\ I_2 &= (i_{1,1}, i_{1,1} + O_1, i_{1,1}, i_{1,1} + O_1, i_{1,2}, i_{1,2}i_{1,2} + O_2, i_{1,2} + O_2, i_2). \end{aligned} \tag{3.3}$$

Table 3.2. Mapping of $i_{1,3}$ into (k_1, k_2) when $P_{CSI-RS} \in \{4, 8\}$, $\nu = 2$. Adapted from [1].

$i_{1,3}$	$N_1 = 2, N_2 = 1$	$N_1 = 4, N_2 = 1$	$N_1 = 2, N_2 = 2$
	(k_1, k_2)	(k_1, k_2)	(k_1, k_2)
0	$(0, 0)$	$(0, 0)$	$(0, 0)$
1	$(O_1, 0)$	$(O_1, 0)$	$(O_1, 0)$
2		$(2O_1, 0)$	$(0, O_2)$
3		$(3O_1, 0)$	(O_1, O_2)

Table 3.3. Mapping of $i_{1,3}$ into (k_1, k_2) when $P_{CSI-RS} \in \{4, 8\}$, $\nu \in \{3, 4\}$. Adapted from [1].

$i_{1,3}$	$N_1 = 2, N_2 = 1$	$N_1 = 4, N_2 = 1$	$N_1 = 2, N_2 = 2$
	(k_1, k_2)	(k_1, k_2)	(k_1, k_2)
0	$(O_1, 0)$	$(O_1, 0)$	$(O_1, 0)$
1		$(2O_1, 0)$	$(0, O_2)$
2		$(3O_1, 0)$	(O_1, O_2)

Table 3.4. The ranges for $i_{1,1}, i_{1,2}, i_2$ for $P_{CSI-RS} \in \{4, 8\}$. Adapted from [1].

	$N_1 = 2, N_2 = 1$	$N_1 = 4, N_2 = 1$	$N_1 = 2, N_2 = 2$
$i_{1,1}$	$0, \ldots, N_1O_1 - 1$	$0, \ldots, \frac{N_1O_1}{2} - 1$	$0, \ldots, N_1O_1 - 1$
$i_{1,2}$	$0, \ldots, N_2O_2 - 1$	0	$0, \ldots, N_2O_2 - 1$
i_2	0,1	0, 1	0, 1

Every precoder is associated with a unique index tuple I defined as

$$I = (l, l', l'', l''', m, m', m'', m''', n) = \begin{cases} I_1, & N_2 = 1 \\ I_2, & N_2 > 1. \end{cases} \tag{3.4}$$

For $P_{CSI-RS} = 4$ and $\nu = 4$, we define tuple I_3 as

$$I_3 = (l, l', m, m', n) = \left(i_{1,1}, i_{1,1} + k_1, i_{1,2}, i_{1,2} + k_2, i_2\right), \tag{3.5}$$

where k_1 and k_2 are determined according to Table 3.3. To avoid using extra indices in the text, I is used to refer to I_3 whenever we are discussing the case $P_{CSI-RS} = 4$.

The entries of I_1, I_2 and I_3 determine the DFT vectors which in turn construct the precoder matrix. Let $\otimes$ be the Kronecker product, $\mathbb{Z}$ the set of integers, and $\omega_k = e^{2\pi j/k}$, with $k \in \mathbb{Z}$, the kth root of unity. The determining DFT vectors are defined as follows

$$\begin{aligned} u_m &= \begin{bmatrix} 1 & \omega_{O_2N_2}^{m} & \cdots & \omega_{O_2N_2}^{m(N_2-1)} \end{bmatrix}, \\ v_l &= \begin{bmatrix} 1 & \omega_{O_1N_1}^{l} & \cdots & \omega_{O_1N_1}^{l(N_1-1)} \end{bmatrix}, \\ v_{l,m} &= \begin{bmatrix} u_m & \omega_{O_1N_1}^{l} u_m & \cdots & \omega_{O_1N_1}^{l(N_1-1)} u_m \end{bmatrix} \\ &= v_l \otimes u_m, \end{aligned} \tag{3.6}$$

alongside with the co-phasing factor $\phi_n = e^{j\pi n/2}$ [1]. Letting $\odot$ be the element-wise matrix product, or the Hadamard product, the full rank precoding matrix for $P_{CSI-RS} = 8$ is defined as

$$\begin{aligned} \mathbf{W}_I^{(8)} &= \frac{1}{\sqrt{8P_{CSI-RS}}} \begin{bmatrix} v_{l,m} & v_{l,m} & v_{l',m'} & v_{l',m'} & v_{l'',m''} & v_{l'',m''} & v_{l''',m'''} & v_{l''',m'''} \\ \phi_n v_{l,m} & -\phi_n v_{l,m} & \phi_n v_{l',m'} & -\phi_n v_{l',m'} & v_{l'',m''} & -v_{l'',m''} & v_{l''',m'''} & -v_{l''',m'''} \end{bmatrix} \\ &= \frac{1}{\sqrt{8P_{CSI-RS}}} \left(\left(\mathbf{1}_{2\times 1} \otimes \begin{bmatrix} v_{l,m} & v_{l',m'} & v_{l'',m''} & v_{l''',m'''} \end{bmatrix} \right) \otimes \mathbf{1}_{1\times 2} \right) \odot \mathbf{\Phi}_n, \end{aligned} \tag{3.7}$$

and for $P_{CSI-RS} = 4$ as

$$\begin{aligned} \mathbf{W}_I^{(4)} &= \frac{1}{\sqrt{4P_{CSI-RS}}} \begin{bmatrix} v_{l,m} & v_{l',m'} & v_{l,m} & v_{l',m'} \\ \phi_n v_{l,m} & \phi_n v_{l',m'} & -\phi_n v_{l,m} & -\phi_n v_{l',m'} \end{bmatrix} \\ &= \frac{1}{\sqrt{4P_{CSI-RS}}} \left(\mathbf{1}_{2\times 2} \otimes \begin{bmatrix} v_{l,m} & v_{l',m'} \end{bmatrix} \right) \odot \mathbf{\Phi}_n \left(:, \begin{bmatrix} 1 & 3 & 2 & 4 \end{bmatrix} \right), \end{aligned} \tag{3.8}$$

where $\mathbf{1}_{1\times 2}$,$\mathbf{1}_{2\times 1}$, and $\mathbf{1}_{2\times 2}$ are 1×2, 2×1,2×2 matrices, respectively, with all entries equal to 1, $\mathbf{\Phi}_n$ is defined as

$$\mathbf{\Phi}_n = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \phi_n & -\phi_n & \phi_n & -\phi_n & 1 & -1 & 1 & -1 \\ \phi_n & -\phi_n & \phi_n & -\phi_n & 1 & -1 & 1 & -1 \\ \phi_n & -\phi_n & \phi_n & -\phi_n & 1 & -1 & 1 & -1 \\ \phi_n & -\phi_n & \phi_n & -\phi_n & 1 & -1 & 1 & -1 \end{bmatrix}, \tag{3.9}$$

and $(:, v)$ is a column selector operator choosing the columns of a matrix indicated by vector v. Without loss of generality, we omit the multiplication by scalars $\frac{1}{\sqrt{8P_{CSI-RS}}}$ and $\frac{1}{\sqrt{4P_{CSI-RS}}}$. This is due to the fact that the scalar multiplication can be carried out once $\mathbf{W}$ is recovered, and it does not thus impact the algorithm for the generation of $\mathbf{W}$.

As discussed in Section 3.2, a Type I single panel CSI precoder can be decomposed as $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2$. With respect to the implementation of codebook generation, this is an important property, as it introduces the possibility of representing $\mathbf{W}$ using potentially simpler matrices. We will prove that $\mathbf{W}_1$ will be characteristic for each index pair $(i_{1,1}, i_{1,2})$, and that $\mathbf{W}_2$ can be viewed as a generator matrices for the codebooks. More importantly, we will prove that $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2$ holds for precoders when $P_{CSI-RS} \in \{4, 8\}$.

Problem statement. For each NR CSI Type I single-panel precoder $\mathbf{W}$ there exist such matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ that

$$\mathbf{W} = \mathbf{W}_1\mathbf{W}_2$$

for defined configurations of $P_{CSI-RS} \in \{4, 8\}$.

Proof. Starting from a more fundamental result, if $p \in \{0, O_1, 2O_1, 3O_2\}$ and $q \in \{0, O_2\}$, from the definitions of v_l and u_m it follows that

$$\begin{cases} v_{l+p} = v_l \odot v_p \\ u_{m+q} = u_m \odot u_q \end{cases} \tag{3.10}$$

as is stated in [7]. Then by the mixed product property of Kronecker and element-wise products we have

$$\begin{aligned} v_{l+p,m+q} &= v_{l+p} \otimes u_{m+q} \\ &= (v_l \odot v_p) \otimes (u_m \odot u_q) \\ &= (v_l \otimes u_m) \odot (v_p \otimes u_q) \\ &= v_{l,m} \odot v_{p,q}. \end{aligned} \tag{3.11}$$

To derive $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2$, we start by finding an expression for $\mathbf{W}_1$. Initially, we find $\mathbf{W}_1$. Later on we will demonstrate that this matrix is unique up to pair $(v, i_{1,3})$. Let us define a diagonal matrix $\mathbf{D}_{l,m}^{(P_{CSI-RS})}$ as

$$\begin{aligned} \mathbf{D}_{l,m}^{(P_{CSI-RS})} &= \mathbf{I}_2 \otimes \operatorname{diag}(v_{l,m}) \\ &= \begin{bmatrix} v_{l,m}^1 & 0 & & \dots & & 0 \\ 0 & \ddots & & & & \\ & & v_{l,m}^{P_{CSI-RS}/2} & & & \vdots \\ \vdots & & & v_{l,m}^1 & & \\ & & & & \ddots & 0 \\ 0 & & \dots & & 0 & v_{l,m}^{P_{CSI-RS}/2} \end{bmatrix}, \end{aligned} \tag{3.12}$$

where $\mathbf{I}_2$ is the 2×2 identity matrix, and diag(.) is the operator mapping a vector into a diagonal matrix with the input as its main diagonal. The

left-multiplication of a matrix by a diagonal matrix is equivalent to element-wise multiplication of each column of the same matrix by the diagonal. Let $p_1, p_2, p_3 \in \{0, O_1, 2O_1, 3O_1\}$ and $q_1, q_2, q_3 \in \{0, O_2\}$, and set

$$\begin{cases} l' = l + p_1, l'' = l + p_2, l''' = l + p_3 \\ m' = m + q_1, m'' = m + q_2, m''' = m + q_3. \end{cases} \tag{3.13}$$

The values of $p_1, p_2, p_3, q_1, q_2, q_3$ depend on N_1 according to (3.2), (3.3), and (3.4). By (3.11) and (3.13) we have

$$\begin{aligned} \begin{bmatrix} v_{l,m} & v_{l',m'} & v_{l'',m''} & v_{l''',m'''} \end{bmatrix} &= \begin{bmatrix} v_{l,m} & v_{l+p_1,m+q_1} & v_{l+p_2,m+q_2} & v_{l+p_3,m+q_3} \end{bmatrix} \\ &= \operatorname{diag}(v_{l,m}) \begin{bmatrix} v_{0,0} & v_{p_1,q_1} & v_{p_2,q_2} & v_{p_3,q_3} \end{bmatrix} \end{aligned} \tag{3.14}$$

for $P_{CSI-RS} = 8$, and similarly for $P_{CSI-RS} = 4$ we have

$$\begin{bmatrix} v_{l,m} & v_{l',m'} \end{bmatrix} = \operatorname{diag}(v_{l,m}) \begin{bmatrix} v_{0,0} & v_{p_1,q_1} \end{bmatrix}. \tag{3.15}$$

Furthermore, by Kronecker product properties we have

$$\begin{aligned} &\mathbf{1}_{2\times 1} \otimes \left(\operatorname{diag}(v_{l,m}) \begin{bmatrix} v_{0,0} & v_{p_1,q_1} & v_{p_2,q_2} & v_{p_3,q_3} \end{bmatrix} \right) \\ &= (\mathbf{I}_2 \mathbf{1}_{2\times 1}) \otimes \left(\operatorname{diag}(v_{l,m}) \begin{bmatrix} v_{0,0} & v_{p_1,q_1} & v_{p_2,q_2} & v_{p_3,q_3} \end{bmatrix} \right) \\ &= \left(\mathbf{I}_2 \otimes \operatorname{diag}(v_{l,m}) \right) \left(\mathbf{1}_{2\times 1} \otimes \begin{bmatrix} v_{0,0} & v_{p_1,q_1} & v_{p_2,q_2} & v_{p_3,q_3} \end{bmatrix} \right) \\ &= \mathbf{D}_{l,m}^{(8)} \left(\mathbf{1}_{2\times 1} \otimes \begin{bmatrix} v_{0,0} & v_{p_1,q_1} & v_{p_2,q_2} & v_{p_3,q_3} \end{bmatrix} \right) \end{aligned} \tag{3.16}$$

for $P_{CSI-RS} = 8$ and

$$\begin{aligned} &\mathbf{1}_{2\times 2} \otimes \left(\operatorname{diag}(v_{l,m}) \begin{bmatrix} v_{0,0} & v_{p_1,q_1} \end{bmatrix} \right) \\ &= (\mathbf{I}_2 \mathbf{1}_{2\times 4}) \otimes \left(\operatorname{diag}(v_{l,m}) \begin{bmatrix} v_{0,0} & v_{p_1,q_1} \end{bmatrix} \right) \\ &= \left(\mathbf{I}_2 \otimes \operatorname{diag}(v_{l,m}) \right) \left(\mathbf{1}_{2\times 2} \otimes \begin{bmatrix} v_{0,0} & v_{p_1,q_1} \end{bmatrix} \right) \\ &= \mathbf{D}_{l,m}^{(4)} \left(\mathbf{1}_{2\times 1} \otimes \begin{bmatrix} v_{0,0} & v_{p_1,q_1} \end{bmatrix} \right) \end{aligned} \tag{3.17}$$

for $P_{CSI-RS} = 4$. Finally, using (3.14), (3.16) and (3.7) can be reformulated as

$$\begin{aligned} \mathbf{W}_I^{(8)} &= \left(\left(1_{2\times 1} \otimes \begin{bmatrix} v_{l,m} & v_{l',m'} & v_{l'',m''} & v_{l''',m'''} \end{bmatrix} \right) \otimes 1_{1\times 2} \right) \odot \mathbf{\Phi}_n \\ &= \mathbf{D}_{l,m}^{(8)} \left(\left(\left(1_{2\times 1} \otimes \begin{bmatrix} v_{0,0} & v_{p_1,q_1} & v_{p_2,q_2} & v_{p_3,q_3} \end{bmatrix} \right) \otimes 1_{1\times 2} \right) \odot \mathbf{\Phi}_n \right). \end{aligned} \tag{3.18}$$

By setting

$$\mathbf{G}_I^{(8)} = \left(\left(1_{2\times 1} \otimes \begin{bmatrix} v_{0,0} & v_{p_1,q_1} & v_{p_2,q_2} & v_{p_3,q_3} \end{bmatrix} \right) \otimes 1_{1\times 2} \right) \odot \mathbf{\Phi}_n, \tag{3.19}$$

and substituting into (3.18), we have

$$\mathbf{W}_I^{(8)} = \mathbf{D}_{l,m}^{(8)} \mathbf{G}_I^{(8)} \tag{3.20}$$

for $P_{CSI-RS} = 8$. Similarly, using (3.17), (3.8) can be reformulated as

$$\mathbf{W}_I^{(4)} = \mathbf{D}_{l,m}^{(4)} \left(\left(\mathbf{1}_{2\times 2} \otimes \begin{bmatrix} v_{0,0} & v_{p_1,q_1} \end{bmatrix} \right) \odot \mathbf{\Phi}_n \left(:, \begin{bmatrix} 1 & 3 & 2 & 4 \end{bmatrix} \right) \right). \tag{3.21}$$

By setting

$$\mathbf{G}_I^{(4)} = \left(\mathbf{1}_{2\times 2} \otimes \begin{bmatrix} v_{0,0} & v_{p_1,q_1} \end{bmatrix} \right) \odot \mathbf{\Phi}_n \left(:, \begin{bmatrix} 1 & 3 & 2 & 4 \end{bmatrix} \right), \tag{3.22}$$

and substituting into (3.21), we have

$$\mathbf{W}_I^{(4)} = \mathbf{D}_{l,m}^{(4)} \mathbf{G}_I^{(4)} \tag{3.23}$$

for $P_{CSI-RS} = 4$.

□

Throughout this thesis, the diagonal matrix $\mathbf{D}_{l,m}^{(P_{CSI-RS})}$ will be referred to as the *characteristic matrix* and $\mathbf{G}_I^{(P_{CSI-RS})}$ as the *generator matrix*. To simplify notation, we will omit the exponent (P_{CSI-RS}) from the diagonal and generator matrices for the remainder of the thesis.

The choice of I is governed by N_1 according to (3.4). Given I, parameters $p_1, p_2, p_3, q_1, q_2, q_3$ are determined according to (3.13), and will be the input of (3.4). The value of n in turn defines $\mathbf{\Phi}_n$ according to (3.9). Once I is completely determined, $\mathbf{G}_I$ is computed according to (3.19). An illustration of the process is presented in Figure 3.5.

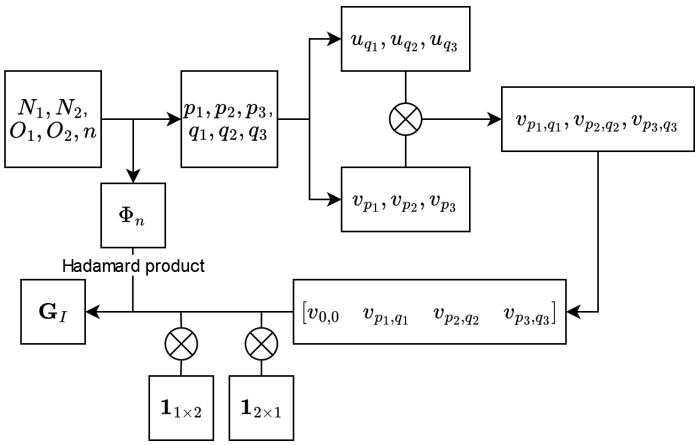

Figure 3.5. Retrieving $\mathbf{G}_I$ based on the configurations for $P_{CSI-RS} = 8$.

The retrieval of $\mathbf{G}_I$ is simpler for $P_{CSI-RS} = 4$ than for $P_{CSI-RS} = 8$, since the it is determined by $v_{0,0}$ and $v_{O_1,0}$ regardless of N_2. Moreover, it is Kronecker-multiplied with $\mathbf{1}_{2\times 2}$ and Hadamard-multiplied by a permutation of first rows

of Φ_n. Apart from these differences, the retrieval closely resembles the case in $P_{CSI-RS} = 8$. The process is depicted in Figure 3.6. When it comes to the

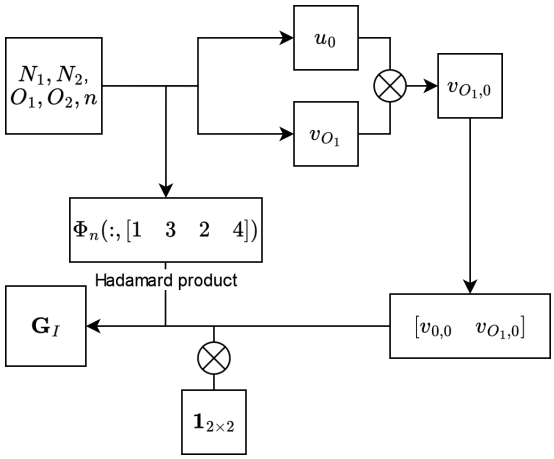

Figure 3.6. Retrieving $\mathbf{G}_I$ based on the configurations for $P_{CSI-RS} = 4$.

characteristic matrix $\mathbf{D}_{l,m}$, configuration (N_1, N_2, O_1, O_2) dictates the used l, m together with I. First, N_1, O_1, l, and N_2, O_2, m determine the entries of v_l and u_m, respectively, according to (3.6). The Kronecker product of v_l and u_m yields $v_{l,m}$, which is then mapped into a diagonal matrix $\text{diag}(v_{l,m})$. Finally, this matrix is Kronecker multiplied from the left by $\mathbf{I}_2$ resulting in $\mathbf{D}_{l,m}$. Figure 3.7 illustrates the process of generating $\mathbf{D}_{l,m}$.

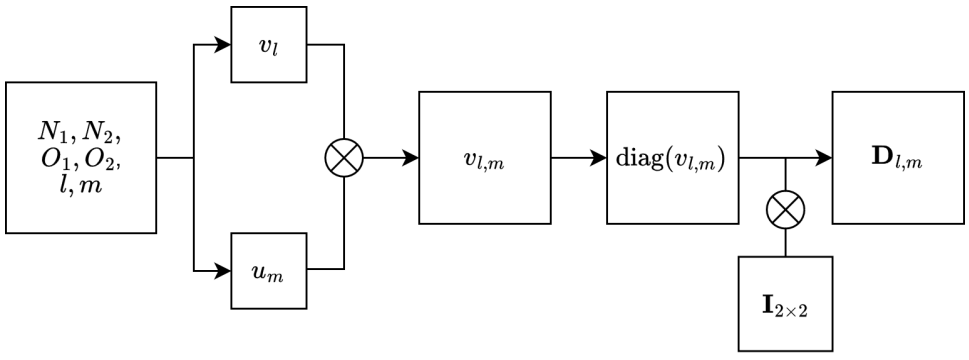

Figure 3.7. A depiction of the process of retrieving $\mathbf{D}_{l,m}$ based on N_1, N_2, O_1, O_2, l, m.

For the non-full rank case, we will prove that the rank ν precoder $\mathbf{W}_I^{(\nu)}$ can be extracted from full-rank matrix $\mathbf{W}_I^{(P_{CSI-RS})}$. As mentioned before, index $i_{1,3}$ also will now have an impact on the precoder for $\nu = 2, 3, 4$. This effect will be taken into account in the form of coefficient matrices and permutation matrices.

Problem statement. For each NR CSI Type I single-panel precoder $\mathbf{W}^{(\nu)}$, where $\nu \leq P_{CSI-RS}$, there exist such matrices $\mathbf{W}_1^{(\nu)}$ and $\mathbf{W}_2^{(\nu)}$ that

$$\mathbf{W}^{(\nu)} = \mathbf{W}_1^{(\nu)}\mathbf{W}_2^{(\nu)}$$

for defined configurations of $P_{CSI-RS} \in \{4, 8\}$. Moreover, for each $\mathbf{W}_2^{(\nu)}$ there exist such matrices $\mathbf{P}$ and $\mathbf{C}$ that

$$\mathbf{W}_2^{(\nu)} = \left(\mathbf{W}_2^{(P_{CSI-RS})}\mathbf{P}\right) \odot \mathbf{C}$$

for all $\nu < P_{CSI-RS}$.

Proof. We begin by looking into the case $P_{CSI-RS} = 8$, and defining a coefficient matrix $\mathbf{C}_{i_{1,3}}^{(\nu)}$ as

$$\mathbf{C}_{i_{1,3}}^{(\nu)} = \begin{cases} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \phi_n & \phi_n & 1 \end{bmatrix} \otimes \mathbf{1}_{4\times 1}, & \nu = 5 \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \phi_n & 1 & \phi_n \end{bmatrix} \otimes \mathbf{1}_{4\times 1}, & \nu = 4, i_{1,3} = 1, 2 \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & \phi_n & 1 \end{bmatrix} \otimes \mathbf{1}_{4\times 1}, & \nu = 3, i_{1,3} = 1, 2 \\ \begin{bmatrix} 1 & 1 \\ 1 & \phi_n \end{bmatrix} \otimes \mathbf{1}_{4\times 1}, & \nu = 2, i_{1,3} = 2, 3 \\ \mathbf{1}_{8\times\nu}, & \text{otherwise.} \end{cases} \tag{3.24}$$

We also need a set of permutation matrices $\mathbf{P}^{(\nu)}$, which can be viewed as a matrix formed by a subset c of columns of $\mathbf{I}_8$. The permutation matrices are listed in Table 3.5 for $\nu \neq 2, 3, 4$, and in Table 3.6 for $\nu = 2, 3, 4$.

Using $\mathbf{P}^{(\nu)}$ and $\mathbf{C}_{i_{1,3}}^{(\nu)}$, we will prove that $\mathbf{W}_I^{(\nu)}$ can be extracted from $\mathbf{W}^{(8)}$ as follows

$$\mathbf{W}_I^{(\nu)} = \left(\mathbf{W}_I^{(8)}\mathbf{P}^{(\nu)}\right) \odot \mathbf{C}_{i_{1,3}}^{(\nu)}, \tag{3.25}$$

where $\mathbf{W}_I^{(\nu)}$ is defined according to Tables 5.2.2.2.1-5 - 5.2.2.2.1-11 of [1]. The idea is to use rank ν, configuration (N_1, N_2, O_1, O_2), and the indices $i_{1,1}, i_{1,2}, i_{1,3}, i_2$ to first retrieve the full rank precoder, permutation matrix and coefficient matrix. These matrices are then used according to (3.25). An illustration of the non-full rank precoder retrieval based on (3.25) is presented in Figure 3.8.

Let us prove (3.25) for $P_{CSI-RS} = 8$. The factor $\frac{1}{\sqrt{\nu P_{CSI-RS}}}$ is neglected without loss of generality.

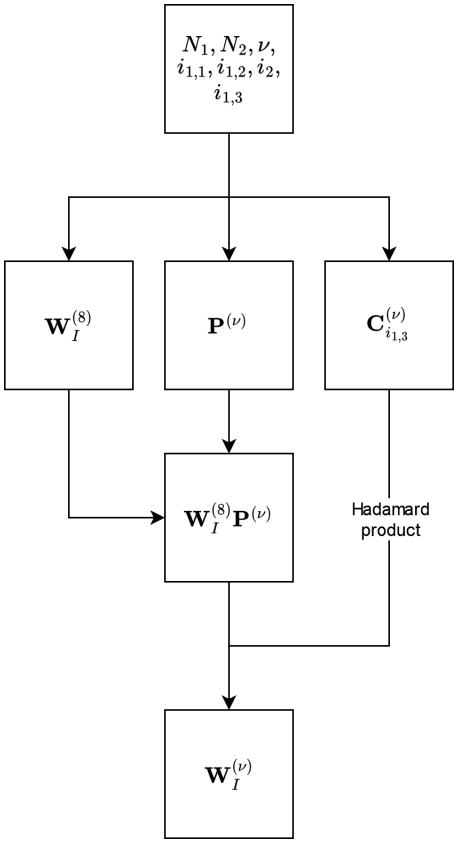

Figure 3.8. Non-full rank precoding matrix retrieval based on the set of indices.

Table 3.5. Permutation matrices for $P_{CSI-RS} = 8$, $\nu \neq 2, 3, 4$.

	$\mathbf{P}^{(\nu)} = \mathbf{I}_8(:, c)$
$\nu = 1, i_2 = 0, 1$	$c = 1$
$\nu = 1, i_2 = 2, 3$	$c = 2$
$\nu = 5, N_2 = 1$	$c = \begin{bmatrix} 1 & 2 & 3 & 4 & 7 \end{bmatrix}$
$\nu = 5, N_2 > 1$	$c = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \end{bmatrix}$
$\nu = 6, N_2 = 1$	$c = \begin{bmatrix} 1 & 2 & 3 & 4 & 7 & 8 \end{bmatrix}$
$\nu = 6, N_2 > 1$	$c = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$
$\nu = 7$	$c = \begin{bmatrix} 1 & 2 & 3 & 5 & 6 & 7 & 8 \end{bmatrix}$

Table 3.6. Permutation matrices for $P_{CSI-RS} = 8$, $\nu = 2, 3, 4$.

	$\mathbf{P}^{(\nu)} = \mathbf{I}_8(:, c)$
$\nu = 2, i_{1,3} = 0, 1, 2, 3$	$c = \begin{bmatrix} 1 & 2 + 2i_{1,3} \end{bmatrix}$
$\nu = 3, i_{1,3} = 0, 1, 2$	$c = \begin{bmatrix} 1 & 3 + 2i_{1,3} & 2 \end{bmatrix}$
$\nu = 4, i_{1,3} = 0, 1, 2$	$c = \begin{bmatrix} 1 & 3 + 2i_{1,3} & 2 & 4 + 2i_{1,3} \end{bmatrix}$

For $\nu = 1$, according to Table 5.2.2.2.1-5 [1], we have $l = i_{1,1}, m = i_{1,2}, n = i_2$, and $I = (l, m, n)$. Then by Tables 3.5 and 3.6, and by (3.24) and (3.25), it follows that

$$\begin{aligned}(\mathbf{W}_I^{(8)}\mathbf{P}^{(1)}) \odot \mathbf{C}_{i_{1,3}}^{(1)} &= \begin{cases} \mathbf{W}_I^{(8)}(:,1), & n = 0,1 \\ \mathbf{W}_I^{(8)}(:,2), & n = 2,3 \end{cases} \\ &= \begin{cases} \begin{bmatrix} v_{l,m} \\ \phi_n v_{l,m} \end{bmatrix}, & n = 0,1 \\ \begin{bmatrix} v_{l,m} \\ -\phi_n v_{l,m} \end{bmatrix}, & n = 2,3 \end{cases} \\ &= \mathbf{W}_I^{(1)}. \end{aligned}$$

For $\nu = 2$, according to Table 5.2.2.2.1-6 of [1] $l = i_{1,1}, l' = i_{1,1} + k_1, m = i_{1,2}, m' = i_{1,2} + k_2$ and $n = i_2$. Then by Tables 3.5 and 3.6, and by (3.24) and (3.25), it follows that

$$\begin{aligned}(\mathbf{W}_I^{(8)}\mathbf{P}^{(2)}) \odot \mathbf{C}_{i_{1,3}}^{(2)} &= \mathbf{W}_I^{(8)}\left(:, \begin{bmatrix} 1 & 2+2i_{1,3} \end{bmatrix}\right) \odot \mathbf{C}_{i_{1,3}}^{(2)} \\ &= \begin{cases} \begin{bmatrix} v_{l,m} & v_{l,m} \\ \phi_n v_{l,m} & -\phi_n v_{l,m} \end{bmatrix}, & i_{1,3} = 0 \\ \begin{bmatrix} v_{l,m} & v_{l',m'} \\ \phi_n v_{l,m} & -\phi_n v_{l',m'} \end{bmatrix}, & i_{1,3} = 1 \\ \begin{bmatrix} v_{l,m} & v_{l'',m''} \\ -\phi_n v_{l,m} & -v_{l'',m''} \end{bmatrix}, & i_{1,3} = 2 \\ \begin{bmatrix} v_{l,m} & v_{l''',m'''} \\ -\phi_n v_{l,m} & -v_{l''',m'''} \end{bmatrix}, & i_{1,3} = 3 \end{cases} \\ &= \mathbf{W}_I^{(2)}. \end{aligned}$$

For case $\nu = 3$, $i_{1,3} = 0, 1, 2$. Then by Tables 3.5, 3.6, and by (3.24) and (3.25), we have

$$\begin{aligned}(\mathbf{W}_I^{(8)}\mathbf{P}^{(3)}) \odot \mathbf{C}_{i_{1,3}}^{(3)} &= \mathbf{W}_I^{(8)}\left(:, \begin{bmatrix} 1 & 3+2i_{1,3} & 2 \end{bmatrix}\right) \odot \mathbf{C}_{i_{1,3}}^{(3)} \\ &= \begin{cases} \begin{bmatrix} v_{l,m} & v_{l',m'} & v_{l,m} \\ \phi_n v_{l,m} & v_{l',m'} & -\phi_n v_{l,m} \end{bmatrix}, & i_{1,3} = 0 \\ \begin{bmatrix} v_{l,m} & v_{l'',m''} & v_{l,m} \\ \phi_n v_{l,m} & v_{l'',m''} & -\phi_n v_{l,m} \end{bmatrix}, & i_{1,3} = 1 \\ \begin{bmatrix} v_{l,m} & v_{l''',m'''} & v_{l,m} \\ \phi_n v_{l,m} & v_{l''',m'''} & -\phi_n v_{l,m} \end{bmatrix}, & i_{1,3} = 2 \end{cases} \end{aligned}$$

$$= \mathbf{W}_I^{(3)}.$$

For case $\nu = 4$, $i_{1,3} = 0, 1, 2$. By Tables 3.5 and 3.6, and by (3.24) and (3.25), we have

$$\begin{aligned}
(\mathbf{W}_I^{(8)}\mathbf{P}^{(4)}) \odot \mathbf{C}_{i_{1,3}}^{(4)} &= \mathbf{W}_I^{(8)}\left(:, \begin{bmatrix} 1 & 3+2i_{1,3} & 2 & 4+2i_{1,3} \end{bmatrix}\right) \odot \mathbf{C}_{i_{1,3}}^{(4)} \\
&= \begin{cases} \begin{bmatrix} v_{l,m} & v_{l',m'} & v_{l,m} & v_{l',m'} \\ \phi_n v_{l,m} & \phi_n v_{l',m'} & -\phi_n v_{l,m} & -\phi_n v_{l',m'} \end{bmatrix}, & i_{1,3} = 0 \\ \begin{bmatrix} v_{l,m} & v_{l'',m''} & v_{l,m} & v_{l'',m''} \\ \phi_n v_{l,m} & v_{l'',m''} & -\phi_n v_{l,m} & -v_{l'',m''} \end{bmatrix}, & i_{1,3} = 1 \\ \begin{bmatrix} v_{l,m} & v_{l''',m'''} & v_{l,m} & v_{l''',m'''} \\ \phi_n v_{l,m} & v_{l''',m'''} & -\phi_n v_{l,m} & -v_{l''',m'''} \end{bmatrix}, & i_{1,3} = 2 \end{cases} \\
&= \mathbf{W}_I^{(4)}.
\end{aligned}$$

For case $\nu = 5$, $i_{1,3} = 0, 1, 2$. By Tables 3.5 and 3.6, and by (3.24) and (3.25), it follows that

$$\begin{aligned}
(\mathbf{W}_I^{(8)}\mathbf{P}^{(5)}) \odot \mathbf{C}_{i_{1,3}}^{(5)} &= \mathbf{W}_I^{(8)}\left(:, \begin{bmatrix} 1 & 2 & 3 & 4 & 7 \end{bmatrix}\right) \odot \mathbf{C}_{i_{1,3}}^{(5)} \\
&= \begin{bmatrix} v_{l,m} & v_{l,m} & v_{l',m'} & v_{l',m'} & v_{l'',m''} \\ \phi_n v_{l,m} & -\phi_n v_{l,m} & \phi_n v_{l',m'} & -\phi_n v_{l',m'} & v_{l'',m''} \end{bmatrix} \\
&= \mathbf{W}_I^{(5)}.
\end{aligned}$$

For case $\nu = 6$, $i_{1,3} = 0, 1, 2$. By Tables 3.5 and 3.6, and by (3.24) and (3.25), it follows that

$$\begin{aligned}
(\mathbf{W}_I^{(8)}\mathbf{P}^{(6)}) \odot \mathbf{C}_{i_{1,3}}^{(6)} &= \mathbf{W}_I^{(8)}\left(:, \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}\right) \\
&= \begin{bmatrix} v_{l,m} & v_{l,m} & v_{l',m'} & v_{l',m'} & v_{l'',m''} & v_{l'',m''} \\ \phi_n v_{l,m} & -\phi_n v_{l,m} & \phi_n v_{l',m'} & -\phi_n v_{l',m'} & v_{l'',m''} & -v_{l'',m''} \end{bmatrix} \\
&= \mathbf{W}_I^{(6)}.
\end{aligned}$$

For case $\nu = 7$, $i_{1,3} = 0, 1, 2$. By Tables 3.5 and 3.6, and by (3.24) and (3.25), it follows that

$$\begin{aligned}
(\mathbf{W}^{(8)}\mathbf{P}^{(7)}) \odot \mathbf{C}_{i_{1,3}}^{(7)} &= \mathbf{W}_I^{(8)}\left(:, \begin{bmatrix} 1 & 2 & 3 & 5 & 6 & 7 & 8 \end{bmatrix}\right) \\
&= \begin{bmatrix} v_{l,m} & v_{l,m} & v_{l',m'} & v_{l'',m''} & v_{l'',m''} & v_{l''',m'''} & v_{l''',m'''} \\ \phi_n v_{l,m} & -\phi_n v_{l,m} & \phi_n v_{l',m'} & v_{l'',m''} & -v_{l'',m''} & v_{l''',m'''} & -v_{l''',m'''} \end{bmatrix} \\
&= \mathbf{W}_I^{(7)}.
\end{aligned}$$

Comparing the results expressions in the above equations with the ones presented in Tables 5.2.2.2.1-5 - 5.2.2.2.1-11 [1], they can be seen to coincide. As

the choice of I was arbitrary, (3.25) holds for $P_{CSI-RS} = 8$ for ranks $\nu = 1, \ldots, 7$. By setting $\mathbf{P}^{(8)} = \mathbf{I}_8$, (3.25) holds for $\nu = 8$ as well.

Substituting (3.20) in (3.25) yields

$$\mathbf{W}_I^{(\nu)} = \left(\mathbf{D}_{l,m}\left(\mathbf{G}_I\mathbf{P}^{(\nu)}\right)\right) \odot \mathbf{C}_{i_{1,3}}^{(\nu)} = \mathbf{D}_{l,m}\left(\left(\mathbf{G}_I\mathbf{P}^{(\nu)}\right) \odot \mathbf{C}_{i_{1,3}}^{(\nu)}\right) =: \mathbf{D}_{l,m}\mathbf{G}_{I,i_{1,3}}^{(\nu)}, \tag{3.26}$$

where $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ can be viewed as the ν-layer generator matrix.

For $P_{CSI-RS} = 4$, the permutation matrices $\mathbf{P}^{(\nu)}$ are defined in Table 3.7. The coefficient matrices are defined as $\mathbf{C}_{i_{1,3}}^{(\nu)} = \mathbf{1}_{4\times\nu}$ for $P_{CSI-RS} = 4$.

Table 3.7. Permutation matrices for $P_{CSI-RS} = 4$.

	$\mathbf{P}^{(\nu)} = \mathbf{I}_4(:, c)$
$\nu = 1, i_2 = 0, 1$	$c = 1$
$\nu = 1, i_2 = 2, 3$	$c = 2$
$\nu = 2, i_{1,3} = 0$	$c = \begin{bmatrix} 1 & 3 \end{bmatrix}$
$\nu = 2, i_{1,3} = 1$	$c = \begin{bmatrix} 1 & 4 \end{bmatrix}$
$\nu = 3$	$c = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

For $\nu = 1$, we have

$$\begin{aligned}(\mathbf{W}_I^{(4)}\mathbf{P}^{(1)}) &= \begin{cases} \mathbf{W}_I^{(8)}(:, 1), & n = 0, 1 \\ \mathbf{W}_I^{(8)}(:, 2), & n = 2, 3 \end{cases} \\ &= \begin{cases} \begin{bmatrix} v_{l,m} \\ \phi_n v_{l,m} \end{bmatrix}, & n = 0, 1 \\ \begin{bmatrix} v_{l,m} \\ -\phi_n v_{l,m} \end{bmatrix}, & n = 2, 3 \end{cases} \\ &= \mathbf{W}_I^{(1)}.\end{aligned}$$

For $\nu = 2$, according to Table 5.2.2.2.1-6 of [1] $l = i_{1,1}, l' = i_{1,1} + k_1, m = i_{1,2}, m' = i_{1,2} + k_2$ and $n = i_2$

$$\begin{aligned}(\mathbf{W}_I^{(4)}\mathbf{P}^{(2)}) &= \begin{cases} \mathbf{W}^{(4)}\left(:, \begin{bmatrix} 1 & 3 \end{bmatrix}\right), & i_{1,3} = 0 \\ \mathbf{W}^{(4)}\left(:, \begin{bmatrix} 1 & 4 \end{bmatrix}\right), & i_{1,3} = 0 \end{cases} \\ &= \begin{cases} \begin{bmatrix} v_{l,m} & v_{l,m} \\ \phi_n v_{l,m} & -\phi_n v_{l,m} \end{bmatrix}, & i_{1,3} = 0 \\ \begin{bmatrix} v_{l,m} & v_{l',m'} \\ \phi_n v_{l,m} & -\phi_n v_{l',m'} \end{bmatrix}, & i_{1,3} = 1 \end{cases}\end{aligned}$$

$$= \mathbf{W}_I^{(2)}.$$

For case $\nu = 3$, $i_{1,3} = 0, 1, 2$ we have

$$\begin{aligned}(\mathbf{W}_I^{(4)}\mathbf{P}^{(2)}) \odot \mathbf{C} &= \mathbf{W}^{(4)}\left(:, \begin{bmatrix} 1 & 3 \end{bmatrix}\right) \\ &= \begin{bmatrix} v_{l,m} & v_{l',m'} & v_{l,m} \\ \phi_n v_{l,m} & v_{l',m'} & -\phi_n v_{l,m} \end{bmatrix} \\ &= \mathbf{W}_I^{(3)}.\end{aligned}$$

Since the derived formulae coincide with the ones presented in Tables 5.2.2.2.1-5 - 5.2.2.2.1-7 [1], (3.26) can be concluded to hold for $P_{CSI-RS} = 4$ as well.

□

A consequence of (3.26) is that the characteristic matrix $\mathbf{D}_{l,m}$ is unique up to pair $(\nu, i_{1,3})$, since the each pair (l, m) corresponds to a unique vector $v_{l,m}$. On the other hand, the fact that the characteristic matrices coincide over all ranks proves to be useful in terms of memory consumption when implementing an algorithm based on (3.26).

As an example, let's consider a case where $P_{CSI-RS} = 8, \nu = 7, (N_1, N_2) = (2, 2)$ and $(i_{1,1}, i_{1,2}, i_2) = (0, 0, 1)$. We have $I = I_2$ and $l = i_{1,1}, m = 1, 2$. By (3.12) and (3.19) we have

$$\mathbf{D}_{0,0} = \mathbf{I}_{2\times 2} \otimes \mathrm{diag}(v_{0,0}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{I}_8$$

and

$$\mathbf{G}_I = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ j & -j & j & -j & -1 & 1 & -1 & 1 \\ j & -j & j & -j & 1 & -1 & 1 & -1 \\ j & -j & -j & j & -1 & 1 & 1 & -1 \\ j & -j & -j & j & 1 & -1 & -1 & 1 \end{bmatrix}.$$

From Table 3.5 and by (3.18) we further have

$$\mathbf{C}_{i_{1,3}}^{(7)} = 1_{8\times 7}, \quad \mathbf{P}^{(7)} = \mathbf{I}_8\left(:, \begin{bmatrix} 1 & 2 & 3 & 5 & 6 & 7 & 8 \end{bmatrix}\right)$$

which eventually yields by (3.20)

$$
\begin{aligned}
\mathbf{W}_I^{(7)} &= \left((\mathbf{D}_{0,0}\mathbf{G}_I)\mathbf{P}^{(7)}\right) \odot \mathbf{C}_{i_{1,3}}^{(7)} \\
&= \mathbf{G}_I\mathbf{P}^{(7)} \\
&= \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & 1 & 1 \\
j & -j & j & -1 & 1 & -1 & 1 \\
j & -j & j & 1 & -1 & 1 & -1 \\
j & -j & -j & -1 & 1 & 1 & -1 \\
j & -j & -j & 1 & -1 & -1 & 1
\end{bmatrix} \\
&= \begin{bmatrix}
v_{0,0} & v_{0,0} & v_{O_1,0} & v_{0,O_2} & v_{0,O_2} & v_{O_1,O_2} & v_{O_1,O_2} \\
\phi_1 v_{0,0} & -\phi_1 v_{0,0} & \phi_1 v_{O_1,0} & v_{0,O_2} & -v_{0,O_2} & v_{O_1,O_2} & -v_{O_1,O_2}
\end{bmatrix}.
\end{aligned}
$$

The mathematical structure of the codebooks enables representation of the precoders in terms of characteristic matrices and generator matrices. The diagonal matrix acts as an identifier up to pair $(\nu, i_{1,3})$, as same characteristic matrices are used for full rank and non-full rank cases. In comparison to the precoder, the corresponding characteristic matrix has ν times fewer non-zero elements. Storing the characteristic matrices and generator matrices allows reducing memory overhead in comparison to a direct implementation, where the precoders would be stored unchanged. With the aim of presenting an implementation based on this observation, we will describe an algorithm justified by the mathematical work presented in this section and by reductions in memory consumption.

3.4 Transforming Precoders Into Binaries And Binaries Into Precoders

Representing $\mathbf{W}_I^{(\nu)}$ in terms of $\mathbf{D}_{l,m}$ and $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ is enabled by (3.26). In the heart of the algorithm is to exploit (3.26) to save memory by storing the characteristic matrices and the generator matrices instead of the full precoder. On a high level, $\mathbf{D}_{l,m}$ and $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ will be algorithmically encoded into binaries, which can then conversely be used to retrieve the precoder $\mathbf{W}_I^{(\nu)}$. In this thesis, the generator matrices will be stored, and only an implementation for retrieving a characteristic matrix from a binary number will be presented. Figure 3.9 presents a high level illustration of this procedure.

The core idea of the algorithm is to retrieve the characteristic matrix of a

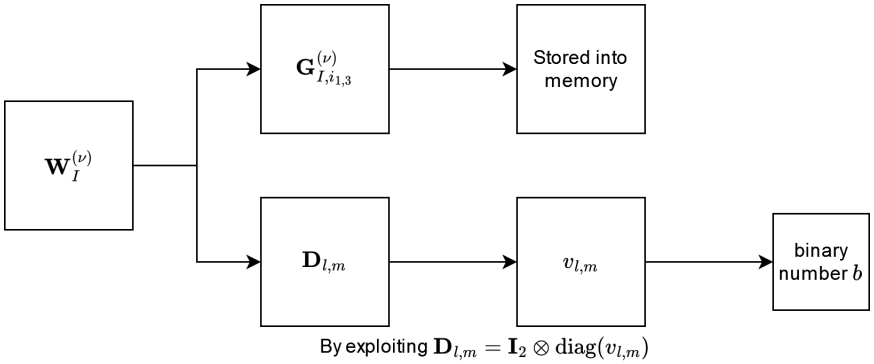

Figure 3.9. Flowchart of the binary encoding.

precoder by using a binary number identified with it. The characteristic matrix is determined by $v_{l,m}$ according to (3.12). Furthermore, we have $v_{l,m} = v_l \otimes u_m$ by (3.6). Since the entries of v_l and u_m are of form $\omega_{N_1O_1}^{n_1}$ and $\omega_{N_2O_2}^{n_2}$, respectively, where $n_1, n_2 \in \mathbb{Z}$, a product of their entries will be a power of a root of unity. Particularly, the entries of $v_{l,m}$ are powers of ω_M, where M is the least common multiple of N_1O_1, N_2O_2, $\mathrm{lcm}(N_1O_1, N_2O_2)$. In fact, for $P_{CSI-RS} \in \{4, 8\}$, we have

$$M = \mathrm{lcm}(N_1O_1, N_2O_2) = \max(N_1O_1, N_2O_2), \tag{3.27}$$

because of both products being powers of 2 based on (3.1). The practical consequence of (3.6) is, that the entries of $v_{l,m}$ can be expressed in terms of the exponents of ω_M. More formally, the elements of $v_{l,m}$ will be mapped into integers according to

$$z \mapsto \frac{-M}{2\pi}\mathrm{Arg}(z), \tag{3.28}$$

where z is a complex number, and Arg(.) returns the argument of a complex number. The range of these exponents will be $0, \ldots, M = \max(N_1O_1, N_2O_2)$, implying the number of needed bits per exponent is $\log_2(M)$. Together with (3.12), this implies that $\mathbf{D}_{l,m}$ can be completely stored into memory using

$$\log_2(\max(N_1O_1, N_2O_2)) \cdot \text{number of entries in } v_{l,m} \tag{3.29}$$

bits, where "$\cdot$" is the multiplication of real numbers. Concatenating the binary numbers corresponding to each exponent, we can identify $\mathbf{D}_{l,m}$ with a single binary number.

Reversing the process yields an algorithm to retrieve $\mathbf{D}_{l,m}$ using a characteristic binary number b. First, binary number b will be split into smaller binaries by taking the $\log_2(M)$ leftmost bits as one binary, and moving on to the next $\log_2(M)$ bits until the end of b. For example, in the case $P_{CSI-RS} = 4$ we have $M = 2 \cdot 4 = 8$, and thus, a single sub-binary length being $\log_2(8) = 3$ bits. A high-level illustration of the binary split is presented in Figure 3.10.

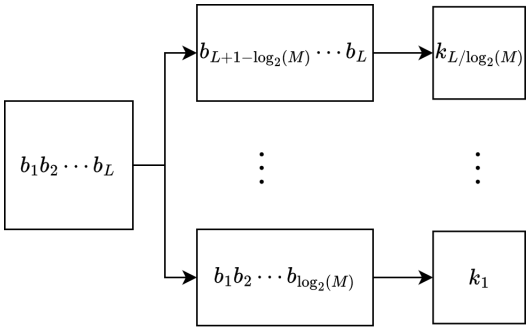

Figure 3.10. Splitting the characteristic binary into exponents.

The next step is to transform the sub-binaries into integer exponents. Using the exponents and information on N_1, N_2, O_1, O_2, the powers of the roots of unity can be computed by reversing (3.28) as follows

$$k \mapsto \omega_M^k, \tag{3.30}$$

where $k \in \mathbb{Z}$. The output corresponds to the values of $v_{l,m}$, allowing to compute $\mathbf{D}_{l,m}$ according to (3.12). Lastly, N_1, N_2, O_1, O_2, ν, and I are used to retrieve corresponding $\mathbf{G}^{(\nu)}_{I,i_{1,3}}$, which can be used with $\mathbf{D}_{l,m}$ according to (3.26) to compute $\mathbf{W}_I^{(\nu)}$. Figure 3.11 illustrates the retrieval of the precoder on the high level.

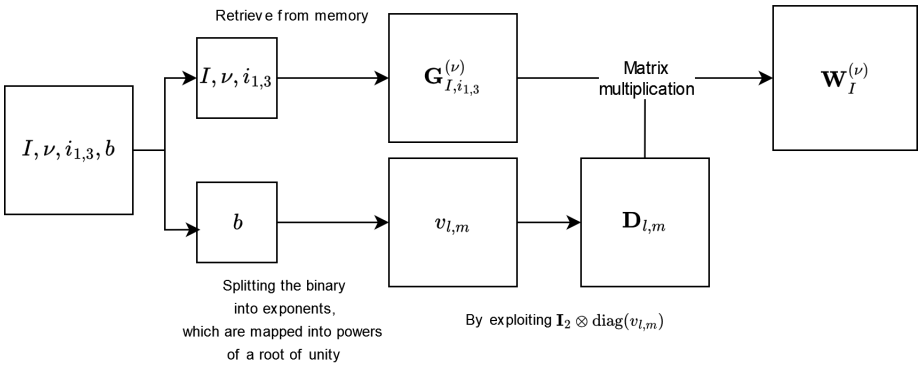

Figure 3.11. Flowchart of retrieving $\mathbf{W}_I^{(\nu)}$ based on I, ν, and binary b.

One of the advantages of the proposed algorithm is the reduction of memory with the cost of computational complexity. From a hardware point of view, $\mathbf{D}_{l,m}$ is more efficient to store in memory than the corresponding $\mathbf{W}_I^{(\nu)}$, since $\mathbf{W}_I^{(\nu)}$ has $8 \cdot \nu$ non-zero elements to be stored. On the other hand, the number of needed generators per ν, N_1, N_2, O_1, O_2 is also smaller than the size of the full codebook. For example, the number of generators is 2 when $(N_1, N_2) = (4, 1)$ and $P_{CSI-RS} = \nu = 8$, since according to (3.3) and (3.19) there is one

generator per n, and by Table 3.4, the range of n in this case is $\{0, 1\}$. For every configuration, each generator matrix and characteristic matrix determine a unique precoder, implying that the number of characteristic matrices is the number of precoders divided by the number of generators. This highlights the potential of the algorithm in mitigating memory use, and advocates the algorithm worth implementing in practice.

4 Proposed Implementation

This chapter discusses the implementation of the generation algorithm described in Chapter 3 and the process of retrieving the precoding matrices using characteristic binaries in an efficient way. We will present a description of the generic definition of the matrices, followed by discussing the implementation of the algorithm. Moreover, the algorithm is validated by comparing the precoding results achieved by the proposed algorithm and by directly implemented matrices. An implementation and tests were conducted for cases $P_{CSI-RS} \in \{4, 8\}$, covering all respective configurations of (N_1, N_2, O_1, O_2) presented in [1]. Additionally, the implementation covers all cases $\nu = 1, \ldots, P_{CSI-RS}$ for all configurations. The number of precoders depends on the ranges of $i_{1,1}, i_{1,1}, i_{1,3},$ and i_2 as follows

$$N = \text{card}(\text{range}(i_{1,1})) \cdot \text{card}(\text{range}(i_{1,2})) \cdot \text{card}(\text{range}(i_{1,3})) \cdot \text{card}(\text{ran}(i_2)), \quad (4.1)$$

where ran(.) is the range of a variable and card(.) is the cardinality of a set. The numbers of precoders for each configuration are listed in Table 4.1. Table 4.1 shows the codebook sizes being large for $P_{CSI-RS} = 8$, and $\nu \in \{2, 3, 4\}$, which is due to the varying values of $i_{1,3}$. However, as presented in (3.26), the non-full rank matrices can be extracted from the full rank precoder using a permutation matrix and a coefficient matrix. Another implication of (3.26) is that the characteristic matrix is the same regardless of the value of $i_{1,3}$ for all $\nu \in \{2, 3, 4\}$. According to (3.26), generator $\mathbf{G}^{(\nu)}_{I,i_{1,3}}$ can be used to represent the effect of the permutation and coefficient matrices. Consequently, it is sufficient to store these generators and characteristic matrices for each configuration, which proves to be significantly more efficient in terms of memory compared to storing all precoders.

Recall that Section 3.4 describes an algorithm for generating precoding matrices based on (3.26). As discussed in the same section, the diagonal elements of $\mathbf{D}_{l,m}$ can be encoded by mapping them to their respective exponents. These exponents can further be transformed into binary numbers using $\log_2(M)$ bits, where M is defined as in (3.27). The algorithm description of Section 3.4 acts as a foundation for the proposed implementation of this thesis.

4.1 Preparation

Considering a common method to reduce complexity in practical systems, we apply look-up tables to retrieve some necessary variables. The values of ω^k_M,

Table 4.1. Numbers of precoders per configuration

$(N_1, N_2, O_1, O_2, \nu)$	Number of Precoders (N)
$(2, 1, 4, 1, 1)$	32
$(2, 1, 4, 1, 2)$	32
$(2, 1, 4, 1, 3)$	16
$(2, 1, 4, 1, 4)$	16
$(2, 2, 4, 4, 1)$	256
$(2, 2, 4, 4, 2)$	512
$(2, 2, 4, 4, 3)$	384
$(2, 2, 4, 4, 4)$	384
$(2, 2, 4, 4, 5)$	128
$(2, 2, 4, 4, 6)$	128
$(2, 2, 4, 4, 7)$	128
$(2, 2, 4, 4, 8)$	128
$(4, 1, 4, 1, 1)$	64
$(4, 1, 4, 1, 2)$	128
$(4, 1, 4, 1, 3)$	96
$(4, 1, 4, 1, 4)$	96
$(4, 1, 4, 1, 5)$	32
$(4, 1, 4, 1, 6)$	32
$(4, 1, 4, 1, 7)$	16
$(4, 1, 4, 1, 8)$	16

$k = 0, \ldots, M-1$, the generators $\mathbf{G}^{(\nu)}_{I,i_{1,3}}$, and the binaries of the diagonal matrix $\mathbf{D}_{l,m}$ are all stored in memory.

The powers $\omega_M^k, M \in \{8, 16\}$, are stored into vector containers *Powers8* and *Powers16*, according to the value of M. In the containers, ω_M^k is assigned as the $(k+1)$th element of the vector. The diagonal matrix $\mathbf{D}_{l,m}$ can be identified with the vector $v_{l,m}$ according to (3.12). The elements of $v_{l,m}$ can be mapped into an integer-valued according to (3.28). The flowchart of transforming the diagonal matrix into a binary is presented in Figure 4.1. Since there will be a unique $\mathbf{D}_{l,m}$ for each pair $(i_{1,1}, i_{1,2})$ up to ν, the binaries can be ordered based on an integer index instead of an index tuple according to the mapping

$$(i_{1,1}, i_{1,2}) \mapsto i_{1,1} \cdot l(i_{1,2}) + i_{1,2} + 1, \tag{4.2}$$

where the term 1 is added for Matlab indexing. The binaries are stored in *Binaries4* for $P_{CSI-RS} = 4$ and for $P_{CSI-RS} = 8$ in *Binaries8_8* and *Binaries8_16* for

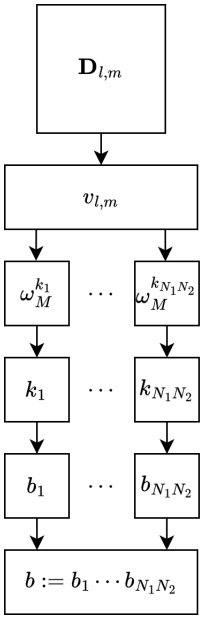

Figure 4.1. Flowchart of the binary encoding.

$N_1 = 2$ and $N_1 = 4$, respectively. The generator matrices $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ are stored in a hash table *GMatrices*. The hash table uses configuration $\left(P_{CSI-RS}, \nu, N_2, i_{1,3}, n\right)$ as a key and matrix $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ is assigned as the corresponding value, that will be returned by the hash table when a key is used. Following the possible combinations of ν and P_{CSI-RS}, the numbers of integrated generators in each case have been listed in Table 4.2. Formula (3.26) implies that the precoders can be completely covered by storing the diagonal matrices $\mathbf{D}_{l,m}$ and total of 76 generator matrices $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$. Moreover, storing the integrated generators enables avoiding some extra computational steps involved with using the coefficient and permutation matrices.

According to Table 4.1 the total number of precoding matrices is 2624. On the other hand, there are a total of 76 generators $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ as listed in Table 4.2, implying that generating the codebooks using the generators and diagonal matrices corresponding to pair (l, m) defined for $\nu = P_{CSI-RS}$ allows significant reduction of memory use.

4.2 Implementation

The PMI acts as the key to retrieve the necessary matrices for generating the precoder according to (3.26) because it is used to retrieve the precoder directly in [1]. The implementation proposed in this thesis is based on the idea of reversing the binary encoding process of $D_{l,m}$ and retrieving the correct generator for PMI. To determine $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ one needs the index tuple $(N_2, O_1, O_2, i_2, i_{1,3})$

Table 4.2. Number of $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ for each layer-port pair.

	$P_{CSI-RS} = 4$	$P_{CSI-RS} = 8$	$P_{CSI-RS} = 8$
$\nu = 1$	4	4	4
$\nu = 2$	4	8	8
$\nu = 3$	2	6	6
$\nu = 4$	2	6	6
$\nu = 5$		2	2
$\nu = 6$		2	2
$\nu = 7$		2	2
$\nu = 8$		2	2
Total	12	32	32

and to determine $\mathbf{D}_{l,m}$ one needs $(i_{1,1}, i_{1,2}, N_1, N_2, O_1, O_2)$. Therefore, a single codebook can be identified with index tuple $\big(N_1, N_2, O_1, O_2, i_{1,1}, i_{1,2}, i_{1,3}, i_2, \nu\big)$.

For each ν, the sets of values for $i_{1,1}, i_{1,2}, i_{1,3}, i_2$ are retrieved on each loop according to ν and the configuration (N_1, N_2, O_1, O_2). The function will return a set of indices from which combinations will be formed based on the PMI mapping in [1]. The index ranges in each respective case can be found from Tables 4.3 - 4.5.

For a given precoding matrix $\mathbf{W}_I^{(\nu)}$ and configuration (N_1, N_2, O_1, O_2) the index pair $(i_{1,1}, i_{1,2})$ is mapped according to (4.2) into an integer number, which is then used to retrieve binary number corresponding to $\mathbf{D}_{l,m}$ associated with $\mathbf{W}_I^{(\nu)}$. For each covered case, the binary will be divided into sub-binaries corresponding to integer exponents of a root of unity. These powers are then concatenated into a vector that is diagonalized using Matlab function diag(.) and the resulting matrix will be mapped into $\mathbf{D}_{l,m}$ by left Kronecker-product with $\mathbf{I}_2$. Finally the desired $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ is retrieved from a lookup table allowing the use of (3.26) to compute the desired precoding matrix. We start by proposing an implementation for $P_{CSI-RS} = 4$.

4.2.1 Implementation for Four Antenna Ports

For $P_{CSI-RS} = 4$ there is only one configuration, that is, $(N_1, N_2, O_1, O_2) = (2, 1, 4, 1)$. Since $M = max(O_1 N_1, O_2 N_2) = 8$, lookup table $Powers8$ is used to retrieve the powers of ω_8. Furthermore, lookup tables *GMatrices* and *Binaries4* are used to retrieve the desired generator matrices and corresponding exponents

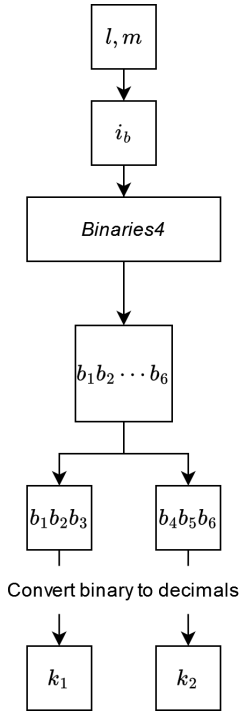

Figure 4.2. Retrieving k_1 and k_2 based on the pair (l, m).

to determine $\mathbf{D}_{l,m}$. The ranges for each determining index are listed in Table 4.3.

Table 4.3. Index ranges per layer for $P_{CSI-RS} = 4$.

$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$
$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$
$i_{1,2} = 0$	$i_{1,2} = 0$	$i_{1,2} = 0$	$i_{1,2} = 0$
$i_2 = 0, \ldots, 3$	$i_2 = 0, \ldots, 1$	$i_2 = 0, \ldots, 1$	$i_2 = 0, \ldots, 1$
-	$i_{1,3} = 0, 1$	$i_{1,3} = 0$	$i_{1,3} = 0$

Starting from retrieving $\mathbf{D}_{l,m}$, the index pair $(l, m) = (i_{1,1}, i_{1,2})$ will be mapped according to (4.2) giving us total of 8 binaries to be decoded into diagonal matrices. The index i_b will be used to retrieve the correct binary b from the lookup table *Binaries4*. Once b is known, it will be decomposed into 2 sub-binaries b_1 and b_2. This is carried out by setting the string of $\log_2(M) = \log_2(8) = 3$ bits as b_1, and then moving to the next string of $\log_2(M)$ bits, which will be set as b_2. The binary numbers b_1 and b_2 are converted into integers k_1 and k_2. The retrieval flow of k_1 and k_2 is illustrated in Figure 4.2. Using k_1 and k_2, the powers $\omega_8^{k_1}$ and $\omega_8^{k_2}$ are retrieved using *Powers8*. Finally,

$\mathbf{D}_{l,m}$ is computed as

$$\mathbf{D}_{l,m} = \begin{bmatrix} \omega_8^{k_1} & 0 & 0 & 0 \\ 0 & \omega_8^{k_2} & 0 & 0 \\ 0 & 0 & \omega_8^{k_1} & 0 \\ 0 & 0 & 0 & \omega_8^{k_2} \end{bmatrix}.$$

Figure 4.3 describes the process of mapping k_1 and k_2 into $\mathbf{D}_{l,m}$. Retrieval of $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ is realized by using look up table *GMatrices* and key $(P_{CSI-RS}, \nu, N_2, i_{1,3}, i_2)$. The precoder $\mathbf{W}_I^{(\nu)}$ can be computed by $\mathbf{D}_{l,m}\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ according to (3.26).

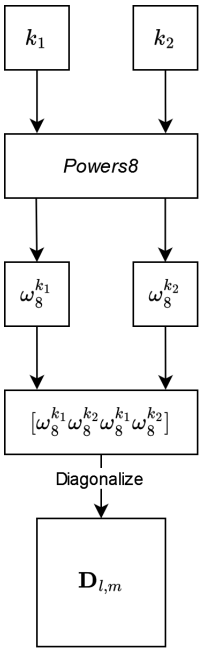

Figure 4.3. Converting k_1 and k_2 into $\mathbf{D}_{l,m}$.

4.2.2 Implementation for Eight Antenna Ports

For $P_{CSI-RS} = 8$, there are two configurations to cover, namely, $(N_1, N_2, O_1, O_2) = (4, 1, 4, 1)$ and $(2, 2, 4, 4)$. The implementation to generate all the precoders and validate them is aligned in both cases with the exception of using separate lookup tables for powers of ω_M and associated binaries. For $(N_1, N_2, O_1, O_2) = (4, 1, 4, 1)$, we have $M = \max(O_1N_1, O_2N_2) = 16$, and thus *Powers16* and *Binaries8_16* are used, whereas for $(N_1, N_2, O_1, O_2) = (2, 2, 4, 4)$ we have $M = \max(O_1N_1, O_2N_2) = 8$ and the lookup tables *Powers8* and *Binaries8_8* are used. The lookup table *GMatrices* for the generators is common in both cases. The index ranges for $i_{1,1}, i_{1,2}, i_{1,3}, i_2$ for each ν for both configurations are listed in Tables 4.4 and 4.5. We begin describing the proposed implementation for $(N_1, N_2, O_1, O_2) = (2, 2, 4, 4)$.

Table 4.4. Index ranges for $P_{CSI-RS} = 8, N_1 = 2$ for codebook mode 1.

$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$
$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$
$i_{1,2} = 0, \ldots, 7$	$i_{1,2} = 0, \ldots, 7$	$i_{1,2} = 0, \ldots, 7$	$i_{1,2} = 0, \ldots, 7$
$i_2 = 0, \ldots, 3$	$i_2 = 0, 1$	$i_2 = 0, 1$	$i_2 = 0, 1$
-	$i_{1,3} = 0, \ldots, 3$	$i_{1,3} = 0, 1, 2$	$i_{1,3} = 0, 1, 2$
$\nu = 5$	$\nu = 6$	$\nu = 7$	$\nu = 8$
$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$
$i_{1,2} = 0, \ldots, 7$	$i_{1,2} = 0, \ldots, 7$	$i_{1,2} = 0, \ldots, 7$	$i_{1,2} = 0, \ldots, 7$
$i_2 = 0, 1$	$i_2 = 0, 1$	$i_2 = 0, 1$	$i_2 = 0, 1$

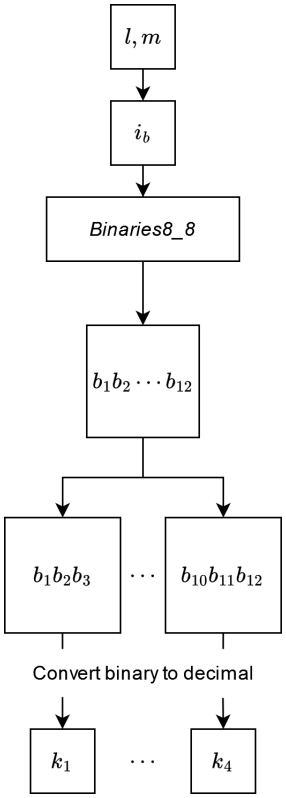

Figure 4.4. Retrieving k_1 and k_2 based on the pair (l, m).

Retrieval of $\mathbf{D}_{l,m}$ is implemented in a similar manner as for $P_{CSI-RS} = 4$ and follows the flow chart presented in Figure 4.4. The pair $(l, m) = (i_{1,1}, i_{1,2})$ is mapped according to (4.2) in this case resulting in a total of 64 unique binaries. A binary number is extracted from the lookup table *Binaries4* based on i_b. It is then decomposed into sub-binaries by looping over its bits using step size of

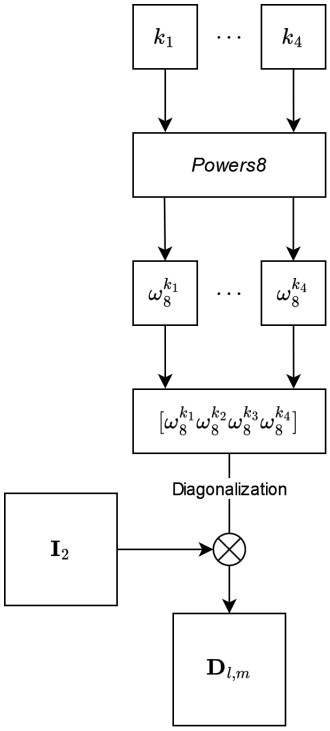

Figure 4.5. Converting k_1 and k_2 into $\mathbf{D}_{l,m}$ when $P_{CSI-RS} = 8$ and $N_2 = 2$.

$\log_2(8) = 3$ bits. giving us 4 sub-binaries $b_1, \ldots, b_4$ of 3 bits.

Once the sub-binaries are extracted, they will be converted into four respective exponents $k_1, \ldots, k_4$. The exponents are used to extract the powers of ω_8 from *Powers8*, which are then concatenated into a vector $[\omega_8^{k_1} \ldots \omega_8^{k_4}]$. This vector is diagonalized using Matlab's diag(.) function and further mapped into $\mathbf{D}_{l,m}$ by left-Kronecker product with $\mathbf{I}_2$. The flow chart of the process is presented in Figure 4.5. The retrieval of $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ is performed similarly as it was for P_{CSI-RS} by retrieving the matrix corresponding to the key $(P_{CSI-RS}, \nu, N_2, i_{1,3}, i_2)$ from *GMatrices* finally allowing computation of the precoder $\mathbf{W}_I^{(\nu)}$ using (3.26).

Finally, covering the configuration $(N_1, N_2, O_1, O_2) = (4, 1, 4, 1)$ for $P_{CSI-RS} = 8$, the steps of computing $\mathbf{D}_{l,m}$ are similar to the scenarios covered previously. Table 4.5 lists the ranges for $i_{1,1}, i_{1,2}, i_{1,3}$, and i_2. A key difference to previous cases is $M = \max(N_1O_1, N_2O_2) = 16$, implying the number of needed bits per sub-binary is $\log_2(16) = 4$.

Figure 4.6 presents a flow chart of the process of mapping the pair (l, m) into exponents $k_1, \ldots, k_4$. The pair $(l, m) = (i_{1,1}, i_{1,2})$ is mapped into i_b using (4.2) which yields a total of 16 unique binary representations of $\mathbf{D}_{l,m}$. The retrieved index i_b is used to extract the correct 16-bit binary b from *Binaries8_16* breaking down b into 4-bit sub-binaries $b_1, \ldots, b_4$ which in turn can be converted into exponents $k_1, \ldots, k_4$.

Table 4.5. Index ranges for $P_{CSI-RS} = 8, N_2 = 1$.

$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$
$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$
$i_{1,2} = 0$	$i_{1,2} = 0$	$i_{1,2} = 0$	$i_{1,2} = 0$
$i_2 = 0, \ldots, 3$	$i_2 = 0, 1$	$i_2 = 0, 1$	$i_2 = 0, 1$
-	$i_{1,3} = 0, \ldots, 3$	$i_{1,3} = 0, 1, 2$	$i_{1,3} = 0, 1, 2$
$\nu = 5$	$\nu = 6$	$\nu = 7$	$\nu = 8$
$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$	$i_{1,1} = 0, \ldots, 7$
$i_{1,2} = 0$	$i_{1,2} = 0$	$i_{1,2} = 0$	$i_{1,2} = 0$
$i_2 = 0, 1$	$i_2 = 0, 1$	$i_2 = 0, 1$	$i_2 = 0, 1$

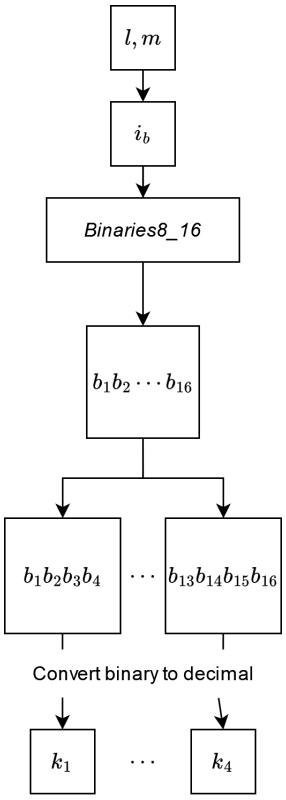

Figure 4.6. Retrieving k_1 and k_2 based on the pair (l, m).

The exponents are then further mapped into powers of ω_{16} using *Powers16* and the powers are concatenated into a vector $[\omega_{16}^{k_1}\omega_{16}^{k_2}\omega_{16}^{k_3}\omega_{16}^{k_4}]$. The vector is diagonalized using the Matlab function diag(.) yielding a diagonal vector that is Kronecker-multiplied from the left side by $\mathbf{I}_2$ finally yielding $\mathbf{D}_{l,m}$. This stage of the algorithm is presented in Figure 4.7. Lastly, the corresponding $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ is extracted from *GMatrices* the key $(P_{CSI-RS}, \nu, N_2, i_{1,3}, i_2)$ and the precoder is

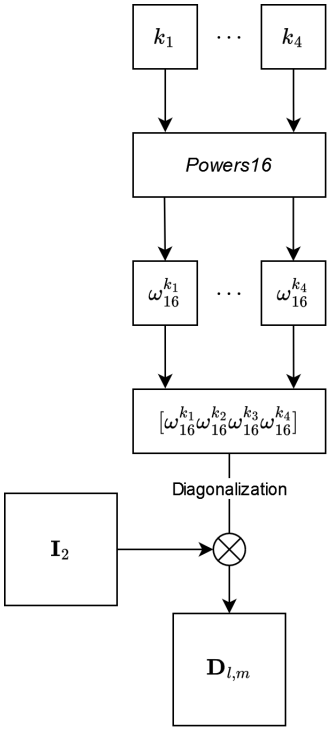

Figure 4.7. Converting k_1 and k_2 into $\mathbf{D}_{l,m}$ when $P_{CSI-RS} = 8$ and $N_2 = 1$.

computed using (3.26).

In this chapter we have presented an implementation to generate NR Type I CSI precoders. The aim of the proposed implementation is to improve the use of memory comparing to the direct implementation of the codebooks. The proposed algorithm enables computing and retrieving $\mathbf{D}_{l,m}$ and $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ using limited data, where only generators, look-up tables and binaries corresponding to a specific diagonal matrix need to be accessed from the memory. However, the efficiency of the proposed algorithm in terms of memory and computational complexity needs to be studied carefully to gain thorough understanding of the algorithm's potential. In Chapter 5 we discuss and assess the complexity and memory use of the proposed algorithm in detail.

5 Efficiency Analysis of the Proposed Algorithm

In this chapter the efficiency of the proposed algorithm is discussed. The complexity of the algorithm is analyzed in terms of multiplication and addition of complex numbers. These operations further compose of real additions and real multiplications. The amount of memory required to store all the binary numbers associated with the characteristic matrices, and the generator matrices is also analyzed in terms of bits. The complexity of the two stage precoding with $\mathbf{D}_{l,m}$ and $\mathbf{G}^{(\nu)}_{I,i_{1,3}}$ is also analyzed. The memory overhead and complexity of the proposed algorithm is compared against those of direct implementation of the precoders. In terms of memory, we assume that the elements of a precoder are stored and retrieved using similar look-up table approach as for the characteristic matrix entries in the proposed algorithm. We will show, that the algorithm enables implementing the precoder generation with less needed memory than by directly storing each precoder.

5.1 Complexity

The complexity of the algorithm shall be determined by the number of computations, i.e., the real additions and multiplications needed in order to retrieve the precoding matrix. We assume the natural definition of complex multiplication, that is,

$$(a + jb)(c + jb) = ac - bd + j(ab + cd),$$

which then has four real multiplications and two real additions. The analysis is also based on the assumption of using lookup tables for calculating the powers of ω_n and retrieving $\mathbf{G}^{(\nu)}_{I,i_{1,3}}$. Storing these, will in turn have an impact on needed memory, which will be discussed in this chapter.

To retrieve a precoding matrix decomposition for index tuple $(i_{1,1}, i_{1,2}, i_2, i_{1,3})$ with the configuration (N_1, N_2, O_1, O_2), the indices are first mapped into a single index i_b according to (4.1) to retrieve the binary numbers representing the given configuration, which adds up to three additions and 3 multiplications. Once the binary is known, the number of needed bits for each diagonal element is calculated by $N_1 \cdot O_1$, which results in one more multiplication. This is followed by looping over the number of ports to retrieve the exponents of ω out of the binary. At each step RotId, the position of starting bit for each exponent

is calculated as

$$(\text{RotId} - 1) \cdot N_1 O_1$$

adding one P_{CSI-RS} multiplications. Knowing the length of each binary representing one power, the binaries are transformed into decimals. Finally, using the lookup table for ω_M^k defined in Section 4.1, the exponents are used to retrieve the first half diagonal of $\mathbf{D}_{l,m}$, which is sufficient to determine $\mathbf{D}_{l,m}$ completely as demonstrated in (3.12), and the configurations are used to extract $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ from the look-up table.

Once the decomposition matrices are computed, the precoder $\mathbf{W}_I^{(\nu)}$ is calculated according to (3.26). The calculation involves one matrix multiplication between $\mathbf{D}_{l,m}$ and $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$. As $\mathbf{D}_{l,m}$ is a diagonal matrix, a left multiplication by it translates to taking the element-wise product of each column of $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ and the diagonal of $\mathbf{D}_{l,m}$. Thus, there is a total of $P_{CSI-RS} \cdot \nu \cdot 4$ real multiplications and $P_{CSI-RS} \cdot \nu \cdot 2$ real additions. In summary, all the computations involved in the matrix multiplication and the binary mapping and decomposition result in a total of $5 \cdot P_{CSI-RS} \cdot \nu + 4$ real multiplications and $2 \cdot P_{CSI-RS} \cdot \nu + 4$ real additions. The needed numbers of computations have been listed in Table 5.1.

Table 5.1. Number of real computations.

	Computations
Mapping the configuration into a correct binary and binary decomposition	$4 + P_{CSI-RS}$ multiplications, 4 additions
Left multiplication by $\mathbf{D}_{l,m}$	$4 \cdot P_{CSI-RS} \cdot \nu$ multiplications, $2 \cdot P_{CSI-RS} \cdot \nu$ additions
Total	$5 \cdot P_{CSI-RS} \cdot \nu + 4$ multiplications, $2 \cdot P_{CSI-RS} \cdot \nu + 4$ additions

From the precoding point of view, the matrix composition and the associativity of the matrix multiplication allows a simplification in the precoding according to (3.26) as

$$\mathbf{W}_I^{(\nu)}\mathbf{x} = \left(\mathbf{D}_{l,m}\mathbf{G}_{I,i_{1,3}}^{(\nu)}\right)\mathbf{x} = \mathbf{D}_{l,m}\left(\mathbf{G}_{I,i_{1,3}}^{(\nu)}\mathbf{x}\right),$$

where $\mathbf{x}$ is a signal vector. Since the elements of $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ are in $\{1, -1, j, -j\}$, there are $P_{CSI-RS} \cdot \nu \cdot 2$ in the multiplication $\mathbf{G}_{I,i_{1,3}}^{(\nu)}\mathbf{x}$. Right-multiplying the product by $\mathbf{D}_{l,m}$ has P_{CSI-RS} complex multiplications, i.e., $4 \cdot P_{CSI-RS}$ real multiplications

and $3 \cdot P_{CSI-RS}$ additions. In contrast, the direct multiplication of $\mathbf{x}$ by $\mathbf{W}_I^{(\nu)}$ has $P_{CSI-RS} \cdot \nu$ complex multiplications, or $4 \cdot P_{CSI-RS} \cdot \nu$ real multiplications and $3 \cdot P_{CSI-RS} \cdot \nu$ additions. The numbers of needed computations have been listed in Table 5.2.

Table 5.2. Number of real computations in different schemes of precoding.

	Computations
Direct precoding by $\mathbf{W}_I^{(\nu)}$	$4 \cdot P_{CSI-RS} \cdot \nu$ multiplications, $3 \cdot P_{CSI-RS} \cdot \nu$ additions
Precoding in parts	$(2 \cdot \nu + 4) \cdot P_{CSI-RS}$ multiplications, $3 \cdot P_{CSI-RS}$ additions
$\frac{\text{Precoding in parts}}{\text{Direct Precoding by } \mathbf{W}_I^{(\nu)}}$	Multiplications: $\frac{1}{2} + \frac{1}{\nu}$, Additions: $\frac{1}{\nu}$

It is worth noting, that when $\nu = 1$, the number of real multiplications is actually larger in precoding in parts than in direct multiplication by $\mathbf{W}_I^{(\nu)}$. However, since the multiplication by $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ is multiplying the elements of $\mathbf{x}$ by either ± 1 or $\pm j$, the factor 2ν is not significant in terms of computations. As ν increases, the number of computations in precoding directly with $\mathbf{W}_I^{(\nu)}$. This means, that the precoding in parts offers the highest improvement, when $\nu = P_{CSI-RS}$. In this case, the number of needed real additions is 12.5 % of of the number of those needed in direct precoding, and the number of needed multiplications is 62.5 % of the number of those needed in direct precoding.

5.2 Memory

The proposed algorithm increases the complexity in terms of real additions and multiplications with the trade-off of reducing the number of needed bits to store the codebooks. As the sizes of the codebooks increases, the number of possible precoders increases accordingly. Furthermore, various combinations of P_{CSI-RS} and ν also increases the number of unique precoders as presented in Table 4.1, rendering the optimization in memory consumption crucial.

In the implementation proposed in Chapter 4, one needs to store the binary number associated with the characteristic matrix $\mathbf{D}_{l,m}$ for each precoder having $\nu = P_{CSI-RS}$ along with generator matrices $\mathbf{G}_I$. According to Table 4.1 we have

16 diagonal matrices for $P_{CSI-RS} = 4$, and $128 + 16 = 144$ diagonal matrices for $P_{CSI-RS} = 8$. Based on (3.12), for $P_{CSI-RS} = 4$, the number of elements in $\mathbf{D}_{l,m}$ that needed for storing is 2, and for $P_{CSI-RS} = 8$, it is 4. Furthermore, the number of needed bits to represent one diagonal element of $\mathbf{D}_{l,m}$ in case of $P_{CSI-RS} = 4$ is 3 as it is for $P_{CSI-RS} = 8, (N_1, N_2) = (2, 2)$. For $P_{CSI-RS} = 8, (N_1, N_2) = (4, 1)$, the number of needed bits per element is 4. Therefore, the total number of needed bits for the characteristic matrices is $128 \cdot 4 \cdot 3 + 16 \cdot 2 \cdot 3 + 16 \cdot 4 \cdot 4 = 1888$ bits, since storing the zeros is redundant.

In the case of generator matrices, it is sufficient to store the integrated generator $\mathbf{G}^{(\nu)}_{I,i_{1,3}}$, adding up to a total of 76 matrices to be stored. However, elements of $\mathbf{G}^{(\nu)}_{I,i_{1,3}}$ are in the set $\{1, -1, j, -j\}$, implying they can be represented using 2 bits per element. The number of bits needed to represent the generator matrices for each each port and rank is calculated as

$$\text{Number of } \mathbf{G}^{(\nu)}_{I,i_{1,3}} \cdot P_{CSI-RS} \cdot \nu \cdot 2. \tag{5.1}$$

In Table 5.3, the numbers of needed bits have been listed for each configuration having taken into account the multiplicity of different P_{CSI-RS} cases. Adding together the number of bits calculated using (5.1) based on the data of Table 5.3, the total number of needed bits to store all generators is 3856 bits. Thus, the total memory needed for storing the needed matrix composition matrices is 5744 bits.

In contrast, storing matrices individually for each covered configuration consumes significantly more memory. With the assumption of retrieving the powers of root of unity using similar look-up table as for the proposed algorithm, the number of needed bits per configuration can be calculated as

$$\text{Number of } \mathbf{W}^{(\nu)}_I \cdot \text{Number of Elements in } \mathbf{W}^{(\nu)}_I \cdot \log_2(M). \tag{5.2}$$

The results in each case are listed in Table 5.4. For $P_{CSI-RS} = 4$ cases, the sum of the corresponding numbers of bits as listed in Table 5.4 is 2496 bits. The sums for $P_{CSI-RS} = 8, (N_1, N_2) = (2, 2)$ and $P_{CSI-RS} = 8, (N_1, N_2) = (4, 1)$ are 87552 bits and 25344 bits, respectively. In summary, a total $2496 + 25344 + 87552 = 115392$ bits is needed to cover all the precoders. Comparing this number with the number of bits needed for the proposed algorithm, the algorithm requires

$$\frac{5744}{115392} \cdot 100\% = 4.9778\% \approx 4.98\%$$

of the bits needed when storing all the precoders directly. This indicates that the proposed scheme requires approximately 95.2% less memory compared to the direct implementation.

Table 5.3. Numbers of needed bits to store each generator for different sizes.

P_{CSI-RS}, ν	Number of matrices	Number of needed bits
$4, 1$	4	32
$4, 2$	4	64
$4, 3$	2	48
$4, 4$	2	64
$8, 1$	8	128
$8, 2$	16	512
$8, 3$	12	576
$8, 4$	12	768
$8, 5$	4	320
$8, 6$	4	384
$8, 7$	4	448
$8, 8$	4	512

Table 5.4. Needed numbers of bits to store all codebooks.

$P_{CSI-RS}, \nu, \log_2(M)$	Number of precoders (N)	Number of needed bits according to (5.2)
4, 1, 3	32	384
4, 2, 3	32	768
4, 3, 3	16	576
4, 4, 3	16	768
8, 1, 3	256	3072
8, 2, 3	512	12288
8, 3, 3	384	13824
8, 4, 3	384	18432
8, 5, 3	128	7680
8, 6, 3	128	9216
8, 7, 3	128	10752
8, 8, 3	128	12288
8, 1, 4	64	1024
8, 2, 4	128	4096
8, 3, 4	96	4608
8, 4, 4	96	6144
8, 5, 4	32	2560
8, 6, 4	32	3072
8, 7, 4	16	1792
8, 8, 4	16	2048

6 Discussion and Conclusions

The generation algorithm can be extended further from 8 ports case into other cases supported by 3GPP by using the approach of diagonal and generator matrices. For case $P_{CSI-RS} \in \{12, 16, 24, 32\}$, formulae (3.12) and (3.26) lay a foundation to use similar generative approach as for cases $P_{CSI-RS} \in \{4, 8\}$. This can be achieved by finding $P_{CSI-RS} \times \nu$ generator matrices $\mathbf{G}^{(\nu)}_{I,i_{1,3}}$ while sustaining the concept of having a diagonal matrix $\mathbf{D}_{l,m}$ for $\nu = 8$. Significant differences rise in the form of extended need of memory and computations, which is expected due to the grown size of the precoders and number of needed bits as function of $\max(N_1O_1, N_2O_2)$ with respect to cases $P_{CSI-RS} \in \{4, 8\}$. One important detail for larger number of antenna ports is that the number of needed bits per element is $\lceil(\log_2(\max(N_1O_1, N_2O_2)))\rceil$, where $\lceil . \rceil$ is the ceiling function. The ceiling function is needed, as for example for $P_{CSI-RS} = 12$ and $(N_1, N_2, O_1, O_2) = (6, 1, 4, 1)$, we have $\max(N_1O_1, N_2O_2) = 24$, which is not an integer power of 2.

For the sake of an example, in case $P_{CSI-RS} = 32$ and $(N_1, N_2, O_1, O_2) = (4, 4, 4, 4)$, there is a total of 256 unique 32×32 diagonal matrices $\mathbf{D}_{l,m}$ and two 32×8 generator matrices $\mathbf{G}^{(8)}_I$. Since $\max(N_1O_1, N_2O_2) = 16$, the needed bits per element of $\mathbf{D}_{l,m}$ is 4. Following the ideas of the proposed algorithm, it is sufficient to store the integer exponents of entries of $v_{l,m}$ giving a total $16 \cdot 4 = 64$ needed bits per $\mathbf{D}_{l,m}$ and $256 \cdot 64 = 16384$ bits to cover all $\mathbf{D}_{l,m}$. The generators $\mathbf{G}^{(8)}_{I,i_{1,3}}$ can be shown to be a QPSK matrix and there are two of them in this case. A single generator in needs $32 \cdot 8 \cdot 2 = 512$ bits meaning 1024 bits are needed to cover all the generators for $\nu = 8$ and 17408 bits to cover the precoders completely. In contrast, if all the precoding matrices were stored individually, the number of needed bits would be $512 \cdot 32 \cdot 8 \cdot 4 = 524288$ implying that the algorithm allows a reduction of

$$\left(1 - \frac{17408}{524288}\right) \cdot 100\% \approx 96.7\%$$

of memory needed to cover the matrices.

Another application to study is the extensibility to cover Type I multi-panel cases, where only $P_{CSI-RS} = 8, 16, 32$ and $\nu = 1, \ldots, 4$ are defined. The configurations are also different, since N_g has been added as an extra parameter. For example, the case $P_{CSI-RS} = 8, (N_g, N_1, N_2, O_1, O_2) = (2, 2, 1, 4, 1)$ utilizes

matrices

$$\mathbf{W}_{l,m,p,n}^{1,N_g,1} = \begin{bmatrix} v_{l,m} \\ \phi_n v_{l,m} \\ \phi_{p_1} v_{l,m} \\ \phi_n \phi_{p_1} v_{l,m} \end{bmatrix}, \mathbf{W}_{l,m,p,n}^{2,N_g,1} = \begin{bmatrix} v_{l,m} \\ -\phi_n v_{l,m} \\ \phi_{p_1} v_{l,m} \\ -\phi_n \phi_{p_1} v_{l,m} \end{bmatrix}$$

to define the $\nu = 4$ precoder as

$$\mathbf{W}_{l,l',m,m',p,n}^{(4)} = \begin{bmatrix} \mathbf{W}_{l,m,p,n}^{1,2,1} & \mathbf{W}_{l',m',p,n}^{1,2,1} & \mathbf{W}_{l,m,p,n}^{2,2,1} & \mathbf{W}_{l',m',p,n}^{2,2,1} \end{bmatrix}.$$

From the definitions it can be seen, that precoders for $P_{CSI-RS} = 4$, $(N_1, N_2, O_1, O_2) = (2, 1, 4, 1)$ are integrated within the multi-panel matrix both as is and by multiplying the single panel matrix by ϕ_{p_1}. Since the proposed algorithm covers the case $P_{CSI-RS} = 4, \nu = 4$, these observations argue the necessity to study extending the algorithm over Type I multi-panel case as well. The investigation undertaken in this thesis further suggests the possibility of using similar algorithmic approach to cover the remaining cases of Type I single panel CSI codebooks. However, this alongside with the extension to multi-panel and Type II codebook generation, requires further research. Another critical factor is to look into different mathematical tools that could be exploited in the generation algorithm.

In summary, this thesis proposes an algorithm and an implementation to cover the generation of NR CSI Type I single panel codebooks for number of ports $P_{CSI-RS} \in \{4, 8\}$. The proposed scheme is compared against the direct implementation, where the codebooks are stored directly and then retrieved. The mathematical structure admitted by the codebooks serves as the basis for deriving a generation algorithm, enabling a representation of a precoder $\mathbf{W}_I^{(\nu)}$ as a product $\mathbf{D}_{l,m}\mathbf{G}_{I,i_{1,3}}^{(\nu)}$, where $\mathbf{D}_{l,m}$ is a characteristic matrix and $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ is a generator matrix. The characteristic matrix is unique up to the pair $(\nu, i_{1,3})$, allowing the identification of precoders up to $(\nu, i_{1,3})$ using the characteristic matrices. The generator matrices $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ are defined over the rank ν, index tuple I and index $i_{1,3}$ while serving as a generator for multiple precoders. This allows to store the characteristic matrices and generator matrices instead of the precoders, which we have proven to save memory. With the cost of computational complexity, the proposed algorithm enables using 95.2 % less memory than in the direct implementation, as discussed in more detail in Chapter 5. Furthermore, in comparison with performing precoding directly with a precoder $\mathbf{W}_I^{(\nu)}$, the two stage matrix multiplication with $\mathbf{D}_{l,m}$ and $\mathbf{G}_{I,i_{1,3}}^{(\nu)}$ enables a reduction of 87.5 % in the number of real additions and 37.5 % in the number of real multiplications in the case of $P_{CSI-RS} \in \{4, 8\}$. While more investigation is needed in comparing the proposed scheme against other codebook generation methods to assess more carefully the benefits of the scheme, it can be nonetheless

concluded that the proposed algorithm and implementation enable significant improvements with respect to the direct storing and retrieval of precoders.

References

[1] 3GPP, NR; Physical layer procedures for data (Release 17). Technical Report TR 38.214, 2022, available at: http://www.3gpp.org

[2] 3GPP, NR; Multiplexing and channel coding (Release 17). Technical Report TR 38.212, 2022, available: http://www.3gpp.org

[3] 3GPP, NR; Physical channels and modulation (Release 17). Technical Report TR 38.211, 2022, available: http://www.3gpp.org

[4] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems", IEEE Commun. Mag., vol. 52, no. 2, pp. 186–195, 2014, DOI: https://doi.org/10.1109/MCOM.2014.6736761

[5] Z. Qin, H. Yin, "A review of Codebooks for CSI Feedback in 5G New Radio and Beyond", arXiv, version 2, 18.2.2023, DOI: https://doi.org/10.48550/arXiv.2302.09222

[6] Ahmadi, S., "5G NR - Architecture, Techonology, Implementation, and Operation of 3GPP New Radio Standards", Academic Press, 2019, ISBN: 978-0-08-102267-2

[7] M. Maleki, J. Jin, M. Haardt, "Low Complexity PMI Selection for BICM-MIMO Rate Maximization in 5G New Radio Systems", 2023 31st European Signal Processing Conference (EUSIPCO), 2023, DOI: https://doi.org/10.23919/EUSIPCO58844.2023.10290121

[8] F. Penna, H. Kwon, D. Bai, J. H. Bae, J. Kim, J. Lee, H. W. Je, "A Wideband Capacity Maximization Approach for CSI Feedback in Frequency Selective Channels", GLOBECOM 2022 - 2022 IEEE Global Communications Conference, 2022, Electronic ISBN: 978-1-6654-3540-6

[9] C. Tsai, et. al., "A Low Complexity PMI Selection Scheme for 3GPP 5G NR FDD Systems", 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2021, Electronic ISBN: 978-988-14768-9-0

[10] X. Fu, et. al., "A Tutorial on Downlink Precoder Selection Strategies for 3GPP MIMO Codebooks", IEEE Access, Vol. 11, 2023, DOI: https://doi.org/10.1109/ACCESS.2023.3338866

[11] M. Maleki, J. Jin, H. Wang, M. Haardt, "Precoding Design and PMI Selection for BICM-MIMO Systems With 5G New Radio Type-I CSI", IEEE Transactions on Communications, Vol. 72 (9), 2024, DOI: https://doi.org/10.1109/TCOMM.2024.3383106

[12] X. Fu, D. Le Ruyet, R. Visoz, T. Clessienne, "On the Performance Analysis of Different Selecting Strategies for Type-II Codebook", 2022 Joint European Conference on Networks and Communications and 6G Summit (EuCNC/6G Summit), 2022, DOI: https://doi.org/10.1109/EuCNC/6GSummit54941.2022.9815725

[13] T. Akyıldız, T. M. Duman, "Search-Free Precoder Selection for 5G New Radio using Neural Networks", 2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 2020, DOI: https://doi.org/10.1109/BlackSeaCom48709.2020.9234966

[14] B. Mondal, V. Sergeev, A. Sengupta, A. Davydov, "5G-NR (New Radio) CSI Computation Algorithm and Performance", 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 2018, DOI: https://doi.org/10.1109/ACSSC.2018.8645339

[15] F. Wang, X. Hou, X. Li, "Plane Wave Expansion-based Codebook Design for 6G Near-field MIMO", 2024 IEEE 29th Asia Pacific Conference on Communications (APCC), 2024, DOI: https://doi.org/10.1109/APCC62576.2024.10768093

[16] P. Madadi, et. al., "PolarDenseNet: A Deep Learning Model for CSI Feedback in MIMO Systems", ICC 2022 - IEEE International Conference on Communications, 2022, DOI: https://doi.org/10.1109/ICC45855.2022.9838669

[17] M. Özateş, M. Kazemi, Ç. Göken, T. M. Duman, "Channel Reconstruction Based Multiuser Precoding with Limited Feedback", 2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 2021, DOI: https://doi.org/10.1109/BlackSeaCom52164.2021.9527824

[18] D. A. U. Villalonga, H. OdetAlla, M. J. Fernández-Getino García, A. Flizikowski, "Performance bounds with precoding matrices compliant with standardized 5G-NR for MIMO transmission", 2022 IEEE Conference on Standards for Communications and Networking (CSCN), 2022, DOI: https://doi.org/10.1109/CSCN57023.2022.10050966

[19] V. Sergeev, et. al., "Enhanced Precoding Design with Adaptive Beam Width for 5G New Radio Systems", 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), 2017, DOI: https://doi.org/10.1109/VTCFall.2017.8287985

[20] S. Moon, I. Hwang, "Hybrid Precoding with Power Allocation for NR MIMO-OFDM Systems", 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), 2019, DOI: https://doi.org/10.1109/ICUFN.2019.8806038

[21] A. Farsaei, U. Gustavsson, A. Alvarado, F. M. J. Willems, "A Low-Complexity Hybrid Linear and Nonlinear Precoder for Line-Of-Sight Massive MIMO With Max-Min Power Control", IEEE Transactions on Wireless Communications, Vol. 20 (12), 2021, DOI: https://doi.org/10.1109/TWC.2021.3092965

[22] E. Onggosanusi, et. al., "Modular and High-Resolution Channel State Information and Beam Management for 5G New Radio", IEEE Commun. Mag. 56 (3) 48-55, 2018, DOI: https://doi.org/10.1109/MCOM.2018.1700761

[23] eCPRI Specification V1.2, Common Public Radio Interface: eCPRI Interface Specification, 2018, available in: http://www.cpri.info/spec.html

[24] B. Clerckx, Y. Zhou, G. C. Chung, "Codebook Generating Method and Apparatus for Generating a Codebook for Mutli-polarized Multiple-Input Multiple-output (MIMO) Systems", European Patent Office, 2019, available: https://patents.google.com/patent/US8144808

[25] A. Elshafie, et. al., "Codebook Generation for Precoding Reconfigurable Intelligent Surface (RIS) Elements", World Intellectual Property Organization, 2022, available: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2022183310

[26] M. A. Khan, et. al., "A Survey on Mobile Edge Computing for Video Streaming: Opportunities and Challenges", arXiv, 13.9.2022, DOI: https://doi.org/10.48550/arXiv.2209.05761

[27] F. Hu, et. al., "Cellular-Connected Wireless Virtual Reality: Requirements, Challenges, and Solutions", arXiv, 13.3.2020, DOI: https://doi.org/10.48550/arXiv.2001.06287