FlexRAN L1 Algorithm Description

Application Note Software v21.03

March 2021

Intel Confidential

Document Number: 610016-8.0

Contents

Figures

Tables

Revision History

Revision Number	Description	Revision Date
8.0	FlexRAN Software Release v21.03 changes: • Revised Section 2.12, Frequency Offset Estimation and Compensation with Multiple DM-RS Symbols • Revised Section 3.3 for UL prefiltering weight calculation • Revised Section 4.1, PUCCH F0 • Revised Section 4.2, De-cyclic Shift • Revised Section 4.3, Antenna Combination • Revised Section 4.4, SNR Estimation and DTX/ACK/SR Detection • Added Section 5.10, PRACH FO Check • Added Chapter 6.0, Common Algorithm	March 2021
7.0	FlexRAN Software Release v20.11 changes: • Updated Section 2.12 for carrier frequency offset estimation and compensation. • Added Section 2.13 for Doppler shift estimation • Added Section 2.14 for Time Domain Interpolation Algorithm Selection	November 2020
6.0	FlexRAN Software Release v20.08 changes: • Retitled Figures 4 and 5 • Section 2.7 added on Improved DM-RS CE • Added Section 2.8, Dual DM-RS based CE • Added Section 2.9 Time direction interpolation algorithm • Retitled Section 2.11 • Added Section 3.1, MMSE CE • Added Section 3.2. DFT Based CE • Added Section 3.4, DL Beamforming Weight Calculation	August 2020
5.0	FlexRAN software release v20.04 changes: • Revised Section 2.6, Channel Estimation on Front-Loaded DM-RS Symbol • Added Figure X, Channel estimation Chart • Added Section 2.6.4, DCT-Based Channel Estimation • Revised Section 3.3.3, Step 1, Interpolation Weight Generation, Step b, and Step 2. Interpolation in Frequency Domain. • Revised Section 4.3.2 PUCCH Carrier Frequency Offset Compensation • Revised Figures 23 and 26 • Revised Section 4.4.2.4, SNR Estimation and DTX/ACK/SR Detection	April 2020
4.0	Updates for release v20.2: • Updated Section 2.1 PUSCH Receiver Processing Flow • Added Section 2.12 Frequency Offset Estimation and Compensation • Added Section 2.15 Transform De-precoding	February 2020

	• Updated Section 2.16 Softbits De-mapping	
3.0	Updates for release v19.10: • Add algorithms for SRS channel estimation (Section 3.1.2) • Modify the algorithm for PUSCH Softbits de-mapping (Section 2.16) • Optimize algorithms for PUCCH format 0,1,2,3,4 (Chapter 4.0)	September 2019
2.0	Section 4.2: Updated And the context for frequency band shift and downsampling.	June 2019
1.0	Initial release for FlexRAN Software Release v19.03	April 2019

§

1.0 Introduction

This document describes the 5GNR Physical Uplink Shared Channel (PUSCH), Sounding Reference Channel (SRS). Physical Uplink Control Channel (PUCCH) and Physical Random Access Channel (PRACH) uplink baseband algorithms used in FlexRAN projects.

1.1 Scope

This document is a description of 5GNR uplink baseband algorithms. The algorithms described are implemented in the FlexRAN project. Layer 1 (L1) algorithms of PUSCH, PUCCH, and PRACH are covered in the document. The detailed feature list of currently supported configurations is described in Table 3.

The current algorithm design is based on Third Generation Partnership Project New Radio (3GPP NR) v15.5.0. Minor updates may occur in the future.

The algorithm adopted for FlexRAN is aimed to be performance calibrated with 3GPP RAN4 minimum requirements. Further performance optimizations may occur in the future.

1.2 Intended Audience

This document serves as a reference for L1 baseband processing. Its intended audience includes designers and engineers working with FlexRAN releases.

1.3 Terminology

Table 1. Terminology

Term	Description
3GPP	Third Generation Partnership Project
CP	Cyclic Prefix
CSI	Channel State Indicator
DM-RS	Demodulation Reference Signal
I/Q	In-Phase/Quadrature
IFFT	Inverse Fast Fourier Transform
L1	Layer 1
LS	Least Square
MIMO	Multiple-Input and Multiple-Output
NR	New Radio
PDP	Power Delay Profile
PRACH	Physical Random Access Channel
PT-RS	Phase Tracking Reference Signal

Term	Description
PUCCH	Physical Uplink Control Channel
PUSCH	Physical Uplink Shared Channel
RB	Resource Block
TA	Timing Advanced
TDD	Time Division Duplexing
ZC	Zadoff-Chu (root sequence)

1.4 Reference Documents and Resources

Table 2. Reference Documents and Resources

Document or Reference	Document No./Location
3GPP TS38.104, NR; Base station radio transmission and reception	https://www.3gpp.org/ftp/Specs/archive/38_series/38.104/
3GPP TS38.211, NR; Physical channels and modulation	http://www.3gpp.org/ftp//Specs/archive/38_series/38.211/
3GPP TS38.212, NR; Multiplexing and channel coding	http://www.3gpp.org/ftp//Specs/archive/38_series/38.212
3GPP TS38.213, NR; Physical layer procedures for control	http://www.3gpp.org/ftp//Specs/archive/38_series/38.213
3GPP TS38.214, NR; Physical layer procedures for data	http://www.3gpp.org/ftp//Specs/archive/38_series/38.214

Note: The specifications used in this document are 3GPP NR v15.5.0

1.5 Configuration and Scope

Table 3. Configuration and Scope

Channel and Feature	3GPP (v15.5.0)	Simulator Status
PUSCH		
UL waveform type	Cyclic Prefix (CP)-OFDM, DFT-S-OFDM	CP-OFDM, DFT-S-OFDM
Subcarrier spacing	15, 30, 60, 120, 240, 480 kHz	Configurable (generally 30KHz for Sub6 and 120 KHz for mmWave)
PUSCH modulation without transform precoding	QPSK, 16QAM, 64QAM, 256QAM	QPSK, 16QAM, 64QAM, 256QAM
DFT-S-OFDM based PUSCH	pi/2-BPSK, QPSK, 16QAM, 64QAM	pi/2-BPSK, QPSK, 16QAM, 64QAM
Cyclic prefix length	Normal CP, Extended CP	Normal CP
Max # layers per CC, per UE	4	4
Multiplexing on DM-RS symbol	Multiplexing, No multiplexing	No multiplexing

Channel and Feature	3GPP (v15.5.0)	Simulator Status
Multiplexing on PUCCH symbol	Multiplexing, No multiplexing	No multiplexing
Frequency hopping	No hopping, Mode 1, Mode 2	No hopping
DM-RS		
# ports	Up to 12 ports defined	Up to 4/6 ports defined
DM-RS config (CP-OFDM)	Type 1: Single-symbol, up to 4 ports Double-symbol, up to 8 ports Type 2: Single-symbol, up to 6 ports Double-symbol, up to 12 ports	Single-symbol Type 1, up to 4 ports Single-symbol Type 2, up to 4 ports
DM-RS symbols combination	1, 1+1, 1+1+1, 1+1+1+1, 2, 2+2	1, 1+1, 1+1+1, 1+1+1+1
PT-RS		
PT-RS configuration	Configured, Not configured	Configurable and enabled in mmWave
# ports	1, 2	1
PT-RS granularity	freq domain: every 2, 4 PRB time domain: every 1, 2, 4 symbols	freq domain: every 2, 4 PRB time domain: every 1, 2, 4 symbols
Channel coding		
Coding scheme	LDPC	LDPC
RV	0,1,2,3	Configurable
Transport block CRC size	24 (TBS>3824), 16 (TBS<=3824)	Same as 3GPP (vv15.5.0) see Table 2
Code block CRC size	24 bits	24 bits
PUSCH rate matching	Full buffer RM, Limited buffer RM	Full buffer RM
PUCCH		
Subcarrier spacing	15, 30, 60, 120, 240, 480 kHz	Configurable
UCI coding scheme	Repetition, Simplex, Reed-Müller, Polar	Repetition, Simplex, Reed-Müller, Polar
DFT-S-OFDM	Format 3, 4	Support
Short PUCCH		
PUCCH duration	1..2 symbols	1..2 symbols
PUCCH formats	Format 0, 2	Format 0, 2
Resource size	F0: 1 PRB F2: 1..16 PRB	F0: 1 PRB F2: 1..16 PRB
Frequency hopping	2 symbol case	Support intra-slot frequency hopping

Channel and Feature	3GPP (v15.5.0)	Simulator Status
Long PUCCH		
PUCCH duration	4...14 symbols	4...14 symbols
PUCCH formats	Format 1, 3, 4	Format 1, 3, 4
Resource size	F1; F4: 1 PRB; F3: 1..6, 8..10, 12 PRB	Same as 3GPP (vv15.5.0) see Table 2
intra-slot Frequency hopping	No, Yes	yes
Inter-slot frequency hopping	No, Yes	no
Index of time-domain OCC	F1: 0..6	Same as 3GPP (vv15.5.0) see Table 2
Index of pre-DFT OCC	F4: 0..1 (length 2), 0..3 (length 4)	Same as 3GPP (vv15.5.0) see Table 2
PRACH		
Preamble format	0-3, A0-A3, B1-B4, C0, C2	0, A0-A3, B1-B4, C0, C2
Preamble sequence length	139, 839	139, 839
Subcarrier spacing	1.25, 5, 15, 30, 60, 120 kHz	1.25, 15, 30, 60, 120 KHz
ZeroCorrelationZoneConfig	0..15	0..15
PRACH configuration index	0..255	Same as 3GPP (vv15.5.0) see Table 2
RACH configuration period	10, 20, 40, 80, 160 ms	Configurable
RACH start symbol	Short formats: Symbol 0, 2 Long format 2: Symbol 0, 7 Long format 0,1,3: Symbol 0	Short formats: Symbol 0, 2 Long format 0: Symbol 0
Restricted set configuration	Long formats: Unrestricted, Restricted Type A, Restricted Type B Short formats: Unrestricted	Unrestricted for long and short formats
RACH procedure	Contention-based, Contention-free	Contention-based

§

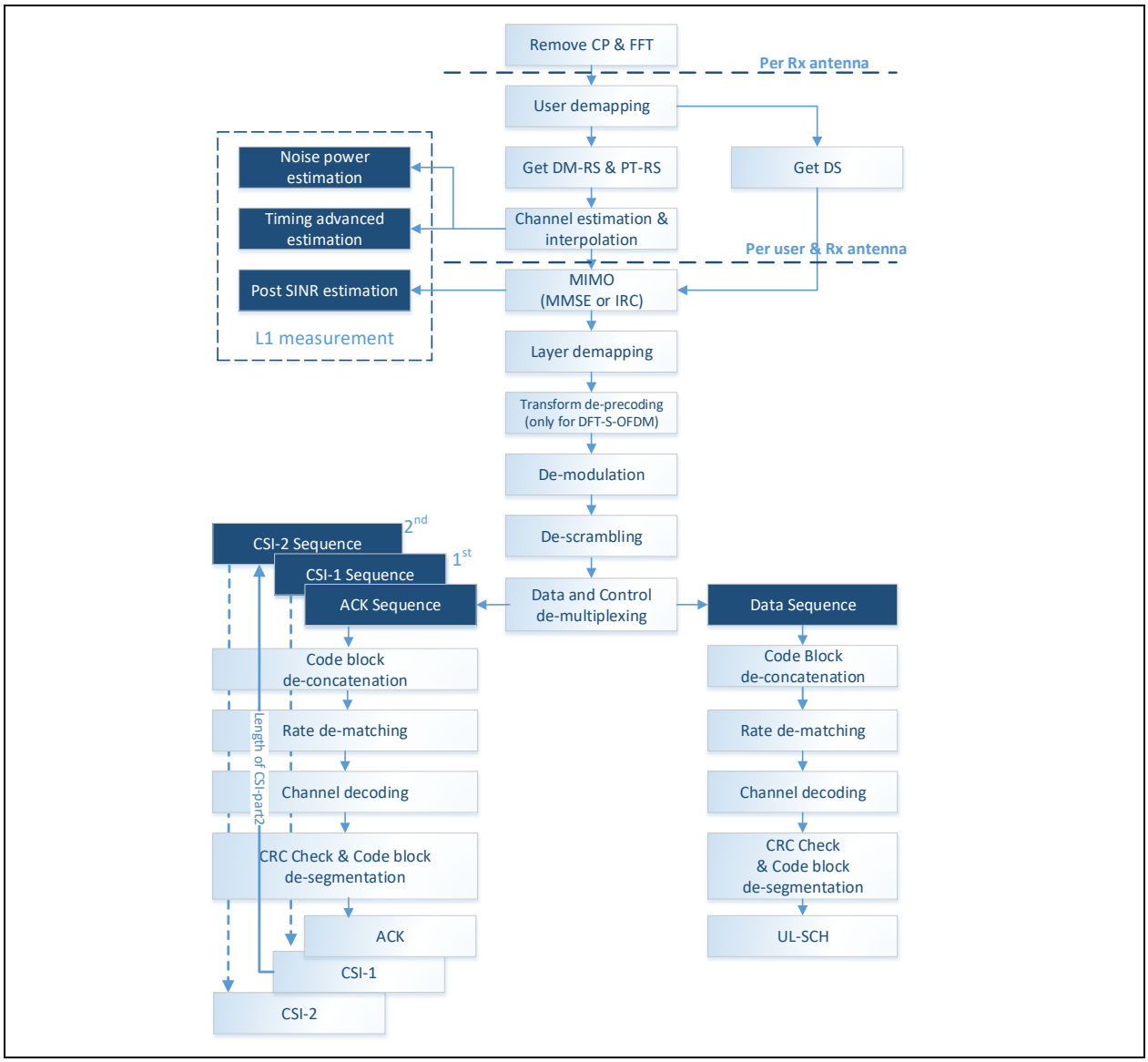
2.0 PUSCH Algorithms

This section introduces the Physical Uplink Shared Channel (PUSCH) baseband algorithms for the FlexRAN project, including the overall processing steps, the introduction of critical modules, input, output, detailed algorithms, and so forth.

2.1 Overview

The PUSCH receiver processing algorithm described below is implemented by software and hardware. It includes the physical layer (L1) signal processing chain from antenna In-Phase/ Quadrature (I/Q) data to decoded bits.

Figure 1. PUSCH Receiver Processing Flow



2.2 Main Module Overview

This section introduces the main modules in order of execution:

1. **Front end processing**: the input is radio samples in the time domain, CP is removed, and samples are transformed into frequency domain data after Fast Fourier Transform (FFT).
2. **User/symbol de-mapping**: data and reference signals of Orthogonal Frequency-Division Multiplexing (OFDM) symbols corresponding to each user are extracted within a slot.
3. **Channel estimation and interpolation**: the channel is estimated based on DM-RS, and then time-domain interpolation is performed to get the estimated channel for all PUSCH data symbols.
4. **Multiple Input and Multiple Output (MIMO)/Equalizer**: multiple-input and multiple-output antenna combined with Minimum Mean Squared Error (MMSE) estimation or MMSE-IRC.
5. **Softbit de-mapping**: Quadrature Amplitude Modulation (QAM) constellation values are converted to soft values, which represent the logarithmic probability of 1 or 0 for each bit.
6. **Reception of UCI on PUSCH**: de-multiplex the UCI (ACK, CSI-part1, and CSI-part2) from the de-scrambled PUSCH and perform rate de-matching, channel decoding, and CRC check in the UCI reception process. Finally, get with ACK, CSI-part1, and CSI-part2 messages and the CRC check results.
7. L1 measurement:
   a. **Noise Power**: averaged noise power among Rx antennas.
   b. **TA**: it measures the time deviation of symbols from different User Equipment (UEs) compared with the preferred arrival, and the measurement is based on the estimated channel of a reference signal (RS).
   c. **Frequency offset**: it measures the frequency offset for each user and is based on different symbols of the reference signal.
   d. **Post Noise Ratio (SINR)**: the estimated SINR after the equalizer.

2.3 CP Removal

This section describes the algorithm for cyclic prefix (CP) removal.

2.3.1 Input

$y(n,r)$ Time-domain samples in Rx antenna r

2.3.2 Output

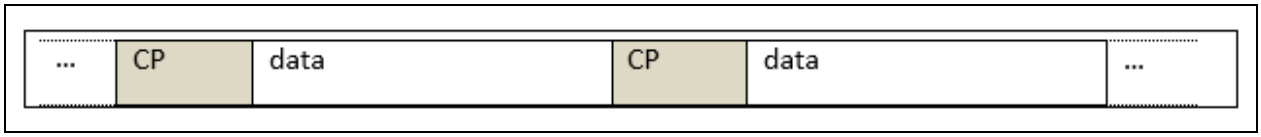
$y_{FFT}(n,r)$ CP removal output in Rx antenna r

2.3.3 Algorithm

Multiple numerologies are supported for the 5G NR subcarrier spacing configuration μ, and the cyclic prefix for a specific bandwidth part is obtained from higher-layer parameters `subcarrierSpacing` and `cyclicPrefix`. Slots are numbered $n_s^{\mu} \in \{0,..., N_{slot}^{subframe,\mu} - 1\}$ in increasing order within a subframe and $n_{s,f}^{\mu} \in \{0,..., N_{slot}^{frame,\mu} - 1\}$ increasing order within a frame. There are N_{symb}^{slot} consecutive OFDM symbols in a slot where N_{symb}^{slot} depend on

the cyclic prefix as given by Tables 4.3.2-1 and 4.3.2-2 in 3GPP TS38.211 (refer to Table 2). The start of the slot n_s^{μ} in a subframe is aligned in time with the start of the OFDM symbol $n_s^{\mu} N_{symb}^{slot}$ in the same subframe. The CP and data starting position of the OFDM symbol can be found in clause 5.3.1 in 3GPP TS38.211 (refer to Table 2).

Figure 2. CP and Data in Time Domain within an OFDM Symbol



2.4 FFT

This section describes the algorithm for Fast Fourier Transform (FFT) operations.

2.4.1 Input

$y_{FFT}(n,r)$ CP removal output, which is also the FFT input.

2.4.2 Output

$Y_{FFT}(k,r)$ FFT output in subcarrier k and Rx antenna r

2.4.3 Algorithm

The time-domain samples are transformed into the frequency domain through FFT.

2.5 De-mapping

This section describes the algorithm for removing mapping information from the data.

2.5.1 User Extraction

This section describes how to remove the mapping from the frequency domain data for each user.

2.5.1.1 Input

$Y_{FFT}(k,r)$ FFT output

$N_{RB,u}$) Number of RBs for user u

N_{FFT} FFT size

$n_{StartRB,u}$ Start position for user u, in terms of RB index

2.5.1.2 Output

$Y_u(k,r)$ Frequency domain data for the user u in subcarrier k and Rx antenna *r*

2.5.1.3 Algorithm

In this module, the frequency domain data is de-mapped for each user.

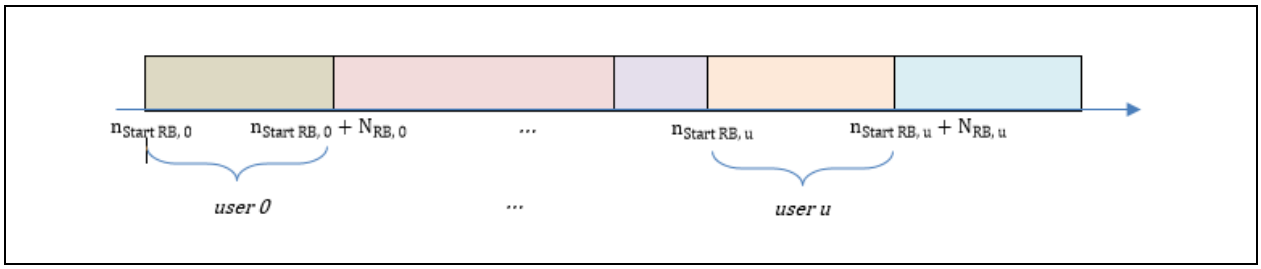
First, extract the used $N_{sc} = N_{RB}^{UL} N_{sc}^{RB}$ subcarriers from N_{FFT} output;

Then, separate users according to the scheduling information.

$Y_u(k,r) = Y_{FFT}(k,r)$,

$k = n_{StartRB,u} \cdot N_{sc}^{RB}, \ldots, (n_{StartRB,u} + N_{RB,u}) \cdot N_{sc}^{RB} - 1, n_{StartRB,u} = 0, \ldots, N_{RB}^{UL}$

Figure 3. User De-mapping



2.5.2 Symbol Extraction

This section describes how to remove DM-RS symbols from frequency domain data.

2.5.2.1 Input

$Y_u(k,l,r)$	Frequency domain data in subcarrier k, symbol l, and Rx antenna r for the user u

2.5.2.2 Output

$Y_{RS,u}(k,l,r)$	Frequency domain DM-RS symbol(s) for the user u
$Y_{DS,u}(k,l,r)$	Frequency domain data symbol(s) for the user u

2.5.2.3 Algorithm

There are one or multiple DM-RS symbols in one slot, namely the first loaded DM-RS symbol and additional DM-RS symbols (configurable).

The reference point for l and the position l_0 of the first DM-RS symbol depends on the mapping type provided in 3GPP TS38.211 (refer to Table 2) :

- For PUSCH mapping type A:
- l is defined relative to the start of the slot if frequency hopping is disabled and relative to the start of each hop in case frequency hopping is enabled
- l_0 is given by the higher-layer parameter dmrs-TypeA-Position.

For PUSCH mapping type B:

- l is defined relative to the start of the scheduled PUSCH resources if frequency hopping is disabled and relative to the start of each hop in case frequency hopping is enabled

- l_0 = 0
- The position(s) of all the DM-RS symbols are given by $\bar{l}$ and:
- The signalled duration between the first OFDM symbol of the slot and the last OFDM symbol of the scheduled PUSCH resources in the slot for PUSCH mapping type A according to Tables 6.4.1.1.3-3 and 6.4.1.1.3-4 in 3GPP TS38.211 (refer to Table 2) if frequency hopping is not used, or:
- The signalled duration of scheduled PUSCH resources for PUSCH mapping type B according to Tables 6.4.1.1.3-3 and 6.4.1.1.3-4 in 3GPP TS38.211 (refer to Table 2) if frequency hopping is not used.

2.6 Channel Estimation on Front-Loaded DM-RS Symbol

There are two-channel estimation methods that can be applied in the single front-loaded DM-RS symbol, frequency-domain channel estimation, and transformed domain channel estimation. Interpolation based on Phase Tracking Reference Signal (PT-RS) and/or additional DM-RS symbol(s) among different data symbols are supported and will be described in later sections of this document. Figure 4 illustrates the procedure of the channel estimation algorithm in the frequency domain first, and then in the time domain. Figure 5 illustrates the procedure of the channel estimation algorithm in the transformed domain first, and then in the time domain.

Figure 4. Channel Estimation Flow Chart with MMSE Based Algorithm

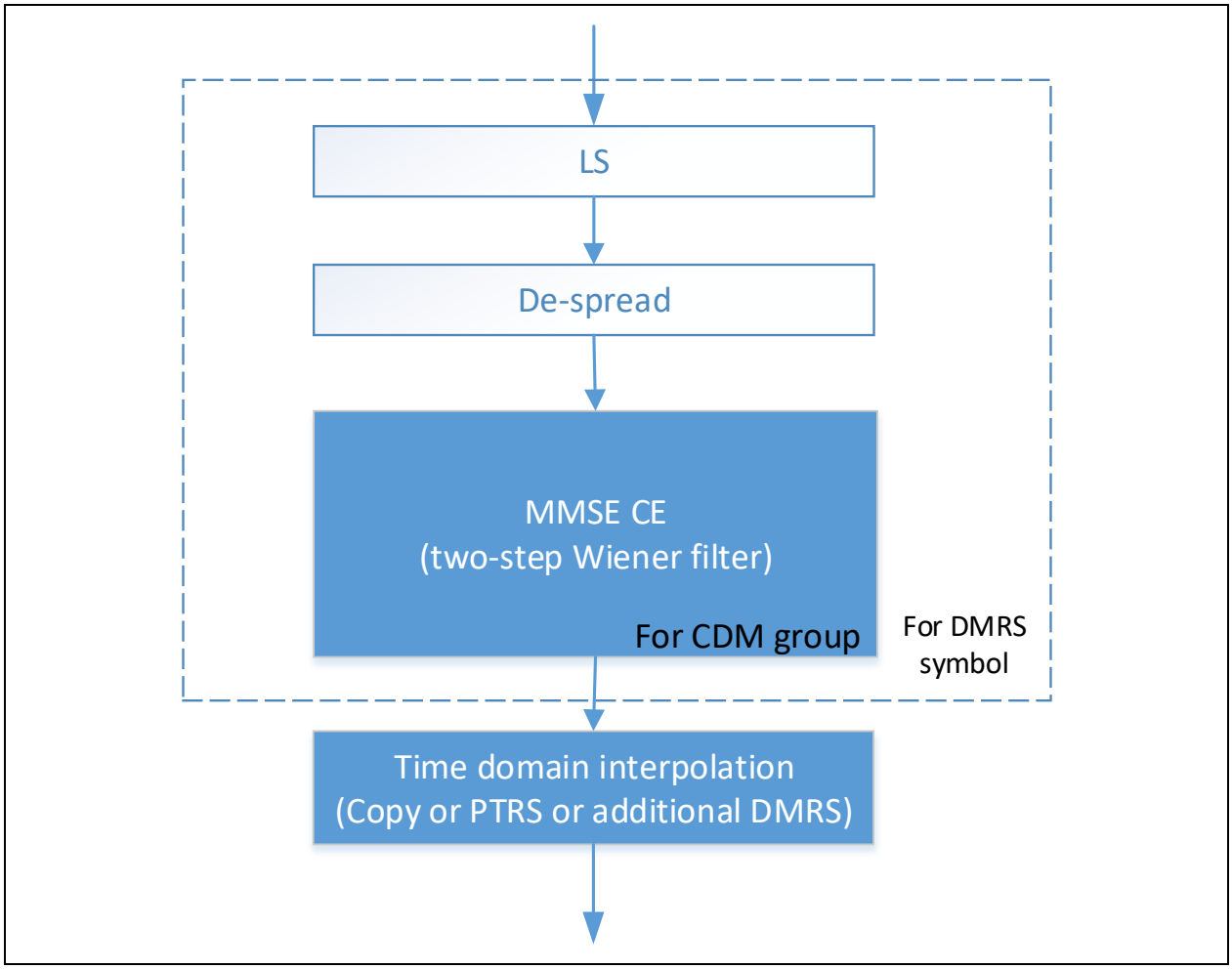
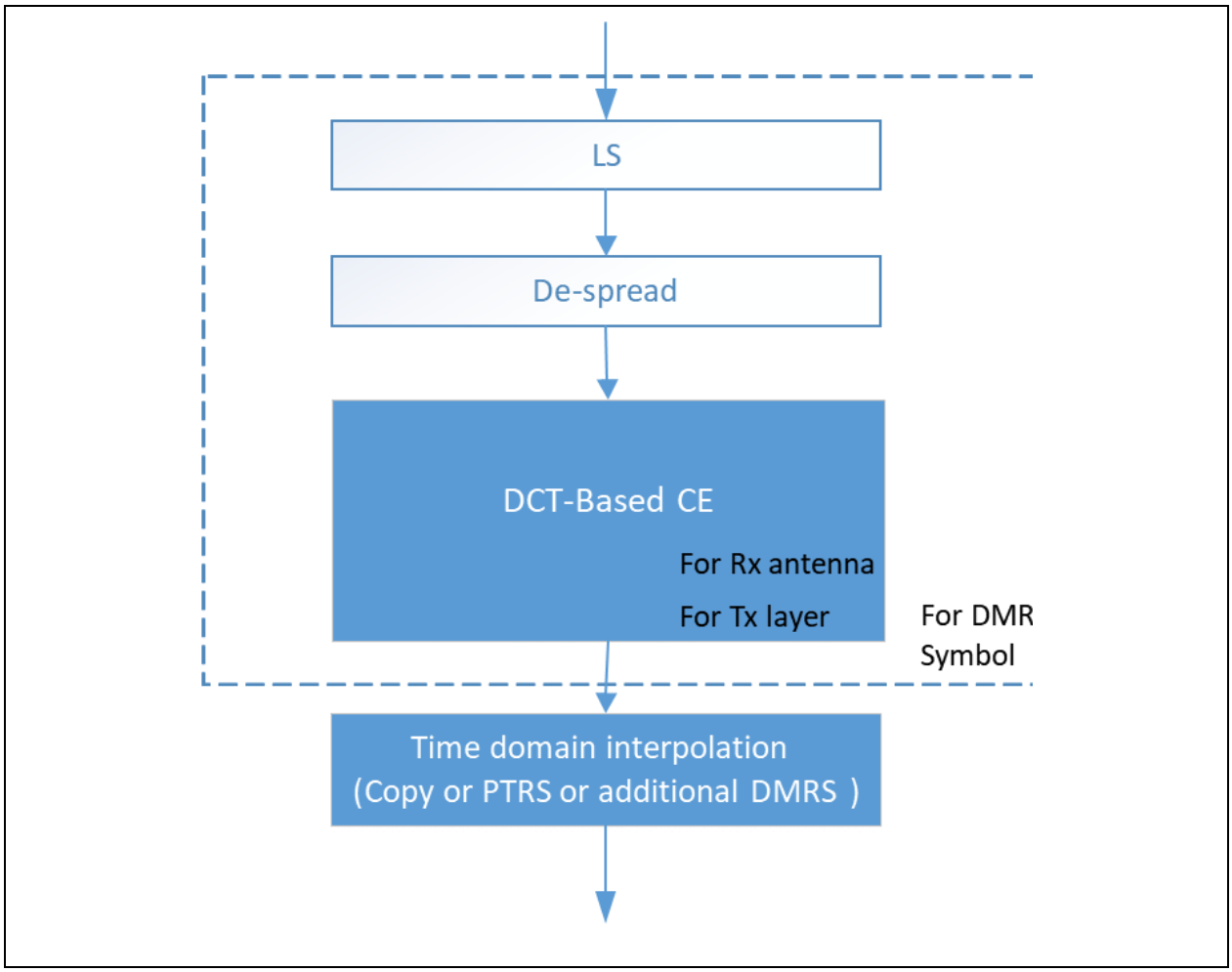


Figure 5. Channel Estimation Flow Chart with DCT Based Algorithm



2.6.1 DM-RS Overview

2.6.1.1 DM-RS Type 1

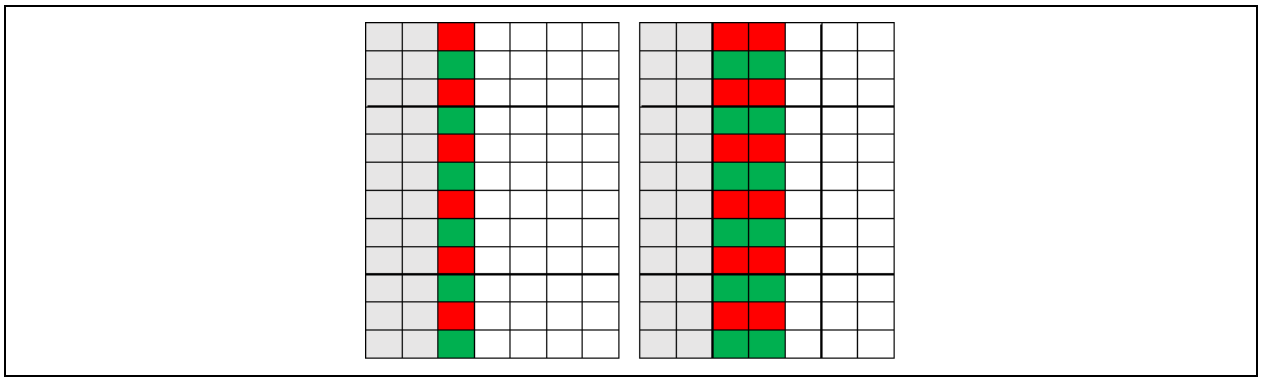
In this pattern, one DM-RS symbol supports up to four antenna ports, and two symbols support up to eight antenna ports.

Possible DM-RS starting positions:

- Third or fourth OFDM symbol in a slot (PDSCH/PUSCH Mapping Type A) or
- The first symbol in the scheduled data symbol durations (PDSCH/PUSCH for Mapping Type B).

The pattern is illustrated in Figure 6.

Figure 6. DM-RS Type 1



2.6.1.2 DM-RS Type 2

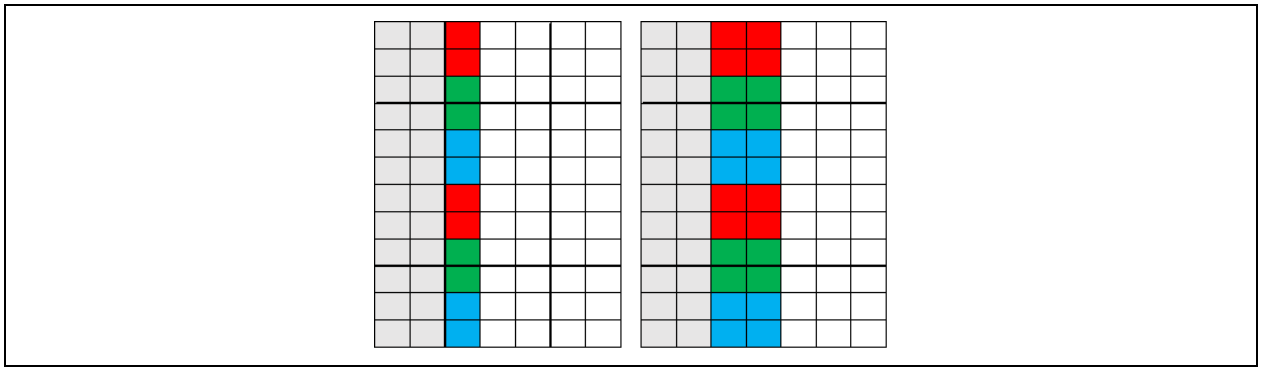
In this pattern, one DM-RS symbol supports up to six antenna ports, while two symbols support up to 12 antenna ports.

Possible DM-RS starting positions:

- Third and fourth OFDM symbol in a slot (PDSCH/PUSCH Mapping Type A) or
- The first symbol in the scheduled data symbol durations (PDSCH/PUSCH Mapping Type B).

The pattern is illustrated in Figure 7.

Figure 7. DM-RS Type 2



2.6.2 LS and De-spread Processing

2.6.2.1 Input

$Y_{RS}(k,l,r)$	Received DM-RS signal on subcarrier k, symbol l and Rx antenna r
$X_{DMRS}(k,l)$	Transmitted DM-RS base sequence on subcarrier k and symbol l
t	DM-RS transmission port index

2.6.2.2 Output

$\hat{H}_{RS}(k,l,r,t)$ Estimated channel response

2.6.2.3 Algorithm

DM-RS Type 1

Assuming the 2Tx case using DM-RS Tx port [0 1] according to the CDM group defined in Table 6.4.1.1.3-1 in 3GPP TS38.211 (refer to Table 2):

$$Y_{RS}(k_0,l,r) = \hat{H}_{RS}(k_0,l,r,t_0)\cdot X_{DMRS}(k_0,l) + \hat{H}_{RS}(k_0,l,r,t_1)\cdot X_{DMRS}(k_0,l) + n$$

$$Y_{RS}(k_2,l,r) = \hat{H}_{RS}(k_2,l,r,t_0)\cdot X_{DMRS}(k_2,l) - \hat{H}_{RS}(k_2,l,r,t_1)\cdot X_{DMRS}(k_2,l) + n$$

$$\hat{H}_{RS}(k_0,l,r,t_0) = \hat{H}_{RS}(k_2,l,r,t_0))$$

$$\hat{H}_{RS}(k_0,l,r,t_1) = \hat{H}_{RS}(k_2,l,r,t_1)$$

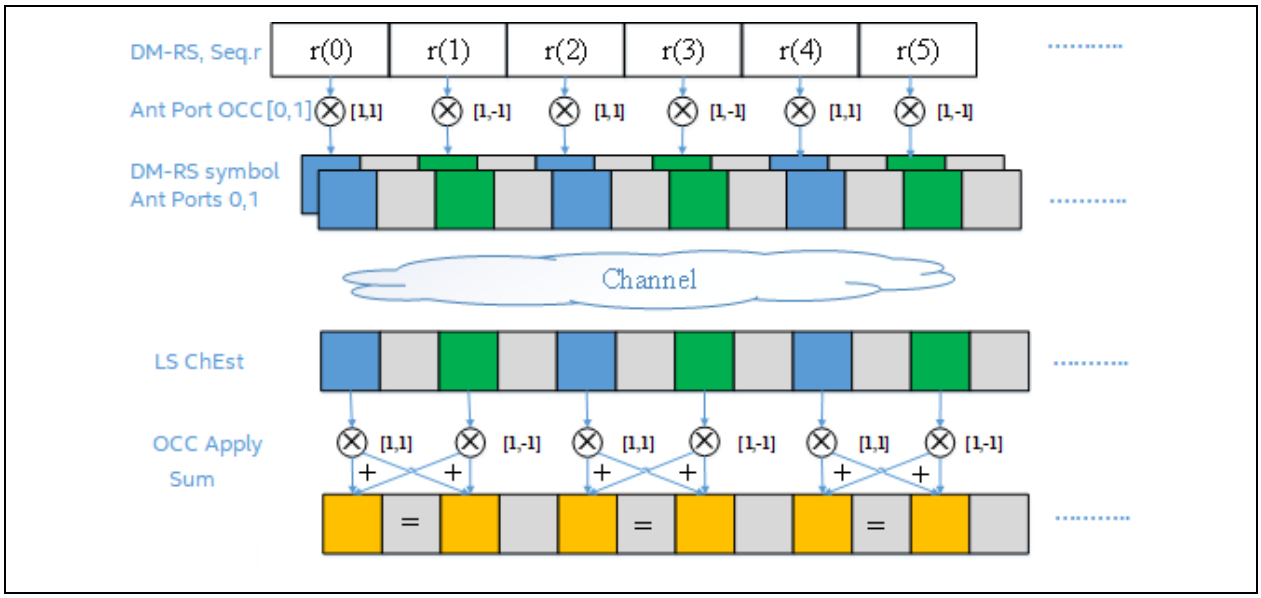
Where k0 and k2 indicate subcarrier 0 and 2, which are consecutive DM-RS subcarriers for DM-RS type 1.

Ignoring *n*, the results are:

$$\hat{H}_{RS}(k_0,l,r,t_0) = \hat{H}_{RS}(k_2,l,r,t_0) = (Y_{RS}(k_0,l,r)/X_{DMRS}(k_0,l) + Y_{RS}(k_2,l,r)/X_{DMRS}(k_2,l))/2$$

$$\hat{H}_{RS}(k_0,l,r,t_1) = \hat{H}_{RS}(k_2,l,r,t_1) = (Y_{RS}(k_0,l,r)/X_{DMRS}(k_0,l) - Y_{RS}(k_2,l,r)/X_{DMRS}(k_2,l))/2$$

Figure 8. LS and De-spread for DM-RS Type 1



DM-RS Type 2

Assuming the 2Tx case using DM-RS Tx port [0 1] according to the CDM group defined in Table 6.4.1.1.3-2 in 3GPP TS38.211 (refer to Table 2):

$$Y_{RS}(k_0,l,r) = \hat{H}_{RS}(k_0,l,r,t_0)\cdot X_{DMRS}(k_0,l) + \hat{H}_{RS}(k_0,l,r,t_1)\cdot X_{DMRS}(k_0,l) + n$$

$$Y_{RS}(k_1,l,r) = \hat{H}_{RS}(k_1,l,r,t_0)\cdot X_{DMRS}(k_1,l) - \hat{H}_{RS}(k_1,l,r,t_1)\cdot X_{DMRS}(k_1,l) + n$$

$$\hat{H}_{RS}(k_0,l,r,t_0) = \hat{H}_{RS}(k_2,l,r,t_0)$$

$$\hat{H}_{RS}(k_0,l,r,t_1) = \hat{H}_{RS}(k_2,l,r,t_1)$$

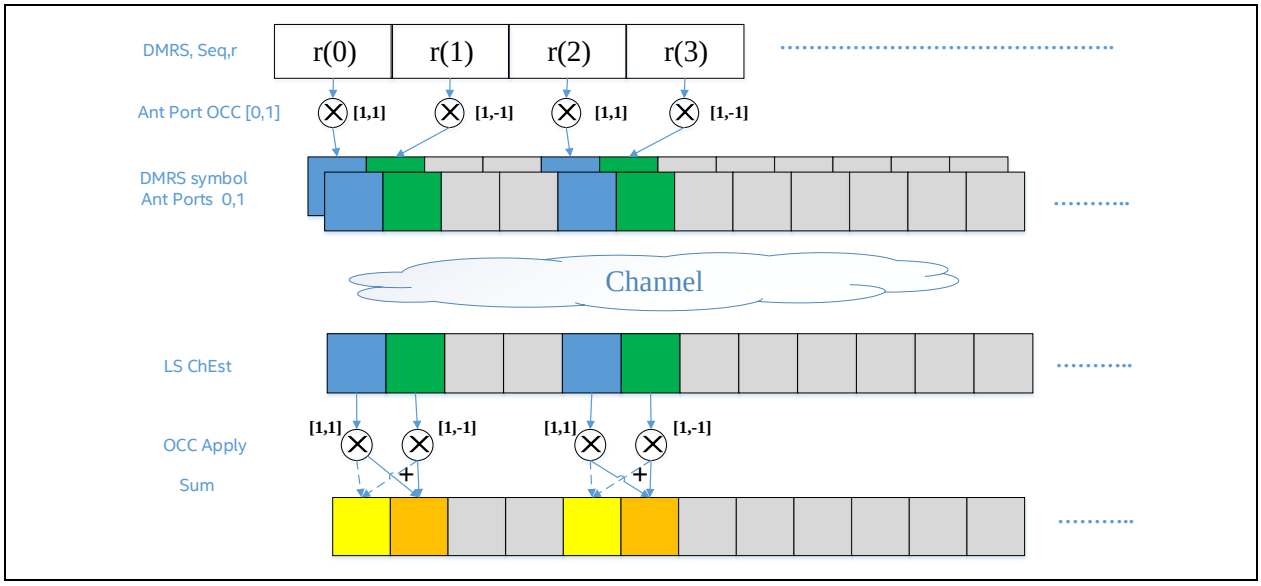
Where k0 and k1 indicate subcarriers 0 and 1, which are consecutive DM-RS subcarriers for DM-RS Type 2.

Ignoring n, the results are:

$$\hat{H}_{RS}(k_0,l,r,t_0) = \hat{H}_{RS}(k_1,l,r,t_0) = (Y_{RS}(k_0,l,r)/X_{DMRS}(k_0,l) + Y_{RS}(k_1,l,r)/X_{DMRS}(k_1,l))/2$$

$$\hat{H}_{RS}(k_0,l,r,t_1) = \hat{H}_{RS}(k_1,l,r,t_1) = (Y_{RS}(k_0,l,r)/X_{DMRS}(k_0,l) - Y_{RS}(k_1,l,r)/X_{DMRS}(k_1,l))/2$$

Figure 9. LS and De-spread Processing for DM-RS Type 2



2.6.3 MMSE-Based Channel Estimation

2.6.3.1 Input

SNR_0	Default initial SNR, for example, 30 dB
$Y_{RS}(k,l,r)$	Received DM-RS signal
$X_{DMRS-OCC}(k,l,t)$	Transmitted DM-RS signal with OCC
$\hat{H}_{RS}(k,l,r,t)$	Estimated channel response on DM-RS symbol l

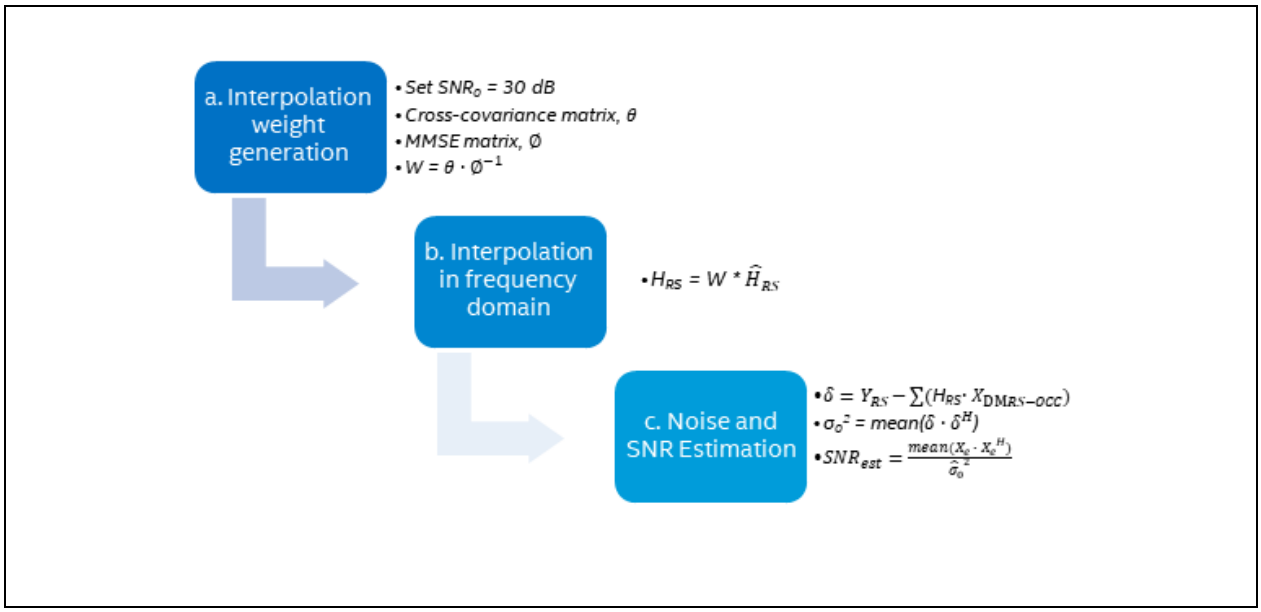
2.6.3.2 Output

σ_u^2	The estimated noise power of user u
W	Frequency domain interpolation weight
$H_{RS}(k,l,r,t)$	Channel estimation of the first loaded DM-RS symbol

2.6.3.3 Algorithms

The estimation of SNR and noise is based on MMSE and processed in the RB group unit, default as 4RBs. The process is illustrated in Figure 10.

Figure 10. SNR Estimation



$$w(k,l;k',l') = \theta(k,l) \cdot \Phi^{-1}(k',l')$$

Where $\theta(k,l)$ is the cross-covariance matrix between the data channel and the reference signal channel of the received signal. It can be calculated by the following two methods:

- Bessel function:

$$\theta(k,l) = \frac{J_0(2\pi f_{D_{max}} T_s(l-l'))}{1 + i \cdot 2\pi\tau_{max}\Delta F(k-k')}$$

- Sinc function (default adopted):

$$\theta(k,l) = sinc(2f_{D_{max}} T_s(l-l')) \cdot sinc(2\tau_{max}\Delta F(k-k'))$$

  In case of 1-D frequency domain filtering, $sinc(2f_{D_{max}} T_s(l-l'))$ is 1.

  Where

  τ_{max} is the delay spread of the wireless channel, unit in seconds.

  ΔF is subcarrier spacing, unit in Hz.

  $k-k'$ is the subcarrier index offset between data and DM-RS.

  $\Phi(k',l')$ is the MMSE matrix. It includes auto-covariance matrix and noise

$$\Phi(k',l') = \frac{1}{SNR_0} \cdot I + \theta(k',l')$$

$$H_{RS}(k,l) = w^T(k,l;k',l') \cdot \hat{H}_{RS}(k',l')$$

  $\delta(k,l,r) = Y_{RS}(k,l,r) - \sum_t X_{DMRS-OCC}(k,l,t) \times H_{RS}(k,l,t,r)$,

  where

  $Y_{RS}(k,l,r)$ is the received reference signals on receive antenna *r*,

  H_{RS} is the estimated channel of the reference signals shown in step b of Figure 10.

Then the noise power is derived by:

$$\hat{\sigma}_0{}^2 = mean(\delta \cdot \delta^H)$$

where mean ( ) is overall RX antennas and all subcarriers which reference signals use. And then the estimated signal power is derived by:

$X_e = \sum(\mathrm{H_{RS}} \cdot X_{DMRS-OCC})$

$$SNR_{est} = \frac{mean(X_e \cdot X_e{}^H)}{\hat{\sigma}_0{}^2}$$

The MMSE CE algorithm is similar to the procedure of SNR estimation, except for the input SNR value.

2.6.4 DCT-Based Channel Estimation

This section describes the algorithm for DCT based channel estimation.

2.6.4.1 Input

$\widehat{H}_{RS}(k,l,r,t)$ Estimated channel response on DM-RS symbol l

2.6.4.2 Output

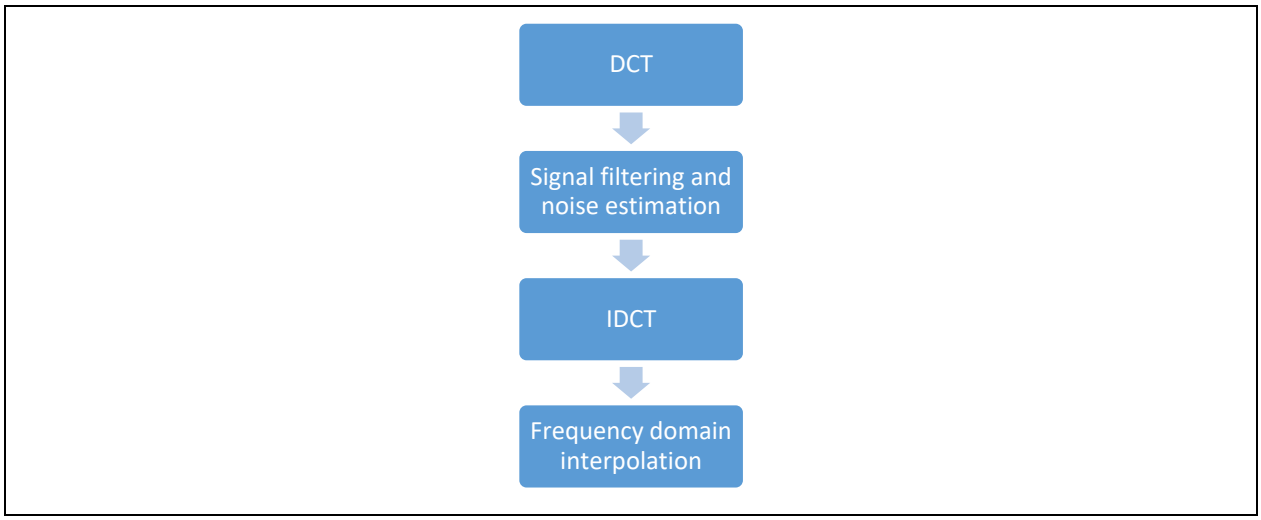
σ^2 The estimated noise power

$H_{RS}(k,l,r,t)$ Channel estimation of the first loaded DM-RS symbol

2.6.4.3 Algorithms

The process of DCT based channel estimation algorithm is illustrated in Figure 11.

Figure 11. DCT-Based Channel Estimation



The first step of the DCT channel is converting LS estimate into the DCT domain by the DCT algorithm. The DCT domain channel response of the LS estimated channel can be given by

$$h_{LS}^{DCT}(n,l,r,t) = DCT(\widehat{H}_{RS}(k,l,r,t))$$

Then, filtering $h_{LS}^{DCT}(n,l,r,t)$ with rectangle window, we can get the filtered DCT domain channel response as

$$h_{filter}^{DCT}(n,l,r,t) = \begin{cases} h_{LS}^{DCT}(n,l,r,t), & 0 \le n \le w_{size} - 1 \\ 0, & otherwise \end{cases}$$

Here w_{size} is the rectangle window size, which scales the maximum possible path delay according to the quotient of the number of scheduled subcarriers and original FFT size.

$$w_{size} = \frac{N_{CP} * N_{DMRS}}{N_{FFT}}$$

Currently, in the fixed window above, we choose to further set the taps to zeros whose energy is smaller than a threshold, where the threshold is decided by the average energy of the noise outside of the window P_{out}.

$$threshold = F_{Scale} * P_{out}$$

$$\hat{h}_{filter}^{DCT}(n,l,r,t) = \begin{cases} h_{filter}^{DCT}(n,l,r,t), & P_n \ge threshold \\ 0, & otherwise \end{cases}, 0 \le n \le w_{size} - 1$$

Meanwhile, the estimated noise power σ^2 can be gotten with the average energy of all taps, which are set to zero.

Thirdly, converting the DCT domain estimate into the frequency domain by the IDCT algorithm.

$$H(k',l,r,t) = IDCT(\hat{h}_{filter}^{DCT}(n,l,r,t)), 0 \le k' < N_{DMRS} - 1$$

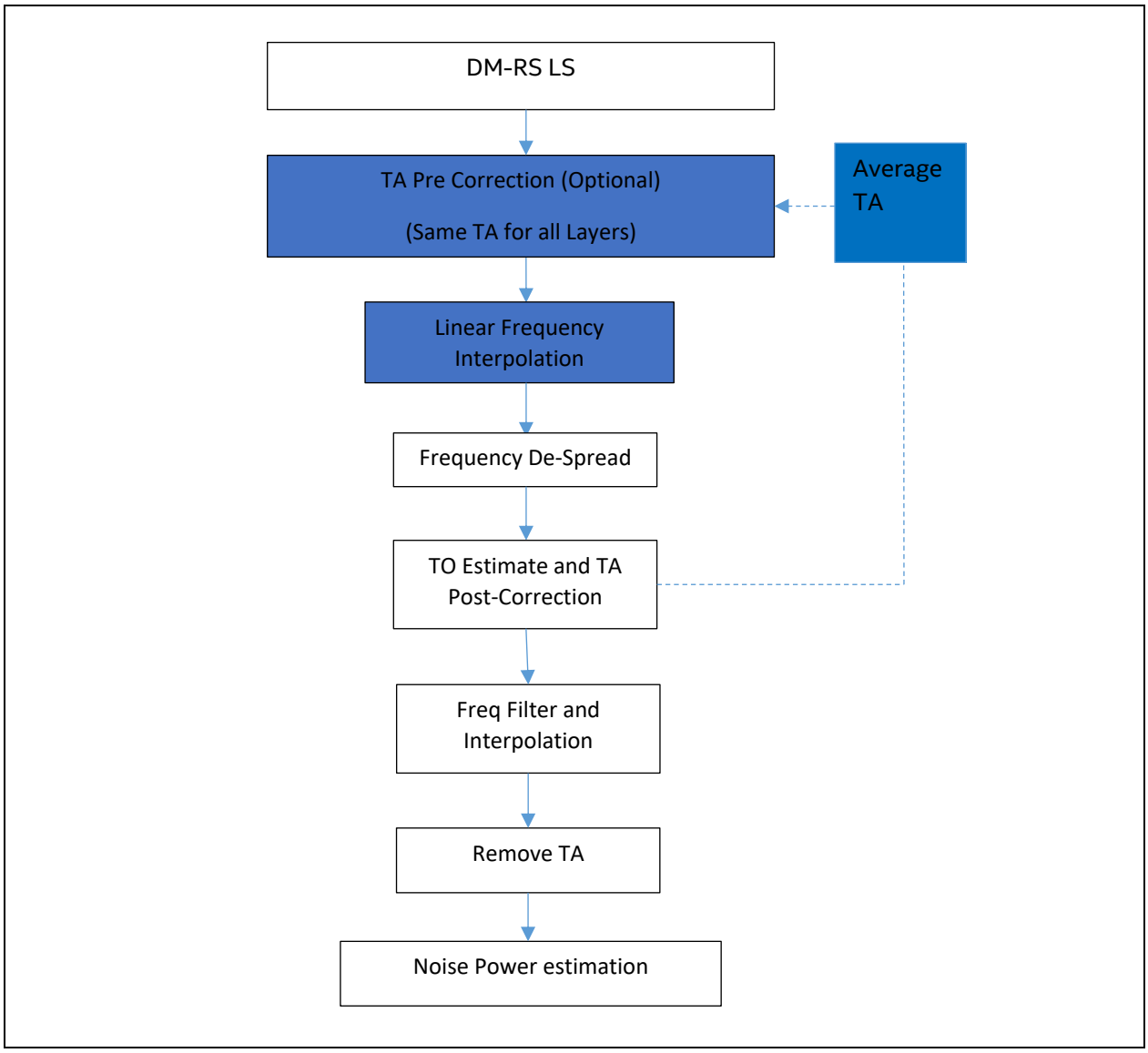
The last step is a frequency-domain interpolation, which will expand the channel estimate from DM-RS subcarriers to scheduled subcarriers. And the final frequency domain channel response can be written as:

$$H_{RS}(k,l,r,t) = linear\_interpolation(H(k',l,r,t)), 0 \le k < N_{used\_sc} - 1$$

2.7 Frequency Interpolation to Improve Accuracy of Channel Estimation on Front-Loaded DM-RS Symbol

In this section, we describe enhancements made to DM-RS channel estimation to improve performance in multipath channel scenarios. Algorithm is captured in Figure 12, and brief description of each module is given in subsequent text.

Figure 12. Channel Estimation Flow Chart with Enhanced DM-RS features



- TA Pre Correction module apply a common Timing offset (TO) adjustment to DM-RS symbol data at the output of LS module. We call this TA pre correction to identify this separately from TA correction described in Section 2.11, Timing Advanced Estimation and Compensation which we refer to as TA post correction. The common timing offset is the average of timing offset estimate for all layers (see Section 2.11, Timing Advanced Estimation and Compensation)
- TA Post correction in Section 2.11, Timing Advanced Estimation and Compensation is per Layer. Because of the common pre correction applied here, the estimate in TO estimate module estimates the residual offset for each layer.
- Frequency De-spread operations give channel estimation per Layer (TX Port). Then the TO estimation is performed for each layer followed by TA-post correction.
- Timing offset output, specified per layer, from DM-RS module into Channel Interpolator (MMSE or DCT) is the sum of the pre and post TO corrections.

$$TA'_{est,u}(n) = TA\_AVE(n) + TA_{est,u}(n)$$

Where n refers to slot number and u refers to layer index.

In Channel Interpolator, these per layer $TA'_{est,u}(n)$ are used to remove timing offset after the frequency interpolation, before the noise estimation process.

- Average of residual offset is added to the current average TO in order to form the new TO for the next slot.

$$TA\_AVE(n+1) = TA\_AVE(n) + mean(TA_{est,u})$$

- For the very first slot, we do not have a $TA\_AVE(1)$ to apply in TA pre correction phase. This issue can be addressed in two different ways.
  1. Initialize $TA\_AVE(1) = 0$ and use this as the estimate for first slot
  2. Do as above, but perform another iteration following the estimate of $TA_{est,u}(1)$ with the updated $TA\_AVE(2)$ applied to first slot as the TO pre correction.

     Apply linear frequency interpolation to TO adjusted DM-RS symbols to align subcarrier.
- Apply linear frequency interpolation to adjusted data to align the subcarrier indices before applying the frequency De-spread
- Perform frequency de-Spread to get per layer channel estimates.

New aspects of the algorithm, where it differs from Section 2.6, Channel Estimation on Front-Loaded DM-RS Symbol is described in following sub-sections.

2.7.1 DM-RS LS

DM-RS LS removes the DM-RS sequence from the sub-carrier data. Assuming the 2Tx case using DM-RS Tx port [0 1] according to the CDM group defined in Table 6.4.1.1.3-1 in 3GPP TS38.211 (refer to Table 2):

Taking DM-RS Type 1 as an example, LS step is given below,

$$Y'_{RS}(k_0,l,r) = \frac{Y_{RS}(k_0,l,r)}{X_{DMRS}(k_0,l)} = H_{RS}(k_0,l,r,t_0) + H_{RS}(k_0,l,r,t_1) + \frac{n(k_0,l,r)}{X_{DMRS}(k_0,l)}$$

$$Y'_{RS}(k_2,l,r) = \frac{Y_{RS}(k_2,l,r)}{X_{DMRS}(k_2,l)} = H_{RS}(k_2,l,r,t_0) - H_{RS}(k_2,l,r,t_1) + \frac{n(k_2,l,r)}{X_{DMRS}(k_2,l)}$$

2.7.2 TA Pre Correction

Apply the common TO correction to DM-RS, Y'_{RS}, taking correct subcarriers indices to account.

$$Y'_{RS,comp}(k,l,r) = Y'_{RS}(k,l,r) \cdot e^{j\frac{2\pi kTA\_AVE}{N_{FFT}}}$$

Taking the two sequences in Section 2.7.1 separately, and ignoring the noise term, this leads to,

$$Y'_{RS,comp}(k_0,l,r) = \left(H_{RS}(k_0,l,r,t_0) + H_{RS}(k_0,l,r,t_1)\right) \cdot e^{j\frac{2\pi k_0 TA\_AVE}{N_{FFT}}}$$

$$Y'_{RS,comp}(k_2,l,r) = \left(H_{RS}(k_2,l,r,t_0) - H_{RS}(k_2,l,r,t_1)\right) \cdot e^{j\frac{2\pi k_2 TA\_AVE}{N_{FFT}}}$$

2.7.3 Frequency Interpolation

Frequency interpolation is aimed at aligning the DM-RS pilot subcarrier indices before the frequency de-Spreading.

We can consider the righthand side of equations in Section 2.7.2, TA Pre Correction as the superposition of channel from the two antenna ports of interest, t_0 and t_1.

The impulse response of the superimposed channel is the coherent sum of the impulse responses of the two channels.

Denote

$$\left(H_{RS}(k_0,l,r,t_0)+H_{RS}(k_0,l,r,t_1)\right)\cdot e^{j\frac{2\pi k_0 TA\_AVE}{N_{FFT}}}=H_{RS}\left(k_0,l,r,t_{0,1}\right)$$

$$\left(H_{RS}(k_2,l,r,t_0)-H_{RS}(k_2,l,r,t_1)\right)\cdot e^{j\frac{2\pi k_2 TA\_AVE}{N_{FFT}}}=H_{RS}\left(k_2,l,r,t_{0,-1}\right)$$

Where $t_{0,1}$ indicates that this is the sum of channels for t_0 and t_1, and $t_{0,-1}$ indicates that this is the difference of channels for t_0 and t_1. First equation gives $t_{0,1}$ composite channel for subcarriers indices 0, 4, ..., and the second equation gives $t_{0,-1}$ composite channel for subcarrier indices 2, 6, ...

Apply the interpolation to this superimposed channel same as for individual channels because both channels are delay spread bounded well within what is required to avoid aliasing.

Doing Time offset pre correction before interpolation further improves accuracy of interpolation by 'centering' the impulse responses. Centering is simply interpolation of the channel and no attempt is made at noise filtering, which will be done in the processing chain using change to: MMSE or DCT based technique described in Sections 2.6.3, MMSE-Based Channel Estimation, and 2.6.4, DCT-Based Channel Estimation.

We interpolate by 1:2 to get channel values at k_0, k_2,... for both $t_{0,-1}$ and $t_{0,+1}$ composite channels.

$$\left[H_{RS}\left(k_0,l,r,t_{0,1}\right),H_{RS}\left(k_4,l,r,t_{0,1}\right),\ldots\right]\overrightarrow{Interp\ 1:2}\left[H_{RS}\left(k_0,l,r,t_{0,1}\right),H_{RS}\left(k_2,l,r,t_{0,1}\right),\ldots\right]$$

$$\left[H_{RS}\left(k_2,l,r,t_{0,-1}\right),H_{RS}\left(k_6,l,r,t_{0,-1}\right),\ldots\right]\overrightarrow{Interp\ 1:2}\left[H_{RS}\left(k_0,l,r,t_{0,-1}\right),H_{RS}\left(k_2,l,r,t_{0,-1}\right),\ldots\right]$$

Now, we have the sub-carrier indices for the two equations perfectly aligned to do the frequency de-Spread.

For DM-RS Type 2, we can apply the same algorithm described above to align the subcarriers indices for $t_{0,-1}$ and $t_{0,+1}$ composite channels. In this case, instead of 1:2 interpolation, appropriate phase of 1:3 interpolation must be applied.

2.7.4 Frequency De-Spread

2.7.4.1 Input

$H_{RS}(\mathrm{k},l,r,t_{0,1}), H_{RS}(\mathrm{k},l,r,t_{0,-1})$	Interpolated composite channel estimates from DM-RS
t	DM-RS transmission port index

2.7.4.2 Output

$\widehat{H}_{RS}(k,l,r,t)$	Estimated channel response

2.7.4.3 Algorithm

Since we have aligned the subcarrier indices for $t_{0,-1}$ and $t_{0,+1}$ composite channels, frequency de Spreading is done using the channel estimates with the same sub-carrier indices.

For example, consider the example that we developed above for DM-RS Type 1,

$$H_{RS}(k_{2n}, l, r, t_0) = \frac{H_{RS}(k_{2n},l,r,t_{0,1})+H_{RS}(k_{2n},l,r,t_{0,-1})}{2}$$

$$H_{RS}(k_{2n}, l, r, t_1) = \frac{H_{RS}(k_{2n},l,r,t_{0,1})-H_{RS}(k_{2n},l,r,t_{0,-1})}{2}$$

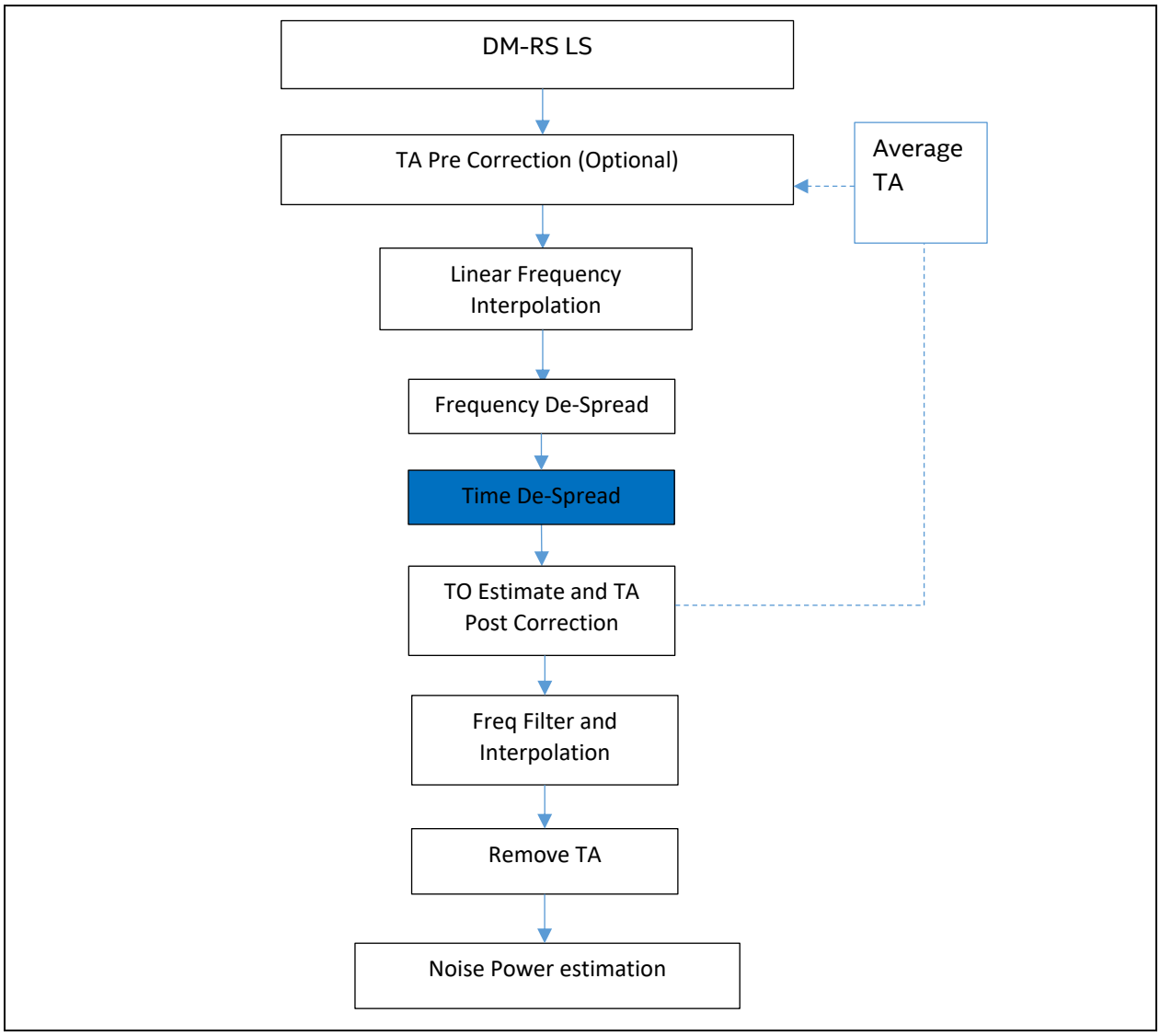
2.8 Dual DM-RS

Dual DM-RS refers to a pair of adjacent DM-RS symbols used to support more than 4 TX antenna ports in case of Type 1 and more than 6 TX antenna ports in the case of Type 2 DM-RS.

In this case, time de-Spread between the two DM-RS forming the dual DM-RS formation is used to add additional TX ports. Refer to Table 2, 38.211, Table 6.4.1.1.3-1 and 6.4.1.1.3-2, these tables describe how the DM-RS sequences are used to get 8 ports, and 12 ports for DM-RS Type 1 and 2 respectively.

Time de-Spread is applied directly after the frequency de-Spread as shown in figure below, where this module is only activated for Dual DM-RS.

Figure 13: Time-De-Spread for Dual DM-RS



Consider DM-RS Type 1 with 8 antenna ports for example. The following tables summarizes the operation of time de-Spread module. Here we are assuming symbol 2, and 3 in the slot corresponds to dual DM-RS symbols. DM-RS sequences seq2 and seq3 are used to modulate pilots for different TX Ports as shown in Table 4.

Table 4: DM-RS sequences assignment for TX ports

Port index	Sequence on symbol2	Sequence on symbol3
0	t0_seq2	t0_seq3
1	t1_seq2	t1_seq3
2	t2_seq2	t2_seq3
3	t3_seq2	t3_seq3

Port index	Sequence on symbol2	Sequence on symbol3
4	t4_seq2	t4_seq3 * -1
5	t5_seq2	t5_seq3 * -1
6	t6_seq2	t6_seq3 * -1
7	t7_seq2	t7_seq3 * -1

In order to get 8 orthogonal channels, the time De Spread is applied as shown in Table 5 below between the symbol 2 and 3 frequency de Spread channel estimates.

Table 5. Time De-Spread for Dual DM-RS

Port index	Channel on port
0	1/2 * (h0_sym2 + h0_sym3)
1	1/2 * (h1_sym2 + h1_sym3)
2	1/2 * (h2_sym2 + h2_sym3)
3	1/2 * (h3_sym2 + h3_sym3)
4	1/2 * (h4_sym2 - h4_sym3)
5	1/2 * (h5_sym2 - h5_sym3)
6	1/2 * (h6_sym2 - h6_sym3)
7	1/2 * (h7_sym2 - h7_sym3)

2.9 Additional DM-RS

The additional DM-RS symbol index relating to the duration of slots/data symbols for mapping type A/B is signalled by a high layer.

Multiple DM-RS symbol resources should be removed in TBS calculation.

Without PT-RS, when there are no additional DM-RS configured, we simply copy the channel estimate of the available DM-RS symbol to all UL data symbols within the same slot.

When there is an additional DM-RS symbol, linear interpolation is introduced to generate the channel estimation in other data symbols.

2.9.1 Input

$H_{RS}(k,l,r,t)$ Channel estimation of the front-loaded DM-RS symbol

i Symbol index of DM-RS symbols, there are in total N DM-RS symbols, including front-loaded and additional DM-RS symbols

j Symbol index of UL data symbols

2.9.2 Output

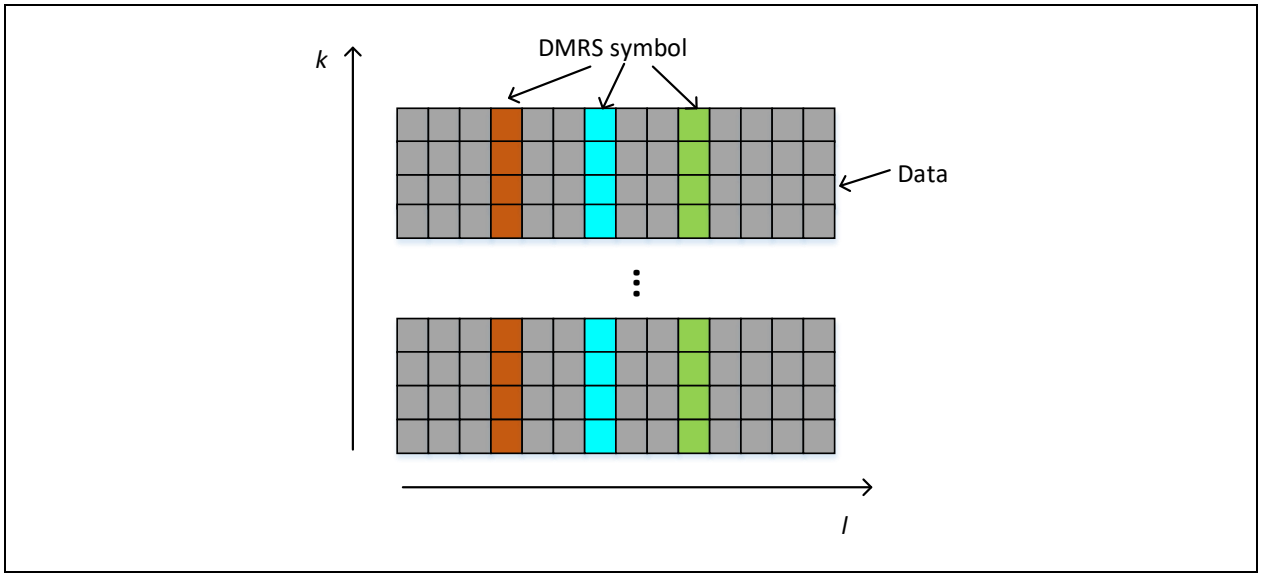
$H_{DS}(k,j,r,t)$ Channel estimation of UL data symbols

2.9.3 Algorithm

For simplicity, we denote $H_{RS,u}(k,i,r,t)$ as h_i where h_i represents the channel response in DM-RS symbol j, and $H_{DS}(k,j,r,t)$ as h_j, where h_j represents the channel response in data symbol j. The linear interpolation is applied as below:

$$h_j = \sum_{i=0}^{N-1} w_i h_i$$

Figure 14. Example of UL Slot with Additional DM-RS Symbols



Option1: Nearest

The simple approach is to copy the nearest channel estimate of the DM-RS symbol to data symbols.

$$w_i = \begin{cases} 1, & argmin|i-j| \\ 0, & others \end{cases}$$

Option 2: Linear Interpolation

- With single additional DM-RS, 1+1

  The weight for the j^{th} data symbol is calculated as:

$$\boldsymbol{w_0} = \frac{i_1 - \boldsymbol{j}}{i_1 - i_0},\ \boldsymbol{w_1} = \frac{\boldsymbol{j} - i_0}{i_1 - i_0}$$

- With double additional DM-RS, 1+1+1

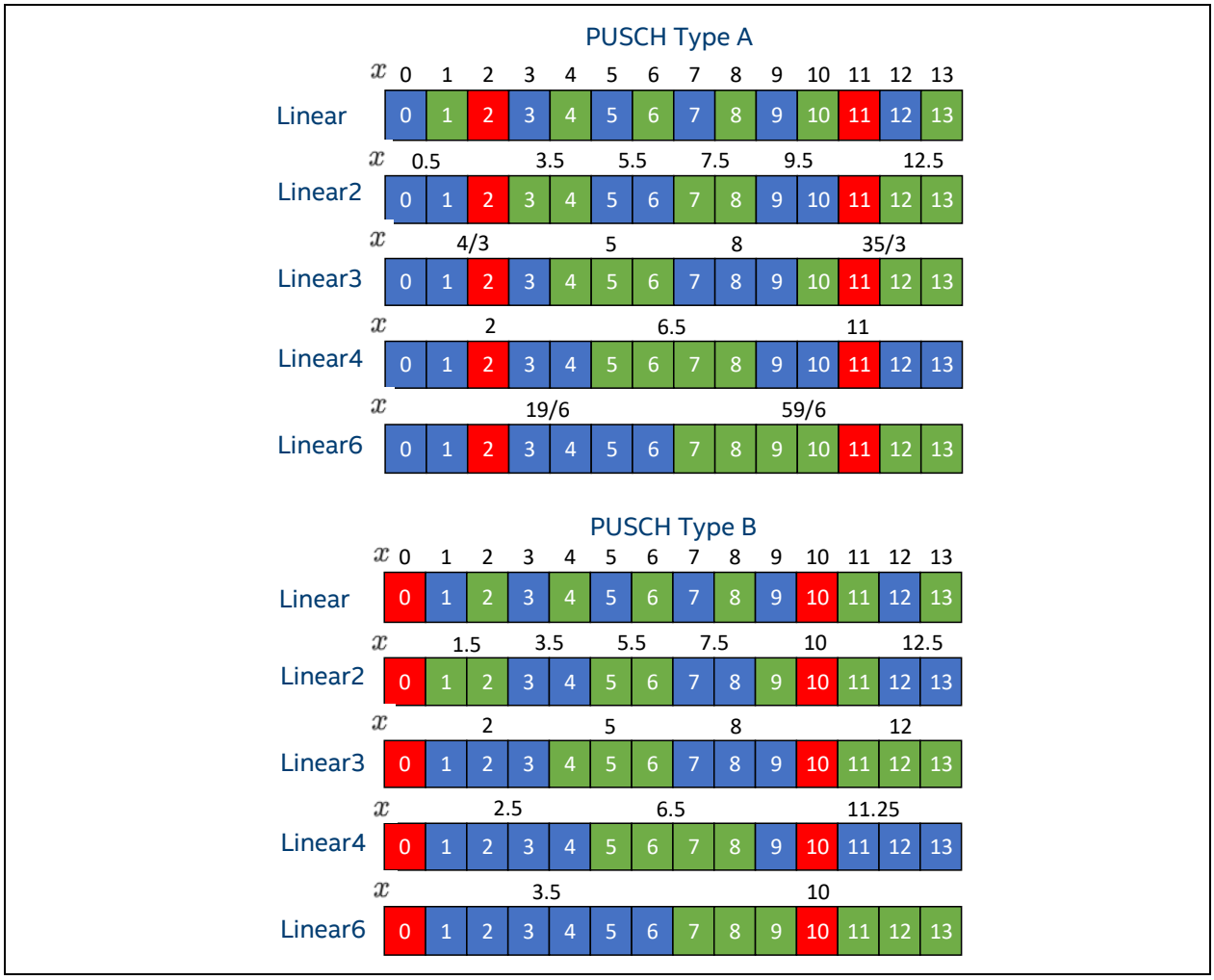
  Similarly, the weight for the j^{th} data symbol could be calculated as:

$$w_0 = \frac{[(j-i_1)(j-i_2)]}{[(i_0-i_1)(i_0-i_2)]}, w_1 = \frac{[(j-i_0)(j-i_2)]}{[(i_1-i_0)(i_1-i_2)]}, w_2 = \frac{[(j-i_0)(j-i_1)]}{[(i_2-i_0)(i_2-i_1)]}$$

Option 3: Grouped Linear interpolation with single additional DM-RS, 1+1

Computation load of per-symbol linear interpolation is a little high especially when taking equalization processing into account. It is reasonable to assume that channel response of several consecutive symbols in the time domain is constant for many practical scenarios. Several granularity examples are listed in Figure 15 and implemented. Those consecutive symbols are put together into one group and share the same channel response that is computed according to its mean of symbol index. The means index x is shown in Figure 15 for PUSCH type A/B and single DM-RS configuration.

Figure 15 Time domain linear interpolation



2.10 PT-RS-based Phase Noise Estimation and Compensation

PT-RS is enabled in FR2 only, and the impact of phase noise is shown in Figure 16 below.

PT-RS supports CP-OFDM waveform only.

Figure 16 are examples of constellation graphs showing the effects of phase noise estimation and compensation based on PT-RS.

Figure 16. Example of Constellation Graph *without* Phase Noise Estimation and Compensation

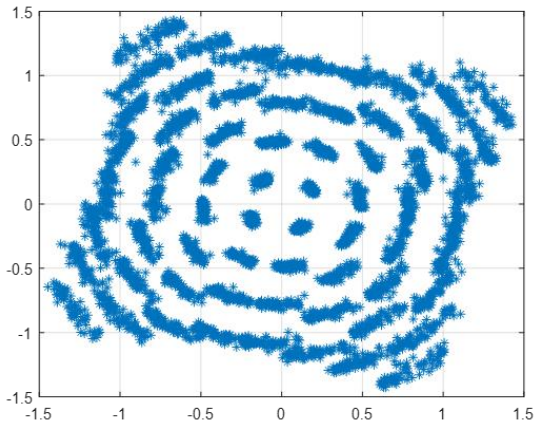
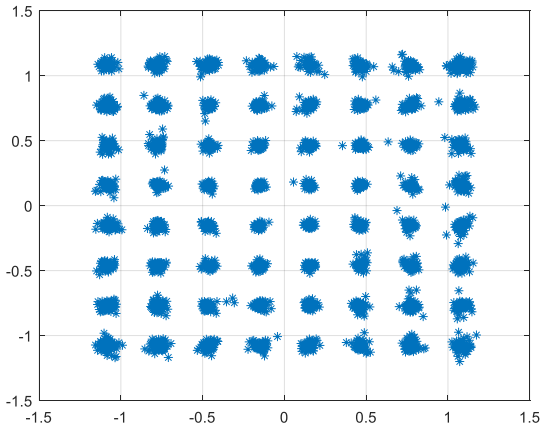


Figure 17. Example of Constellation Graph *with* Phase Noise Estimation and Compensation



Y_{PTRS}	Received PT-RS signal in the frequency domain
X_{PTRS}	Transmitted PT-RS sequence
H_{DMRS}	The estimated channel of front-loaded single DM-RS symbol

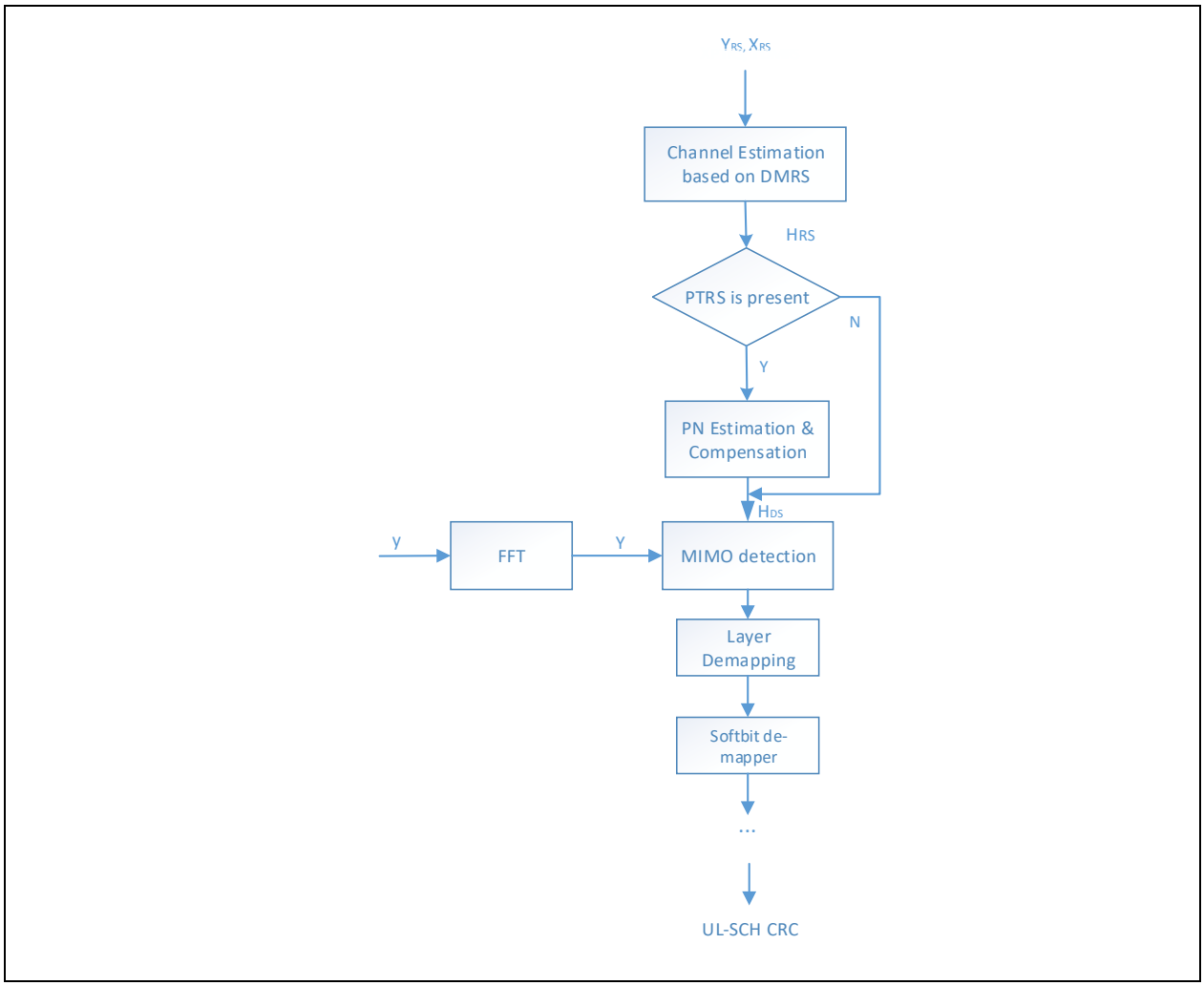
2.10.1 Output

H_{PTRS} The estimated channel of PT-RS

2.10.2 Algorithm

The constellation diagram (refer to Figure 18 and Figure 19 Is corrected using the PT-RS based phase noise estimate and compensation. The phase error is estimated based on PT-RS and then compensated.

Figure 18. PT-RS Processing in the Receiver



The reference signal channel can be estimated using the LS algorithm shown below:

$$H_{PTRS}(k,l) = [X_{PTRS}(k,l)]^* \cdot Y_{PTRS}(k,l)$$

where $(\,)^*$ is the complex conjugate.

Only the front-loaded DM-RS transmitted in one OFDM symbol is supported. Data is transmitted using other UL data symbols. The channel that is estimated for subcarrier *k* and data symbol *l* can be obtained by using:

$$H_{DS}(k,l) = H_{DMRS}(k,l')e^{j\theta_l}$$

where H_{DMRS} denotes the channel estimated from the front-loaded DM-RS.

θ_l can be obtained using:

$$\theta_l = \frac{1}{N}\sum_{j=1}^{N} \arg\{H_{PTRS}(k_j, l)H_{DMRS}(k_j, l')^*\}$$

where N denotes the total number of PT-RS subcarriers in data symbol l; indicates PT-RS subcarrier j.

To simplify the implementation:

Set $d_l = \sum_{j=1}^{N} \{H_{PTRS}(k_j, l)H_{DMRS}(k_j, l')^*\}$

$$= \sum_{j=1}^{N} \{H_{PTRS}(k_j, l)H_{DMRS}(k_j, l')^*| \cdot e^{j\theta_{lj}}\}$$

According to the nature of common phase error, $\theta_{lj} \approx \theta_l$, then

$d_l = e^{j\theta_l} \cdot \sum_{j=1}^{N} \{|H_{PTRS}(k_j, l)H_{DMRS}(k_j, l')^*|\} = e^{j\theta_l} \cdot |d_l|$,

Then,

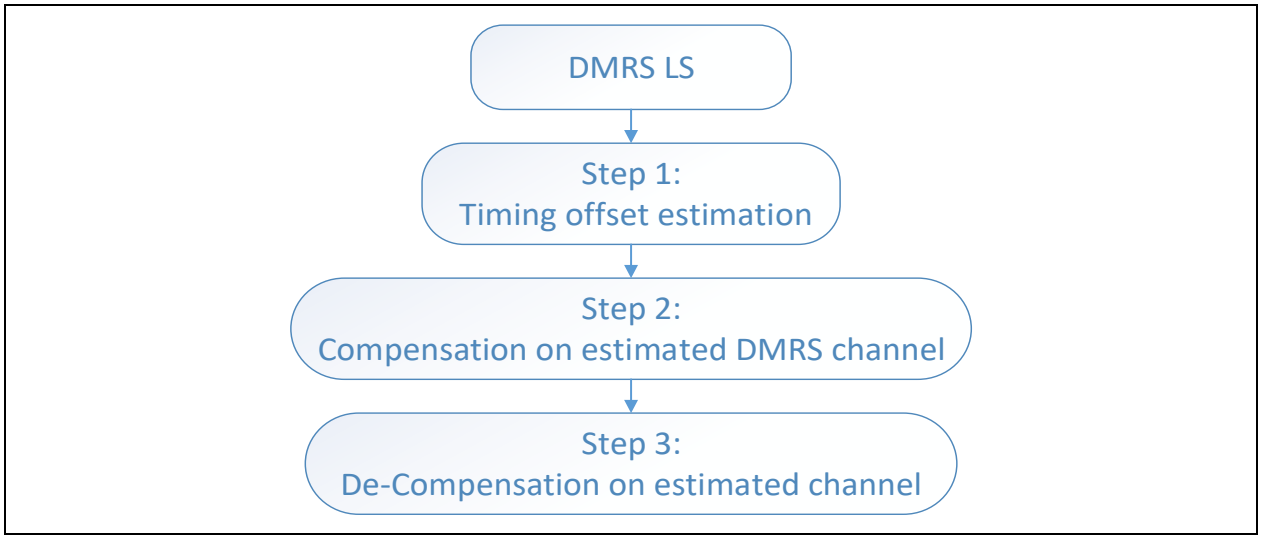
$e^{j\theta_l} = \frac{d_l}{|d_l|}$

2.11 Timing Advanced Estimation and Compensation

The advanced timing process (TA) includes estimation and compensation, as presented in Figure 19. The first two steps are conducted after DM-RS LS and before entering the channel estimation calculation so that time is aligned with the interpolation matrix.

In Figure 19, **Step 3** is performed before calculation of the noise power to keep the time alignment between the received data and estimated channel, and therefore Step 3 needs to be conducted twice since the channel estimation is a two-step-interpolation.

Figure 19. Illustration of TA Process



2.11.1 Timing Offset Estimation

This section describes the algorithm for estimation of timing offsets.

2.11.1.1 Input

$\hat{H}_{RS,u}(k,r)$	The matched filter channel estimate of RS for the u-th user
N_{FFT}	FFT size

2.11.1.2 Output

$TA_{est,l}$	The estimated TA for each layer

2.11.1.3 Algorithms

There are two steps to realize the timing offset value estimation.

1. Phase shift estimation:

$$e_u(r) = \sum_{k=0}^{K-2} \hat{H}_{RS,u}(k,r)\ (\ \hat{H}_{RS,u}(k+2,r))^H$$

where k is the DM-RS subcarrier index and $k = 0 : K-1$.

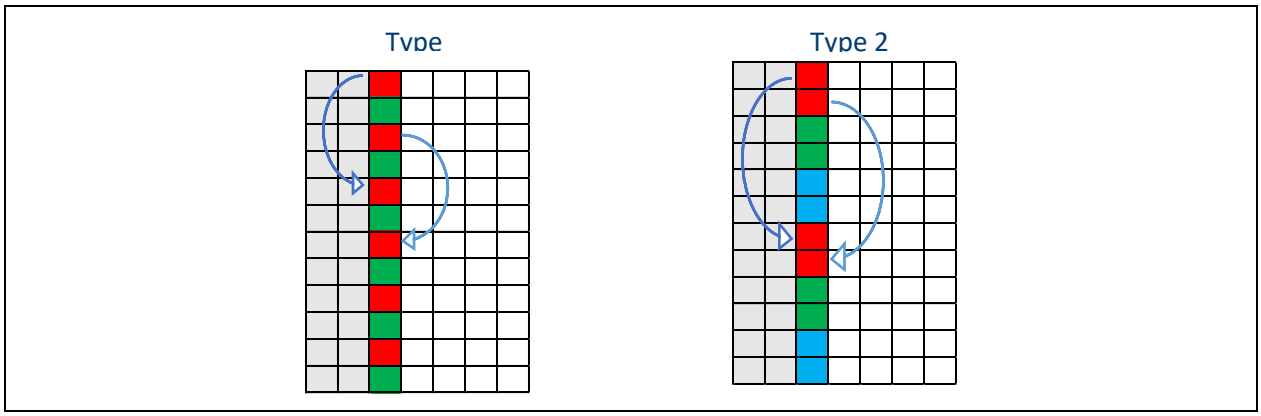
2. Convert phase shift to timing offset:

$$TA_{est,u} = \frac{angle(\sum_{r=0}^{R-1} e_u(r)) \cdot N_{FFT}}{2 \cdot \pi \cdot L}$$

where N_{FFT} is FFT size and , L is adjacent calculated DM-RS index offset as illustrated in Figure 20. L differs with the DM-RS type as below:

   a. L = 4 for DM-RS type 1
      - DM-RS index in a RB: 1,3,5,7,9,11
      - Calculate the rotation between 1 and 5, 3 and 7
   b. L = 6 for DM-RS type 2
      - DM-RS index in a RB: 1,2,7,8
      - Calculate the rotation between 1 and 7, 2 and 8.
      - For CDM case via OCC, the 1 and 2 have same value after LS

Figure 20. Illustration of the Angle Rotation Method



2.11.2 Compensation on DM-RS in LS

This section describes the algorithm for decompensating a DM-RS channel.

2.11.2.1 Input

$\widehat{H}_{RS,u}(k,r)$	The matched filter channel estimate of RS for the u-th user
$TA_{est,l}$	The estimated TA for each layer
N_{FFT}	FFT size

2.11.2.2 Output

$\widehat{H}_{RS,u,comp}(k,r)$	The compensated DM-RS channel

2.11.2.3 Algorithm

This step is to compensate the estimated timing offset value on DM-RS LS result in frequency domain.

$$\widehat{H}_{RS,u,comp}(k,r) = \widehat{H}_{RS,u}(k,r) \cdot e^{j\frac{2\pi tTA_{est,l}}{N_{FFT}}}$$

2.11.3 De-compensation on Estimated Channel

This section describes the algorithm for decompensating an estimated channel.

2.11.3.1 Input

$H_{RS,u}(k,r)$	The estimated channel after MMSE
$TA_{est,l}$	The estimated timing offset value per transmit layer
N_{FFT}	FFT size

2.11.3.2 Output

$H_{RS,u,decomp}(k,r)$	The estimated channel after de-compensation

2.11.3.3 Algorithm

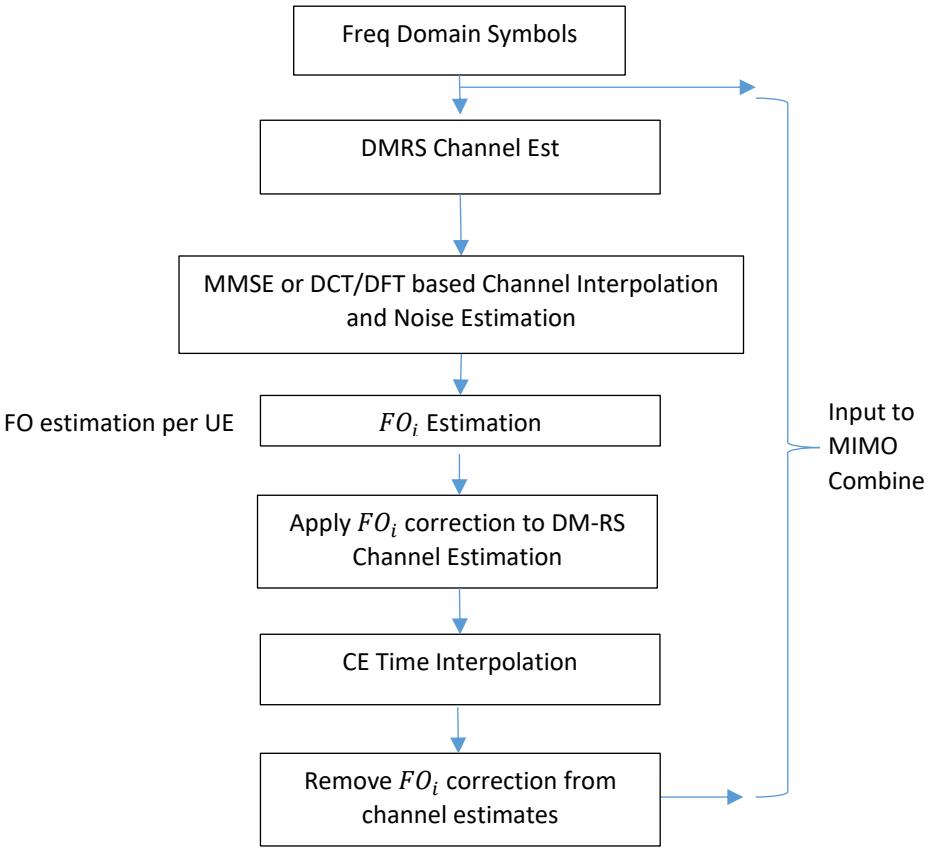
De-compensate on DM-RS in CE MMSE when calculating the noise power to keep time alignment of channel and received signal in noise calculation. CE adopts two-step MMSE, and therefore should be de-compensated two times, once for each of the two steps.

$$H_{RS,u,decomp}(k,r) = H_{RS,u}(k,r) \cdot e^{-j\frac{2\pi t TA_{est,l}}{N_{FFT}}}$$

2.12 Frequency Offset Estimation and Compensation with Multiple DM-RS Symbols

This section describes the algorithm for frequency offset (FO) estimation and compensation with multiple DM-RS symbols, as presented in Figure 21. The input to this module is the noise filtered and interpolated DM-RS channel estimates. The FO compensation is applied to channel estimates before time interpolation, and then reintroduced to align the phase of channel estimate with FO impacted (phase rotated) data symbols.

Figure 21. Illustration of FO Estimation and Compensation Process



2.12.1 Frequency Offset Estimation with Multiple DM-RS Symbols

This section describes the algorithm for the estimation of frequency offsets with multiple DM-RS symbols.

2.12.1.1 Input

$\widehat{H}_{RS,u}(k,i_l)$	DM-RS channel estimate for subcarrier k in DMRS symbol l
K_{dmrs}	The total number of DM-RS REs in one symbol
d_{dmrs}^{l}	The distance between two adjacent DM-RS symbols, l, and l -1l
M_R	The total number of receive antennas
M_p	The total number of transmitted layer numbers from UE
L	The total number of DM-RS symbols

2.12.1.2 Output

θ_i	The averaged estimated phase offset per UE (or Layer)

2.12.1.3 Algorithm

Assuming there are multiple DM-RS symbols in a slot, there are total L DM-RS symbols within the slot, and each DM-RS port occupy K_{dmrs} subcarriers. θ_i is estimated per UE, UE_i.

For a given UE, and for a given DM-RS symbol pair, phasors from all RX antennas should be the same.
Estimate phaser, P_p^m, for phase offset, θ_p^m, between two DM-RS symbols for received antenna m and transmitted layer p. (phasor $P_p^m = A_p^m e^{j\theta_p^m}$)

$$P_p^m(l-1,l,k) = \widehat{H}_{RS,u}(k,i_l) * conj(\widehat{H}_{RS,u}(k,i_{l-1}))$$

Where

$\widehat{H}_{RS,u}(k,i_l)$ DM-RS channel estimate for subcarrier k in DMRS symbol l

Symbol number index i_l for $l = 1,..,L$ is the symbol number of DM-RS symbol within the slot (range 0, ..., 13).

Since we have the knowledge of which Layers (TX antenna ports) map to which user, we can further average over TX ports from same UE.

Assume we have M UEs where $UE_i$is assigned ports in set S_i

Calculate the Sum of all phasors correspond to UE_i for DM-RS symbols l_1, and l_2 (with symbol indices i_{l_1} and i_{l_2} where $i_{l_1} < i_{l_2}$)

$$P_i(l_1,l_2) = \frac{1}{K_{dmrs}|S_i| * M_R} \sum_{p \in S_i} \left\{ \sum_{m=1}^{M_R} \left\{ \sum_{k=1}^{K_{dmrs}} P_p^m(l_1,l_2,k) \right\} \right\}$$

Where $|S_i|$is the cardinality of set S_i

The scale factor of $\frac{1}{K_{dmrs}|S_i| * M_R}$ is not needed in a floating-point implementation

Calculate all $P_i(l_1,l_2)$ for $i_{l_1} < i_{l_2}$ (i.e. $l_1 < l_2$)

Example 1: If only 2 DM-RS symbols, we have $P_i(1,2)$

Example 2: If we have 3 DM-RS symbols then, we have $P_i(1,2)$, $P_i(2,3)$, and $P_i(1,3)$

Convert the phasors to phase and average over all DM-RS pairs, considering different distances.

$$\theta_i = \frac{\sum_{l_1<l_2} angle\{P_i(l_1,l_2)\}}{\sum_{l_1<l_2} d_{dmrs}^{l_1,l_2}}$$

Where $d_{dmrs}^{l_1,l_2} = (i_{l_2} - i_{l_1})$

The above equation implements MRC combining of angle estimates to get optimal estimate of θ_i.

We repeat this calculation process for other UEs

Pass through a one-tap IIR filter to smooth the estimated value

IIR filter weight:

QPSK: current weight: 0.5; history weight: 0.5.

16QAM: current weight: 0.25; history weight: 0.75.

64QAM: current weight: 0.1; history weight: 0.9.

2.12.2 Frequency Offset Compensation

This section describes the algorithm for frequency offsets compensation.

2.12.2.1 Input

θ_i	The averaged estimated phase offset
$H_{UE_i}(k,l,r)$	DMRS channel estimate for subcarrier k, symbol l and Rx antenna r

2.12.2.2 Output

$H_{TIout,UE_i}(k,l,r)$	Time Interpolated channel estimates for subrrier k, symbol l, and Rx antenna r, with FO been taken care of.

2.12.2.3 Algorithm

We cannot apply the FO to OFDM symbol data before MIMO combine as it contains a mix of signals from all UEs.

Instead, we send the channel estimate with the correct phase rotation corresponding to FO_i into MIMO Combine. Because both the data symbols and channel estimation is subjected to the same phase rotation, overall, there is no relative phase error.

We need to compensate for FO induced phase rotation from DM-RS channel estimates to avoid linear time interpolator (TI) severely getting impacted by phase rotation between DM-RS symbols.

Firstly, apply FO correction to DM-RS channel estimates,

$$H'_{UE_i}(k,l,r) = H_{UE_i}(k,l,r) * e^{-ji_l\theta_i}$$

Then, the time interpolation is applied, as described in section 2.9, to $H'_{UE_i}(k,l,r)$ to get interpolated channel $H'_{TIout,UE_i}(k,l,r)$. Following Time Interpolation (TI), FO is reintroduced to channel estimate on every symbol, so that the channel estimation phase is aligned with symbol phase,

$$H_{TIout,UE_i}(k,l,r) = H'_{TIout,UE_i}(k,l,r) * e^{jl\theta_i}$$

2.13 Doppler Shift Estimation

The section describes the algorithm for Doppler shift estimation. The algorithm works when there are 2 DMRS symbols in one slot. When there are more than 2 DMRS symbols in one slot, the algorithm repeats for each consecutive DMRS symbol pair.

2.13.1 Input

$\widehat{H}_{RS}(k,l,r,t)$	Estimated channel response on DM-RS symbol l
σ^2	The estimated noise power
K	number of DMRS subcarriers
R	number of receiving antennas
l_0, l_1	index of two DMRS symbols

2.13.2 Output

$f_d(t)$	Doppler shift estimation for t-th layer

2.13.3 Algorithm

1. Compute auto-correlation of DMRS:

$$R_0 = \frac{1}{2KR}\sum_{r=0}^{R-1}\sum_{p=0}^{1}\sum_{k=0}^{K-1}|\widehat{H}_{RS}(k,l_p,r,t)\,|^2 - 0.1\sigma^2$$

2. Compute cross correlation of DMRS

$$R_1 = \frac{1}{KR}\sum_{r=0}^{R-1}\sum_{k=0}^{K-1}|\widehat{H}^*_{RS}(k,l_0,r,t)\widehat{H}_{RS}(k,l_1,r,t)|$$

3. Estimate Doppler shift

$$f_d = \frac{1}{2\pi T_d}\sqrt{\frac{1-R_1/R_0}{0.223}}$$

Where T_d is the time difference between the two consecutive DMRS symbols.

2.14 Time Domain Interpolation Algorithm Selection

This section describes how to select time interpolation algorithm to trade-off between complexity and performance for given input parameters. The input parameters in consideration are doppler shift, MCS, and antenna configuration. For each combination of these input parameters, the algorithm to select the suitable interpolation method is shown.

2.14.1 Input

d Doppler shift index where $d = \{0, 1, 2, 3, 4\}$ represents doppler shift values $0, 100, 200, 300, 400$ Hz respectively. The index is selected to choose the value that is closest to the estimated doppler shift.

m MCS index where $m = \{0, 1, 2, 3\}$ represents doppler shift values $4, 10, 19, 27$ respectively.

a Antenna configuration index. The supported antenna configurations are $1x2, 1x16, 2x2, 2x4, 2x16, 4x4, 4x16$.

2.14.2 Output

$T_a(m, d)$ Time interpolation table for antenna configuration a. Each element represents the selected method for the mth MCS and the dth doppler shift. Each elements has a value i, where $i = \{0, 1, 2, 3, 4\}$ to represent interpolation methods $Linear, Linear2, Linear3, Linear4, Linear6$ respectively.

2.14.3 Algorithm

1. For each channel model (TDL-A, TDL-B, TDL-C):
   a. Simulate throughput percent vs. SNR for all interpolation methods.
   b. At 70% throughput select the method with least SNR and call it method A.
   c. Select the least complicated method within 0.2 dB (at 70% throughput) from method A and call it method B. There will be method B for each channel model.

2. Select method B with least SNR among all channel models. This will be selected as the method for the given combination of doppler shift, MCS and antenna configuration.

2.15 MIMO Detection

This section describes three algorithms for detecting Multiple Input and Multiple Output.

2.15.1 Input

$Y_{DS}(k, r)$ Frequency domain data symbol

$H_{DS}(k, r, t)$ The estimated channel response of data symbols

σ_u^2 Estimated noise power

2.15.2 Output

Z_u The equalized data of the user u

β Equalization gain for user u of each subcarrier

2.15.3 Algorithms

There are three available algorithms for MIMO detection. All are implemented for FlexRAN.

2.15.3.1 MMSE

This approach is the current default used for FlexRAN.

The MIMO MMSE is based on the output of FFT and channel estimation result and aimed to get the optimized SNR.

A MIMO system can be described as:

$$Y_{DS}(k,r) = H_{DS}(k,r,t) \cdot X(k,t) + \sigma_u^2$$

where

$$H_{DS}(k,r,t) = \begin{bmatrix} H_{DS}(k,0,0) & H_{DS}(k,0,1) & \cdots & H_{DS}(k,0,N_{tx}-1) \\ H_{DS}(k,1,0) & H_{DS}(k,1,1) & \cdots & H_{DS}(k,1,N_{tx}-1) \\ \vdots & \vdots & \ddots & \vdots \\ H_{DS}(k,N_{rx}-1,0) & H_{DS}(k,N_{rx}-1,1) & \cdots & H_{DS}(k,N_{rx}-1,N_{tx}-1) \end{bmatrix}$$

where N_{tx} denotes the number of DM-RS transmission ports and N_{rx} denotes the number of RX antennas.

The MIMO MMSE equalization matrix W is given as

$$W = (H^H \cdot H + \sigma_u^2 \cdot I_{Nt})^{-1} \cdot H^H$$

Then the output of MIMO MMSE (the estimated TX signal) can be obtained by

$$Z_u = W \cdot Y_{DS}(k,r) = (H^H \cdot H + \sigma_u^2 \cdot I_{Nt})^{-1} \cdot H^H \cdot Y_{DS}(k,r)$$

2.15.3.2 MMSE IRC

The MMSE-IRC algorithm approach is also based on the frequency domain data and channel estimation result. The interference plus noise is estimated to get the optimized SINR.

The system should be described as:

$$Y_{DS}(k,r) = H_{DS}(k,r,t) \cdot X(k,t) + n + I$$

And I denotes the sum of interference from all interference users.

The MMSE-IRC equalization matrix W is given as

$$W = H^H \cdot (H \cdot H^H + R_{nn})^{-1}$$

$$R_{nn} = \frac{1}{n}\sum R_{IpN}$$

$$R_{IpN} = (Y_{DS} - HX_{DMRS})(Y_{DS} - HX_{DMRS})^H$$

The covariance matrix of interference plus noise R_{IpN} here is obtained in every DM-RS subcarrier. According to DM-RS patterns in frequency domains, interpolation is necessary to derive the covariance matrix in all scheduled subcarriers.

$$n = N_{inter\_block} \times N_{DMRS\_RB}$$

N_{inter_block} is the matrix averaging unit, in terms of RBs. N_{DMRS_RB} is the total number of DM-RS subcarriers of each RB for this DM-RS pattern.

When additional DM-RS symbol(s) are available, we need to carry out an interpolation to obtain a more accurate R_{IpN} of a different symbol. If the nearest approach is used, the value of R_{nn} is set to the value calculated from the nearest available DM-RS symbol.

Then the output of this function can be obtained by

$$Z_u = W \cdot Y$$

2.15.3.3 Equalization Gain

Equalization gain β is calculated based on estimated channel *H* and antenna combination weight *W*.

$$\beta = Real\{diag(WH)\}$$

It can be averaged in RB groups for simplicity.

2.16 Layer De-mapping

This section describes the algorithms for de-mapping MIMO detection output. There are different algorithms used for cases of one to four layers.

2.16.1 Input

Z_u	MIMO detection output, [nLayer x Nsymbols]

2.16.2 Output

r	Layer-demapped output

2.16.3 Algorithm

$nLayers = 1,$

$$z_u = Z_u$$

$nLayers = 2,$

$$z_u(1:2:end) = Z_u(1,:)$$

$$z_u(2:2:end) = Z_u(2,:)$$

$nLayers = 3,$

$$z_u(1:3:end) = Z_u(1,:)$$

$$z_u(2:3:end) = Z_u(2,:)$$

$$z_u(3:3:end) = Z_u(3,:)$$

$nLayers = 4,$

$$z_u(1:4:end) = Z_u(1,:)$$

$$z_u(2:4:end) = Z_u(2,:)$$

$$z_u(3:4:end) = Z_u(3,:)$$

$$z_u(4:4:end) = Z_u(4,:)$$

$$r = z_u$$

2.17 Transform De-precoding

This section works only when transform precoding is enabled for DFT-S-OFDM.

2.17.1 Input

r	Layer-demapped output

2.17.2 Output

r^{tp}	Transform de-precoding output

2.17.3 Algorithm

Transform de-precoding is to perform IDFT for the sequence, which is the reverse process of transmission, as defined in clause 6.3.1.4 in *3GPP TS38.211* (refer to Figure 2).

2.18 Softbits De-mapping

This section describes the algorithm for determining soft values representing the logarithmic probability of one and zero for each bit in each layer.

2.18.1 Input

r or r^{tp}	Layer de-mapping output or transform de-precoding output
β	Equalization gain for user u of each subcarrier

2.18.2 Output

q	Soft values; refPHY has saturated truncation from 16 bits to 8 bits for LLR soft bits

2.18.3 Algorithms

In this section, constellation values are converted to soft values representing the logarithmic probability of one and zero for each bit. These are further processed in the following functions, de-scrambling, channel de-interleaver before channel decoder. The input data has been biased due to `MIMO MMSE` equalization. The normalized factor is as follows:

$$c = \frac{\beta}{\sqrt{2}}(\text{pi/2 BPSK, QPSK}), \quad \frac{\beta}{\sqrt{10}}(16QAM), \quad \frac{\beta}{\sqrt{42}}(64QAM), \quad \text{or} \quad \frac{\beta}{\sqrt{170}}(256QAM)$$

As a result, the constant in the demodulation algorithm should also be biased by the normalized factor.

- For pi/2 BPSK, the soft bit can be calculated as:

$$\begin{bmatrix} q_0 & q_1 \end{bmatrix} = [\sqrt{2} \cdot \frac{1}{1-\beta} \cdot (re(r) + im(r)), \quad \sqrt{2} \cdot \frac{1}{1-\beta} \cdot (-re(r) + im(r))]$$

- For QPSK, the soft bit can be calculated as:

$$\left[q_0 \;\; q_1\right] = [2\sqrt{2}\cdot\frac{1}{1-\beta}\cdot re(r), 2\sqrt{2}\cdot\frac{1}{1-\beta}\cdot im(r)]$$

- For 16QAM, the soft bit can be calculated as: $\left[q_0 \;\; q_1 \;\; q_2 \;\; q_3\right]$

$$\sigma_n^2 = \frac{1}{2}\cdot(\beta-\beta^2)$$

$$q_0 = \begin{cases} \dfrac{8c(re(r)+c)}{2\sigma_n^2}, & -\infty < re(r) \le -2c \\ \dfrac{4c\cdot re(r)}{2\sigma_n^2}, & -2c < re(r) \le 2c \\ \dfrac{8c(re(r)-c)}{2\sigma_n^2}, & 2c < re(r) \le \infty \end{cases}$$

$$q_1 = \begin{cases} \dfrac{8c(im(r)+c)}{2\sigma_n^2}, & -\infty < im(r) \le -2c \\ \dfrac{4c\cdot im(r)}{2\sigma_n^2}, & -2c < im(r) \le 2c \\ \dfrac{8c(im(r)-c)}{2\sigma_n^2}, & 2c < im(r) \le \infty \end{cases}$$

$$q_2 = \begin{cases} \dfrac{4c(re(r)+2c)}{2\sigma_n^2}, & -\infty < re(r) \le 0 \\ \dfrac{4c(-re(r)+2c)}{2\sigma_n^2}, & 0 < re(r) \le \infty \end{cases}$$

$$q_3 = \begin{cases} \dfrac{4c(im(r)+2c)}{2\sigma_n^2}, & -\infty < im(r) \le 0 \\ \dfrac{4c(-im(r)+2c)}{2\sigma_n^2}, & 0 < im(r) \le \infty \end{cases}$$

- For 64QAM, the soft bit can be calculated as: $\left[q_0 \;\; q_1 \;\; q_2 \;\; q_3 \;\; q_4 \;\; q_5\right]$

  As done with 16QAM, q_0 and q_1, q_2 and q_3, q_4 and q_5 use the same equation while with the input of the real part of r and imaginary part of r, respectively.

$$\sigma_n^2 = \frac{1}{2}\cdot(\beta-\beta^2)$$

$$q_0 = \begin{cases} \dfrac{16c(re(r)+3c)}{2\sigma_n^2}, & -\infty < re(r) \le -6c \\ \dfrac{12c(re(r)+2c)}{2\sigma_n^2}, & -6c < re(r) \le -4c \\ \dfrac{8c(re(r)+c)}{2\sigma_n^2}, & -4c < re(r) \le -2c \\ \dfrac{4c \cdot re(r)}{2\sigma_n^2}, & -2c < re(r) \le 2c \\ \dfrac{8c(re(r)-c)}{2\sigma_n^2}, & 2c < re(r) \le 4c \\ \dfrac{12c(re(r)-2c)}{2\sigma_n^2}, & 4c < re(r) \le 6c \\ \dfrac{16c(re(r)-3c)}{2\sigma_n^2}, & 6c < re(r) \le \infty \end{cases}$$

$$q_2 = \begin{cases} \dfrac{8c(-|re(r)|+3c)}{2\sigma_n^2}, & -\infty < |re(r)| \le 2c \\ \dfrac{4c \cdot (-|re(r)|+4c)}{2\sigma_n^2}, & 2c < |re(r)| \le 6c \\ \dfrac{8c(-|re(r)|+5c)}{2\sigma_n^2}, & 6c < |re(r)| \le \infty \end{cases}$$

$$q_4 = \begin{cases} \dfrac{4c(|re(r)|-2c)}{2\sigma_n^2}, & -\infty < |re(r)| \le 4c \\ \dfrac{4c(-|re(r)|+6c)}{2\sigma_n^2}, & 4c < |re(r)| \le \infty \end{cases}$$

- For 256QAM, the softbits can be calculated as: $\begin{bmatrix} q_0 & q_1 & q_2 & q_3 & q_4 & q_5 \end{bmatrix}$

  Like 16QAM, q_0 and q_1, q_2 and q_3, q_4 and q_5 use the same equation while with the input of the real part of r and imaginary part of r, respectively.

$$\sigma_n^2 = \frac{1}{2} \cdot (\beta - \beta^2)$$

$$q_0 = \begin{cases} \frac{64c(re(r)+7c)}{2\sigma_n^2}, & -\infty < re(r) \le -14c \\ \frac{56c(re(r)+6c)}{2\sigma_n^2}, & -14c < re(r) \le -12c \\ \frac{48c(re(r)+5c)}{2\sigma_n^2}, & -12c < re(r) \le -10c \\ \frac{40c(re(r)+4c)}{2\sigma_n^2}, & -10c < re(r) \le -8c \\ \frac{32c(re(r)+3c)}{2\sigma_n^2}, & -8c < re(r) \le -6c \\ \frac{24c(re(r)+2c)}{2\sigma_n^2}, & -6c < re(r) \le -4c \\ \frac{16c(re(r)+c)}{2\sigma_n^2}, & -4c < re(r) \le -2c \\ \frac{8c \cdot re(r)}{2\sigma_n^2}, & -2c < re(r) \le -2c \\ \frac{16c(re(r)-c)}{2\sigma_n^2}, & 2c < re(r) \le 4c \\ \frac{24c(re(r)-2c)}{2\sigma_n^2}, & 4c < re(r) \le 6c \\ \frac{32c(re(r)-3c)}{2\sigma_n^2}, & 6c < re(r) \le 8c \\ \frac{40c(re(r)-4c)}{2\sigma_n^2}, & 8c < re(r) \le 10c \\ \frac{48c(re(r)-5c)}{2\sigma_n^2}, & 10c < re(r) \le 12c \\ \frac{56c(re(r)-6c)}{2\sigma_n^2}, & 12c < re(r) \le 14c \\ \frac{64c(re(r)-7c)}{2\sigma_n^2}, & 14c < re(r) \le \infty \end{cases}$$

$$q_2 = \begin{cases} \frac{8c(-4 * |re(r)| + 20c)}{2\sigma_n^2}, & 0 < |re(r)| \le 2c \\ \frac{6c \cdot (-4 * |re(r)| + 24c)}{2\sigma_n^2}, & 2c < |re(r)| \le 4c \\ \frac{4c \cdot (-4 * |re(r)| + 28c)}{2\sigma_n^2}, & 4c < |re(r)| \le 6c \\ \frac{2c \cdot (-4 * |re(r)| + 32c)}{2\sigma_n^2}, & 6c < |re(r)| \le 10c \\ \frac{4c \cdot (-4 * |re(r)| + 36c)}{2\sigma_n^2}, & 10c < |re(r)| \le 12c \\ \frac{6c \cdot (-4 * |re(r)| + 40c)}{2\sigma_n^2}, & 12c < |re(r)| \le 14c \\ \frac{8c(-4 * |re(r)| + 44c)}{2\sigma_n^2}, & 14c < |re(r)| \le \infty \end{cases}$$

$$q_4 = \begin{cases} \frac{16c \cdot (|re(r)| - 3c)}{2\sigma_n^2}, & 0 < |re(r)| \leq 2c \\ \frac{8c \cdot (|re(r)| - 4c)}{2\sigma_n^2}, & 2c < |re(r)| \leq 6c \\ \frac{16c \cdot (|re(r)| - 5c)}{2\sigma_n^2}, & 6c < |re(r)| \leq 8c \\ \frac{16c \cdot (-|re(r)| + 11c)}{2\sigma_n^2}, & 8c < |re(r)| \leq 10c \\ \frac{8c \cdot (-|re(r)| + 12c)}{2\sigma_n^2}, & 10c < |re(r)| \leq 14c \\ \frac{16c \cdot (-|re(r)| + 13c)}{2\sigma_n^2}, & 14c < |re(r)| \leq \infty \end{cases}$$

$$q_6 = \begin{cases} \frac{8c \cdot (|re(r)| - 2c)}{2\sigma_n^2}, & 0 < |re(r)| \leq 4c \\ \frac{8c \cdot (-|re(r)| + 6c)}{2\sigma_n^2}, & 4c < |re(r)| \leq 8c \\ \frac{8c \cdot (|re(r)| - 10c)}{2\sigma_n^2}, & 8c < |re(r)| \leq 12c \\ \frac{8c \cdot (-|re(r)| + 14c)}{2\sigma_n^2}, & 12c < |re(r)| \leq \infty \end{cases}$$

2.19 De-scrambling

This section describes the algorithm for de-scrambling output.

2.19.1 Input

q	Soft values

2.19.2 Output

$B_{de-scramble}$	Output values after de-scramble

2.19.3 Algorithm

This function is the opposite processing of scrambling in chapter 6.3.1.1 in *3GPP TS38.211* (refer to Table 2).

The calculation method of scramble sequence c is given by clause 5.2.1 in *3GPP TS38.211* (refer to Table 2).

$$B_{de-scramble} = q \cdot (1 - 2 \cdot c)$$

2.20 Data and Control De-multiplexing

This section describes the algorithm for de-multiplexing output, which has been de-scrambled.

2.20.1 Input

$B_{de-scramble}$	De-scrambled bit codes output
C	Total number of code blocks for UCI
N_L	The number of transmission layers.
Q_m	Order of modulation, 2 for QPSK, 4 for 16QAM, and 6 for 64QAM
$N_{\text{symb,all}}^{\text{PUSCH}}$	Total number of OFDM symbols of the PUSCH
$M_{\text{sc}}^{\text{PUSCH}}$	Total number of subcarriers

2.20.2 Output

g_{UL-SCH}	De-multiplexed UL-SCH code bits sequence
g_{ACK}^{UCI}	De-multiplexed ACK code bits sequence
$g_{\text{CSI-part1}}^{UCI}$	De-multiplexed CSI part 1 code bits sequence
$g_{\text{CSI-part2}}^{UCI}$	De-multiplexed CSI part 2 code bits sequence

2.20.3 Algorithm

The input is the de-scrambled bits sequence $B_{de-scramble}$ with data and control codes multiplexed together.

The process of de-multiplexing is the reverse process of data and control multiplexing, which includes six steps. To Refer to Chapter 6.2.7 in *TS38.212* listed in Table 2.

The sequence g_{UL-SCH}, g_{ACK}^{UCI}, $g_{\text{CSI-part1}}^{UCI}$ and $g_{\text{CSI-part2}}^{UCI}$ can be obtained after the six steps.

Code Block de-concatenation, rate de-matching, channel decoding, code block de-segmentation, and CRC check are conducted in the following for the de-multiplexed control bits, as illustrated in Figure 1. The algorithm used for control bits is similar to the PUCCH process, which is described in Chapter 4.0, PUCCH Algorithms.

2.21 Rate De-matching

This section describes the algorithm for rate de-matching.

2.21.1 Input

g_{UL-SCH}	De-multiplexed data code bits sequence of UL-SCH
N_{cb}	Number of coded bits of the code block – no consideration of LBRM
$Basegraph$	Base graph of LDPC channel coding, 1 or 2
Z_c	The minimum value of Z
f	Number of filler bits
r	Code block index, starting from 0
C	Total number of code blocks
G	Total number of bits available for the transmission of one transport block
N_L	The number of transmission layers.

Q_m	Order of modulation, 2 for QPSK, 4 for 16QAM, and 6 for 64QAM
$RVid$	Redundancy version index for this transmission, [0, 1, 2, 3]
K	The number of bits to encode

2.21.2 Output

B_{de-rm} Rate de-matched values of the code block

2.21.3 Algorithm

First, the de-scrambled soft values g_{UL-SCH} are reshaped to B_{in} in each code block according to the rate matching size of the code block. Then this function is done in every code block loop, and it is the opposite processing of rate matching for LDPC code in clause 5.4.2 in *3GPP TS38.212* (refer to Table 2); hence bit de-interleaving and bit de-selection are applied.

- De-interleaver

  M = reshape(B_{in}, Qm, []);

  e = reshape(M',1,[]);

- Bit de-selection

  The starting position of different redundancy versions k_0 in *3GPP TS38.212* (refer to Table 2):

  $Basegraph = 1$:

$$k_0 = [0, \quad \left\lfloor \frac{17N_{cb}}{66Z_c} \right\rfloor \cdot Z_c, \left\lfloor \frac{33N_{cb}}{66Z_c} \right\rfloor \cdot Z_c, \left\lfloor \frac{56N_{cb}}{66Z_c} \right\rfloor \cdot Z_c \,]$$

  $Basegraph = 2$:

$$k_0 = [0, \quad \left\lfloor \frac{13N_{cb}}{50Z_c} \right\rfloor \cdot Z_c, \left\lfloor \frac{25N_{cb}}{50Z_c} \right\rfloor \cdot Z_c, \left\lfloor \frac{43N_{cb}}{50Z_c} \right\rfloor \cdot Z_c \,]$$

  If $\leq C - mod\left(\frac{G}{N_L * Q_m}, C\right) - 1$,

$$E = N_L \cdot Q_m \cdot \lfloor G/(N_L * Q_m * C) \rfloor$$

  else

$$E = N_L \cdot Q_m \cdot \lceil G/(N_L * Q_m * C) \rceil$$

  Set k = 1, j = 0;

  B_{de-rm} = zeros(1,E);

  while k < E + 1

    a = mod(k_0(RVid + 1) + j, Ncb) + 1;

    if B_{in}(a) ~= nullbit

      B_{de-rm}(k) = B_{in}(a);

      k = k + 1;

    end

    j = j + 1;

end

2.22 LDPC Decoding

The rate de-matched bits of each code block are decoded here.

This function is done by third-party provided hardware, and thus will not be introduced here.

2.23 CRC Check

This section describes the algorithm for Transport Block (TB) Cyclic Redundancy Check (CRC).

2.23.1 Input

$Buf_{decoded}$	LDPC decoded bits of the transport block
parity_bits	CRC polynomial, 24A, 24B, 24C, 16, 11, 6

2.23.2 Output

sig	Decoded bits without CRC
crc_check	CRC check result, 0 or 1

2.23.3 Algorithm

This function covers CRC check of the transport block. Certain parity bits are used according to coding rate and number of information bits. The parity bits are generated by one of the cyclic generator polynomials defined in Chapter 5.1 in *3GPP TS38.212* (refer to Table 2). Calculate the remainder by dividing $Buf_{decoded}$ by polynomial. If the remainder is zero, then it passes the CRC check. Otherwise, it fails the CRC check. Finally, remove the redundancy.

Let n = length($Buf_{decoded}$), m = length(*parity_bits*) - 1;

for *i* = 1: *n*

  z = *register*(m)* *parity_bits*(m + 1);

  for p = m: -1: 2

    register(p) = xor(register(p-1), z * parity_bits(p));

  end

  register(1) = xor($Buf_{decoded}$ (i), z * parity_bits(1));

end

if sum(*register*) == 0

  crc_check = 1;

else

  crc_check = 0;

end

$sig = Buf_{decoded}[1 : end - length(parity\_bits)]$

§

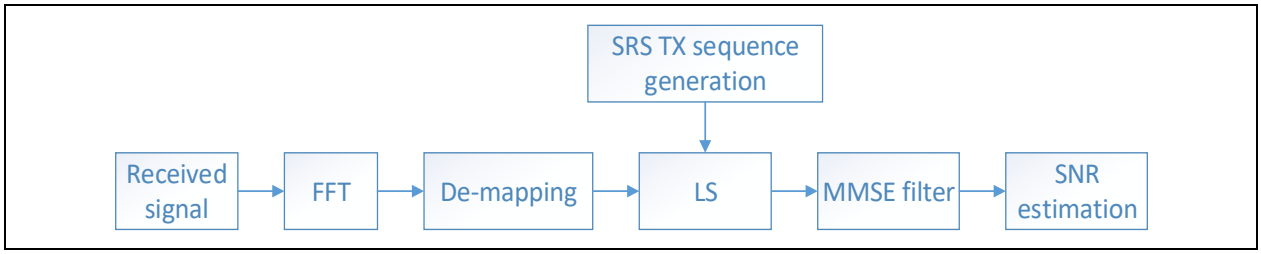
3.0 SRS Algorithms

This section introduces two methods of the Sounding Reference Signal (SRS) channel estimation algorithm for FlexRAN project, which are MMSE CE and DFT-based CE.

3.1 MMSE CE

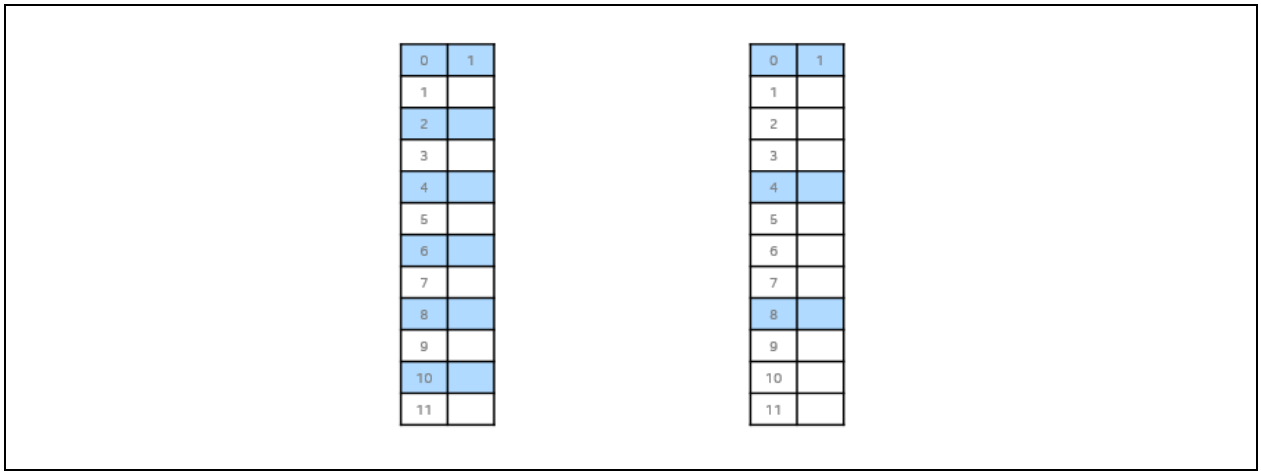
This document describes the receiver processing of SRS, including channel estimation and SNR measurement algorithms used in FlexRAN projects. The overall receiver process is illustrated in Figure 22.

Figure 22. Illustration of SRS Receiver



There are two types of transmission comb level for SRS, and the type is defined by high layer parameter, K_{TC}, which is equal to 2 or 4. An example of a mapping pattern within an RB in the frequency domain is illustrated in Figure 23 with the subcarrier starting from 0. The left part is applied when $K_{TC} = 2$, and the right part is applied when
$K_{TC} = 4$.

Figure 23. Example of SRS Transmission Comb Level Mapping Pattern



The related parameters are also different per the comb level. The maximum number of cyclic shifts is $n_{SRS}^{cs,max} = 12$ if $K_{TC} = 4$ and $n_{SRS}^{cs,max} = 8$ if $K_{TC} = 2$.

3.1.1 LS

This section describes the Least Square (LS) method to get the rough estimated channel response. LS needs to perform with all used cyclic shifts for each receive antenna in MMSE CE.

3.1.1.1 Input

$Y_{SRS}(k,l,r)$	Received SRS signal on subcarrier k, symbol l and Rx antenna r
$X_{SRS}(k,l,p)$	Transmitted SRS sequence on subcarrier k, symbol l, and port index p
N_u	Number of users

3.1.1.2 Output

$\widetilde{H}_{SRS}(k,l,r,p)$	Estimated channel response after LS

3.1.1.3 Algorithms

The $X_{SRS}(k,l,p)$ can be generated in local as defined in section 6.4.1.4.2 in *3GPP TS38.211*. α_i (refer to Table 2) is one of the most important parameters in the generation of $X_{SRS}(k,l,p)$ and is defined to differentiate users and ports, as shown in the below equations:

$$\alpha_i = 2\pi \frac{n_{SRS}^{cs,i}}{n_{SRS}^{cs,max}}$$

$$n_{SRS}^{cs,i} = \left(n_{SRS}^{cs} + \frac{n_{SRS}^{cs,max}(p_i - 1000)}{N_{ap}^{SRS}}\right) mod\ n_{SRS}^{cs,max}$$

$$n_{SRS}^{cs} \in \{0,1,\cdots,n_{SRS}^{cs,max} - 1\}$$

n_{SRS}^{cs} is user-specific and is contained in a high layer message. p_i is the port index, and the cyclic shift for different ports are the most orthogonal ones, which can be derived from the above equation. For clarity of this document, the specific definition for other parameters can refer to chapter 6.4.1.4.2 in *3GPP TS38.211* (refer to Table 2).

The first step of channel estimation is to get a rough channel $\widetilde{H}'_{SRS}(k,l,r,p)$ via LS.

$$\widetilde{H}'_{SRS}(k,l,r,p) = Y_{SRS}(k,l,r) \cdot \mathrm{conj}(X_{SRS}(k,l,p))$$

The transmitted α_i can be removed through this step; however, the interference from other layers and users are still present in $\widetilde{H}'_{SRS}(k,l,r,p)$. To remove the interference as well as to keep the highest accuracy of the estimated channel, averaging $\widetilde{H}'_{SRS}(k,l,r,p)$ in every N_m subcarriers are necessary due to the propriety of the Zadoff-Chu sequence.

$$\widetilde{H}_{SRS}(k,l,r,p) = \frac{1}{N_m} \sum_{every N_m\ SC} \widetilde{H}'_{SRS}(k,l,r,p)$$

Where $N_m = N_{ap}^{SRS} \cdot 2^{\lceil \log_2 N_u \rceil}$, and N_m cannot be larger than $n_{SRS}^{cs,max}$. This algorithm applies under the condition that the scheduler should schedule the most orthogonal n_{SRS}^{cs} for different users, which is generally considered as the basic function for the MAC scheduler. For example, if two users are scheduled with 1-layer-SRS, the scheduler allocates n_{SRS}^{cs} for one user, and it should allocate $(n_{SRS}^{cs} + n_{SRS}^{cs,max}/2)\ mod\ n_{SRS}^{cs,max}$ for the other user.

3.1.2 MMSE Filter

This section describes SRS channel estimation by using Minimum Mean Square Error (MMSE) method to get the estimated channel response.

3.1.2.1 Input

SNR_0	Default initial SNR, for example, 30 dB
$Y_{SRS}(k,l,r)$	Received SRS signal on subcarrier k, symbol l and Rx antenna r
$X_{SRS}(k,l,p)$	Transmitted SRS sequence on subcarrier k, symbol l, and port index p
$\widetilde{H}_{SRS}(k,l,r,p)$	Estimated channel response after LS

3.1.2.2 Output

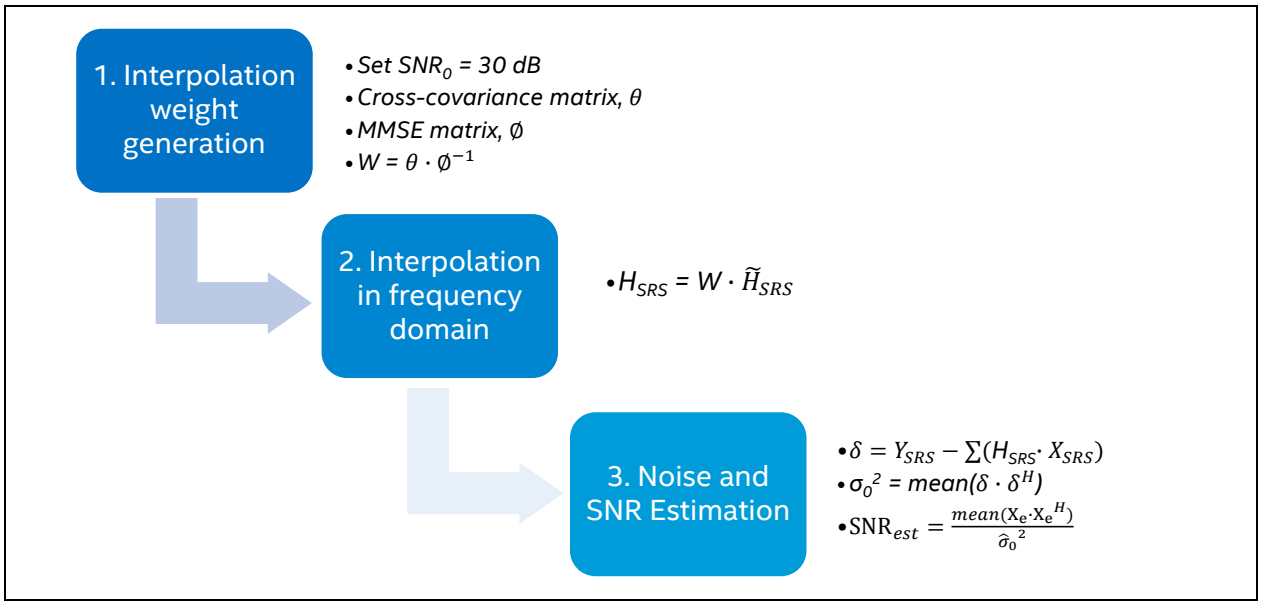
σ_u^2	The estimated noise power of user u
SNR_{est}	Estimated SNR value
$H_{SRS}(k,l,r,p)$	The estimated channel of SRS symbol

3.1.2.3 Algorithms

MMSE filter is used here for frequency domain interpolation and noise filtering. The algorithm is processed in the RB group unit, default as 4RBs.

The whole step conducts this MMSE filter two times. For the first time, the input SNR is 30 dB, and it follows the process of three steps illustrated in Figure 24 to estimate the real SNR value. For the second time, the input SNR is the estimated one in the first MMSE filter, and then the final SRS channel is obtained.

Figure 24. MMSE Filter Illustration



1. Interpolation Weight Generation:

   The interpolation weight is included in two parts: cross-covariance matrix θ and MMSE matrix Φ. Their relation is as

$$w(k,l;k',l') = \theta(k,l) \cdot \Phi^{-1}(k',l')$$

   a. The cross-covariance matrix can be calculated by the Bessel function and Sinc function. The general expression is as below:

   - Bessel function:

$$\theta(k,l) = \frac{J_0(2\pi f_{D_{max}} T_s(l-l'))}{1 + i \cdot 2\pi\tau_{max}\Delta F(k-k')}$$

   - Sinc function:

$$\theta(k,l) = sinc(2f_{D_{max}} T_s(l-l')) \cdot sinc(2\tau_{max}\Delta F(k-k'))$$

     The expression above includes both frequency domain interpolation and time-domain interpolation. For the SRS channel estimation, only frequency domain interpolation is used. Therefore, the interpolation expression is simplified as

   - Bessel function:

$$\theta(k) = \frac{1}{1 + i \cdot 2\pi\tau_{max}\Delta F(k-k')}$$

   - Sinc function (default adopted in the project):

$$\theta(k) = sinc(2\tau_{max}\Delta F(k-k'))$$

     Where τ_{max} is the maximum time delay spread, and ΔF is the subcarrier spacing, k is the subcarrier index of the whole allocated bandwidth (including the non-SRS position), k' is the subcarrier index of SRS.

   b. The MMSE matrix includes auto-covariance matrix and noise:

$$\Phi(k') = \frac{1}{SNR_0} \cdot I + \theta(k')$$

      Therefore, the interpolation weight is equal to

$$w(k;k') = \theta(k) \cdot \Phi^{-1}(k')$$

2. Interpolation in Frequency Domain

   Interpolation in the frequency domain is to multiply the interpolation weight matrix with the estimated channel after LS. Now we have two interpolation methods in the frequency domain.

   - RE level interpolation:

$$H_{SRS}(k,l,r,p) = w^T(k;k') \cdot \widetilde{H}_{SRS}(k',l,r,p)$$

   - RB level interpolation:

     To reduce the calculation of interpolation in the frequency domain, we simplify the above formula.

     First, averaging w for each RB, the averaged interpolation weight is equal to

$$w(k: k^{rb}) = \frac{1}{ScNum} \sum_{every\ SC/RB} w(k: k')$$

Then averaging $\tilde{H}_{SRS}$ for each RB, the averaged channel estimation results are equal to

$$\tilde{H}_{SRS}(k^{rb}, l, r, p) = \frac{1}{ScNum} \sum_{every\ SC/RB} \tilde{H}_{SRS}(k', l, r, p)$$

The RB level interpolation in the frequency domain is shown below:

$$H_{SRS}(k, l, r, p) = w^T(k; k^{rb}) \cdot \tilde{H}_{SRS}(k^{rb}, l, r, p)$$

3. Noise Estimation

$$\delta(k, l, r) = Y_{SRS}(k, l, r) - \sum_p X_{SRS}(k, l, p) \times H_{SRS}(k, l, r, p),$$

To calculate the noise, the receiver needs to remove all the users' information.

Then the noise power is derived by:

$$\hat{\sigma}_0^{\,2} = mean(\delta \cdot \delta^H)$$

where $mean(\,)$ is overall the receiving antennas and subcarriers which reference signals are allocated. And then the estimated signal power is derived by:

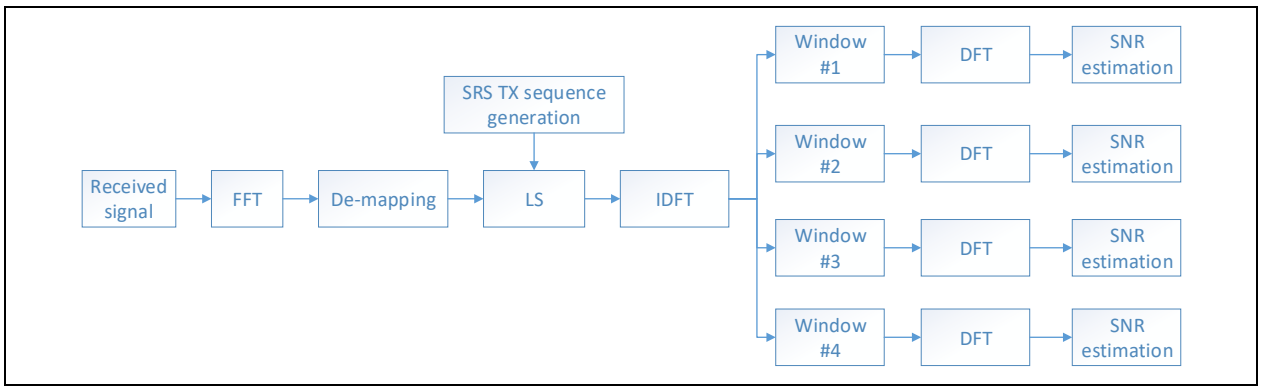
$$X_e = \sum_p X_{SRS}(k, l, p) \times H_{SRS}(k, l, r, p)$$

$$\mathrm{SNR}_{est} = \frac{mean(\mathrm{X_e} \cdot \mathrm{X_e}^H)}{\hat{\sigma}_0^{\,2}}$$

3.2 DFT based CE

The process of the DFT based CE shows in the Figure 22.

Figure 22 Illustration of SRS DFT based CE Receiver



3.2.1 LS

This section describes the LS method to get the rough estimated channel response with DFT-based CE. It only needs to perform once for each receive antenna by using the base sequence of cyclic shift 0.

3.2.1.1 Input

$Y_{SRS}(k,l,r)$	Received SRS signal on subcarrier *k*, symbol *l* and Rx antenna *r*
$X_{SRS}(k,l,0)$	Transmitted SRS sequence on subcarrier *k*, symbol *l*, and cyclic shift index 0

3.2.1.2 Output

$\widetilde{H}_{LS}(k,l,r)$	Estimated channel response after LS on subcarrier *k*, symbol *l* and Rx antenna *r*

3.2.1.3 Algorithms

The first step of channel estimation is to get a rough channel $\widetilde{H}_{LS}(k,l,r)$ via LS.

$$\widetilde{H}_{LS}(k,l,r) = Y_{SRS}(k,l,r) \cdot \mathrm{conj}\big(X_{SRS}(k,l,0)\big) = \sum_{i=0}^{P} \widetilde{H}(k,l,r,i) \cdot X_{SRS}(k,l,i) \cdot conj(X_{SRS}(k,l,0))$$

$\widetilde{H}(k,l,r,i)$ is the channel frequency response for subcarrier *k*, symbol *l*, Rx antenna *r* and cyclic shift i. $P = N_{ap}^{SRS} \cdot n_{SRS}^{cs}$. As $r_{u,v}^{(\alpha,\delta)}(n) = e^{j\alpha n}\bar{r}_{u,v}(n), \quad 0 \le n < M_{ZC}$ and $\alpha_i = 2\pi \frac{n_{SRS}^{cs,i}}{n_{SRS}^{cs,max}}$, then

$$\widetilde{H}_{LS}(k,l,r) = \sum_{i=0}^{P} (\widetilde{H}(k,l,r,i) \cdot e^{j2\pi \frac{n_{SRS}^{cs,i}}{n_{SRS}^{cs,max}} k})$$

3.2.2 IDFT

This section describe the IDFT processing in SRS channel estimation. IDFT will transform the channel response from frequency domain to time domain.

3.2.2.1 Input

$\widetilde{H}_{LS}(k,l,r)$	Channel frequency response on subcarrier *k*, symbol *l* and Rx antenna *r*

3.2.2.2 Output

$\tilde{h}(n,l,r)$	Discrete time domain CIR on sample n, symbol l, Rx antenna r

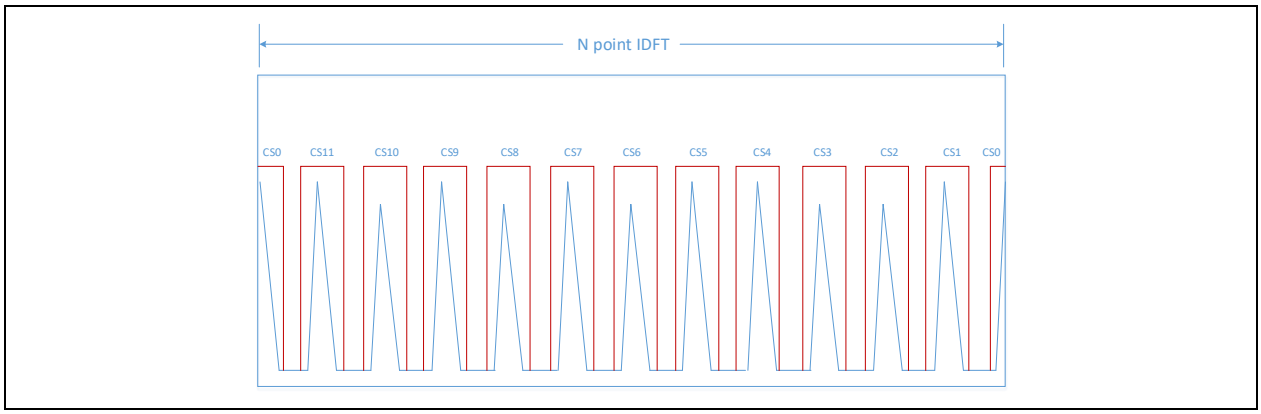
3.2.2.3 Algorithm

$$\tilde{h}(n,l,r) = IDFT\left\{\sum_{i=0}^{P} (\widetilde{H}(k,l,r,i) \cdot e^{j2\pi \frac{n_{SRS}^{cs,i}}{n_{SRS}^{cs,max}} k})\right\} = \sum_{i=0}^{P} \tilde{h}((n + \frac{N_{IDFT}}{n_{SRS}^{cs,max}} \cdot n_{SRS}^{cs,i}), l, r, i)$$

$$N_{IDFT} = 12/K_{TC} \cdot N_{RB}$$

$\tilde{h}(n,l,r,i)$ is the time domain CIR on sample n, symbol l, Rx antenna r and cyclic shift i. The formula shows the $\tilde{h}(n,l,r,i)$ for cyclic shift i has $\frac{N_{IDFT}}{n_{SRS}^{cs,max}} \cdot n_{SRS}^{cs,i}$ sample shift compare with $\tilde{h}(n,l,r,0)$ as Figure 23 shows.

Figure 23 Diagram of time domain CIR for each cyclic shift



The IDFT could use the Radix 2/3/5 to implement. Then the $N_{IDFT} \geq 12/K_{TC} \cdot N_{RB}$, zero padding for $\tilde{H}_{LS}(k,l,r)$ at the end of the sequence to get N_{IDFT} length for input to the IDFT.

3.2.3 UE selection and DFT

This section describes the UE selection method by using DFT-based CE, which include DFT of LS results. UE selection, noise estimation, and IDFT transmission to get the estimated channel response.

3.2.3.1 Input

$\tilde{h}(n,l,r)$	Discrete time domain CIR on sample n, symbol l, Rx antenna r

3.2.3.2 Output

SNR_{est}	Estimated SNR value
$H_{SRS}(k,l,r,i)$	The estimated channel of SRS symbol

3.2.3.3 Algorithm

DFT-based channel estimation exploits the property of OFDM system having the symbol period much longer than the duration of the CIR. Since the estimated CIR from LS has most of its power concentrated on a few first samples, the DFT-based estimation reduces the noise power that exists in only outside of the CIR part.

1. UE selection

   From the Section 3.2.3.3, Algorithm, the center of cyclic shift i should be at

$$N_{center,i} = (N_{IDFT} - \left\lfloor N_{IDFT} \cdot \frac{n_{SRS}^{cs,i}}{n_{SRS}^{cs,max}} \right\rfloor) \; mod \; N_{IDFT}$$

2. Cut-off window define

   A de-noise filter is applied in time domain to reduce noise. The cut-off window for cyclic shift 0 of $win_0(n)$ can be designed by keeping the time domain energy concentration region as useful CIR samples and setting the samples at the noise only region to be zeros.

$$win_0(n) = \begin{cases} 1, & 0 \le n \le f_c/2 - 1, N_{IDFT} - f_c/2 \le n \le N_{IDFT} - 1 \\ 0, & otherwise \end{cases}$$

Where f_c is the "cut-off" point of the time domain filter, which is commonly chosen as CP length if there is no knowledge about channel length.

For cyclic shift i, the center of the win_i should be at the point $N_{center,i}$.

- Window size option 1

  Set f_c according to CP length

  $$f_c = [N_{CP} \cdot K_{TC} \cdot N_{IDFT}/N_{FFT}],$$

  then the maximum cyclic shift number supported by this window limits to $N_{\max\_cs} = N_{IDFT}/f_c = \frac{N_{FFT}}{N_{CP} \cdot K_{TC}}$. For $\mu = 1$, bandwidth 100MHz, $N_{FFT} = 4096, N_{CP} = 288$. When $K_{TC} = 4$, the maximum cyclic shift number is 3.5. When $K_{TC} = 2$, the maximum cyclic shift number is 7.1.

- Window size option 2

  To support more cyclic shift detected, set

  $$f_c = \lceil N_{IDFT}/n_{SRS}^{cs,max} \rceil$$

  From the spec $n_{SRS}^{cs,max} = 12$ if $K_{TC} = 4$ and $n_{SRS}^{cs,max} = 8$ if $K_{TC} = 2$. Then the window size is defined as

  $$f_c = \begin{cases} [N_{CP} \cdot K_{TC} \cdot N_{IDFT}/N_{FFT}], & K_{TC} = 2, and\ n_{SRS}^{cs} \le 4, K_{TC} = 4 \\ \lceil N_{IDFT}/n_{SRS}^{cs,max} \rceil, & n_{SRS}^{cs} > 4, K_{TC} = 4 \end{cases}$$

3. Noise window define

   Estimate the channel length as

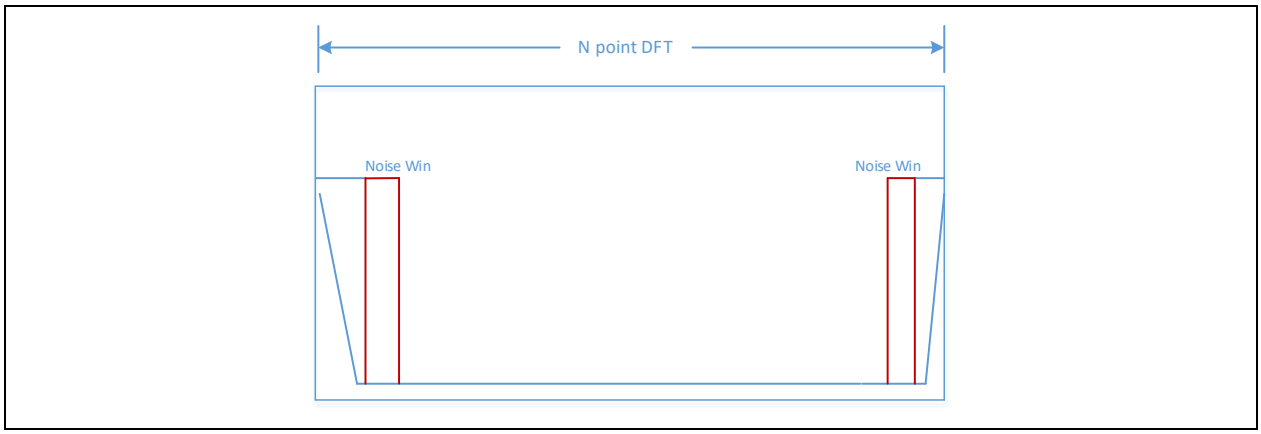
   $$\tilde{f}_L = \left\lceil f_c \cdot \frac{N_{total\_RB} \cdot N_{RB}^{sc}}{N_{FFT}} \right\rceil$$

   The noise window will be in the edge of the cut-off window as

   $$win_{noise,0}(n) = \begin{cases} 1, & \tilde{f}_L/2 \le n \le f_c/2 - 1, N - f_c/2 \le n \le N - \tilde{f}_L/2 \\ 0, & otherwise \end{cases}$$

   The diagram of the noise window for cyclic shift 0 is as Figure 24 shows:

Figure 24 Diagram of noise window for cyclic shift 0



For cyclic shift i, the center of the $win_{noise,i}$ should be at the point $N_{center,i}$.

The noise power in time domain is

$$\sigma_i^2 = E\left[\left|\tilde{h}(n,l,r)\right|^2\right], \quad n \in win_{noise,i}$$

4. Threshold and DFT

   Set threshold in each "cut-off" window for noise elimination, the threshold is at as below. Then the estimated CIR for cyclic shift i should be

$$threshold = 2\sigma_i^2$$

$$\tilde{h}_{SRS}(n,l,r,i) = \begin{cases} \tilde{h}(n,l,r), & \left|\tilde{h}(n,l,r)\right|^2 > threshold, n \in (win_i - win_{noise,i}) \\ 0, & otherwise \end{cases}$$

   Perform the RB number DFT for each cyclic shift to get the RB level channel estimation results.

$$H_{SRS}(k,l,r,i) = DFT\{\tilde{h}_{SRS}(n,l,r,i)\}$$

$$N_{DFT} = N_{RB}$$

5. SNR estimation is calculated in time domain as

$$\mathrm{SNR}_{est} = \frac{\sum_i E\left[\left|\tilde{h}_{SRS}(n,l,r,i)\right|^2\right] - \sum_i \sigma_i^2}{\sum_i \sigma_i^2}$$

3.3 Prefiltering Weight Calculation

This section describes the prefiltering weight calculation based on SRS channel estimated results by using the Zero Forcing (ZF) method or DFT-codebook based method.

3.3.1 Input

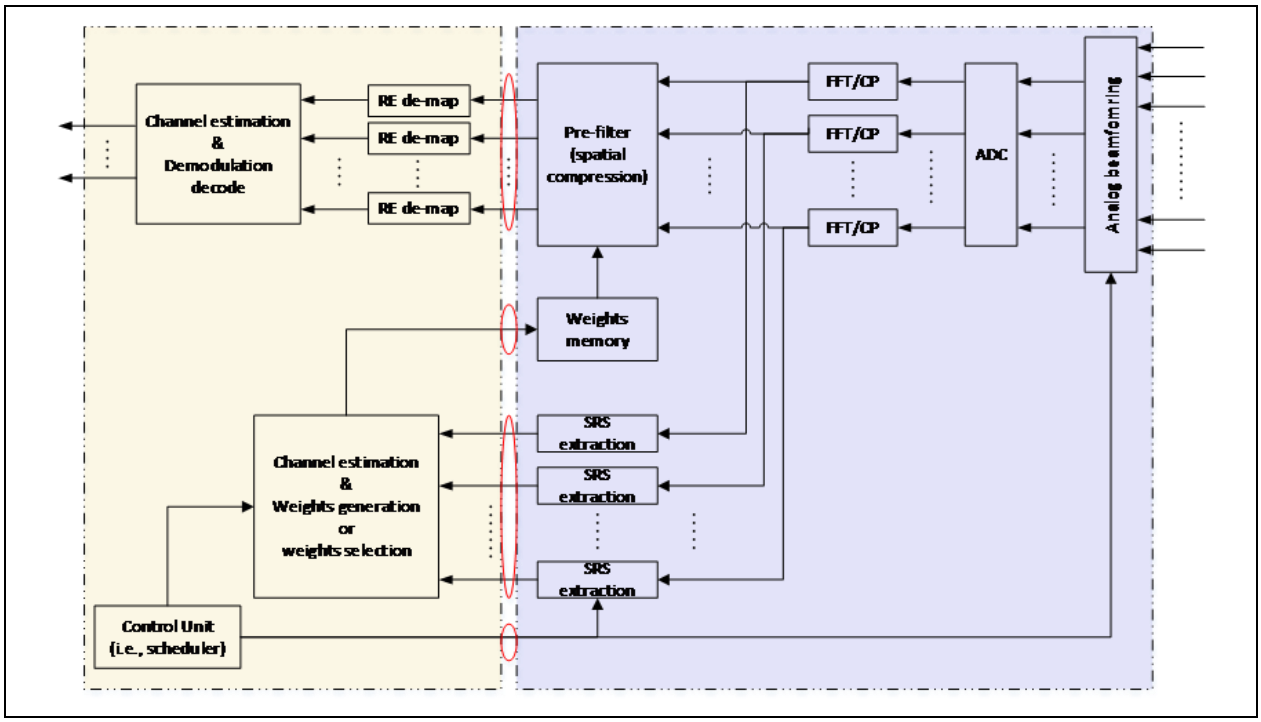
$H_{SRS}(k,l,r,p)$	The estimated channel of SRS symbol
P	Number of the transmission layer
S	Number of the aimed stream

3.3.2 Output

$W_{prefiltering}$	Prefiltering weight for spatial compression

3.3.3 Algorithms

Figure 25. Prefiltering Structure



Spatial compression is adopted in a massive MIMO scenario to lower the processing complexity in baseband and the link burden on front-haul. Figure 25 presents the overall process, and we named the process prefiltering. Zero forcing's are used to calculate the prefiltering weight. The dimension of the signal is compressed from the received antenna number to the transmission layer number.

3.3.3.1 ZF based Prefiltering Weight Calculation

Zero forcing algorithm is used to calculate the prefiltering weight. The dimension of the signal is compressed from the received antenna number to the transmission layer number.

$$W_{prefiltering} = (H_{SRS}^{H} \cdot H_{SRS})^{-1} \cdot H_{SRS}^{H}$$

For ZF prefiltering weight generation method, $S = P$.

3.3.3.2 DFT-codebook based Prefiltering Weight Calculation

The DFT- codebook based prefiltering weight calculation algorithm is deployed for candidate. r_v and r_v denote the number of vertical elements of the receiving antennas and the number of horizontal elements of the receiving antennas, respectively.

The DFT codebook $W_{CB}{}'$ is constructed as below.

$$W_{CB}{}' = W_V \otimes W_H$$

Where W_V denotes the weight matrix for vertical dimension and W_H denotes the weight matrix for horizontal dimension. $\otimes$ denotes Kronecker product.

The codeword $w(l)$ of the lth vector of the vertical weight matrix W_V is expressed as

$$w(l) = \frac{1}{\sqrt{r_v}}\left[1, e^{-j2\pi\cdot 1\cdot l}, e^{-j2\pi\cdot 2\cdot l}, \dots, e^{-j2\pi\cdot(r_v-1)\cdot l}\right]^T, l = 0, 1, 2, \dots, (r_v - 1).$$

Seam as the vertical weight codebook matrix, the codeword $w(l)$ of the lth vector of the horizontal weight matrix W_H is expressed as

$$w(l) = \frac{1}{\sqrt{r_h}}\left[1, e^{-j2\pi\cdot 1\cdot l}, e^{-j2\pi\cdot 2\cdot l}, \dots, e^{-j2\pi\cdot(r_h-1)\cdot l}\right]^T, l = 0, 1, 2, \dots, (r_h - 1).$$

For single polarized receiving antennas, we got the final DFT-based codebook $W_{CB} = W_{CB}{}'$. The codewords denoted as $w(i), i = 0,1,2, \dots, (r_v \cdot r_h - 1)$.

In addition, for dual-polarized receiving antennas, the final DFT-based codebook W_{CB} is expanded as below.

$$W_{CB} = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} W_{CB}{}' & \mathbf{0} \\ \mathbf{0} & W_{CB}{}' \end{bmatrix}$$

The codewords denoted as $w(i), i = 0,1,2, \dots, (2 \cdot r_v \cdot r_h - 1)$.

The S number of codewords $w(s)$ of the prefiltering weight matrix are picked from the codebook W_{CB}, where

$$\{w(s)\} = \underset{i=0,1,\dots,I}{\operatorname{argmax}}(w(i) \cdot H_{SRS} \cdot H_{SRS}^H \cdot w(i)^H)$$

We have the prefiltering weight matrix $W_{prefiltering} = \begin{bmatrix} w(0)^T \\ \dots \\ w(S-1)^T \end{bmatrix}$. For DFT-codebook based prefiltering weight generation method, the stream number S can be bigger than the transmission layer number P.

3.4 DL Beamforming Weight Calculation

This section describes the DL beamforming weight calculation based on SRS channel estimated results by using the Zero Forcing (ZF) method.

3.4.1 Input

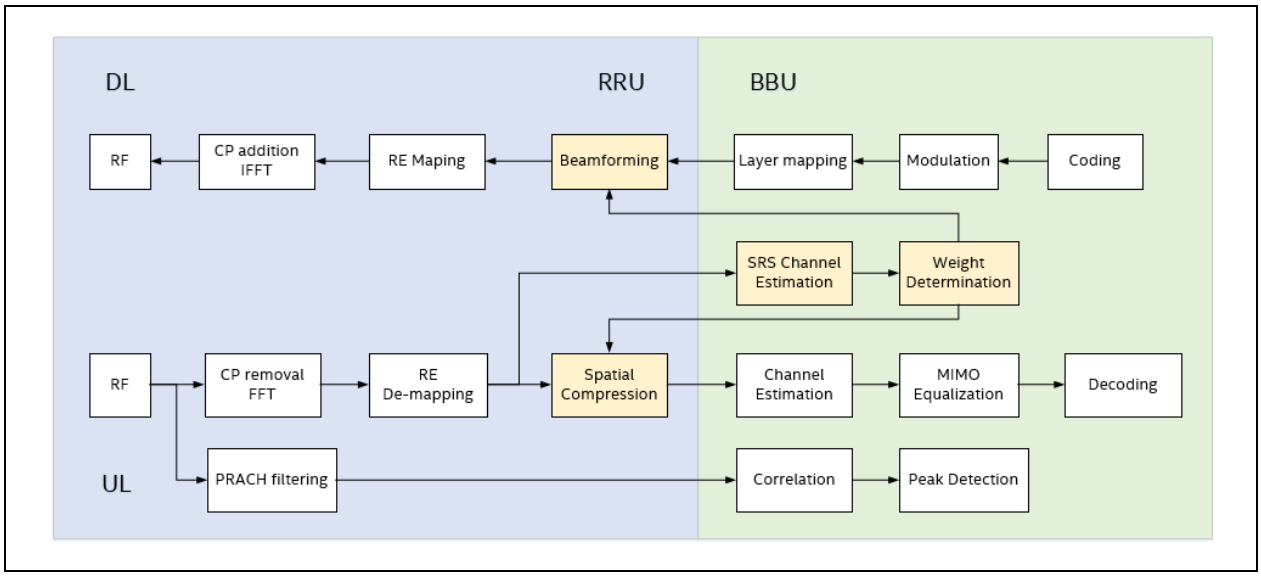
$H(k,l,t,p)$	The estimated channel of SRS symbols for scheduled Users
P	Number of the transmission layer

3.4.2 Output

$W_{beamforming}$	Beamforming weight for spatial compression

3.4.3 Algorithms

Figure 26. Massive MIMO Processing Flow



DL beamforming is adopted in a massive MIMO scenario, the channel reciprocity is used to get the DL channel information from SRS channel results. And the beamforming weight matrix will be updated when the scheduled users are changed, or the channel response matrix is changed.

In current FlexRAN massive MIMO solution, Zero Forcing (ZF) algorithm is used to calculate the beamforming weight matrix. The dimension of beamforming process will change from transmission layer number to transmit antenna number.

With ZF beamforming method, the beamforming weight generation can be expressed as

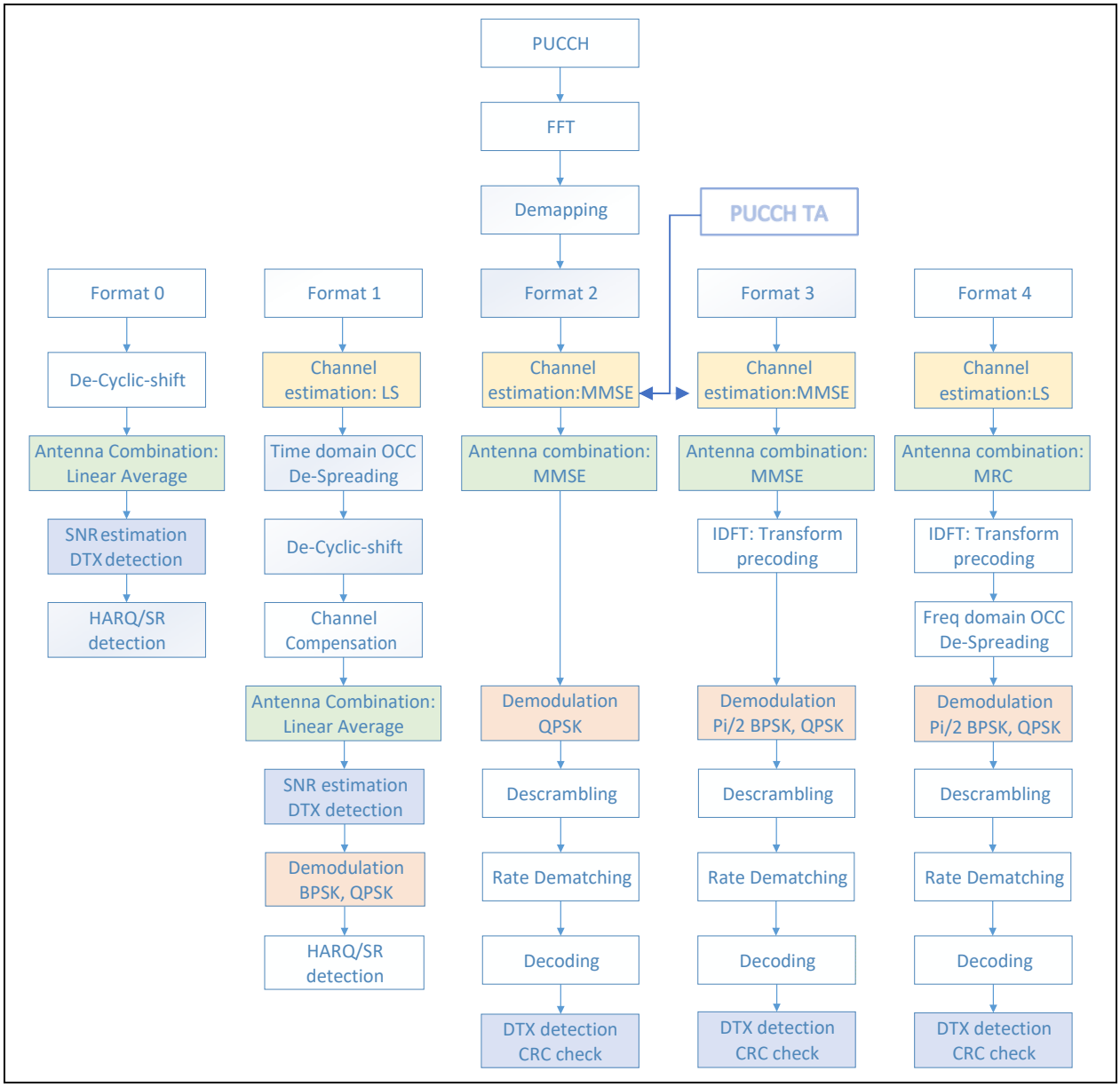
$$W_{beamforming} = H^H \cdot (H \cdot H^H)^{-1}$$

§

4.0 PUCCH Algorithms

This section introduces the Physical Uplink Control Channel (PUCCH) baseband algorithms for FlexRAN, including the overall processing steps, the introduction of key modules, input, output, and detailed algorithms. Figure 22 presents the overall receiver of different PUCCH formats.

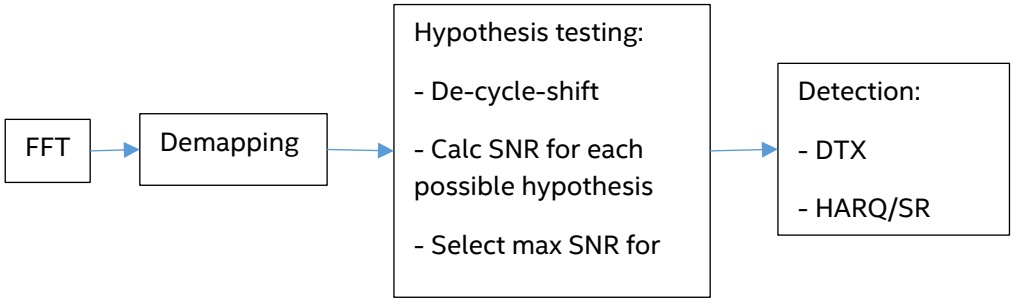
Figure 27. Overview of PUCCH Receiver



4.1 PUCCH F0

The NR PUCCH F0 receiver algorithm processing is illustrated in Figure 28. After FFT processing, the PUCCH data is extracted, and correlation is performed with the local reference sequence. Finally, the ACK/SR is detected.

Figure 28. PUCCH FO Receiver



4.2 De-cyclic Shift

The cyclic shift removal is done via correlation with the local reference sequence. There are three kinds of parameters that co-define the cyclic shift value, as defined in chapter 6.3.2.2.2 in *3GPP TS38.211* (Refer to Table 2).

4.2.1 Input

$Y(k,l,r)$ Received PUCCH signal at subcarrier k, symbol l and received antenna r

$X_i(k,l)$ PUCCH sequence at subcarrier k, symbol l with cyclic shift i

4.2.2 Output

$C_i(r)$ Correlation value for cyclic shift i and received antenna r

4.2.3 Algorithm

$$C_i(r,l) = \frac{1}{K}\sum_{k=0\ldots11} Y(k,l,r)X_i(k,l)^H$$

Where,

Cyclic shift index i

Subcarrier index $k = 0,..K-1$

Symbol index $l = 0,..L-1$

RX antenna index $r = 0,..N-1$

L is the symbol number used in format 0, which varies from 1 to 2.

The de-cyclic shift is divided into two parts, as presented below:

- Correlate with n_{cs} and m_0 sequence, firstly, the following processing is per user detection.

- Correlate with pre-defined m_{cs} set.

If SR only transmission, m_{cs} = 0, the de-cyclic shift is divided into two parts as presented below:

- Correlate with n_{cs} sequence
- Correlate with pre-defined m_0 set.

The second part of the de-cyclic shift is to correlate with a pre-defined set with at least four elements, and the correlated power is used for UCI detection and noise estimation. The least number of an element is an experience value which is derived via simulation to have a good performance of noise estimation and DTX detection.

There are five scenarios in format 0, differed by the number of UCI bit and SR transmission on-off key. For each scenario, the correlation set is explained as below

- for 1-bit-UCI:
  - If SR is scheduled, $\mathrm{m_{cs}}$ set = {0,6,3,9} , where {3, 9} for Positive SR detection and {0, 6} for Negative SR detection
  - If SR is not scheduled, $\mathrm{m_{cs}}$ set = {0,6} &{{5,11}-current UE's $\mathrm{m_0}$} ($\mathrm{m_0}$=5 are reserved for noise estimation)
  - for 2-bit-UCI:
  - If SR is scheduled, $\mathrm{m_{cs}}$ set = {0,3,6,9,1,4,7,10}, where {0,3,6,9} for Positive SR detection and {1,4,7,10} for Negative SR detection
  - If SR is not scheduled, $\mathrm{m_{cs}}$ set= {0,3,6,9}
- for SR only:

  $m_{cs} = 0$, reserve m_0 set ={9, 10, 11} for noise estimation

It can be observed that if the elements in the correlation set are not enough, user-specific parameter m_0 will be reserved to help reach the requirement. Therefore, the MU capability in some scenarios cannot meet its 100% capability. Table 6 summarizes the MU capability for format 0 in our PUCCH receiver.

Table 6. MU Capability for PUCCH Format 0

Scenario	MU Capability in Spec	MU Capability in the Receiver	Notes
SR only	12 UEs	9 UEs	3 cyclic shift reserved
1 bit UCI	6 UEs	5 UEs	2 cyclic shift reserved
2 bits UCI	3 UEs	3 UEs	-
1 bit UCI+SR	3 UEs	3 UEs	-
2 bit UCI+SR	1 UE	1 UE	-

4.3 Antenna Combination

This section describes the algorithm of antenna combination after de-cyclic shift function.

4.3.1 Input

$C_i(r,l)$ Correlation value for cyclic shift i and received antenna r for symbol l

4.3.2 Output

$P_i(r)$	Correlation power for cyclic shift i for received antenna r
$N_i(r)$	Noise power for cyclic shift i for received antenna r

4.3.3 Algorithm

For Hypothesis of sequence i being transmitted, we calculate the signal power, $P_i(r)$, for that sequence per RX antenna, r, as follows. If frequency hop is disabled, given that we are processing adjacent symbols in time direction in the same slot, we can assume the channel to be very close to each other for same subcarrier locations for the same RX antenna. Hence, we can do a coherent sum between symbols per sub-carrier basis.

$$P_i(r) = \left| \frac{1}{L} \sum_{l=0 \ldots L-1} C_i(r,l) \right|^2$$

If frequency hope is enabled, we can only do incoherent power sum between symbols

$$P_i(r) = \frac{1}{L} \sum\nolimits_{l=0:L-1} |C_i(r,l)|^2$$

In case of single symbol (i.e. L=1), either of the above equations reduce to the correct expression for $P_i(r)$.

The set, S, of sequence indices, i , is defined to have two subsets,

S_H: Subset of indices corresponds to the hypothesis we want to test (i.e. possible sequences used by UE)

S_N: Subset of indices dedicated for noise power estimation

$$S = \{S_H, S_N\}$$

$C_i(r,l)$ are calculated for the full set of sequence in set S.

$P_i(r)$ are calculated only for the sequences in the hypothesis set, S_H

For calculating the noise power, we have two modes:

- If the parameter PF0_CommonNoiseVarForAllRxAnt is set to 1 indicating a common noise power across all antennas, we **average** the remaining $|C_j(r,l)|^2$ across all antennas and symbols to get common noise power.

$$N_i(r) = N_i = \frac{1}{L \times (I-1) \times N} \sum_{r=0:N-1:} \sum_{\{j\} \setminus i} \sum_{l=0:L-1} |C_j(r,l)|^2$$

  Where the sum over sequence index j is done excluding hypothesis sequence index i. And I is the cardinality of set S

- If the parameter PF0_CommonNoiseVarForAllRxAnt is set to 0 indicating per RX Antenna noise power estimation, we average the remaining $|C_j(r,l)|^2$ across all symbols and sequences to get pre RX antenna noise power.

$$N_i(r) = \frac{1}{L\times(l-1)}\sum_{\{j\}\setminus i}\sum_{l=0:L-1}\left|C_j(r,l)\right|^2 \qquad (11)\text{-B}$$

Where the sum over sequence index j is done excluding hypothesis sequence index i.

4.4 SNR Estimation and DTX/ACK/SR Detection

4.4.1 Input

$P_i\,(r)$	Correlation power for cyclic shift i for received antenna r
$N_i\,(r)$	Noise power for cyclic shift i for received antenna r

4.4.2 Output

SNR	Estimated SNR value
DTX	If this is DTX transmission
$\hat{A}$	Detected UCI bit
SR	If this is SR transmission and negative/positive SR

4.4.3 Algorithm

The process PNR Estimation and DTX/ACK/SR Detection are interconnected, and therefore the description is combined.

- PNR for hypothesis i is given by,

$$PNR_i = \sum_{r=0:N-1}\frac{P_i(r)}{N_i(r)}$$

- Sequence giving the highest PNR is taken as the most likely transmitted sequence.

$$PNR_{i\_max} = \max\{PNR_i | i \in S_H\}$$

- The PNR_{i_max} value is compared with the pre-defined DTX threshold for DTX detection

  if PNR_{i_max} > threshold

  not DTX signal

  The ACK/SR value can be detected via the index, i_max, corresponding to the largest PNR

  else

  DTX signal

  end

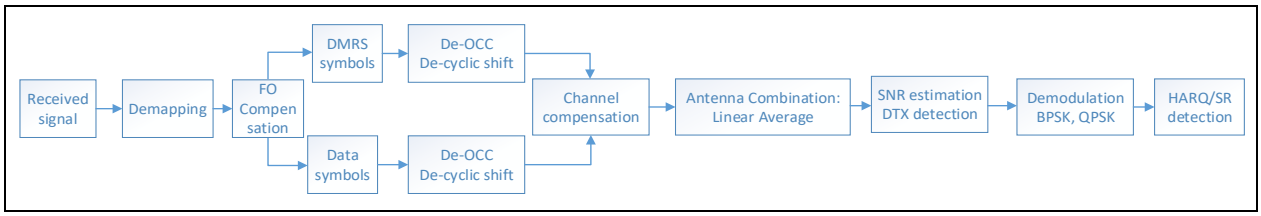
The estimated noise power for the reserved cyclic shift is used for each RB.

The DTX threshold is a statistical value via several simulations. Its value differs per received antenna number, symbol number, and if intra-slot frequency hopping is enabled.

4.5 PUCCH F1

The NR PUCCH F1 receiver algorithm processing is illustrated in Figure 28. After FFT processing and carrier frequency offset compensation, the PUCCH data is extracted, and channel estimation is performed with the local reference sequence. The data part is de-spread and compensated with the channel estimation. Finally, the ACK/NACK is detected. Format 1 can support user multiplexing via OCC and CS.

Figure 29. PUCCH F1 Processing Flow



4.6 Carrier Frequency Offset Compensation

This section describes the algorithm for frequency offsets estimation compensation. After Demapping, all received data will optional do the carrier frequency offset compensation, based on the availability of the averaged estimated angle offset, which calculated by PUSCH Frequency Offset Estimation with Multiple DM-RS Symbols Module. This module is also valid with PUCCH F3.

4.6.1 Input

θ The averaged estimated angle offset (From PUSCH multiple DM-RS symbol estimation)

$Y(k,l,r)$ Received signal on subcarrier k, symbol l and Rx antenna r

N_{fft} FFT size

4.6.2 Output

$Y'(k,l,r)$ Compensated received signal on subcarrier k, symbol l, and Rx antenna r

4.6.3 Algorithm

1. Calculate the specific offset times of signal $Y(k,l,r)$, symbol number l starts from 1.

$$N_{times} = \left((l-1) * \left(N_{fft} + CPSize\right) + CPSize1\right) + k$$

2. Compensate the received signal:

$$Y'(k,l,r) = Y(k,l,r) * e^{\left(-j*\left(\frac{\theta}{N_{fft}}\right)*N_{times}\right)}$$

4.7 De-OCC and De-cyclic Shift

After Carrier frequency offset compensation, DM-RS signals and data signals are extracted, and they are conducted de-OCC and de-cyclic shift separately. The process has no correlation with the local reference sequence.

4.7.1 Input

$Y^{DMRS}(k,l,r)$	PUCCH DM-RS received signals at subcarrier k, symbol l and received antenna r
$Y^{Data}(k,l,r)$	PUCCH data received signals at subcarrier k, symbol l and received antenna r
$X_i(k,l)$	PUCCH TX sequence at subcarrier k, symbol l

4.7.2 Output

$C_i^{DMRS}(r)$	A correlation value of DM-RS signals for cyclic shift i and received antenna r
$C_i^{Data}(r)$	A correlation value of data signals for cyclic shift i and received antenna r

4.7.3 Algorithm

$$C_i^{DMRS}(r) = \frac{1}{K*L}\sum_{k=0\ldots K;l=1\ldots L} Y^{DMRS}(k,l,r)\ X_i(k,l)^H$$

$$C_i^{Data}(r) = \frac{1}{K*L}\sum_{k=0\ldots K;l=1\ldots L} Y^{Data}(k,l,r)\ X_i(k,l)^H$$

where $X_i(k,l) = r_{u,v}^{\alpha,\delta}(k)w_i(l)$.

The processes of de-OCC and de-cyclic shifts are combined. When conducting de-cyclic shift, another three cyclic shifts m_0 = 9, 10, 11 are reserved for noise estimation. Therefore, this process needs to run another three times for each RB, no matter how many users are multiplexed.

4.8 Channel Compensation

After de-OCC and de-cyclic shift, modulated data symbol can be obtained via channel compensation for each RX antenna.

4.8.1 Input

$C_i^{DMRS}(r)$	A correlation value of DM-RS signals for cyclic shift i and received antenna r
$C_i^{Data}(r)$	A correlation value of data signals for cyclic shift i and received antenna r

4.8.2 Output

$d_i(r)$	Compensated data symbol for cyclic shift i and received antenna r

4.8.3 Algorithm

$$d_i(r) = C_i^{Data}(r) * conj(C_i^{DMRS}(r))$$

After this step, the modulated data symbol can be obtained.

4.9 Antenna Combination

Combine data symbol after channel compensation from each RX antenna.

4.9.1 Input

$d_i(r)$ Compensated data symbol for cyclic shift i and received antenna r

4.9.2 Output

d_i' Compensated data symbol for cyclic shift i for combined antennas

4.9.3 Algorithm

The linear average is adopted here for antenna combination.

$$d_i' = \frac{1}{N}\sum_{r=1\ldots N} d_i(r)$$

4.10 SNR Estimation and DTX/ACK/SR Detection

Based on combined data symbols, SNR estimation can be done by power estimation. The DTX detector can conclude from SNR. ACK/SR value can be detected via demodulating data symbols.

4.10.1 Input

d_i' Compensated data symbol for cyclic shift i for combined antennas

4.10.2 Output

SNR Estimated SNR value

DTX If this is DTX transmission

$\hat{A}$ Detected UCI bits

SR If this is SR transmission and negative/positive SR

4.10.3 Algorithm

The processes SNR Estimation and DTX/ACK/SR Detection are interconnected, and therefore the description is combined.

1. The power of the compensated data symbol is calculated as $P_i = \|d_i'\|^2$
2. The power corresponding to the user-specific m_0 is considered as the signal power; the averaged power corresponding to m_0 = 9, 10, 11 is considered as noise power. Therefore, the linear SNR is calculated as $= signal\ power/noise\ power$.
3. The estimated SNR value is compared with the pre-calculated DTX threshold to determine if this is DTX transmission. The DTX threshold is a statistical value via several simulations. Its value differs per received antenna number, symbol number, and if intra-slot frequency hopping is enabled.

if SNR > threshold

not DTX signal

else

DTX signal

end

If this is not DTX transmission, the ACK/SR value can be detected via demodulating d_i'.

If 1 bit UCI is transmitted:

$$b_0 = \begin{cases} 1 & if\ (re(d_i') < 0) \&\& \ (im(d_i') < 0) \\ 0 & otherwise \end{cases}$$

If 2 bits UCI are transmitted:

$$e_0 = re(d_i')$$

$$e_1 = im(d_i')$$

$$b_i = \begin{cases} 1 & if\ (e_j < 0) \\ 0 & otherwise \end{cases} \qquad for\ j = 0,1$$

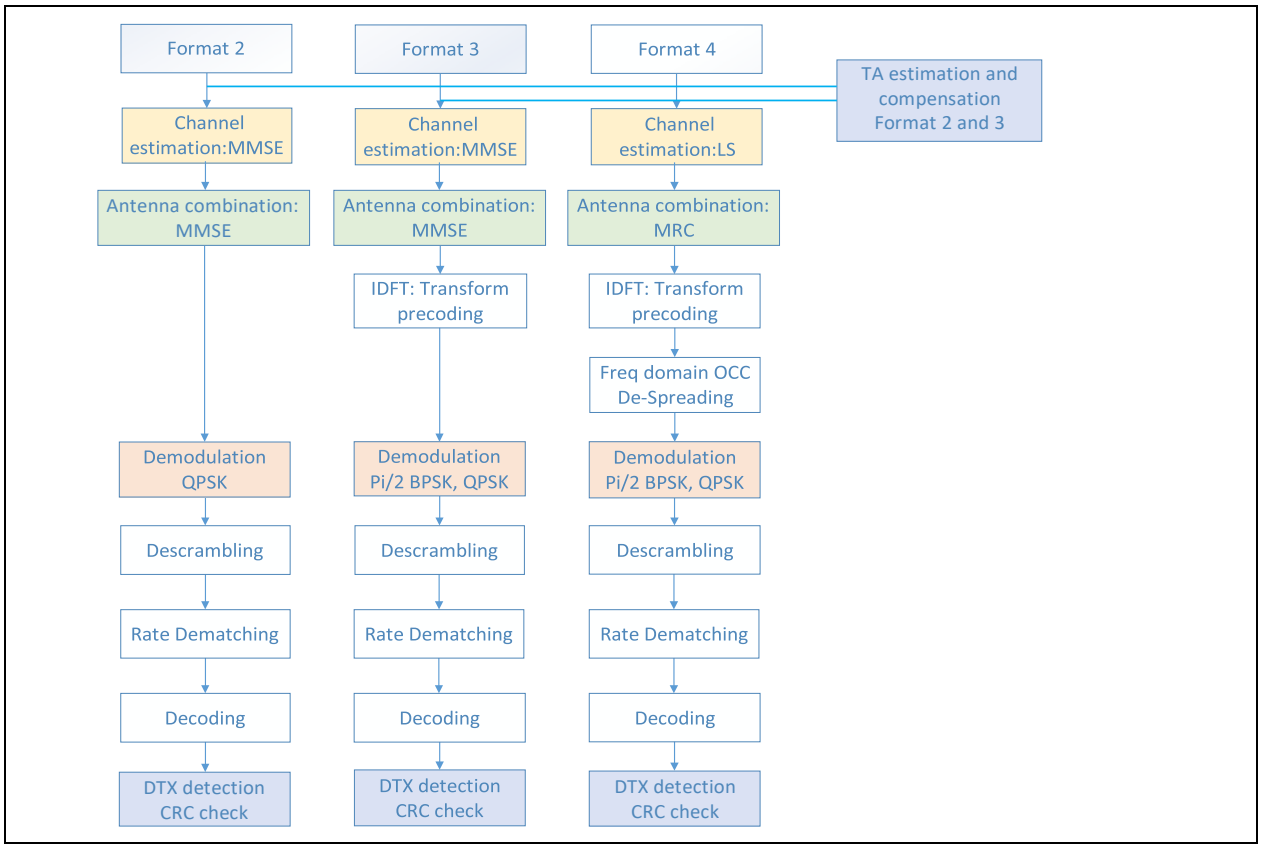
Here $re(d_i')$ and $im(d_i')$ are the real part and imaginary part of d_i'respectively.

4.11 PUCCH F2, F3, F4

The NR PUCCH receiver algorithm processing of formats 2, 3, and 4 are illustrated in Figure 29. It can be noticed that the majority of these three format processes are similar. Therefore, these three formats are described together to facilitate the comparison of common places and different places.

Format 2 is designed for large UCI bits transmission and not considered for low PAPR case. Therefore, the DM-RS sequence is the PN sequence. While for format 3 and format 4, transform precoding is designed, and these two formats are suitable for low PAPR. Format 4 adds OCC to enable user multiplexing compared with format 3.

Figure 30. PUCCH Processing Flow for F2, F3, and, F4



4.12 Channel Estimation

The PUCCH format 2/3/4 channel estimation is MMSE based in frequency domain. DM-RS definitions of format 2 and format 3/4 are not the same, so channel estimation will fit to each DM-RS definition.

4.12.1 Input

$Y(k,l,r)$	Received PUCCH signal at subcarrier k, symbol l and received antenna r
$X(k,l)$	DM-RS sequence at subcarrier k, symbol l

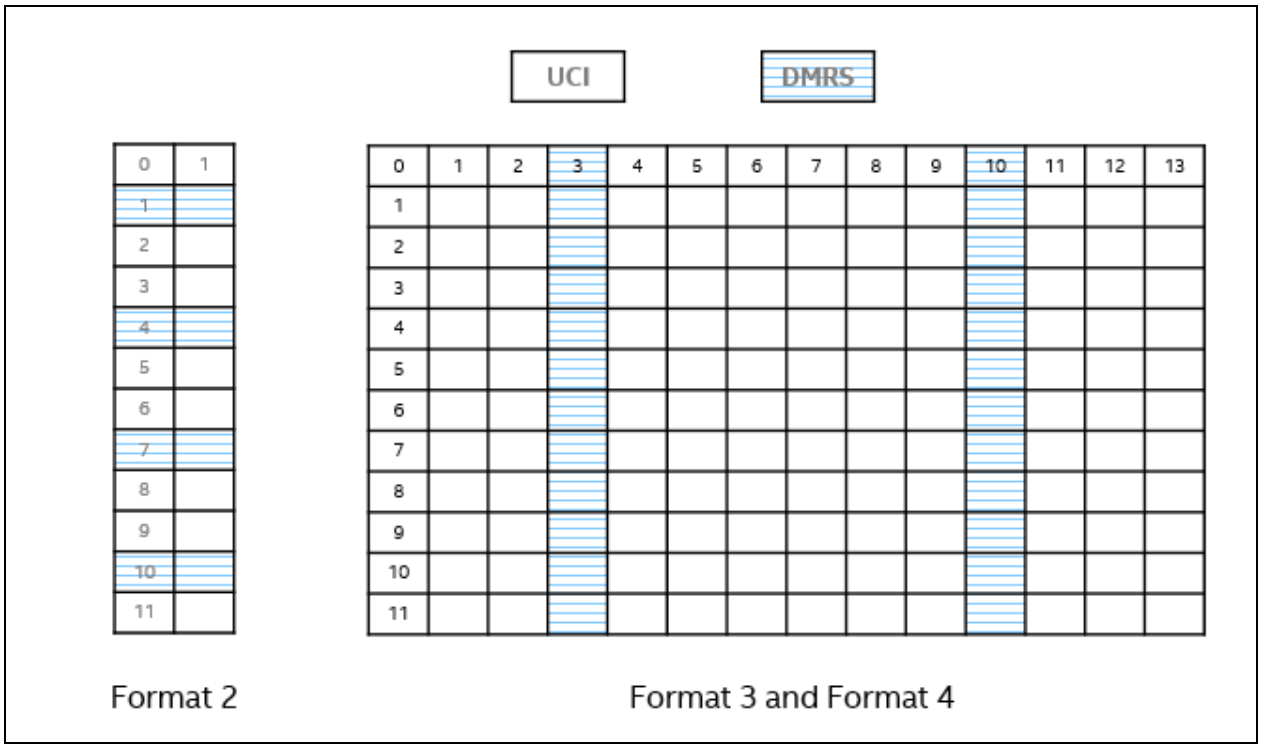
4.12.2 Output

$H(k,l,r)$	Estimated channel response at subcarrier k, symbol l and received antenna r
σ^2	The estimated noise power (only apply for format 2 and format 3)

4.12.3 Algorithm

LS + MMSE is used for channel estimation of format 2 and 30. LS is applied for format 4. The algorithm of LS and MMSE is the same with that part of PUSCH, referring to Chapter 2.6.2, LS and De-spread Processing and Chapter 2.6.3., MMSE-Based Channel Estimation.

Figure 31. Resource Mapping Example for Format 2,3, and, 4



As shown in Figure 31, the resource mapping for formats 2, 3, and 4 are different, and therefore the channel estimation parameter configuration is different.

Format 2: for frequency domain, MMSE filtering is used for interpolation, and the interpolation block size here is one PRB. No specific operation performed for the time domain.

Format 3: for frequency domain, MMSE filtering is used, and the interpolation block size is one PRB. For the time domain, the nearest linear interpolation is used, which is to copy the nearest channel estimate of the DM-RS symbol to data symbols.

Format 4: for frequency domain, LS is applied, and the estimated channel is averaged per RB to remove other users' interference. For the time domain, the nearest linear interpolation is used as format 3. The equation of LS is:

$$H\ (k,l,r) = \frac{1}{N_{sc_{RB}}} \sum_{k=0}^{N_{sc_{RB}}-1} Y(k,l,r)X(k,l,r)^*$$

Where $N_{sc_{RB}}$ is the subcarrier number per RB, which is equal to 12. $(\ )^*$means the conjugate operation.

4.13 Timing Advance Estimation and Compensation

The timing advance estimation and compensation in PUCCH is provided only for Formats 2 and 3. The same algorithm is used, which is implemented for PUSCH. The difference is in the DM-RS positions and the distance between neighbour DM-RS subcarriers. Therefore, the procedure follows steps described in Chapter 2.11.1, Timing Offset Estimation of this document with the following differences:

1. Phase shift estimation:

$$e_u(r) = \sum_{k=0}^{K-2} \hat{H}_{RS,u}(k,r)\ (\hat{H}_{RS,u}(k+n,r))^H$$

Where k is the DM-RS subcarrier index and $k = 0 : K-1$ and $n = 1$ for Format 3 and n = 3 for Format 2.

2. Convert phase shift to timing offset:

$$TA_{est,u} = \frac{angle(\sum_{r=0}^{R-1} e_u(r)) \cdot N_{FFT}}{2 \cdot \pi \cdot L}$$

where N_{FFT} is FFT size and, L is adjacently calculated DM-RS index offset as illustrated in Figure 30. L differs with the PUCCH format as below:

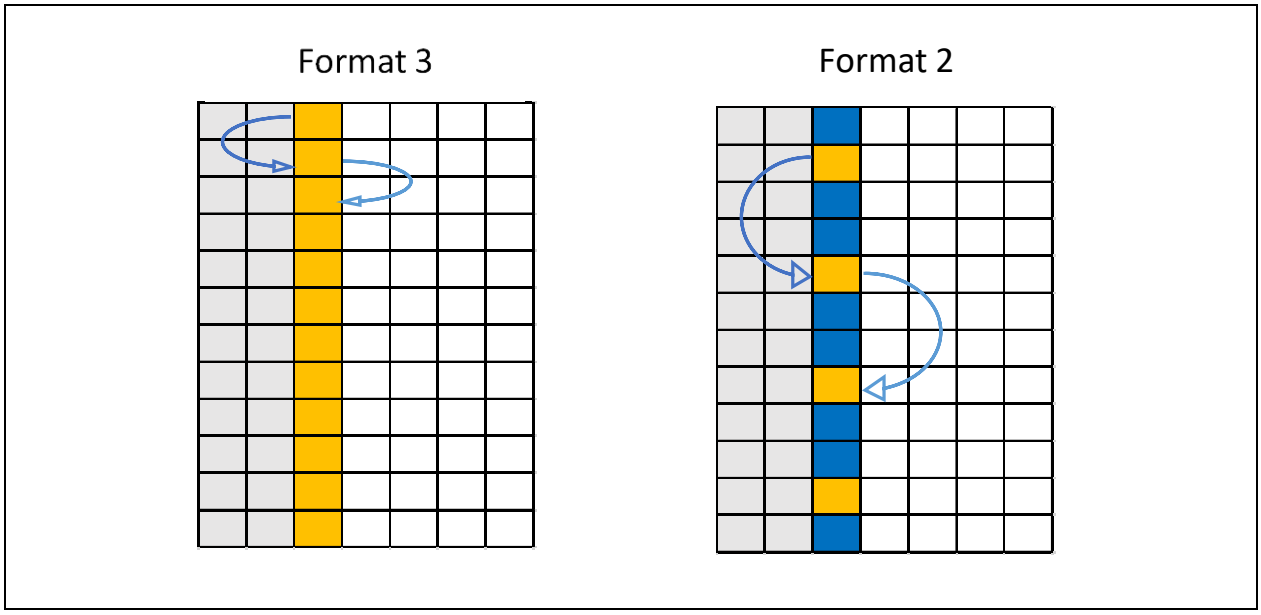
- L = 1 for format 3

  Calculate the rotation between the DM-RS index in RBs: 0,1,2...

- L = 3 for format 2

  Calculate the rotation between DM-RS index in RBs: 1,4,7,10

Figure 32. Rotation Angle for Format 3 and Format 2



The remaining steps follow Sections 2.11.2, Compensation on DM-RS in LS, and 2.11.3, De-compensation on Estimated Channel.

4.14 Antenna Combination

There are two methods, antenna combination or aka equalization. For format 2 and format 3, and the MMSE is used; For format 4, MRC is used.

4.14.1 Input

$Y^{Data}(k,l,r)$ PUCCH data received signals at subcarrier k, symbol l and received antenna r

$H(k,l,r)$	Estimated channel response at subcarrier k, symbol l and received antenna r
σ^2	The estimated noise power (only apply for format 2 and format 3)

4.14.2 Output

$z\,(k,l)$	The equalized data at subcarrier k, symbol l

4.14.3 Algorithm

For format 2 and format 3, MMSE is used for antenna combination. For the detailed equation, refer to Section 2.12.2.32.12.1.3.

For format 4, the maximum ratio combination (MRC) is used for antenna combination. The operation equation is:

$$z\,(k,l) = \frac{\sum_r H(k,l,r)^* Y^{Data}(k,l,r)}{\sum_r \|H(k,l,r)\|}$$

4.15 IDFT

This part only applies to format 3 and 4 for transform precoding.

4.16 De-Spreading

This part only applies to format 4. The block-wise spreading is added to the frequency domain to differentiate users. Two kinds of orthogonal codes are defined in Table 2, TS38.211, 6.3.2.6.3 in [1] to support 2 users and 4 users multiplexing separately.

The de-spreading is done with the local spreading sequence and is averaged to lessen the noise's effect.

4.16.1 Input

$w(k)$	Spreading sequence defined in Table 2, 3GPP TS38.211, Section 6.3.2.6.3 in [1]
$z\,(k,l)$	The equalized data at subcarrier k, symbol l

4.16.2 Output

$z'\,(k,l)$	The sequence after de-spreading

4.16.3 Algorithm

$$z'\,(k,l) = \frac{1}{N_{SF}} \sum_{k=0}^{N_{SF}-1} z\,(k,l)\, w(k)^*$$

Where N_{SF} is equal to 2 or 4 as the orthogonal code type.

4.17 Demodulation

After de-spreading, `softbits` are decoded via demodulation.

4.17.1 Input

$z\,(k,l)$	The equalized data at subcarrier k, symbol l

4.17.2 Output

q	Soft values

4.17.3 Algorithms

In this section, constellation values are converted to soft values representing the logarithmic probability of one and zero for each bit.

The modulation scheme is slightly different among these three formats.

- Format 2: QPSK
- Format 3 and 4: Pi/2 BPSK, QPSK

For pi/2 BPSK, the soft bit can be calculated as:

$$q = \begin{cases} real(z(i)) + imag\big(z(i)\big), & \text{if i is even} \\ -1 * real(z(i)) + imag(z(i)), & \text{if i is odd} \end{cases}$$

For *QPSK*, the soft bit can be calculated as:

$$\begin{bmatrix} q_0 & q_1 \end{bmatrix} = [re\big(z\big) \;\; im(z)]$$

4.18 De-scrambling

The data bit is scrambled at the transmitter side. De-scrambling is done after demodulation at the receiver side.

4.18.1 Input

q	Soft values

4.18.2 Output

$B_{de-scramble}$	Output values after de-scramble

4.18.3 Algorithm

This function is the opposite processing of scrambling in chapter 6.3.2.5.1 in *3GPP TS38.211* (refer to Table 2). The calculation method of scramble sequence c is given by clause 5.2.1 in *3GPP TS38.211* (refer to Table 2). This part is the same for formats 2, 3, and 4.

$$B_{de-scramble} = q \cdot (1 - 2 \cdot c)$$

4.19 Rate De-matching

Polar code is used for the sequence that is larger than 11 bits. Reed-Muller code is used for the sequence that is between 3~11 bits. This part is the same for formats 2, 3, and 4.

4.19.1 Input

$B_{de-scramble}$	De-scrambled output
G	Total number of bits available for the transmission of one transport block
N	Length of de-scrambled bits, which is equal to 32 for RM code and the code block size for polar code.

4.19.2 Output

B_{de-rm}	Rate de-matched values of the code block

4.19.3 Algorithm for RM Code

For the case of code block repetition (G ≥ N), two steps are performed as follows

1. Zero paddings at the end of the sequence.

$$t(i) = \begin{cases} B_{de-scramble}(i), & i = 0, \dots, G-1 \\ 0, & i = G, \dots, G'-1 \end{cases}$$

   Where $G' = \left\lceil \frac{G}{N} \right\rceil N$ and is the sequence size after zero paddings to let the sequence is the integral multiple of the code block size.

2. Average the value for $\frac{N}{G'}$ blocks.

$$B_{de-rm}(i) = \begin{cases} \frac{N}{G'} \sum t(i + jN), & i = 0, \dots, N - \left(G^{'} - G\right) - 1;\ j = 0, \dots, \frac{G'}{N} - 1 \\ \frac{N}{G' - N} \sum t(i + jN), & i = N - \left(G^{'} - G\right), \dots, N-1;\ j = 0, \dots, \frac{G'}{N} - 2 \end{cases}$$

4.19.4 Algorithm for Polar Code

1. Bits de-interleaving

   Triangle de-interleaver is used, whose size is $T * T$. T is the smallest integer such that T $\leq \frac{1}{2}\sqrt{1 + 8 * G}$ instead of interleaving, the $B_{de-scramble}$ is written into the de-interleaver matrix column by column first, and then the bits are read out row by row.

2. Bit selection:

   There are three cases for bit selection: repetition, puncturing, and shortening. The processing methods are different.

   - For the case of repetition, the bit is added together for the corresponding repeated ones.

- For the case of puncturing, the bit sequence is padded with a null sequence to meet the code block size. The null sequence is added before the bit sequence.
- For the case of shortening, the bit sequence is also padded with a null sequence. Different from the puncturing, the null sequence is added after the bit sequence.

3. Sub-block de-interleaving

The coded bits are divided into 32 sub-blocks, and the sub-blocks are interleaved as the interleaver pattern defined in Table 5.4.1.1-1 in [2] (refer to Table 2). Therefore, the output sequence of rate de-matching needs to be de-interleaved by the table. The simplified pattern is as P.

$$P = [0\ 1\ 2\ 4\ 3\ 5\ 6\ 7\ 8\ 16\ 9\ 17\ 10\ 18\ 11\ 19\ 12\ 20\ 13\ 21\ 14\ 22\ 15\ 23\ 24\ 25\ 26\ 28\ 27\ 29\ 30\ 31];$$

The operation is explained as below:

for $i = 1:32$

$B_{de-rm}(:, P(i)) = B_{de-selection}(:, i)$

end

4.20 Channel Decoding

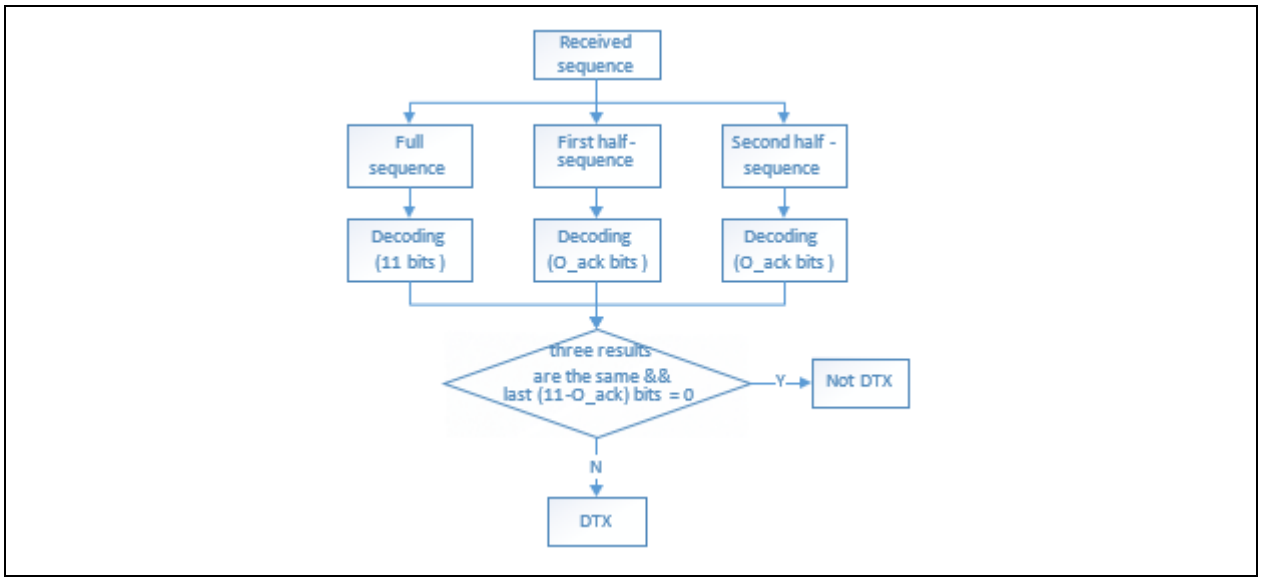
The third-party decoding component is used, so the processing is not described here.

4.21 DTX Detection and CRC Check

RM Code

For RM code, DTX detection is necessary to detect if this is DTX transmission to avoid the wrong resource schedule. The process is illustrated in Figure 30.

Figure 33. Illustration of DTX Detection for RM Code



The received sequence is divided into the first half and second half, and then rate de-matching and decoding are conducted again. The final decoded results of three times are compared to check if they are equal. If the results are the same, the detection is valid. While if the results are different, the transmission is considered as DTX.

To support this kind of DTX detection, the Reed-Muller code rate is limited up to 0.17 for format 2, 3, 4. If the code rate is large than 0.17, the Polar code is recommended for use in our design.

Note: The Reed-Muller code rate is limited up to 0.7 for format 2, 3, 4. If the code rate is larger than 0.17, the Polar code is recommended to use in our design to support this kind of DTX detection.

Polar Code

For the Polar code, a CRC check is needed to determine if the code is received correctly and could instead of the operation of DTX detection. This processing of part is the same as that of the PUSCH but with a different polynomial. Given E is the UCI bit length and L is the CRC length. For $12 \leq \mathrm{E} \leq 19$, $\mathrm{L} = 6$ and for $\mathrm{E} \geq 20$, $\mathrm{L} = 11$

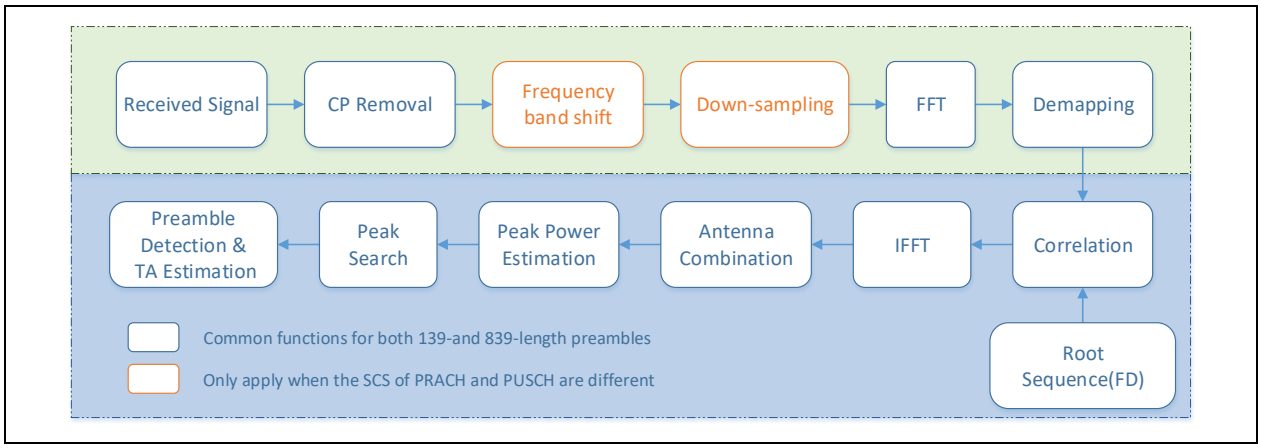
§

5.0 PRACH Algorithms

This section introduces the Physical Random Access Channel (PRACH) baseband algorithms for FlexRAN, including the overall processing steps, the introduction of key modules, input, output, detailed algorithms, and so on.

The UL Random Access (RA) baseband receiver algorithm processing is illustrated in Figure 34.

Figure 34. PRACH Processing Flow



The frequency band shift and down-sampling parts only apply when the subcarrier spacing (SCS) of PRACH and PUSCH are different. The other processing functions are common for 139- and 839-sequence preambles. The location band of PRACH is shifted to the lowest frequency band because the filter in downsampling is a low-pass filter. The down-sampling part is realized by cascade filters to avoid aliasing after decimation.

In order of operation, the received signal must undergo the following processes:

1. Remove the cyclic prefix (CP).
2. Transform into the frequency domain to perform de-mapping to extract PRACH signals.
3. Correlate the extracted signals with the generated frequency domain Zadoff-Chu root sequence.
4. Transform the correlated signals to the time domain using inverse fast Fourier transform (IFFT).
5. Perform non-coherent combining overall antennas and repeated symbols
6. Obtain the power delay profile (PDP) to conduct the noise estimation, peak-search function over cyclic shift specific windows, and threshold comparison.
7. Estimate the timing advanced (TA) value for each RA preamble that will feedback to the UE in message 2.

5.1 CP Removal

This section describes the algorithm for cyclic prefix (CP) removal.

5.1.1 Input

$y(n,r)$	The received signal in the time domain in Rx antenna r

5.1.2 Output

$y_{deCP}(n,r)$	The received signal after removing CP in Rx antenna *r*

5.1.3 Algorithm

This part is conducted in the time domain to extract the PRACH signal. The CP length for different formats is referred to as Table 6.3.3.1-1 and Table 6.3.3.1-2 in *3GPP TS38.211* (refer to Table 2).

5.2 Frequency band shift

This section describes the algorithm for cyclic prefix (CP) removal.

5.2.1 Input

$y_{deCP}(n,r)$	Received signal after removing CP in Rx antenna r
N_{shift}	The shift size

5.2.2 Output

$y_{shift}(n,r)$	The signal that has been shifted to the lowest frequency band

5.2.3 Algorithm

This function only applies to 839 sequence formats.

The frequency band shift is to shift the location band of PRACH to lowest frequency band, and the shift expression is:

$$y_{shift}(n,r) = y_{deCP}(n,r) * e^{-j2\pi N_{shift}\Delta f_{\mathrm{RA}}\left(t-N_{\mathrm{CP},l}^{\mathrm{RA}}T_{\mathrm{c}}-t_{\mathrm{start}}^{\mathrm{RA}}\right)}$$

$$N_{shift} = Kk_1$$

The shift function is the inverse process of the OFDM signal generation equation, which refers to chapter 5.3.2 in *3GPP TS38.211* (refer to Table 2). Therefore,

$$K = \Delta f / \Delta f_{\mathrm{RA}}$$

$$k_1 = k_0^{\mu} + \left(N_{\mathrm{BWP},i}^{\mathrm{start}} - N_{\mathrm{grid}}^{\mathrm{start},\mu}\right) N_{\mathrm{sc}}^{\mathrm{RB}} + n_{\mathrm{RA}}^{\mathrm{start}} N_{\mathrm{sc}}^{\mathrm{RB}} + n_{\mathrm{RA}} N_{\mathrm{RB}}^{\mathrm{RA}} N_{\mathrm{sc}}^{\mathrm{RB}} - N_{\mathrm{grid}}^{\mathrm{size},\mu} N_{\mathrm{sc}}^{\mathrm{RB}}/2$$

$$k_0^{\mu} = \left(N_{\mathrm{grid}}^{\mathrm{start},\mu} + N_{\mathrm{grid}}^{\mathrm{size},\mu}/2\right) N_{\mathrm{sc}}^{\mathrm{RB}} - \left(N_{\mathrm{grid}}^{\mathrm{start},\mu_0} + N_{\mathrm{grid}}^{\mathrm{size},\mu_0}/2\right) N_{\mathrm{sc}}^{\mathrm{RB}} 2^{\mu_0-\mu}$$

For conciseness, the specific meanings of each parameter are not listed here. Refer directly to the Chapter 5.3.2 in 3GPP TS38.211. The value $\bar{k}$ means the guard subcarrier size in the lower side, and its value is described in Table 6.3.3.2-1 in *3GPP TS38.211* (refer to Table 2). Since the passband of downsampling filter is set to full bandwidth, $\bar{k}$ is not considered in the shift.

5.3 Down-Sampling

This section describes the algorithm for frequency band shift.

5.3.1 Input

$y_{shift}(n,r)$ The signals that have been shifted to the lowest frequency band

5.3.2 Output

$y_{ds}(n,r)$ The signals that have been decimated to the same sampling rate as the data channel

5.3.3 Algorithm

This function only applies to 839 sequence formats.

For format 0, the PRACH sub-carrier spacing is 1.25 kHz, while the sub-carrier for PUSCH is 30 kHz. To maintain the same sampling rate, 24 times FFT size is needed for the signal transformation to the frequency domain at the receiver. Decimation is necessary to decrease the needed FFT size. The polyphase filters are used for linear filter and decimation. In general, cascade filters are used to implement the polyphase filters for different transmission formats, as illustrated in Table 7.

Table 7. Overview of 839-Length PRACH Decimation (100 M, 30 kHz)

PRACH format	Format 0,1,2	Format 3
Transmission bandwidth	100 MHz	100 MHz
PUSCH FFT size	4096	4096
PUSCH subcarrier	30 kHz	30 kHz
System sampling rate	122.88 MHz	122.88 MHz
PRACH subcarrier	1.25 kHz	5 kHz
Decimation rate	4x3x2	3x2

For format 0, three-stage cascaded filters are used to realize the downsampling rate 24. The low pass filter is designed with the help of Matlab* toolbox **Filter Design** to generate the filter coefficient. The detailed configuration of each filter is presented in Table 8.

Table 8. Design of Cascade Filter for Format 0

T	Stage 1	Stage 2	Stage 3
Decimation rate	4	3	2
A pass (dB)	0.1	0.1	0.1
A stop (dB)	60	60	60

T	Stage 1	Stage 2	Stage 3
Fs (MHz)	122.88	30.72	10.24
Target Fs (MHz)	30.72	10.24	5.12
F pass (MHz)	0.54	0.54	0.54
F stop (MHz)	30.18	9.7	2.56
Filter order	12	6	12

Note:

- A pass and A stop: the amplitude parameter in low pass filter and their relation is presented in Figure 34
- Fs is the sampling rate of the data channel, which equals to 4096 * 30 kHz = 122.88 MHz
- F pass is half of the frequency resource used for PRACH (864 subcarriers), which is equal to 864 * 1.25 kHz / 2 = 1.08 MHz / 2 = 0.54 MHz
- F stop:
  - For the first two-stage filters: F stop = Target Fs – F pass, as in Figure 35.
  - For the last stage filter that has very strict requirements, it should be equal to half of the FFT size, as in Figure 36.
- Filter order must be an even number to implement the polyphase filter.

Figure 35. Illustration of Low Pass Filter

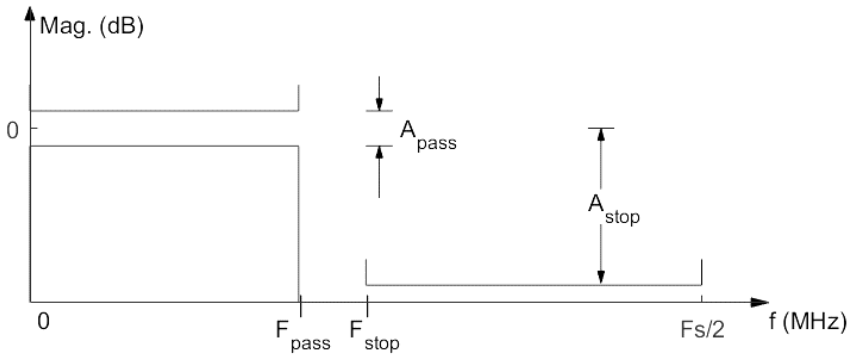


Figure 36. Illustration of First Two-Stage Low Pass Filter

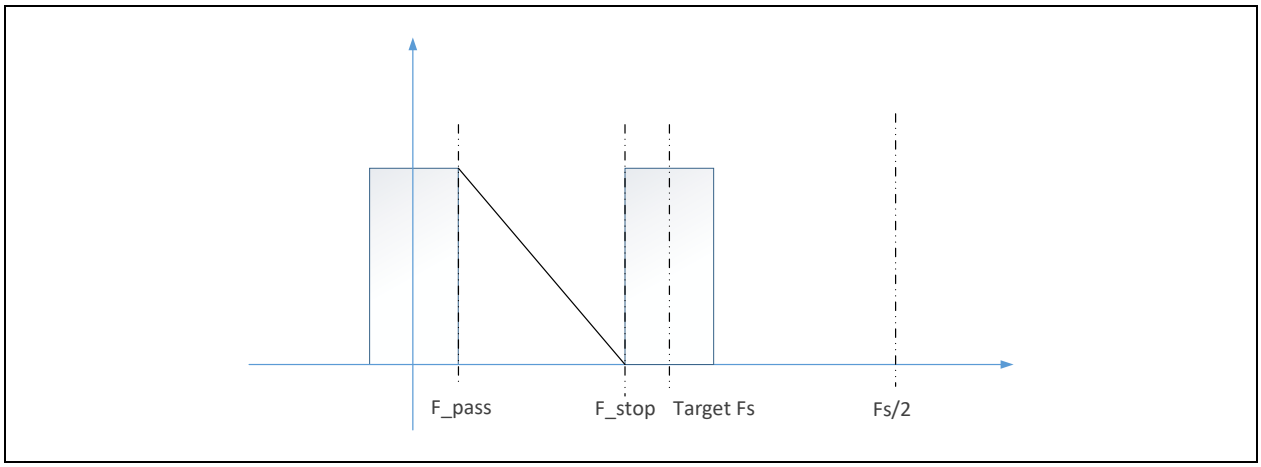
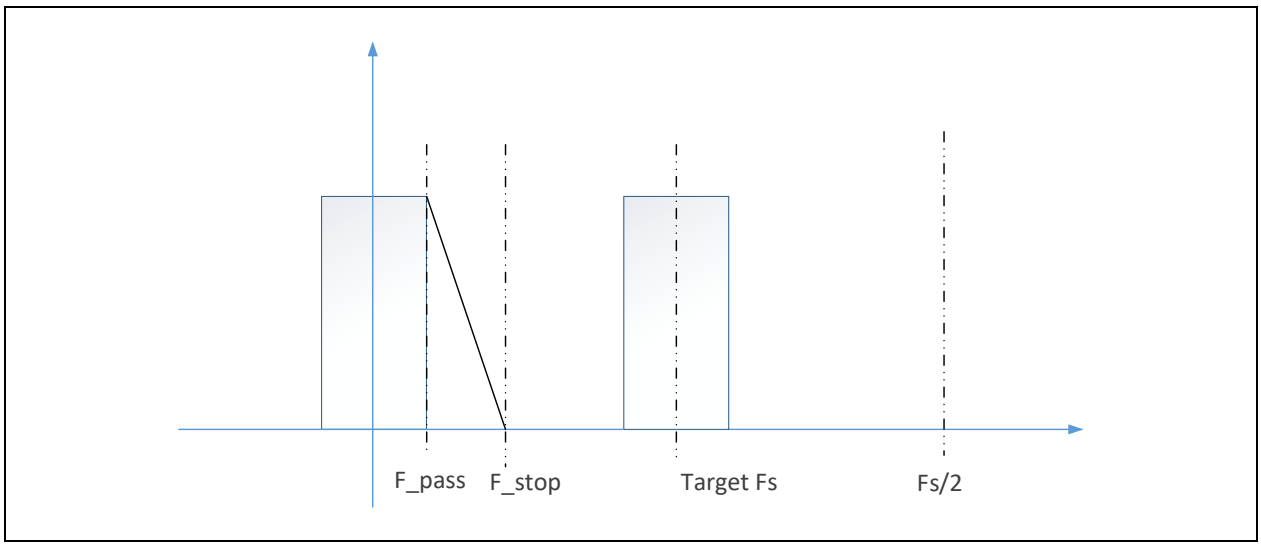


Figure 37. Illustration of Last Stage Filter



5.4 FFT/De-mapping

This section describes the algorithm for Down-Sampling to decrease the needed Fast Fourier Transform (FFT) size.

5.4.1 Input

$y_{ds}(n,r)$ The signal that has been decimated to the same sampling rate as the data channel

5.4.2 Output

$Y(k,r)$ The de-mapped received RA signal in the frequency domain after FFT.

5.4.3 Algorithm

The FFT sizes for 139 sequence preamble and 839 sequence preamble are different due to the numerology.

1024 size FFT is conducted on 139 sequence preamble, and 4096 size FFT is applied for 839 sequence preamble.

The de-mapping part is referred to in Tables 6.3.3.1-1 and 6.3.3.1-2 in 3GPP TS38.211 spec (refer to Table 2.

5.5 Generation of Zadoff-Chu Sequence

This section describes the algorithm for correction with root ZC sequence.

5.5.1 Input

u	Root values
L_{RA}	The length of the preamble sequence

5.5.2 Output

X_u	Frequency domain complex-valued sample of the Zadoff-Chu root sequence

5.5.3 Algorithm

The Zadoff-Chu (ZC) root sequence is defined in the time domain according to 6.3.3.1 in *3GPP TS38.211* (refer to Table 2) as:

$$x_u(i) = e^{-j\frac{\pi ui(i+1)}{L_{RA}}}, i = 0,1,\cdots,L_{RA}-1$$

There could be one or several root sequences. The frequency-domain representation shall be generated according to

$$X_u(i) = \sum_{m=0}^{L_{RA}-1} x_u(m) \cdot e^{-j\frac{2\pi mi}{L_{RA}}}$$

Where $L_{RA} = 839$ or $L_{RA} = 139$ depending on the PRACH preamble format.

5.6 Correlation with Root ZC Sequence

This section describes the algorithm for correction with root ZC sequence.

5.6.1 Input

$Y(k,r)$	The received RA signal in the frequency domain in Rx antenna r
X_u	The root ZC sequence X_u (cell-specific, one or several root values)

5.6.2 Output

$C_u(k,r)$	The correlation results in the frequency domain for different root values

5.6.3 Algorithm

Correlation in the frequency domain

$$C_u(k,r) = conj(\,Y(\mathrm{k,r})\,) * X_u$$

5.7 Inverse Fast Fourier Transform (IFFT)

This section describes the algorithm for Inverse Fast Fourier Transform (IFFT) operations for correlation output.

5.7.1 Input

$C_u(k,r)$	The correlation result
N_{ifft}	IFFT size

5.7.2 Output

$c_u(n,r)$	The time-domain correlation result

5.7.3 Algorithm

Each correlation output is transformed to the time domain employing zero padding and inverse fast Fourier transform (IFFT). Zero paddings are used to interpolate the time domain signal by expanding the L_{RA} frequency domain samples to N_{ifft} samples before IFFT.

The IFFT size for 139 sequence preamble and 839 sequence preamble is different for keeping a reasonable oversampling rate.

1024 size IFFT is conducted on 139 sequence preamble, and 2048 size IFFT is applied for 839 sequence preamble.

The IFFT size adopts 1024 to align with the data channel. The oversampling rate is N_{ifft}/L_{RA}.

$$c_u(n,r) = iFFT(C_u(\mathrm{n,r})) * \sqrt{N_{ifft}}$$

5.8 PDP Calculation and Combination

This section describes the algorithm for Power Delay Profile (PDP) calculation and combination.

5.8.1 Input

$c_u(\mathrm{n,r})$	The time-domain correlation result in Rx antenna r

5.8.2 Output

$P_u(n)$	The combined time-domain correlation result

5.8.3 Algorithm

The PDP is calculated as below:

$$P_u(n) = \frac{1}{N_{os}} \sum_{1}^{N_{os}} \frac{1}{N_{ant}} \sum_{1}^{N_{ant}} |c_u(\mathrm{n}, \mathrm{r})|^2$$

Then the power is combined over the repeated symbols and receiving antennas to lower the noise level. The combination method is a linear average. The core of PRACH detection is the accumulation of correlation power and the reduction of noise power.

5.9 Peak search and Threshold Comparison

This section describes the algorithm for peak search on summed PDP by pre-determined threshold comparison.

5.9.1 Input

$P_u(n)$	The combined time-domain correlation result
N_{CS}	Cyclic shift value
Thr	Threshold set

5.9.2 Output

RAPID Random access preamble index, the detected preamble ID

5.9.3 Algorithm

The preamble is detected by searching the peak in the summed PDP within each search window by comparing it with the pre-determined threshold, as shown in Figure 38.

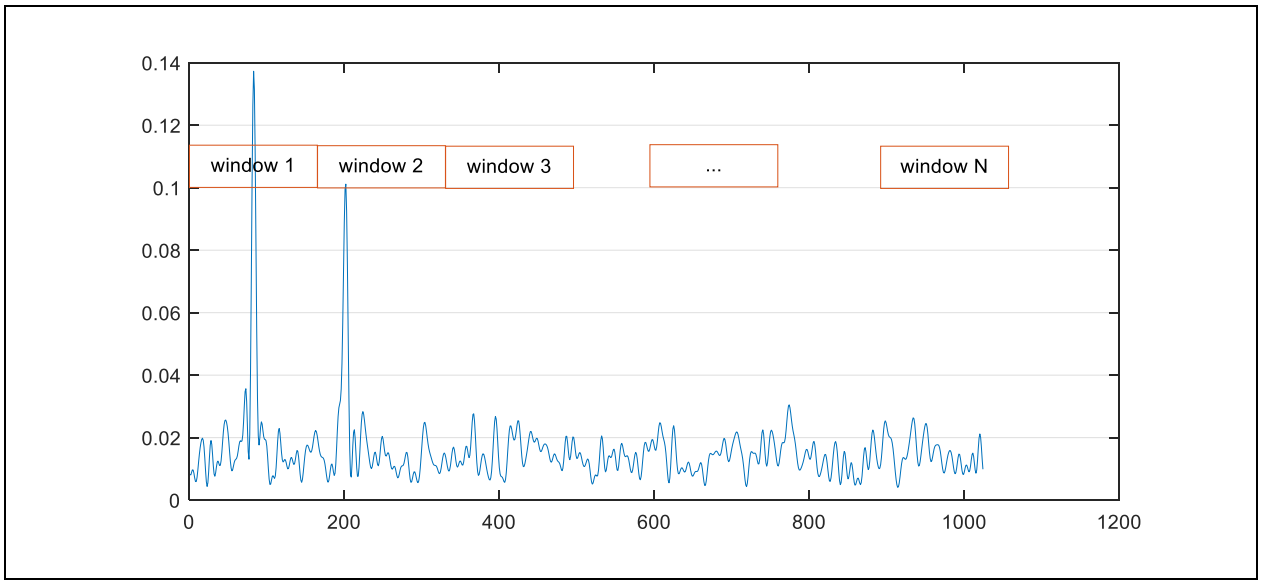
The search window depends on the cyclic shift's value.

$$SearchWindowLength = N_{CS} \cdot \left\lfloor \frac{Correlation\ Size}{PreambleLength} \right\rfloor$$

With a certain CS value v detected, the preamble index can be obtained.

N_{CS} depends on the configuration of ZeroCorrelationZoneConfig, which is explained in Table 6.3.3.1-7 in *3GPP TS38.211* (refer to Table 2).

Figure 38. Illustration of the Search Window



Each search window corresponds to a preamble index, and therefore there are 64 search windows for all the root values. The search window number for each root can be different. This part is related closely with the preamble set generation part.

Two thresholds, $Thr_{u,A}$ and $Thr_{u,B}$, are used in peak detection, and the process is divided into four steps:

1. Calculate the average PDP power for each root value of each occasion.

$$P_{u,avg} = \frac{1}{N_{IFFT}} \sum_{n=0}^{N_{IFFT}-1} P_u(n)$$

2. Calculate $Thr_{u,A}$ to obtain the preamble power.

$$Thr_{u,A} = Thr \text{ x } P_{u,avg}$$

   a. Where Thr is calculated by a chi-square distribution

$$Thr = G_{soft}^{-1}\left[(1 - P_{FA})^{\frac{1}{L}}\right], L \text{ is IFFT size}, P_{FA} = 0.1\%$$

$$G_{soft}(Thr) = F_{\chi^2}(Thr, 2N_{ant}N_{nca}, 0) = 1 - e^{-N_{ant}N_{nca}Thr} \sum_{k=0}^{N_{ant}N_{nca}-1} \frac{1}{k!}(N_{ant}N_{nca}Thr)^k$$

   b. Where Thr is the threshold, N_{ant} is the receiver antenna number, and N_{nca} represents the non-correlation accumulation, which is equal to the repeated times of each preamble format. The itemized deduction is in Table 9, showing that the Thr can be pre-calculated and stored. These thresholds need to be modified slightly (adding 1-3) to adapt the implementation according to a large number of simulations.

   c. The modified values used in the current receiver appear in Figure 7.

Table 9. Threshold Table for Different Formats

Repeated symbol number	Preamble type	Received antenna = 1	Received antenna = 2	Received antenna = 4
1	C1	16.8390	11.3570	8.3450
2	A1, B1	11.3570	8.3450	6.6500
4	A2,B2,C2	8.3450	5.1500	4.1660
6	A3,B3	7.2410	5.0130	4.2870
12	B4	6.0130	5.2870	4.8430
1	Format 0	17.5320	11.7240	8.5440

3. Estimate the preamble power P_{signal} based on $Thr_{u,A}$.

$$P_{u,signal} = \frac{1}{N_{u,S}} \sum_{n=0}^{N_{IFFT}-1} (P_u(n) > Thr_{u,A})$$

$N_{u,S}$ is the number of samples whose value is above $Thr_{u,A}$.

4. Use $Thr_{u,B}$ to detect the peak value in the search window.

$$Thr_{u,B} = P_{u,signal}$$

5.10 PRACH FO Check

This section describes the algorithm for FO peak search on summed PDP by pre-determined FO threshold comparison.

5.10.1 Input

$P_u(n)$	The combined time-domain correlation result
N_{CS}	Cyclic shift value
d_u	Frequency Offset Peak distance related with root seq index u
RAPID	Random access preamble index, the detected preamble ID
N_ThresholdFO	noise threshold in FO scenario
Peak_ThresholdFO	peak threshold in FO scenario

5.10.2 Output

RAPID Random access preamble index, the detected preamble ID

5.10.3 Algorithm

For certain C_v (for the certain preamble UE choose from the total 64 of the current cell), BS will detect the correct one peak at (around) position: $\lceil C\_v * \text{CorrelationSize/PreambleLength} \rceil$ of the PDP (Power Deploy Profile) .

a pair of side lobes appears at the d_u location.

$$d_u = (m \cdot PreambleLength - 1)/u$$

where *m* is the smallest integer that makes d_u an integer.

The Noise Peaks will show around: [($C\_v \pm d\_u$)∗CorrelationSize/PreambleLength], PRACH FO check will remove FO peaks and only report true peak after PRACH FO Check algorithm.

** 839 PRACH test with FO-peak Check Algorithm enabled; N_ThresholdFO = 2.0, Peak_ThresholdFO = 1.05 in PRACH FO-check Algorithm.

6.0 Common Algorithm

6.1 DFT/IDFT Algorithm

Generally, the frequency domain sequence $X_0, X_1, ..., X_{N-1}$ is transformed into a time domain sequence $x_0, x_1, ..., x_{N-1}$ according to $x_k = \sum_{n=0}^{N-1} X_n \cdot W_N^{nk}, where W_N = e^{i2\pi/N}$. But considering N is composite number in 5GNR system, so mixed radix IDFT is introduced to reduce computing efforts.

For example, $N = p \cdot q$, $k = k_1 \cdot p + k_0$, where $k_1 = 0,1,...,q-1$ and $k_0 = 0,1,..., p-1$, $W_N = e^{i2\pi/N}$, then

$$x_k = x(k) = x(p \cdot k_1 + k_0) = x(k_1, k_0) = \sum_{n=0}^{N-1} X_n \cdot W_N^{nk}$$

$$= \sum_{n_0=0}^{p-1} \sum_{n_1=0}^{q-1} X(q \cdot n_0 + n_1) \cdot W_N^{(q \cdot n_0 + n_1)(pk_1 + k_0)}$$

$$= \sum_{n_0=0}^{p-1} \sum_{n_1=0}^{q-1} X(n_0, n_1) \cdot W_N^{qn_0k_0} W_N^{pn_1k_1} W_N^{n_1k_0} W_N^{pqn_0k_1}$$

$$= \sum_{n_1=0}^{q-1} \left\{ \left[\sum_{n_0=0}^{p-1} X(n_0, n_1) \cdot W_p^{n_0k_0} \right] W_N^{n_1k_0} \right\} W_q^{n_1k_1}$$

$$= \sum_{n_1=0}^{q-1} \left[x_1(k_0, n_1) W_N^{n_1k_0} \right] W_q^{n_1k_1}$$

$$= \sum_{n_1=0}^{q-1} x_1'(k_0, n_1) W_q^{n_1k_1} = x_2(k_0, k_1)$$

If p and/or q are composite number too, then the formula above can be further parsed to reduce computing efforts more.

6.1.1 DFT/IDFT Fixed-Point Algorithm

In the current code, DFT/IDFT is implemented in the IA platform by using fixed-Point algorithm, so, the algorithm can be derived as the below.

$$X(k) = \sum_{n=0}^{N} x(n) W_N^{nk}$$

$$if\ N = N1 * N2, then$$

$$X(k) = X(k1, k0) = \sum_{n_0=0}^{N_2-1} \left\{ \left[\sum_{n_1=0}^{N_1-1} x(n_1, n_0) W_{N_1}^{n_1k_0} \right] W_N^{n_0k_0} \right\} W_{N_2}^{n_0k_1}$$

Derive from the original DFT transformation formula $X(k) = \sum_{n=0}^{N} x(n) W_N^{nk}$

, N will be split into N1 and N2, certainly, N1 and N2 can also be further split.

As known, M complex data accumulation will increase the M times amplitude of real and imaginary part and multiplying a twiddle factor will bring $\sqrt{2}$ times amplitude increasing of real and imaginary part. But due to the

random signal, $\sqrt{2}$ times amplitude will not be considered here. So, shifting $\mathrm{ceil}(\log2(\mathrm{M}))$ bits is necessary after M data accumulation.

Shifting fixed bits after each M data accumulation will bring very few cycles increase. But it is not good for next level signal amplitude, because the input signal might have lower amplitude or uneven distribution, after several bit shift, the output accuracy would be cut down due to over bit shift.

So dynamic bit shifting will be considered here, selecting the maximum input data to do the scale factor to avoid over bit shift.

To current design, the maximum data would be the max value in both real/imaginary data in 4 symbols.

The below formula can be derived.

$$\sum_{n=0}^{N} x(n)W_N^{nk} \le \left|\sum_{n=0}^{N} x(n)W_N^{nk}\right| \le \sum_{n=0}^{N} \left|x(n)W_N^{nk}\right| = \sum_{n=0}^{N} |x(n)| \le \sum_{n=0}^{N} \sqrt{y^2+y^2} = \sqrt{2}Ny$$

If$\sqrt{2}Ny < 2^{15}$, then y is limited as$y < 2^{15-log2(\sqrt{2}N)}$, namely, y needs to right shift$15 - log2(\sqrt{2}N)$, $\sqrt{2}$ is also not considered here. So, y needs to right shift $15 - log2(N)$ bits.

Likewise, if there are two levels operation N1 and N2, the biggest bit shift equals$15 - log2(N) = 15 - log2(N_1 * N_2)$.

Ideally, selecting maximum data should be done in each level to get the best precision, but the precision and cycles should also be trade-off.

Currently design is that the 1st level operation will be fixed bits shifting and dynamic bit shifting will be applied in the 2nd level operation.

Take 768 length as an example, 768 can be split 24 * 32.

The 1st radix is 24, shifting the fixed bits, the 2nd radix is 32, shifting the dynamic bits by using the logic condition.

Stages of Radix 24	Right shift scaling bits
2	1
3	2
4	1

Stages of Radix 32	Right shift scaling bits
2	1 bit if (y2>>14)>0 0 bit otherwise
4	2 bits if (y2>>12)>1 1 bit if(y2>>12)==1 0 bit otherwise

4	Fixed to 2 bits

6.1.2 Input

$x(n)$ Time or frequency domain data which needs to DFT/IDFT transformation

6.1.3 Output

$X(k)$ Time or frequency domain data DFT/IDFT transformation's output

Appendix A PRACH Detection Threshold Calculation

This appendix contains the method of calculating the PRACH detection threshold.

Event A represents the event that there is no preamble transmission.

Event B represents the corrected detection. All the PDP values are under the set threshold when there is the condition A that is no preamble transmission.

$$\mathrm{P}\{\mathrm{B}\} = P\{C_{MAX} \le Thr \mid A\}$$

Where C_MAX is the maximum value of averaged PDP value; Th is the threshold that is desired.

$$\mathrm{P}\{\mathrm{B}\} = P\{C_{MAX} \le Thr \mid A\} = P\{C_1 \le Thr\} \cdot P\{C_2 \le Thr\} \cdots P\{C_n \le Thr\}$$

$$= \mathrm{G}^L(Thr) \quad n \in 1, \ldots, L$$

$G(Thr)$ follows the cumulative distribution function (CDF) of a central chi-square distribution with 2N degrees of freedom. L is equal to the IFFT size.

L is the sum of all search window lengths, and this value depends on the cyclic shift values. While through simulation, the threshold slightly changes when L changes from 1024 to 1000,900. Therefore, L equals the IFFT size directly.

The probability of a false alarm is

$$P_{FA} = P\{C_i > Thr \mid A\}$$

Where one or some of the PDP values are above the threshold.

$$P_{FA} = 1 - P\{B\} = 1 - \mathrm{G}^L(Thr)$$

$$G(Thr) = (1 - P_{FA})^{\frac{1}{L}}$$

$$G(Thr) = F_{\chi^2}(Thr, 2N_{ant}N_{nca}, 0) = 1 - e^{-N_{ant}N_{nca}Thr} \sum_{k=0}^{N_{ant}N_{nca}-1} \frac{1}{k!}(N_{ant}N_{nca}Thr)^k = (1 - P_{FA})^{\frac{1}{L}}$$

Where N_ant is the receiver antenna number, and N_nca represents the non-correlation accumulation, which is equal to the number of times of each preamble format is repeated. Thr is the threshold that will be used in Section 5.9, Peak search and Threshold Comparison for peak search.

§