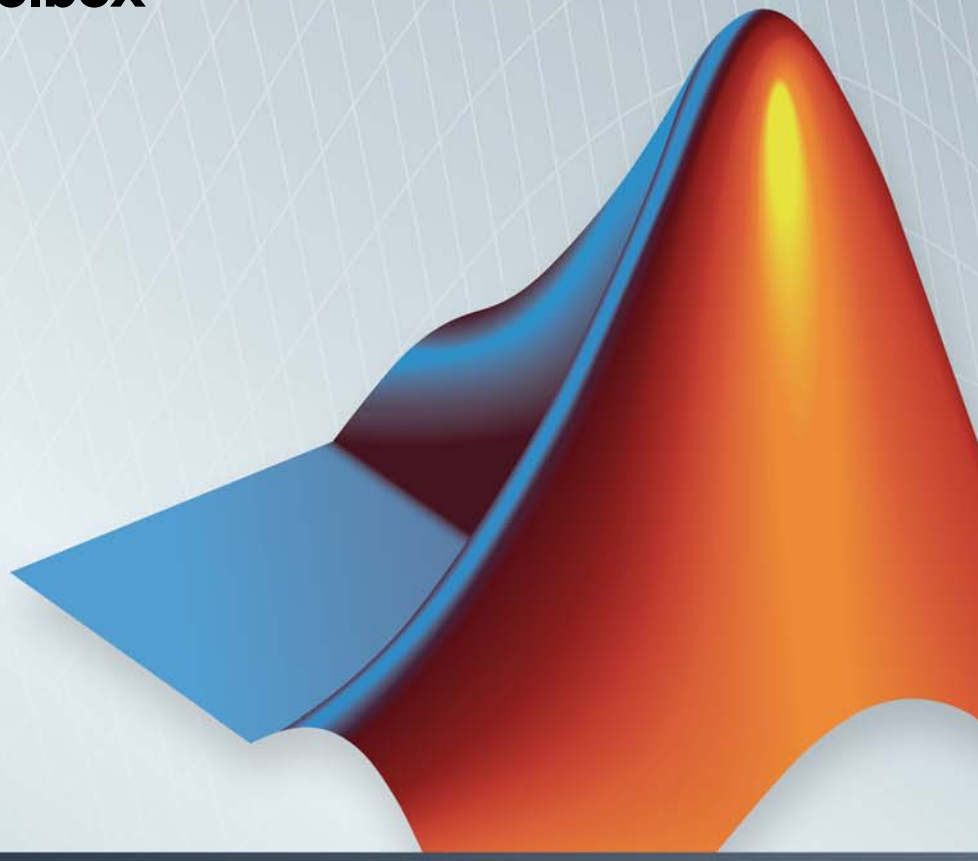
LTE System Toolbox™

User's Guide

R2014**a**



MATLAB®

MathWorks®

How to Contact MathWorks

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`www.mathworks.com/contact_TS.html`	Technical Support

`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

508-647-7000 (Phone)

508-647-7001 (Fax)

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

LTE System Toolbox™ User's Guide

© COPYRIGHT 2013–2014 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Revision History

September 2013	Online only	Revised for Version 1.0 (Release 2013b)
March 2014	Online only	Revised for Version 1.1 (Release 2014a)

Contents

1 LTE Physical Layer Concepts

FDD and TDD Duplexing **1-2**
Create Resource Grid for Cyclic Prefix Choice **1-3**
Create Frame with CellRS in Subframes **1-3**
Frame Structure Type 1: FDD **1-4**
Generate PSS Indices for FDD Mode **1-5**
Frame Structure Type 2: TDD **1-6**
Generate CellRS Indices for TDD Mode **1-8**
Dimension information related to duplexing **1-9**

Synchronization Signals (PSS and SSS) **1-11**
Cell Identity Arrangement **1-11**
Synchronization Signals and Determining Cell Identity . . **1-12**
Primary Synchronization Signal (PSS) **1-12**
Secondary Synchronization Signal (SSS) **1-15**

Sounding Reference Signal (SRS) **1-19**
Sounding Reference Signals **1-19**
Sounding Reference Signals Generation **1-20**

Resource Element Groups (REGs) **1-26**
Resource Element Group Indexing **1-26**
Size and Location of REGs **1-27**
Antenna Port Configurations **1-27**
REG Arrangement with a Normal Cyclic Prefix **1-27**
REG Arrangement with an Extended Cyclic Prefix **1-29**

Control Format Indicator (CFI) Channel **1-32**
Control Format Indicator Values **1-32**
PCFICH Resourcing **1-32**
CFI Channel Coding **1-32**
The PCFICH **1-33**

HARQ Indicator (HI) Channel **1-41**

HARQ Indicator 1-41
PHICH Groups 1-41
HARQ Indicator Channel Coding 1-43
The PHICH 1-43

Downlink Control Channel 1-57
DCI Message Formats 1-57
PDCCH Restructuring 1-58
DCI Message Generation 1-58
DCI Coding 1-58
The PDCCH 1-63

Random Access Channel 1-76
RACH Coding 1-76
The PRACH 1-76
PRACH Conformance Tests 1-79

Uplink Control Channel Format 1 1-81
Uplink Control Information on PUCCH Format 1 1-81
PUCCH Format 1, 1a, and 1b 1-82
Demodulation Reference Signals on PUCCH Format 1 ... 1-83
PUCCH Format 1 Resource Element Mapping 1-87

Uplink Control Channel Format 2 1-91
Uplink Control Information on PUCCH Format 2 1-91
PUCCH Format 2 1-93
Demodulation Reference Signals on PUCCH Format 2 ... 1-96
PUCCH Format 2 Resource Element Mapping 1-100

Downlink Shared Channel 1-104
DL-SCH Coding 1-104
The PDSCH 1-113

Uplink Shared Channel 1-130
UL-SCH Coding 1-130
The PUSCH 1-145
Demodulation Reference Signals (DRS) on the PUSCH ... 1-148

Propagation Channel Models 1-155
Multipath Fading Propagation Conditions 1-155
High Speed Train Condition 1-157

Moving Propagation Condition 1-164
MIMO Channel Correlation Matrices 1-166

Channel Estimation 1-169
Channel Estimation Overview 1-169
Get Pilot Estimates Subsystem 1-173
Pilot Average Subsystem 1-173
Create Virtual Pilots Subsystem 1-178
Interpolation Subsystem 1-181
Noise Estimation 1-181

2 Examples and Demos

Create an Empty Resource Grid 2-2

Map Reference Signal to Resource Grid 2-3

Generate a Test Model 2-5
E-UTRA Test Models 2-5
Generate Test Model Waveform 2-6

Generate LTE Downlink RMC Waveforms 2-8
LTE Downlink RMC Generator Overview 2-8
Open LTE Downlink RMC Generator App 2-8
Dialog Box Inputs and Outputs 2-10
Examples 2-14

Generate LTE Uplink RMC Waveforms 2-18
LTE Uplink RMC Generator Overview 2-18
Open LTE Uplink RMC Generator App 2-18
Dialog Box Inputs and Outputs 2-20
Examples 2-23

Generate LTE Test Model Waveforms 2-27
LTE Test Model Generator Overview 2-27
Open the LTE Test Model Generator App 2-27
Dialog Box Inputs and Outputs 2-29

Examples 2-30

Analyze Throughput for PDSCH Demodulation Performance Test 2-35
LTE Throughput Analyzer Overview 2-35
Open LTE Throughput Analyzer App 2-35
Dialog Box Inputs and Outputs 2-37
Examples 2-39

3 LTE Physical Layer Examples

Create Synchronization Signals 3-2

Model CFI and PCFICH 3-4

Model HARQ Indicator and PHICH 3-7

Model DCI and PDCCH 3-11

Model PUCCH Format 1 3-15

Model PUCCH Format 2 3-17

Model DL-SCH and PDSCH 3-19

Model UL-SCH and PUSCH 3-25

Simulate Propagation Channels 3-27

Find Channel Impulse Response 3-30

4 Selected Bibliography

TS 36.101 . . . . . **4-2**

TS 36.104 . . . . . **4-3**

TS 36.141 . . . . . **4-4**

TS 36.211 . . . . . **4-5**

TS 36.212 . . . . . **4-6**

TS 36.213 . . . . . **4-7**

TS 36.214 . . . . . **4-8**

TS 36.321 . . . . . **4-9**

1

LTE Physical Layer Concepts

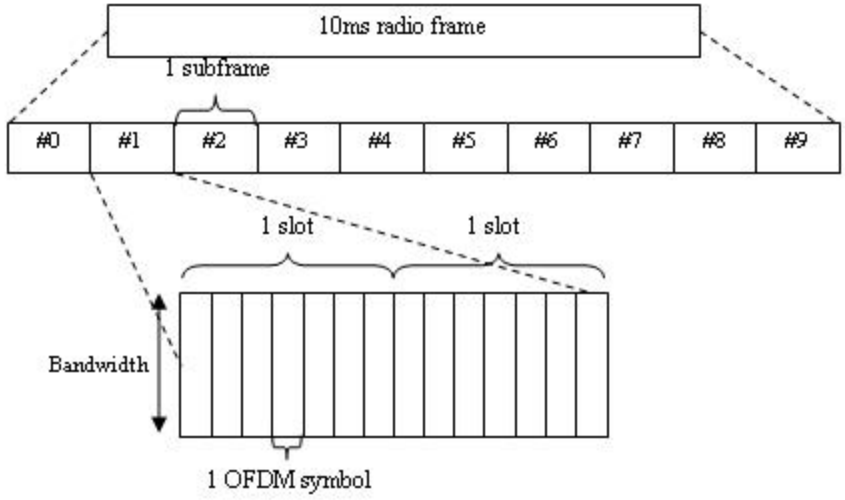
- "FDD and TDD Duplexing" on page 1-2
- "Synchronization Signals (PSS and SSS)" on page 1-11
- "Sounding Reference Signal (SRS)" on page 1-19
- "Resource Element Groups (REGs)" on page 1-26
- "Control Format Indicator (CFI) Channel" on page 1-32
- "HARQ Indicator (HI) Channel" on page 1-41
- "Downlink Control Channel" on page 1-57
- "Random Access Channel" on page 1-76
- "Uplink Control Channel Format 1" on page 1-81
- "Uplink Control Channel Format 2" on page 1-91
- "Downlink Shared Channel" on page 1-104
- "Uplink Shared Channel" on page 1-130
- "Propagation Channel Models" on page 1-155
- "Channel Estimation" on page 1-169

FDD and TDD Duplexing

In this section...

"Create Resource Grid for Cyclic Prefix Choice" on page 1-3
"Create Frame with CellRS in Subframes" on page 1-3
"Frame Structure Type 1: FDD" on page 1-4
"Generate PSS Indices for FDD Mode" on page 1-5
"Frame Structure Type 2: TDD" on page 1-6
"Generate CellRS Indices for TDD Mode" on page 1-8
"Dimension information related to duplexing" on page 1-9

The LTE System Toolbox™ product can generate and manipulate signals for the duplexing arrangements specified in the LTE standard. In LTE, downlink and uplink transmissions are organized into radio frames of duration 10ms consisting of 10 consecutive subframes, each consisting of a number of consecutive OFDM symbols, as shown in the following figure.



Create Resource Grid for Cyclic Prefix Choice

This example shows how to create a resource grid for either normal or extended cyclic prefix. The number of OFDM symbols within one subframe is either 14 for normal cyclic prefix, or 12 for extended cyclic prefix.

Create the cell-wide settings structure.

```
enb.CyclicPrefix = 'Normal';
enb.NDLRB = 9;
enb.CellRefP = 1;
```

Get subframe resource grid dimensions.

```
dims = lteDLResourceGridSize(enb)
```

```
   108    14     1
```

Switch to extended cyclic prefix.

```
enb.CyclicPrefix = 'Extended';
dims = lteDLResourceGridSize(enb)
```

```
   108    12     1
```

The second dimension of the output of `lteDLResourceGridSize` is the number of symbols in the subframe.

Create Frame with CellRS in Subframes

This example shows how to create a frame containing the cell-specific reference signals (CellRS) in each subframe. The radio frame is represented in the LTE System Toolbox product by the use of a succession of 10 cell-wide settings structures with the `NSubframe` field set from 0 through 9 in each case.

Modify the `NDLRB` parameter to set the number of resource blocks. Modify `CellRefP` to set one transmit antenna port. Modify `NCellID` to set the cell ID. Specify normal cyclic prefix and antenna port zero.

```
enb.NDLRB = 6;
enb.CellRefP = 1;
enb.NCellID = 1;
```

```
enb.CyclicPrefix = 'Normal';
antenna = 0;
```

Create an empty resource grid to be populated with subframes.

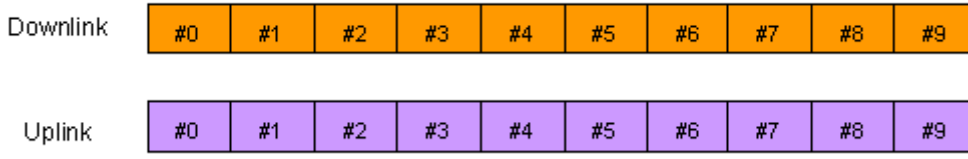
```
txGrid = [];
```

Create an empty resource grid for each subframe and set the current subframe number. Then, generate cell-specific reference signal symbols and indices. Finally, map the cell-specific reference signal to the grid and append the subframe to the grid to be transmitted.

```
for sf = 0:9
    subframe = lteDLResourceGrid(enb);
    enb.NSubframe = sf;
    cellRsSym = lteCellRS(enb,antenna);
    cellRsInd = lteCellRSIndices(enb,antenna);
    subframe(cellRsInd) = cellRsSym;
    txGrid = [txGrid subframe];
end
```

Frame Structure Type 1: FDD

In the FDD duplexing mode, all 10 subframes within a radio frame contain downlink or uplink subframes depending on the link direction.



The uplink and downlink transmitters have separate bandwidths in which to make their transmissions. Therefore, each can transmit at all times.

Within the LTE System Toolbox product, you can create signals or indices for FDD duplexing mode simply by setting the `NSubframe` field of the cell-wide settings structure to the appropriate subframe number. Functions whose behavior depends on the duplexing mode have the `DuplexMode` field, which

you can set to `'FDD'` or `'TDD'`. If you do not specify this field, `'FDD'` is used by default.

Generate PSS Indices for FDD Mode

This example shows how to generate the primary synchronization signal (PSS) indices in subframe 0 using the FDD duplexing mode.

First, create the cell-wide settings structure.

```
enb.CyclicPrefix = 'Normal';
enb.NDLRB = 9;
enb.NCellID = 1;
enb.NSubframe = 0;
enb.DuplexMode = 'FDD';
```

Next, create PSS indices.

```
ind = ltePSSIndices(enb)
```

```
ind=
         672
         673
         674
```

If the same call is made for subframe 1 instead, then the result is an empty matrix.

```
enb.NSubframe = 1;
```

An empty matrix indicates that the PSS is not present in subframe 1. By calling the functions for indices and values for subframes 0 through to 9 by setting the `NSubframe` field, the appropriate transmissions across a radio frame can be formed.

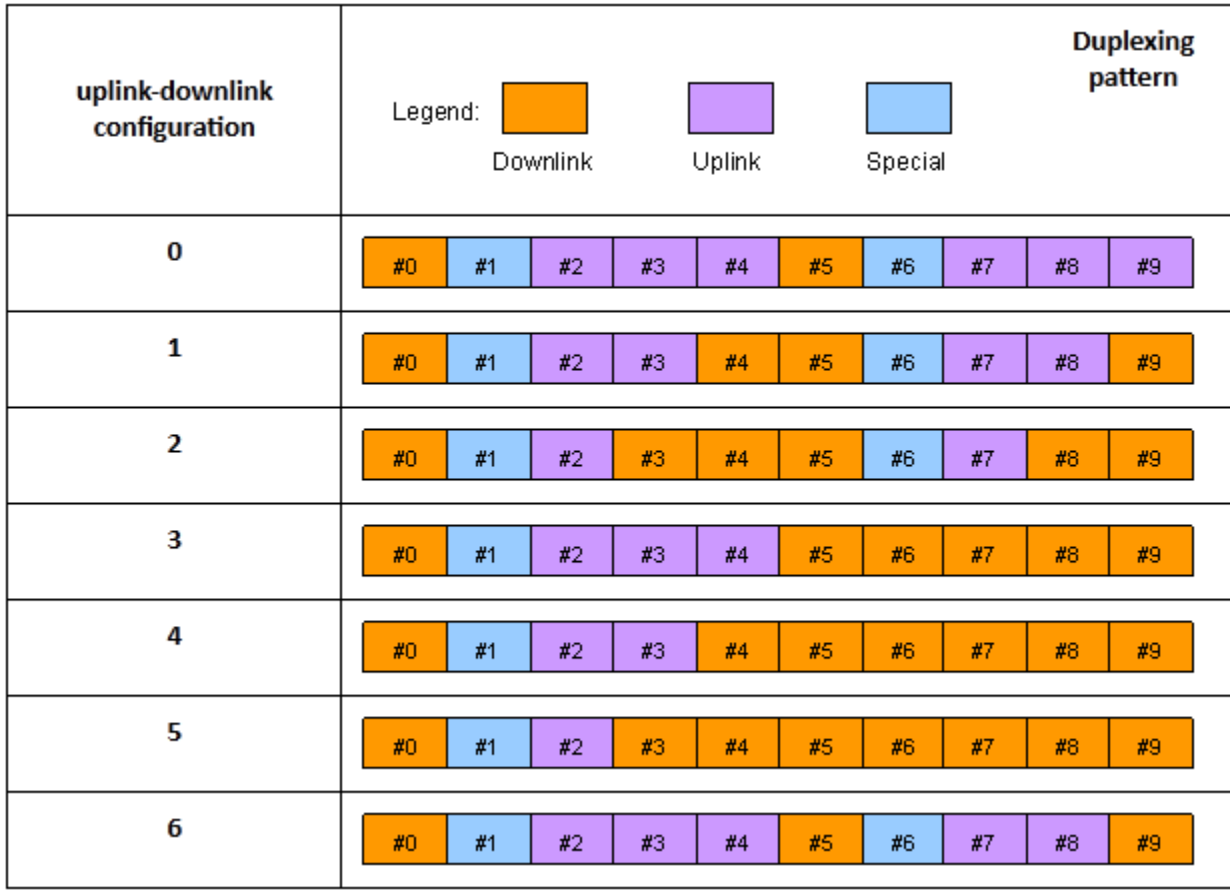
```
ind = ltePSSIndices(enb)
```

```
ind =

   Empty matrix: 0-by-1
```

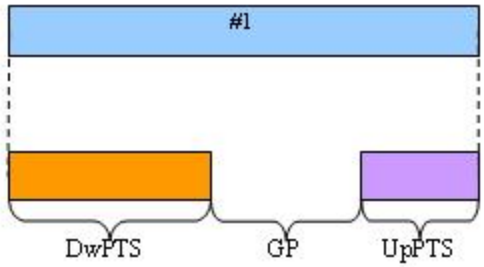
Frame Structure Type 2: TDD

In the TDD duplexing mode, a single bandwidth is shared between uplink and downlink, with the sharing being performed by allotting different periods of time to uplink and downlink. In LTE, there are 7 different patterns of uplink-downlink switching, termed uplink-downlink configurations 0 through 6, as shown in the following figure.



The special subframe (subframe 1 in every uplink-downlink configuration, and subframe 6 is uplink-downlink configurations 0, 1, 2 and 6) contains a

portion of downlink transmission at the start of the subframe (the Downlink Pilot Time Slot, DwPTS), a portion of unused symbols in the middle of the subframe (the Guard Period) and a portion of uplink transmission at the end of the subframe (the Uplink Pilot Time Slot, UpPTS), as shown in the following figure.



The lengths of DwPTS, GP, and UpPTS can take one of 9 combinations of values, termed special subframe configurations 0 through 8. The LTE standard specifies the lengths in terms of the fundamental period of the OFDM modulation, but the lengths can be interpreted in terms of OFDM symbols as shown in the following table.

Special subframe configuration	Normal cyclic prefix in downlink			Extend	
	DwPTS	UpPTS		DwPTS	UpPTS
		Normal cyclic prefix in uplink	Extended cyclic prefix in uplink		Normal cyclic pr
0	3	1	1	3	1
1	9			8	
2	10			9	
3	11			10	
4	12			3	2
5	3	2	2	8	
6	9			9	
7	10			-	-
8	11			-	-

Thus, in effect, the special subframe is both a downlink subframe and an uplink subframe, with some restriction placed on the number of OFDM symbols that are occupied in each case.

To specify TDD operation, in the cell-wide settings structure, set the optional `DuplexMode` field to `'TDD'`. When you use this setting, functions that require `DuplexMode` also require that you specify the uplink-downlink configuration (0,...,6) in the `TDDConfig` field, the subframe number in the `NSubframe` field, and the special subframe configuration (0,...,9) in the `SSC` field.

Generate CellRS Indices for TDD Mode

This example shows how to create the subscripts for the positions of the cell-specific reference signal (CellRS) for antenna port 0 in subframe 6 for uplink-downlink configuration 2 and special subframe configuration 4 with extended cyclic prefix.

First, create the parameter structure.

```
enb.NDLRB        = 9;
enb.NCellID      = 1;
enb.DuplexMode   = 'TDD';
enb.NSubframe    = 6;
enb.TDDConfig    = 2;
enb.SSC          = 4;
enb.CyclicPrefix = 'Extended';
```

Next, create the cell-specific RS indices.

```
sub = lteCellRSIndices(enb,0,'sub')
```

```
sub =

           2           1           1
           8           1           1
          14           1           1
          20           1           1
          26           1           1
          32           1           1
          38           1           1
          44           1           1
          50           1           1
          56           1           1
          62           1           1
          68           1           1
```

```
    74     1     1
    80     1     1
    86     1     1
    92     1     1
    98     1     1
   104     1     1
```

The second column, which gives the OFDM symbol number (1-based) within the subframe, has values of 1 indicating that only the 1st OFDM symbol will contain Cell-Specific Reference signals in this case. This is because the chosen subframe is a special subframe with DwPTS of length 3 and therefore the other Cell-Specific Reference signal elements (in OFDM symbols 4, 7 and 10) which would be present in full downlink subframes are not generated.

To confirm this theory, change the duplex mode to FDD.

```
enb.DuplexMode   = 'FDD';
sub = lteCellRSIndices(enb,0,'sub');
unique(sub(:,2))
```

```
ans =

     1
     4
     7
    10
```

In this case, the switch over to FDD means that the now irrelevant fields, `TDDConfig` and `SSC`, are ignored.

Dimension information related to duplexing

This example shows how to extract information from a parameter structure. To facilitate working with different duplexing arrangements, the LTE System Toolbox product provides the `lteDuplexingInfo` information function. This function takes a cell-wide settings structure containing the fields mentioned in the preceding sections. It returns a structure that indicates the type of the current subframe and the number of symbols in the current subframe.

First, create a parameter structure.

```
enb.NDLRB        = 9;
enb.NCellID      = 1;
enb.DuplexMode   = 'TDD';
enb.NSubframe    = 6;
enb.TDDConfig    = 2;
enb.SSC          = 4;
enb.CyclicPrefix = 'Extended';
```

Next, extract the dimension information.

```
lteDuplexingInfo(enb)
```

```
ans =

         NSymbols: 12
     SubframeType: 'Special'
       NSymbolsDL: 3
    NSymbolsGuard: 7
       NSymbolsUL: 2
```

Finally, change the `NSubframe` property and extract the dimension information again.

```
enb.NSubframe    = 0;
lteDuplexingInfo(enb)
```

```
ans =

         NSymbols: 12
     SubframeType: 'Downlink'
       NSymbolsDL: 12
    NSymbolsGuard: 0
       NSymbolsUL: 0
```

This function provides direct access to the uplink-downlink configuration patterns via the `SubframeType` field and special subframe DwPTS, GP and UpPTS lengths via the `NSymbolsDL`, `NSymbolsGuard` and `NSymbolsUL` fields.

Synchronization Signals (PSS and SSS)

In LTE, there are two downlink synchronization signals which are used by the UE to obtain the cell identity and frame timing.

- Primary synchronization signal (PSS)
- Secondary synchronization signal (SSS)

The division into two signals is aimed to reduce the complexity of the cell search process.

In this section...

"Cell Identity Arrangement" on page 1-11

"Synchronization Signals and Determining Cell Identity" on page 1-12

"Primary Synchronization Signal (PSS)" on page 1-12

"Secondary Synchronization Signal (SSS)" on page 1-15

Cell Identity Arrangement

The physical cell identity, N_{ID}^{cell}, is defined by the equation:

$$N_{ID}^{CELL} = 3N_{ID}^{(1)} + N_{ID}^{(2)}$$

- $N_{ID}^{(1)}$ is the physical layer cell identity group (0 to 167).
- $N_{ID}^{(2)}$ is the identity within the group (0 to 2).

This arrangement creates 504 unique physical cell identities.

Synchronization Signals and Determining Cell Identity

The primary synchronization signal (PSS) is linked to the cell identity within the group ($N_{ID}^{(2)}$). The secondary synchronization signal (SSS) is linked to the cell identity group ($N_{ID}^{(1)}$) and the cell identity within the group ($N_{ID}^{(2)}$).

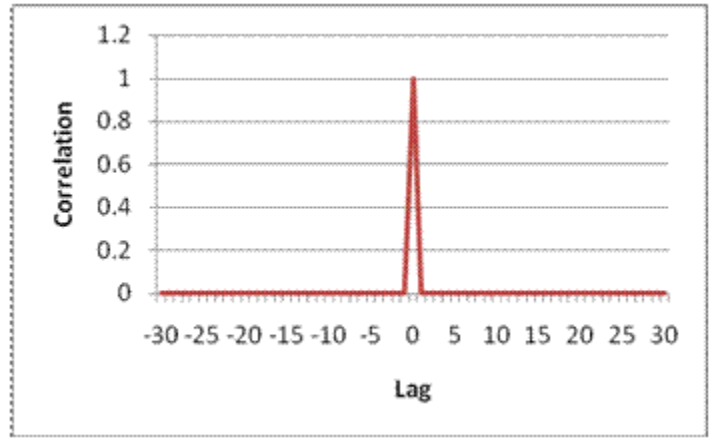
You can obtain $N_{ID}^{(2)}$ by successfully demodulating the PSS. The SSS can then be demodulated and combined with knowledge of $N_{ID}^{(2)}$ to obtain $N_{ID}^{(1)}$. Once you establish the values of $N_{ID}^{(1)}$ and $N_{ID}^{(2)}$, you can determine the cell identity (N_{ID}^{cell}).

Primary Synchronization Signal (PSS)

The primary synchronization signal (PSS) is based on a frequency-domain Zadoff-Chu sequence.

Zadoff-Chu Sequences

Zadoff-Chu sequences are a construction of Frank-Zadoff sequences defined by D. C. Chu in [1]. These codes have the useful property of having zero cyclic autocorrelation at all nonzero lags. When used as a synchronization code, the correlation between the ideal sequence and a received sequence is greatest when the lag is zero. When there is any lag between the two sequences, the correlation is zero. This property is illustrated in the following figure.



PSS Generation

The PSS is a sequence of complex symbols, 62 symbols long.

The sequence $d_u(n)$ used for the PSS is generated according to the following equations.

$$d_u(n) = e^{-j\frac{\pi u(n+1)}{63}}, \text{ for } n = 0,1,\ldots,30$$

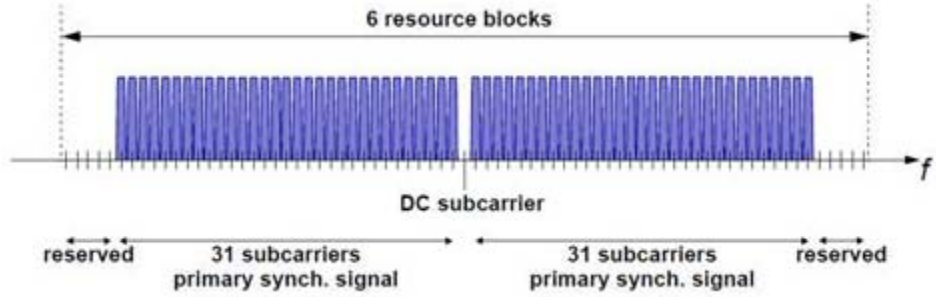
$$d_u(n) = e^{-j\frac{\pi u(n+1)(n+2)}{63}}, \text{ for } n = 31,32,\ldots,61$$

In the preceding equation, u is the Zadoff-Chu root sequence index and depends on the cell identity within the group $N_{ID}^{(2)}$.

$N_{ID}^{(2)}$	**Root index *u***
0	25
1	29
2	34

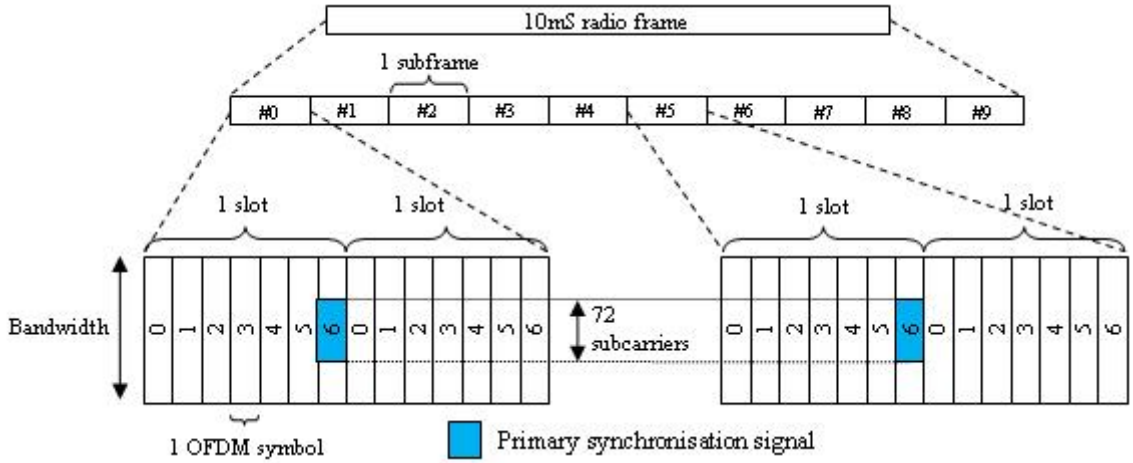
Mapping of the PSS

The PSS is mapped into the first 31 subcarriers either side of the DC subcarrier. Therefore, the PSS uses six resource blocks with five reserved subcarriers each side, as shown in the following figure.

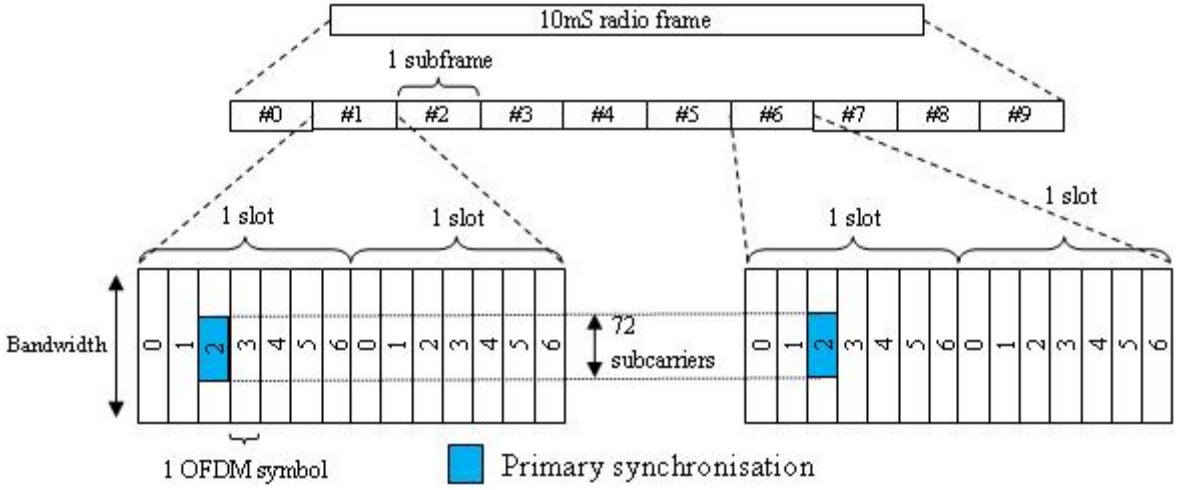


As the DC subcarrier contains no information in LTE this corresponds to mapping onto the middle 62 subcarriers within an OFDM symbol in a resource grid. $d(n)$ is mapped from lowest subcarrier to highest subcarrier. The PSS is mapped to different OFDM symbols depending on which frame type is used. Frame type 1 is frequency division duplex (FDD), and frame type 2 is time division duplex (TDD).

- **FDD** — The PSS is mapped to the last OFDM symbol in slots 0 and 10, as shown in the following figure.



- **TDD** — The PSS is mapped to the third OFDM symbol in subframes 1 and 6, as shown in the following figure.



Secondary Synchronization Signal (SSS)

The secondary synchronisation signal (SSS) is based on maximum length sequences (*m*-sequences).

M-Sequence Definition

An m-sequence is a pseudorandom binary sequence which can be created by cycling through every possible state of a shift register of length resulting in a sequence of length . Three m-sequences, each of length 31, are used to generate the synchronization signals denoted $\tilde{s}$, $\tilde{c}$ and $\tilde{z}$.

SSS Generation

Two binary sequences, each of length 31, are used to generate the SSS. Sequences $s_0^{(m_0)}$ and $s_1^{(m_1)}$ are different cyclic shifts of an m-sequence, $\tilde{s}$. The indices m_0 and m_1 are derived from the cell-identity group, $N_{ID}^{(2)}$ and determine the cyclic shift. The values can be read from table 6.11.2.1-1 in [2].

The two sequences are scrambled with a binary scrambling code ($c_0(n)$, $c_1(n)$), which depends on $N_{ID}^{(2)}$.

The second SSS sequence used in each radio frame is scrambled with a binary scrambling code ($z_1^{(m_0)}$, $z_1^{(m_1)}$) corresponding to the cyclic shift value of the first sequence transmitted in the radio frame.

Binary Sequence Generation

$s_0^{(m_0)}$ and $s_1^{(m_1)}$ are given by the following equations.

$$s_0^{(m_0)} = \tilde{s}((n + m_0) \bmod 31)$$

$$s_0^{(m_1)} = \tilde{s}((n + m_1) \bmod 31)$$

$\tilde{s}$ is generated from the primitive polynomial $x^5 + x^2 + 1$ over the finite field GF(2).

$c_0(n)$ and $c_1(n$ are given by the following equations.

$$c_0(n) = \tilde{c}((n + N_{ID}^{(2)}) \bmod 31)$$

$$c_1(n) = \tilde{c}((n + N_{ID}^{(2)} + 3) \bmod 31)$$

$\tilde{c}$ is generated from the primitive polynomial $x^5 + x^3 + 1$ over the finite field GF(2).

$z_1^{(m_0)}$ and $z_1^{(m_1)}$ are given by the following equations.

$$z_1^{(m_0)} = \tilde{z}((n + (m_0 \bmod 8)) \bmod 31)$$

$$z_1^{(m_1)} = \tilde{z}((n + (m_1 \bmod 8)) \bmod 31)$$

$\tilde{z}$ is generated from the primitive polynomial $x^5 + x^4 + x^2 + x + 1$ over the finite field GF(2).

Mapping of the SSS

The scrambled sequences are interleaved to alternate the sequence transmitted in the first and second SSS transmission in each radio frame. This allows the receiver to determine the frame timing from observing only one of the two sequences; if the first SSS signal observed is in subframe 0 or subframe 5, synchronization can be achieved when the SSS signal is observed in subframe 0 or subframe 5 of the next frame.

The SSS is transmitted in the same subframe as the PSS but one OFDM symbol earlier. The SSS is mapped to the same subcarriers (middle 72 subcarriers) as the PSS.

The SSS is constructed using different scrambling sequences when mapped to even and odd resource elements.

- Even resource elements:
  - Subframe 0: $d(2n) = s_0^{(m_0)}(n)c_0(n)$
  - Subframe 5: $d(2n) = s_1^{(m_1)}(n)c_0(n)$
- Odd resource elements:
  - Subframe 0: $d(2n+1) = s_1^{(m_1)}(n)c_1(n)z_1^{(m_0)}(n)$

- Subframe 5: $d(2n+1) = s_0^{(m_0)}(n)c_1(n)z_1^{(m_1)}(n)$

$d(n)$ is mapped from lowest subcarrier to highest subcarrier.

References

[1] Chu, D. C. "Polyphase codes with good periodic correlation properties." *IEEE Trans. Inf. Theory*. Vol. 18, Number 4, July 1972, pp. 531–532.

[2] 3GPP TS 36.211. "Physical Channels and Modulation." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

See Also

`ltePSS` | `ltePSSIndices` | `lteSSS` | `lteSSSIndices` | `lteDLResourceGrid` | `lteZadoffChuSeq`

Related Examples

- "Create Synchronization Signals" on page 3-2

Sounding Reference Signal (SRS)

In this section...
"Sounding Reference Signals" on page 1-19
"Sounding Reference Signals Generation" on page 1-20

Sounding reference signals (SRS) are transmitted on the uplink and allow the network to estimate the quality of the channel at different frequencies.

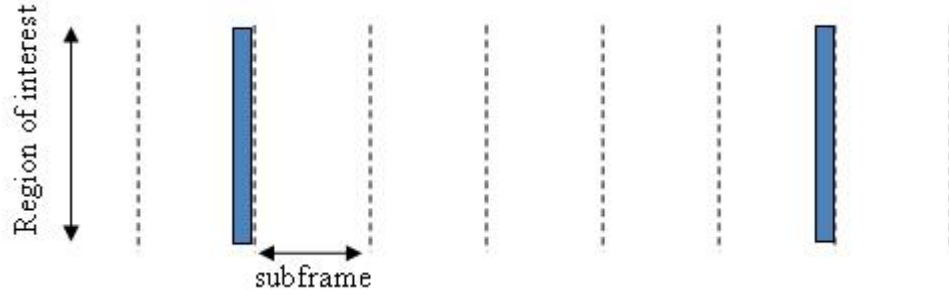
Sounding Reference Signals

The SRS is used by the base station to estimate the quality of the uplink channel for large bandwidths outside the assigned span to a specific UE. This measurement cannot be obtained with the DRS since these are always associated to the PUSCH or PUCCH and limited to the UE allocated bandwidth. Unlike the DRS associated with the physical uplink control and shared channels the SRS is not necessarily transmitted together with any physical channel. If the SRS is transmitted with a physical channel then it may stretch over a larger frequency band. The information provided by the estimates is used to schedule uplink transmissions on resource blocks of good quality.

SRS can be transmitted as often as every second subframe (2 ms) or as infrequent as every 16th frame (160ms). The SRS are transmitted on the last symbol of the subframe.

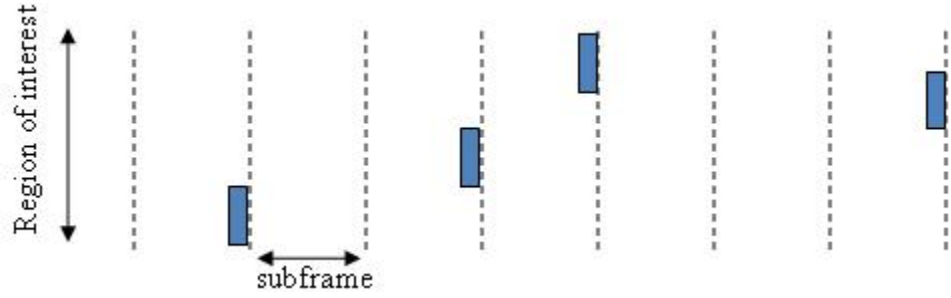
There are two methods of transmitting the SRS:

- **Wideband mode** — one single transmission of the SRS covers the bandwidth of interest. The channel quality estimate is obtained within a single SC-FDMA symbol. However, under poor channel conditions such as deep fade and high path loss, using this mode can result in a poor channel estimate.



Non-frequency-hopping SRS

- **Frequency-hopping mode** — the SRS transmission is split into a series of narrowband transmissions that will cover the whole bandwidth region of interest; this mode is the preferred method under poor channel conditions.



Frequency-hopping SRS

Sounding Reference Signals Generation

The sounding reference signals are generated using a base sequence denoted by $r_{u,v}^{SRS}(n)$. This base sequence is discussed further in "Base sequence" on page 1-21. This base sequence, used expressly to denote the SRS sequence, is defined by the following equation.

$$r_{u,v}^{SRS}(n) = r_{u,v}^{(\alpha)}(n)$$

It is desirable for the SRS sequences to have small power variations in time and frequency, resulting in high power amplifier efficiency and comparable

channel estimation quality for all frequency components. Zadoff-Chu sequences are good candidates as they exhibit constant power in time and frequency. However the number of Zadoff-Chu sequences is limited which makes them unsuitable for use on their own. The generation and mapping of the SRS are discussed further in this section.

Base sequence

The sounding reference signals are defined by a cyclic shift, α, of a base sequence, r.

The base sequence, r, is represented in the following equation.

$$r_{u,v}^{(\alpha)} = e^{j\alpha n} r_{u,v}(n)$$

The preceding equation contains the following variables.

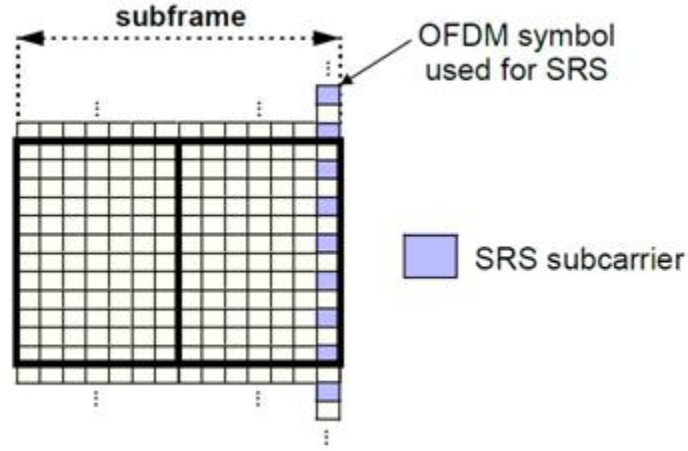
- $n = 0,...,M_{SC}^{RS}$, where M_{SC}^{RS} is the length of the reference signal sequence.
- $U = 0,\ldots,29$ is the base sequence group number.
- $V = 0,1$ is the sequence number within the group and only applies to reference signals of length greater than 6 resource blocks.

A cyclic shift in the time domain (post IFFT in the OFDM modulation) is equivalent to a phase rotation in the frequency domain (pre-IFFT in the OFDM modulation). The base sequence is cyclic shifted to increase the total number of available sequences. For frequency non-selective channels over the 12 subcarriers of a resource block it is possible to achieve orthogonality

between SRS generated from the same base sequence if $\alpha = 2\pi \frac{n_{SRS}}{8}$, where $n_{SRS}^{CS} = 0,1,2,3,4,5,6,7$ (configured for each UE by higher layers) and assuming the SRS are synchronized in time.

The orthogonality can be exploited to transmit SRS at the same time, using the same frequency resources without mutual interference. Generally, SRS generated from different base sequences will not be orthogonal; however they will present low cross-correlation properties.

Similar to the uplink demodulation reference signals for the PUCCH and PUSCH, the SRS are time multiplexed. However, they are mapped to every second subcarrier in the last symbol of a subframe, creating a comb-like pattern, as illustrated in the following figure.



The minimum frequency span covered by the SRS in terms of bandwidth is 4 resource blocks and larger spans are covered in multiples of 4 resource blocks. This means that the minimum sequence length is 24. To maximize the number of available Zadoff-Chu sequences, a prime length sequence is needed. The minimum length sequence, 24, is not prime.

Therefore, Zadoff-Chu sequences are not suitable by themselves. Effectively, there are the following two types of base reference sequences:

- those with a sequence length ≥ 48 (spanning 8 or more resource blocks), which use a cyclic extension of Zadoff-Chu sequences
- those with a sequence length = 24 (spanning 4 resource blocks), which use a special QPSK sequence

Base sequences of length 48 and larger

For sequences of length 48 and larger (i.e. $M_{sc}^{RS} \geq 3N_{sc}^{RS}$), the base sequence is a repetition, with a cyclic offset of a Zadoff-Chu sequence of length N_{zc}^{RS}, where N_{zc}^{RS} is the largest prime such that $N_{zc}^{RS} < M_{sc}^{RS}$. Therefore, the base sequence will contain one complete length N_{zc}^{RS} Zadoff-Chu sequence plus a fractional repetition appended on the end. At the receiver the appropriate de-repetition can be done and the zero autocorrelation property will hold across the length N_{zc}^{RS} vector.

Base sequences of length 24

For sequences of length 24 (i.e. $M_{sc}^{RS} = 12,24$), the sequences are a composition of unity modulus complex numbers drawn from a simulation generated table. These sequences have been found through computer simulation and are specified in the LTE specifications.

SRS Grouping

There are a total of 30 sequence groups, $u \in \{0,1,\ldots,29\}$, each containing one sequence for length less than or equal to 60. This corresponds to transmission bandwidths of 1,2,3,4 and 5 resource blocks. Additionally, there are two sequences (one for v = 0 or 1) for length ≥ 72; corresponding to transmission bandwidths of 6 resource blocks or more.

Note that not all values of m are allowed, where m is the number of resource blocks used for transmission. Only values for m that are the product of powers of 2, 3 and 5 are valid, as shown in the following equation.

$m = 2^{\alpha_0} \times 3^{\alpha_1} \times 5^{\alpha_2}$, where α_i are positive integers

The reason for this restriction is that the DFT sizes of the SC-FDMA precoding operation are limited to values which are the product of powers of 2, 3 and 5. The DFT operation can span more than one resource block, and since each resource block has 12 subcarriers, the total number of subcarriers fed to the DFT will be $12m$. Since the result of $12m$ has to be the product of

powers of 2, 3 and 5 this implies that the number of resource blocks must themselves be the product of powers of 2, 3 and 5. Therefore values of m such as 7, 11, 14, 19, etc. are not valid.

For a given time slot, the uplink reference signal sequences to use within a cell are taken from one specific sequence group. If the same group is to be used for all slots then this is known as fixed assignment. On the other hand, if the group number u varies for all slots within a cell this is known as *group hopping*.

Fixed group assignment. When fixed group assignment is used, the same group number is used for all slots. The group number u is the same as for PUCCH and is a function of the cell identity number modulo 30.

$$u = N_{ID}^{cell} \bmod 30, \text{ with } N_{ID}^{cell} = 0,1,\ldots,503$$

Group hopping. If group hopping is used, the pattern is applied to the calculation of the sequence group number.

This pattern is defined as the following equation.

$$f_{gh}(n_s) = \sum_{i=0}^{7} c(8n_s + i) \cdot 2^i \bmod 30$$

As shown in the preceding equation, this group hopping pattern is a function of the slot number n_s and is calculated making use of a pseudorandom binary sequence $c(k)$, generated using a length-30 *Gold* code. To generate the group hopping pattern, the PRBS generator is initialized with the following value at the start of each radio frame.

$$c_{init} = \left\lfloor \frac{N_{ID}^{cell}}{30} \right\rfloor$$

For SRS with group hopping, the group number, u, is given by the following equation.

$$u = \left(f_{gh}(n_s) + N_{ID}^{cell} \bmod 30 \right) \bmod 30$$

See Also

`lteSRS` | `lteSRSIndices` | `lteSRSInfo` | `lteULResourceGrid`

Related Examples

- "Uplink Waveform Modeling Using SRS and PUCCH"

Resource Element Groups (REGs)

In this section...

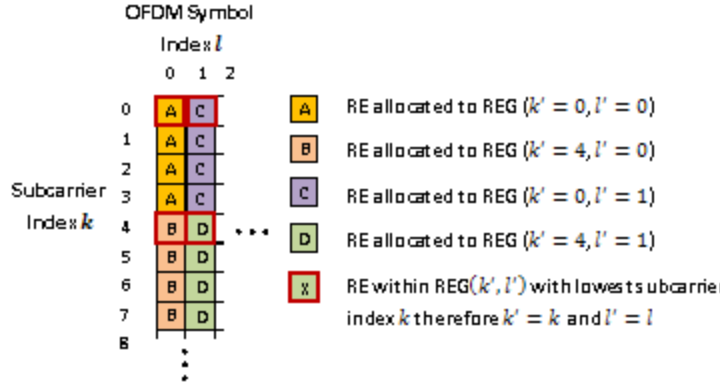
"Resource Element Group Indexing" on page 1-26
"Size and Location of REGs" on page 1-27
"Antenna Port Configurations" on page 1-27
"REG Arrangement with a Normal Cyclic Prefix" on page 1-27
"REG Arrangement with an Extended Cyclic Prefix" on page 1-29

Resource-element groups (REG) are used to define the mapping of control channels to resource elements (RE).

REGs are blocks of consecutive REs within the same OFDM symbol. The REGs within a subframe are located in the first four OFDM symbols and are identical in size and number for each corresponding subframe on every antenna port.

Resource Element Group Indexing

REGs are represented by an index pair (k',l'). The index k' is the subcarrier index of the RE within the REG with the lowest subcarrier index k. The index l' is the OFDM symbol index of the REG (l). This index pair is illustrated in the following figure.



Size and Location of REGs

The number of REs within a REG is such that a REG contains four REs which are not occupied by a cell specific reference signal on any antenna port in use.

All REs within a resource block in one of the first four OFDM symbols are allocated to a REG. Therefore the number of REs within each REG and the number of REGs within an OFDM symbol is affected by the number of cell-specific reference signals present on all antenna ports.

The number and location of cell specific reference signals are dependent on the number of antenna ports and the type of cyclic prefix used.

Antenna Port Configurations

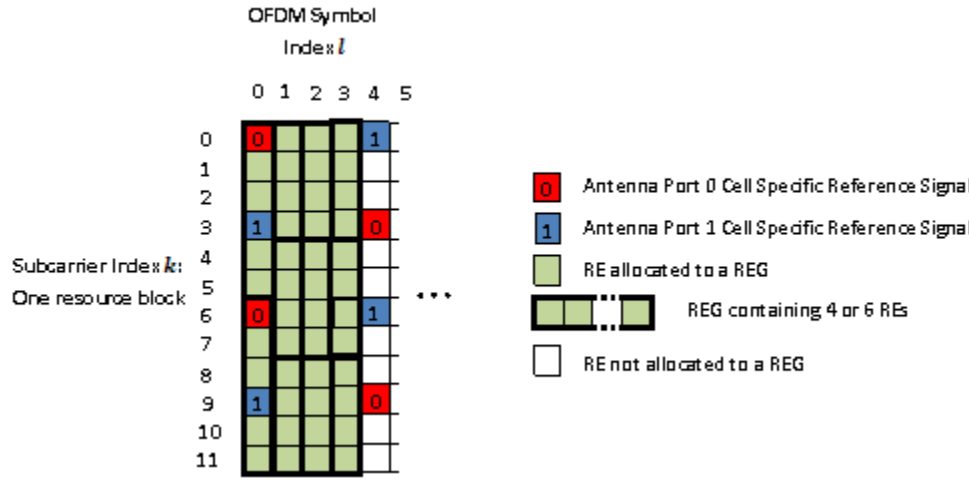
Each antenna port has a unique cell specific reference signal associated with it. As the REG arrangement is affected by cell specific reference signals, the REG arrangement for a one or two antenna port configuration or four antenna port configuration is different. The REG arrangement for each resource block within a subframe and for every antenna port is identical.

REG Arrangement with a Normal Cyclic Prefix

The REG arrangement for each antenna port configuration is described below for a normal cyclic prefix.

One or Two Antenna Port Configuration

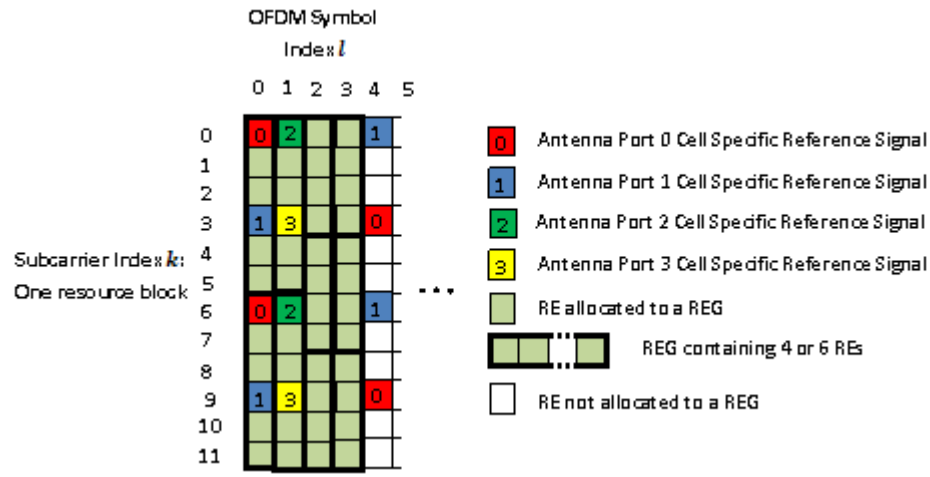
When antenna port 0 or ports 0 and 1 are used it is assumed the cell specific reference signal is present on both antenna ports 0 and 1. This leads to a REG arrangement for each resource block as shown in the following figure.



Cell-specific reference signals are present within the first OFDM symbol. As four REs not containing cell specific reference signals are required in a REG, the twelve REs in the first symbol are divided into two REGs, each containing six REs (two containing cell specific reference signals and four empty). In the second and third OFDM symbols no cell specific reference signal is present therefore the twelve REs in each symbol are divided bewteen three REGs, each containing four REs.

Four Antenna Port Configuration

The REG arrangement in each resource block for four antenna port configuration is shown in the following figure.



The REG allocation within the first OFDM symbol is the same as for a one or two antenna port configuration. Four cell specific reference signals are present in the second OFDM symbol therefore eight REs are available for the mapping of control data. The twelve REs are divided into two REGs, each containing six REs. The third and fourth OFDM symbols contain no reference signals so three REGs are available.

REG Arrangement with an Extended Cyclic Prefix

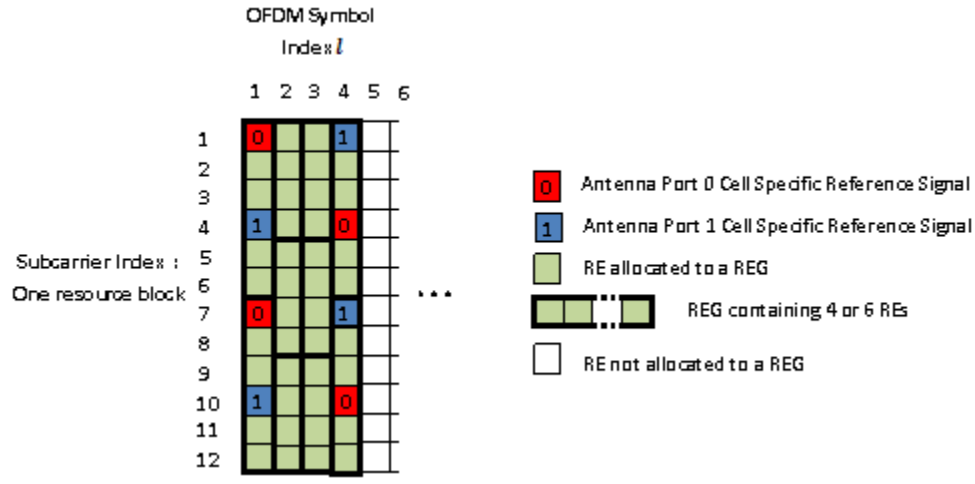
An extended cyclic prefix subframe contains twelve OFDM symbols as opposed to fourteen for a normal cyclic prefix. As the number of cell specific reference signals in a normal or extended cyclic prefix subframe is the same, the limited number of OFDM symbols in an extended cyclic prefix subframe requires the ODFM symbol spacing of the cell specific reference signals to be reduced compared to when using a standard cyclic prefix.

This reduction is spacing causes cell specific reference signals to be present within the fourth OFDM symbol of an extended cyclic prefix subframe whilst in a normal cyclic prefix subframe no cell specific reference signals are present. Therefore when an extended cyclic prefix is used two REGs, each containing six REs, are present in the fourth OFDM symbol.

The number of cell specific reference symbols within the first three OFDM symbols is identical for normal or extended cyclic prefix therefore the REG configurations are identical.

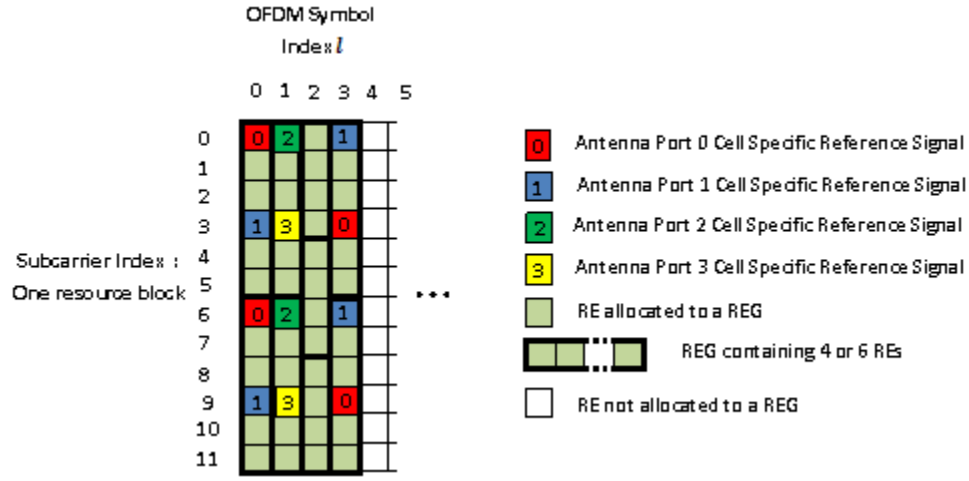
One or Two Antenna Port Configuration

The REG arrangement for a one or two antenna port configuration when using an extended cyclic prefix is shown in the following figure.



Four Antenna Port Configuration

The REG arrangement for a four antenna port configuration when using an extended cyclic prefix is shown in the following figure.



See Also

`ltePDCCH` | `ltePHICH` | `ltePCFICH`

Concepts

- "Data Structures"

Control Format Indicator (CFI) Channel

In this section...
"Control Format Indicator Values" on page 1-32
"PCFICH Resourcing" on page 1-32
"CFI Channel Coding" on page 1-32
"The PCFICH" on page 1-33

When transmitting data on the downlink in an OFDM communication system, it is important to specify how many OFDM symbols are used to transmit the control channels so the receiver knows where to find control information. In LTE, the Control Format Indicator (CFI) value defines the time span, in OFDM symbols, of the Physical Downlink Control Channel (PDCCH) transmission (the control region) for a particular downlink subframe. The CFI is transmitted using the Physical Control Format Indicator Channel (PCFICH).

Control Format Indicator Values

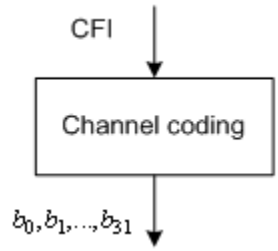
The CFI is limited to the value 1, 2, or 3. For bandwidths greater than ten resource blocks, the number of OFDM symbols used to contain the downlink control information is the same as the actual CFI value. Otherwise, the span of the downlink control information (DCI) is equal to *CFI*+1 symbols.

PCFICH Resourcing

The PCFICH is mapped in terms of Resource Element Groups (REGs) and is always mapped onto the first OFDM symbol. The number of REGs allocated to the PCFICH transmission is fixed to 4 i.e. 16 Resource Elements (REs). A PCFICH is only transmitted when the number of OFDM symbols for PDCCH is greater than zero.

CFI Channel Coding

The CFI value undergoes channel coding to form the PCFICH payload, as shown in the following figure.

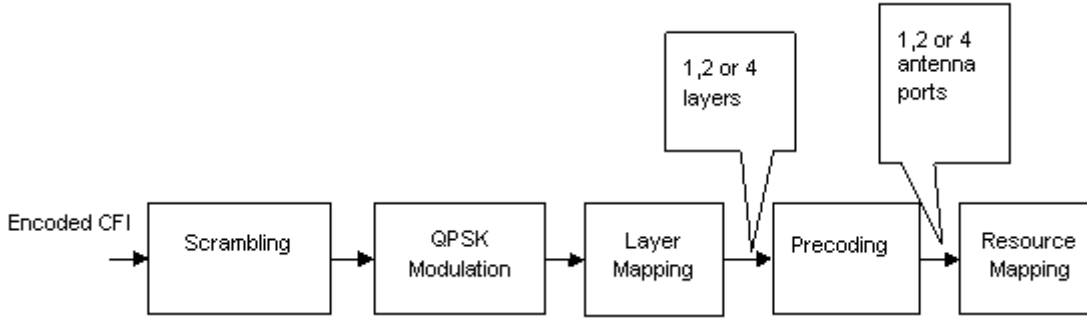


Using the following table contains the CFI codeword for each CFI value. Using these codewords corresponds to a block encoding rate of 1/16, changing a two bit CFI value to a 32 bit codeword.

CFI	CFI codeword <b_0, b_1, ... , b_{31}>
1	<0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1>
2	<1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0>
3	<1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1>
4 (Reserved)	<0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0>

The PCFICH

The coded CFI is scrambled before undergoing QPSK modulation, layer mapping and precoding as shown in the following figure.



- “Scrambling” on page 1-34

- "Modulation" on page 1-34
- "Layer Mapping" on page 1-68
- "Precoding" on page 1-69
- "Mapping to Resource Elements" on page 1-39

Scrambling

The 32-bit coded CFI block undergoes a bit-wise exclusive-or (XOR) operation with a cell-specific scrambling sequence. The scrambling sequence is a pseudo-random sequence created using a length-31 Gold sequence generator. At the start of each subframe, it is initialized using the slot number within the radio frame, n_s, and the cell ID, N_{ID}^{cell}.

$$c_{init} = \left(\left\lfloor \frac{n_s}{2} \right\rfloor + 1 \right) \times \left(2N_{ID}^{cell} + 1 \right) \times 2 + N_{ID}^{cell}$$

Scrambling with a cell specific sequence serves the purpose of intercell interference rejection. When a UE descrambles a received bit stream with a known cell specific scrambling sequence, interference from other cells will be descrambled incorrectly and will only appear as uncorrelated noise.

Modulation

The scrambled bits are then QPSK modulated to create a block of complex-valued modulation symbols.

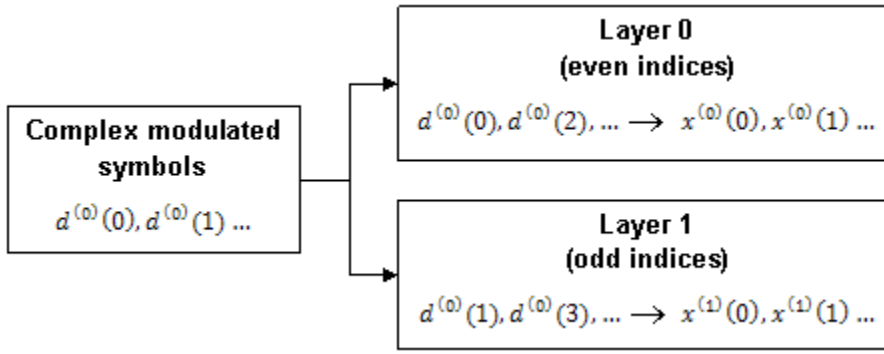
Layer Mapping

The complex symbols are mapped to one, two, or four layers depending on the number of transmit antennas used. The complex modulated input symbols, $d^{(0)}(i)$, are mapped onto v layers, $x^{(0)}(i), x^{(1)}(i), \ldots, x^{(v-1)}(i)$.

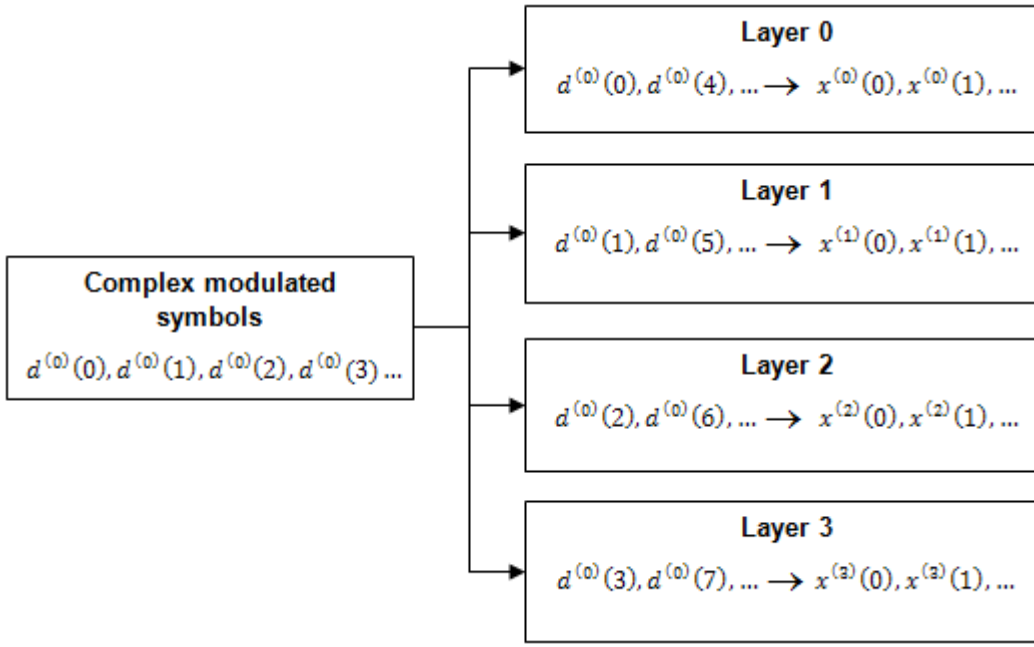
If a single antenna port is used, only one layer is used. Therefore, $x^{(0)}(i) = d^{(0)}(i)$.

If transmitter diversity is used, the input symbols are mapped to layers based on the number of layers.

- **Two Layers** — Even symbols are mapped to layer 0 and odd symbols are mapped to layer 1, as shown in the following figure.



- **Four Layers** — The input symbols are mapped to layers sequentially, as shown in the following figure.



Precoding

- "Two Antenna Port Precoding" on page 1-70
- "Four Antenna Port Precoding" on page 1-70

The precoder takes a block from the layer mapper, $x^{(0)}(i), x^{(1)}(i), \ldots, x^{(v-1)}(i)$, and generates a sequence for each antenna port, $y^{(p)}(i)$. The variable p is the transmit antenna port number, and can assume values of {0}, {0,1}, or {0,1,2,3}.

For transmission over a single antenna port, no processing is carried out, as shown in the following equation.

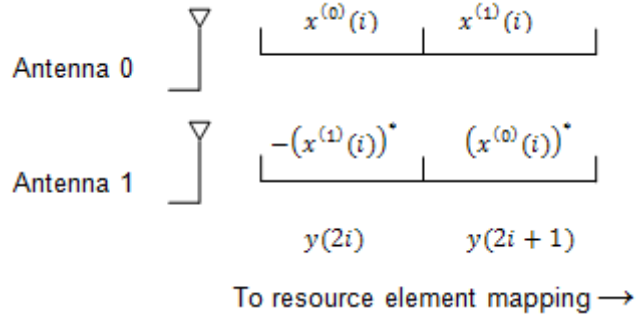
$$y^{(p)}(i) = x^{(0)}(i)$$

Precoding for transmit diversity is available on two or four antenna ports.

Two Antenna Port Precoding. An Alamouti scheme is used for precoding, which defines the relationship between input and output as shown in the following equation.

$$\begin{pmatrix} y^{(0)}(2i) \\ y^{(1)}(2i) \\ y^{(0)}(2i+1) \\ y^{(1)}(2i+1) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & j & 0 \\ 0 & -1 & 0 & j \\ 0 & 1 & 0 & j \\ 1 & 0 & -j & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}\left\{x^{(0)}(i)\right\} \\ \mathrm{Re}\left\{x^{(1)}(i)\right\} \\ \mathrm{Im}\left\{x^{(0)}(i)\right\} \\ \mathrm{Im}\left\{x^{(1)}(i)\right\} \end{pmatrix}$$

In the Alamouti scheme, two consecutive symbols, $x^{(0)}(i)$ and $x^{(1)}(i)$, are transmitted in parallel using two antennas with the following mapping, where the asterisk symbol (*) denotes the complex conjugate operation.

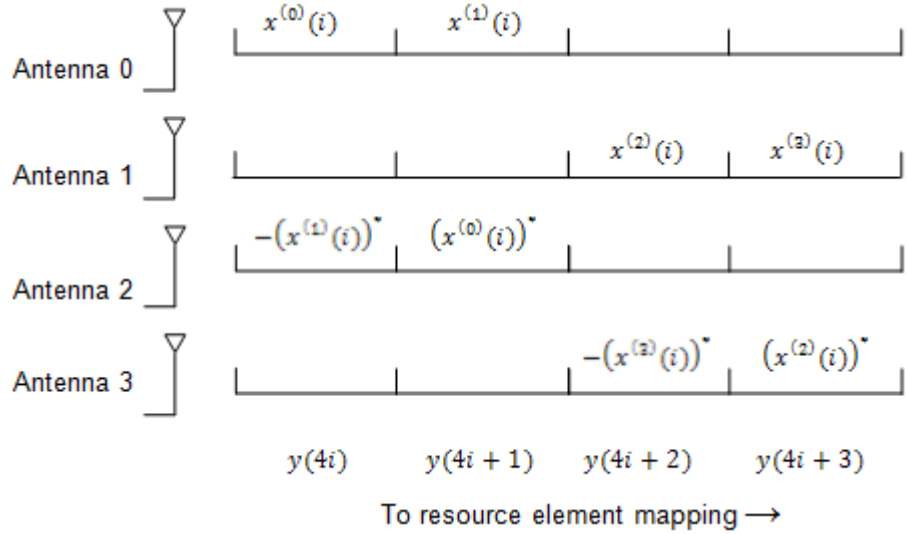


As any two columns in the precoding matrix are orthogonal, the two symbols, $x^{(0)}(i)$ and $x^{(1)}(i)$, can be separated at the UE.

Four Antenna Port Precoding. Precoding for the four antenna port case defines the relationship between the input and output as shown in the following equation.

$$\begin{pmatrix} y^{(0)}(4i) \\ y^{(1)}(4i) \\ y^{(2)}(4i) \\ y^{(3)}(4i) \\ y^{(0)}(4i+1) \\ y^{(1)}(4i+1) \\ y^{(2)}(4i+1) \\ y^{(3)}(4i+1) \\ y^{(0)}(4i+2) \\ y^{(1)}(4i+2) \\ y^{(2)}(4i+2) \\ y^{(3)}(4i+2) \\ y^{(0)}(4i+3) \\ y^{(1)}(4i+3) \\ y^{(2)}(4i+3) \\ y^{(3)}(4i+3) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & j & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -j & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}\left\{x^{(0)}(i)\right\} \\ \mathrm{Re}\left\{x^{(1)}(i)\right\} \\ \mathrm{Re}\left\{x^{(2)}(i)\right\} \\ \mathrm{Re}\left\{x^{(3)}(i)\right\} \\ \mathrm{Im}\left\{x^{(0)}(i)\right\} \\ \mathrm{Im}\left\{x^{(1)}(i)\right\} \\ \mathrm{Im}\left\{x^{(2)}(i)\right\} \\ \mathrm{Im}\left\{x^{(3)}(i)\right\} \end{pmatrix}$$

In this scheme, two consecutive symbols are transmitted in parallel in two symbol periods using four antennas with the following mapping, where the asterisk symbol (*) denotes the complex conjugate operation.



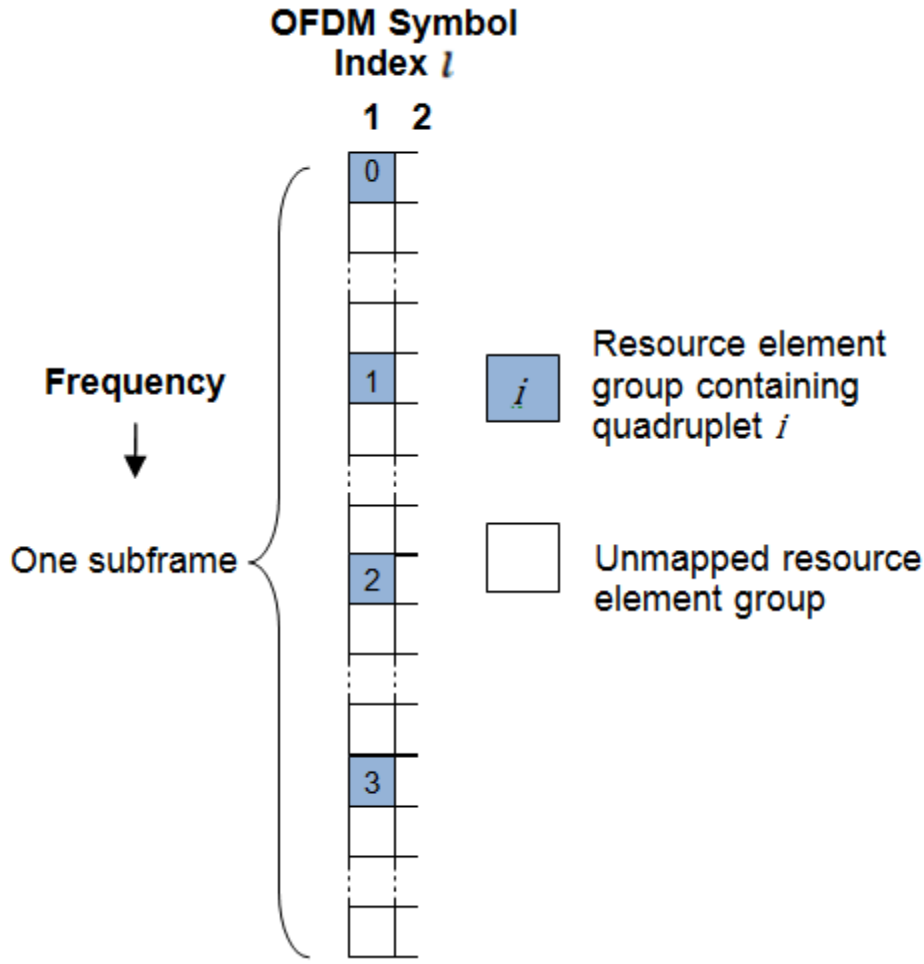
Mapping to Resource Elements

The complex valued symbols for each antenna are divided into quadruplets for mapping to resource elements. Each quadruplet is mapped to a Resource element Group (REG) within the first OFDM symbol. There are sixteen complex symbols to be mapped therefore four quadruplets are created.

The first quadruplet is mapped onto a REG with subcarrier index $k = \bar{k}$, given by the following equation.

$$\bar{k} = \left(\frac{N_{sc}^{RB}}{2}\right) \times \left(N_{ID}^{cell} \bmod 2N_{RB}^{DL}\right)$$

The subsequent three quadruplets are mapped to REGs spaced at intervals of $\left\lfloor N_{RB}^{DL} / 2 \right\rfloor \times \left(N_{sc}^{RB} / 2\right)$ from the first quadruplet and each other. This spreads the quadruplets, and hence the PCFICH, over the entire subframe as illustrated in the following figure.



See Also

`lteCFI` | `ltePCFICH` | `ltePCFICHInfo` | `ltePCFICHIndices` | `lteDLResourceGrid` | `lteSymbolModulate` | `lteSymbolDemodulate` | `ltePCFICHPRBS` | `lteLayerMap` | `lteLayerDemap` | `lteDLPrecode` | `lteDLDeprecode`

Related Examples

- “Model CFI and PCFICH” on page 3-4

HARQ Indicator (HI) Channel

In this section...

"HARQ Indicator" on page 1-41

"PHICH Groups" on page 1-41

"HARQ Indicator Channel Coding" on page 1-43

"The PHICH" on page 1-43

LTE uses a hybrid automatic repeat request (HARQ) scheme for error correction. The eNodeB sends a HARQ indicator to the UE to indicate a positive acknowledgement (ACK) or negative acknowledgement (NACK) for data sent using the uplink shared channel. The channel coded HARQ indicator codeword is transmitted through the Physical Hybrid Automatic Repeat Request Indicator Channel (PHICH).

HARQ Indicator

A HARQ indicator of '0' represents a NACK and a '1' represents an ACK.

PHICH Groups

Multiple PHICHs are mapped to the same set of resource elements (REs). This set of REs constitutes a PHICH group. The PHICHs within a PHICH group are separated through different orthogonal sequences.

A PHICH resource is identified by the index pair $\left(n_{PHICH}^{group}, n_{PHICH}^{seq}\right)$. The variable n_{PHICH}^{group} is the number of the PHICH group and the variable n_{PHICH}^{seq} is the orthogonal sequence index within the group. For more information about the orthogonal sequences, see "Scrambling" on page 1-44.

The number of PHICH groups varies based on whether the frame structure is type one, frequency division duplex (FDD), or type two, time division duplex (TDD).

Frame Structure Type 1: FDD

The number of PHICH groups is constant in all subframes and is given by the following equation.

$$N_{PHICH}^{group} = \begin{cases} \left\lceil N_g \left(N_{RB}^{DL} \right) / 8 \right\rceil, & \text{for normal cyclic prefix} \\ 2 \times \left\lceil N_g \left(N_{RB}^{DL} \right) / 8 \right\rceil, & \text{for extended cyclic prefix} \end{cases}$$

The set $N_g \in \left\{ \frac{1}{6}, \frac{1}{2}, 1, 2 \right\}$ is provided by higher layers and is a scaling factor to control the number of PHICH groups.

The index of the PHICH group n_{PHICH}^{group} ranges from 0 to $n_{PHICH}^{group} - 1$.

Frame Structure Type 2: TDD

The number of PHICH groups varies depending on the number of the downlink subframe and the uplink/downlink time division duplex

configuration. The number of groups is given by the expression $m_i \times N_{PHICH}^{group}$.

The variable n_{PHICH}^{group} is the number of PHICH groups for a frame structure type 1. The variable m_i is dependent on the subframe. The value for m_i for each uplink-downlink configuration and subframe number is given in the following table.

Uplink-downlink configuration	Subframe number *i*									
	0	1	2	3	4	5	6	7	8	9
0	2	1	—	—	—	2	1	—	—	—
1	0	1	—	—	1	0	1	—	—	1
2	1	0	—	—	—	0	0	0	1	1
3	1	0	—	—	—	0	0	0	1	1
4	0	0	—	—	0	0	0	0	1	1

Uplink-downlink configuration	Subframe number *i*									
5	0	0	—	0	0	0	0	0	1	0
6	1	1	—	—	—	1	1	—	—	1

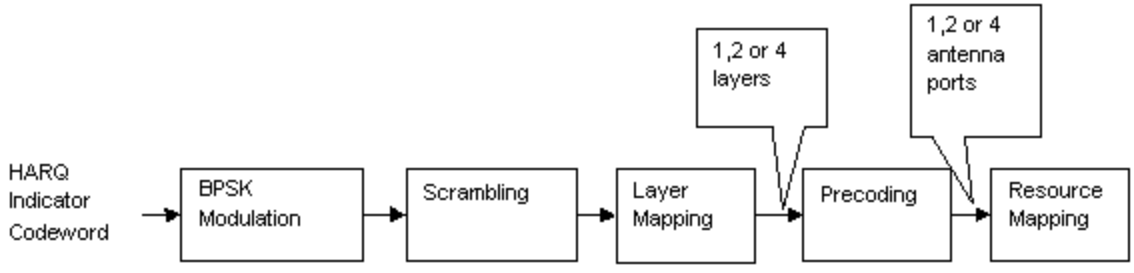
HARQ Indicator Channel Coding

The HARQ Indicator undergoes repetition coding to create a HARQ indicator codeword made up of three bits, $\langle b_0, b_1, b_2 \rangle$

HARQ Indicator	HARQ Indicator Codeword $\langle b_0, b_1, b_2 \rangle$
0 — Negative acknowledgement	$\langle 0,0,0 \rangle$
1 — Positive acknowledgement	$\langle 1,1,1 \rangle$

The PHICH

The HARQ Indicator codeword undergoes BPSK modulation, scrambling, layer mapping, precoding, and resource mapping as shown in block diagram in the following figure.



Modulation

The HARQ indicator codeword undergoes BPSK modulation resulting in a block of complex-valued modulated symbols, $z(0)$, $z(1)$, $z(2)$.

Scrambling

The block of modulated symbols is bitwise multiplied with an orthogonal sequence and a cell-specific scrambling sequence to create a sequence of symbols, $d(0),\ldots,d(M_{symb}-1)$. The number of symbols, M_{symb}, is given by the equation $M_{symb}=3\times N_{SF}^{PHICH}$. The PHICH spreading factor, N_{SF}^{PHICH}, is 4 for a normal cyclic prefix and 2 for an extended cyclic prefix.

The orthogonal sequence allows multiple PHICHs to be mapped to the same set of resource elements.

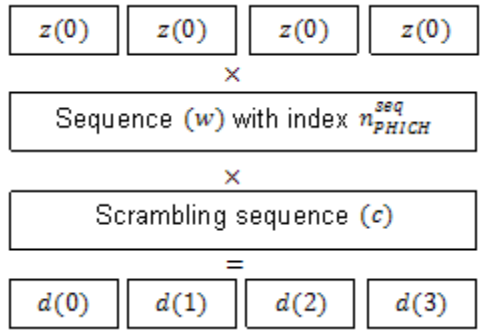
Scrambling with a cell-specific sequence serves the purpose of intercell interference rejection. When a UE descrambles a received bitstream with a known cell specific scrambling sequence, interference from other cells will be descrambled incorrectly and therefore only appear as uncorrelated noise.

The complex scrambled symbols, $d(0),\ldots,d(M_{symb}-1)$, are created according to the following equation.

$$d(i)=w\left(i \bmod N_{SF}^{PHICH}\right)\times\left(1-2c(i)\right)\times z\left(\left\lfloor i/N_{SF}^{PHICH}\right\rfloor\right)$$

The first term, $w\left(i \bmod N_{SF}^{PHICH}\right)$, is the orthogonal sequence symbol with index N_{SF}^{PHICH}. The second term, $\left(1-2c(i)\right)$, is the cell-specific scrambling sequence symbol. The third term, $z\left(\left\lfloor i/N_{SF}^{PHICH}\right\rfloor\right)$, is the modulated HARQ indicator symbol.

The three modulated symbols, $z(0)$, $z(1)$, $z(2)$, are repeated N_{SF}^{PHICH} times and scrambled to create a sequence of six or twelve symbols depending on whether a normal or extended cyclic prefix is used. When using a normal cyclic prefix, the first four scrambled symbols are created as shown in the following figure.



The variable w is an orthogonal scrambling sequence with index n_{PHICH}^{seq}. The sequences are given in the following table.

Sequence Index n_{PHICH}^{seq}	**Orthogonal Sequence,** $w(0),\ldots,w\left(N_{SF}^{PHICH}-1\right)$	
	Normal Cyclic Prefix, $N_{SF}^{PHICH}=4$	**Extended Cyclic Prefix,** $N_{SF}^{PHICH}=2$
0	[+1 +1 +1 +1]	[+1 +1]
1	[+1 –1 +1 –1]	[+1 –1]
2	[+1 +1 –1 –1]	[+j +j]
3	[+1 –1 –1 +1]	[+j –j]
4	[+j +j +j +j]	—
5	[+j –j +j –j]	—
6	[+j +j –j –j]	—
7	[+j –j –j +j]	—

The variable c is a cell-specific pseudo-random scrambling sequence created using a length-31 Gold sequence. The scrambling sequence is initialized using the slot number within the radio frame,n_s, and the cell ID, N_{ID}^{cell}.

$$c_{init} = \left(\lfloor n_s / 2 \rfloor + 1\right) \times \left(2N_{ID}^{cell} + 1\right) \times 2^9 + N_{ID}^{cell}$$

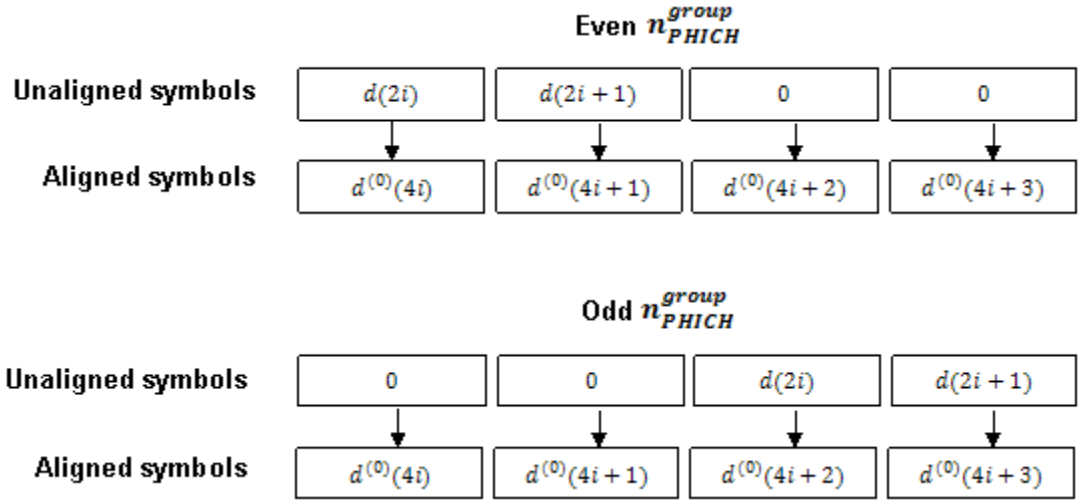
Resource Group Alignment

As resource element groups (REGs) contain four resource elements (each able to contain one symbol) the blocks of scrambled symbols are aligned to create blocks of four symbols.

In the case of a normal cyclic prefix, each of the original complex modulated symbols, $z(0)$, $z(1)$, $z(2)$, is represented by four scrambled symbols. Therefore, no alignment is required, as shown in the following equation.

$$d^{(0)}(i) = d(i)$$

In the case of an extended cyclic prefix each of the original complex modulated symbols, $z(0)$, $z(1)$, $z(2)$, is represented by two scrambled symbols. To create blocks of four symbols, zeros are added before or after blocks of two scrambled symbols depending on whether the PHICH index is odd or even. This allows two groups to be combined during the resource mapping stage and mapped to one REG. Groups of four symbols, $d^{(0)}$, are formed as shown in the following figure.



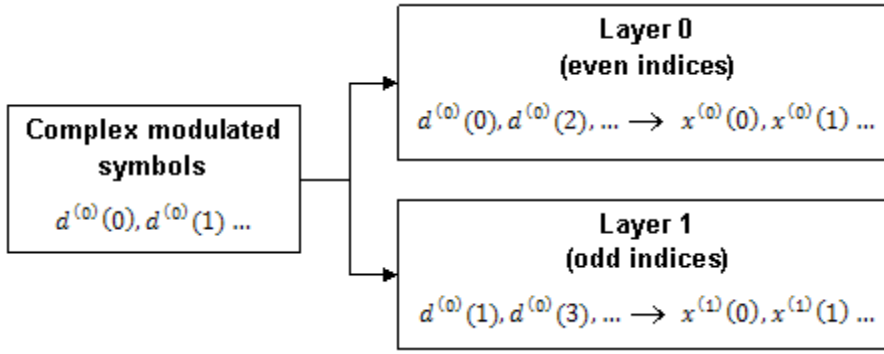
Layer Mapping

The complex symbols are mapped to one, two, or four layers depending on the number of transmit antennas used. The complex modulated input symbols, $d^{(0)}(i)$, are mapped onto v layers, $x^{(0)}(i), x^{(1)}(i), \ldots, x^{(v-1)}(i)$.

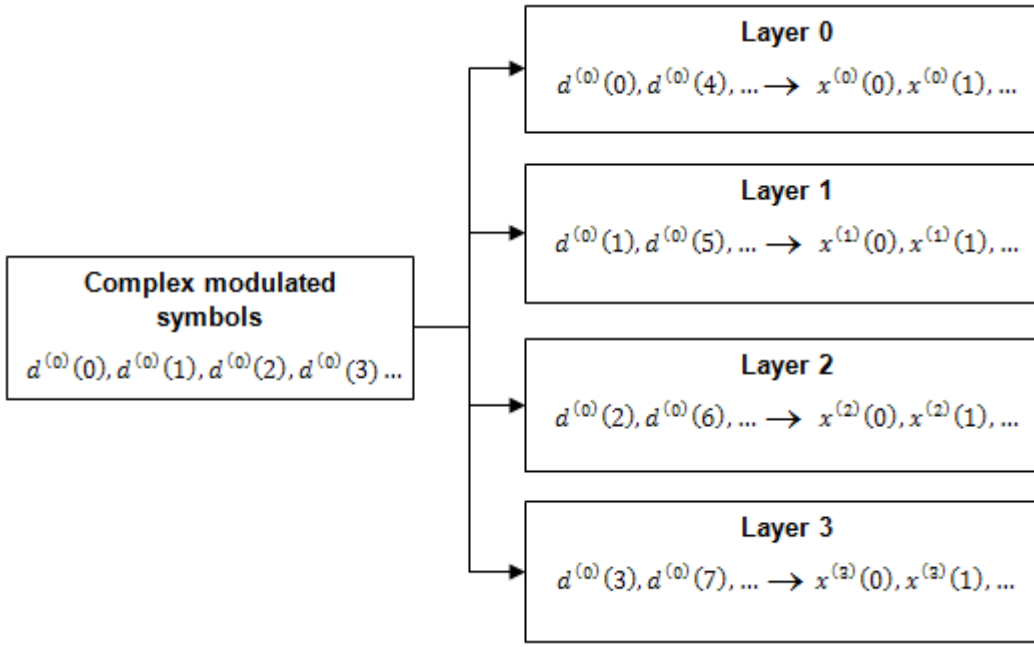
If a single antenna port is used, only one layer is used. Therefore, $x^{(0)}(i) = d^{(0)}(i)$.

If transmitter diversity is used, the input symbols are mapped to layers based on the number of layers.

- **Two Layers** — Even symbols are mapped to layer 0 and odd symbols are mapped to layer 1, as shown in the following figure.



- **Four Layers** — The input symbols are mapped to layers sequentially, as shown in the following figure.



Precoding

The precoder takes a block from the layer mapper, $x^{(0)}(i), x^{(1)}(i), \ldots, x^{(\upsilon-1)}(i)$, and generates a sequence for each antenna port, $y^{(p)}(i)$. The variable p is the transmit antenna port number, and can assume values of {0}, {0,1}, or {0,1,2,3}.

For transmission over a single antenna port, no processing is carried out, as shown in the following equation.

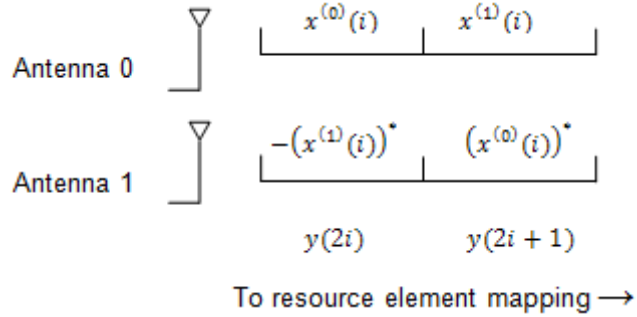
$$y^{(p)}(i) = x^{(0)}(i)$$

Precoding for transmit diversity is available on two or four antenna ports.

Two Antenna Port Precoding. An Alamouti scheme is used for precoding, which defines the relationship between input and output as shown in the following equation.

$$\begin{pmatrix} y^{(0)}(2i) \\ y^{(1)}(2i) \\ y^{(0)}(2i+1) \\ y^{(1)}(2i+1) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & j & 0 \\ 0 & -1 & 0 & j \\ 0 & 1 & 0 & j \\ 1 & 0 & -j & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}\left\{x^{(0)}(i)\right\} \\ \mathrm{Re}\left\{x^{(1)}(i)\right\} \\ \mathrm{Im}\left\{x^{(0)}(i)\right\} \\ \mathrm{Im}\left\{x^{(1)}(i)\right\} \end{pmatrix}$$

In the Alamouti scheme, two consecutive symbols, $x^{(0)}(i)$ and $x^{(1)}(i)$, are transmitted in parallel using two antennas with the following mapping, where the asterisk symbol (*) denotes the complex conjugate operation.

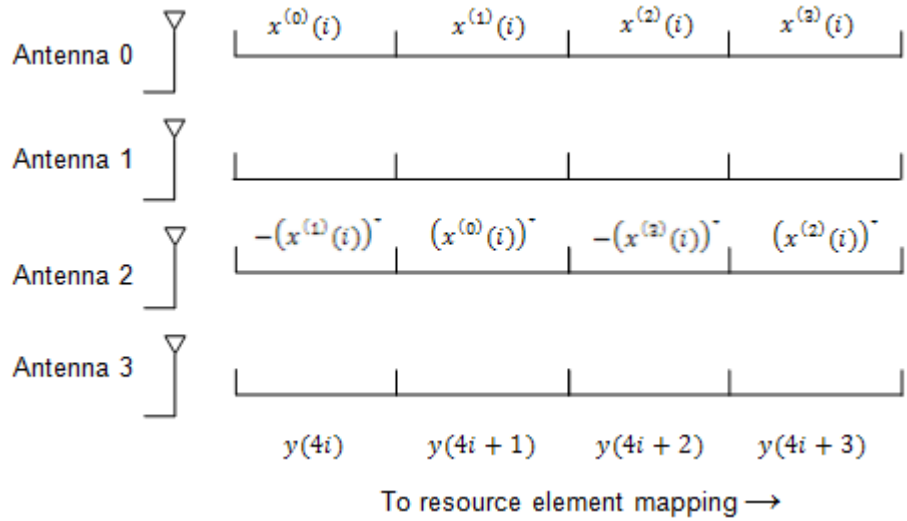


As any two columns in the precoding matrix are orthogonal, the two symbols, $x^{(0)}(i)$ and $x^{(1)}(i)$, can be separated at the UE.

Four Antenna Port Precoding. Precoding for the four antenna port case depends on the index of the PHICH group, n_{PHICH}^{group}. If $n_{PHICH}^{group} + i$ is even for a normal cyclic prefix or if $\left\lfloor n_{PHICH}^{group} / 2 \right\rfloor + i$ is even for an extended cyclic prefix, the relationship between the input and output is defined by the following equation.

$$\begin{pmatrix} y^{(0)}(4i) \\ y^{(1)}(4i) \\ y^{(2)}(4i) \\ y^{(3)}(4i) \\ y^{(0)}(4i+1) \\ y^{(1)}(4i+1) \\ y^{(2)}(4i+1) \\ y^{(3)}(4i+1) \\ y^{(0)}(4i+2) \\ y^{(1)}(4i+2) \\ y^{(2)}(4i+2) \\ y^{(3)}(4i+2) \\ y^{(0)}(4i+3) \\ y^{(1)}(4i+3) \\ y^{(2)}(4i+3) \\ y^{(3)}(4i+3) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & j & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -j & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}\left\{x^{(0)}(i)\right\} \\ \mathrm{Re}\left\{x^{(1)}(i)\right\} \\ \mathrm{Re}\left\{x^{(2)}(i)\right\} \\ \mathrm{Re}\left\{x^{(3)}(i)\right\} \\ \mathrm{Im}\left\{x^{(0)}(i)\right\} \\ \mathrm{Im}\left\{x^{(1)}(i)\right\} \\ \mathrm{Im}\left\{x^{(2)}(i)\right\} \\ \mathrm{Im}\left\{x^{(3)}(i)\right\} \end{pmatrix}$$

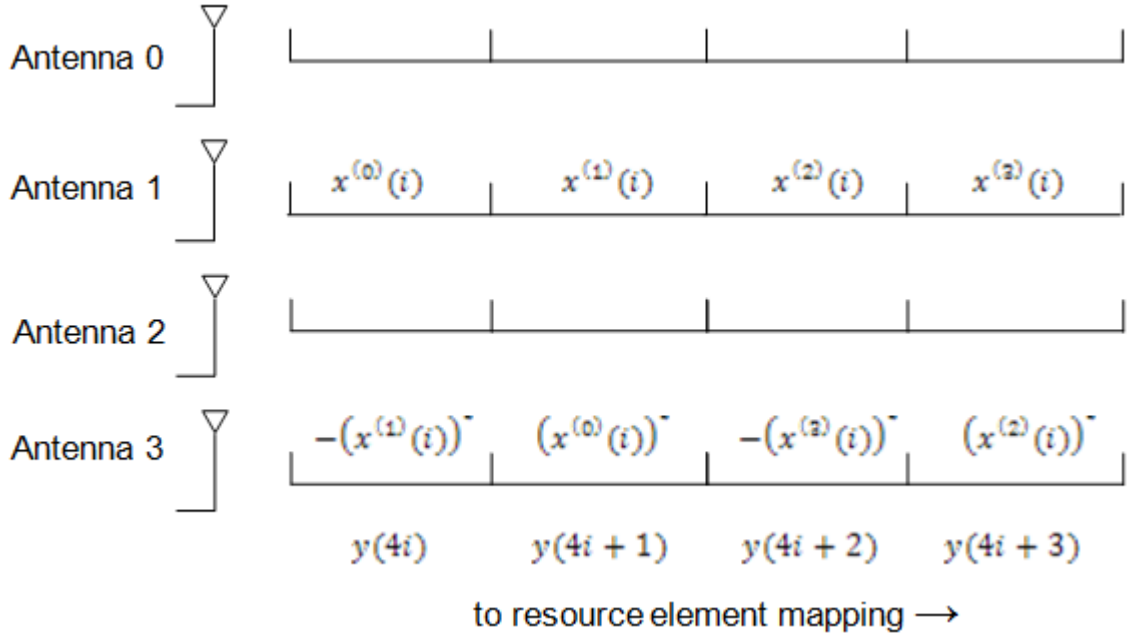
In this scheme, two consecutive symbols are transmitted in parallel in two symbol periods using four antennas with the following mapping, where the asterisk symbol (*) denotes the complex conjugate operation.



If $n_{PHICH}^{group} + i$ is odd for a normal cyclic prefix or if $\left\lfloor n_{PHICH}^{group} / 2 \right\rfloor + i$ is odd for an extended cyclic prefix, the relationship between the input and output is defined by the following equation.

$$\begin{pmatrix} y^{(0)}(4i) \\ y^{(1)}(4i) \\ y^{(2)}(4i) \\ y^{(3)}(4i) \\ y^{(0)}(4i+1) \\ y^{(1)}(4i+1) \\ y^{(2)}(4i+1) \\ y^{(3)}(4i+1) \\ y^{(0)}(4i+2) \\ y^{(1)}(4i+2) \\ y^{(2)}(4i+2) \\ y^{(3)}(4i+2) \\ y^{(0)}(4i+3) \\ y^{(1)}(4i+3) \\ y^{(2)}(4i+3) \\ y^{(3)}(4i+3) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & j & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -j & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}\left\{x^{(0)}(i)\right\} \\ \mathrm{Re}\left\{x^{(1)}(i)\right\} \\ \mathrm{Re}\left\{x^{(2)}(i)\right\} \\ \mathrm{Re}\left\{x^{(3)}(i)\right\} \\ \mathrm{Im}\left\{x^{(0)}(i)\right\} \\ \mathrm{Im}\left\{x^{(1)}(i)\right\} \\ \mathrm{Im}\left\{x^{(2)}(i)\right\} \\ \mathrm{Im}\left\{x^{(3)}(i)\right\} \end{pmatrix}$$

In this scheme, two consecutive symbols are transmitted in parallel in two symbol periods using four antennas with the following mapping, where the asterisk symbol (*) denotes the complex conjugate operation.



Mapping to Resource Elements

PHICH Duration. The number of OFDM symbols used to carry the PHICH is configurable by the PHICH duration.

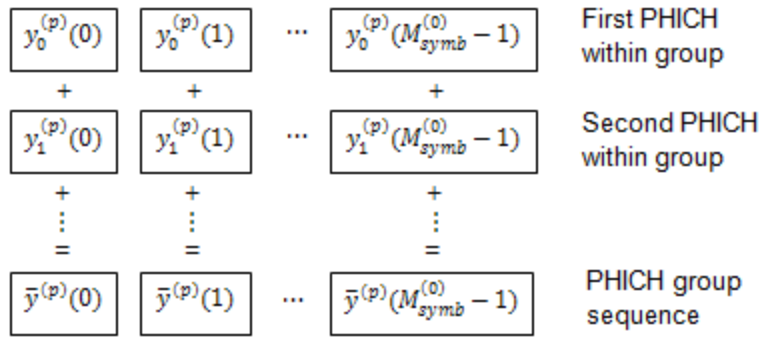
The PHICH duration is either normal or extended. A normal PHICH duration causes the PHICH to be present in only the first OFDM symbol. In general an extended PHICH duration causes the PHCH to be present in the first three OFDM symbols but there are some exceptions. The PHICH is present in the first two OFDM symbols under the following exceptions.

- Within subframe 1 and 6 when frame structure type 2 (TDD) is used
- Within MBSFN subframes

Relationship between CFI and PHICH Duration. Since the control format indicator (CFI) configures how many OFDM symbols are used for mapping the physical downlink control channel (PDCCH) and hence which OFDM symbols are available for the physical downlink shared channel (PDSCH), care must be taken when using an extended PHICH duration so the PHICH is not mapped into the same region as the PDSCH.

For example, when using an extended PHICH in subframe zero of a frame structure type 1 (FDD) 10MHz subframe the first three OFDM symbols will contain PHICH. Therefore, the CFI must be set to 3 so the PDSCH is not mapped to OFDM symbols 0, 1, or 2, and so does not overlap with the PHICH.

Combining PHICH Sequences. Corresponding elements of each PHICH sequence are summed to create the sequence for each PHICH group, $\left(\bar{y}^{(p)}\right)$. This process is illustrated in the following figure.



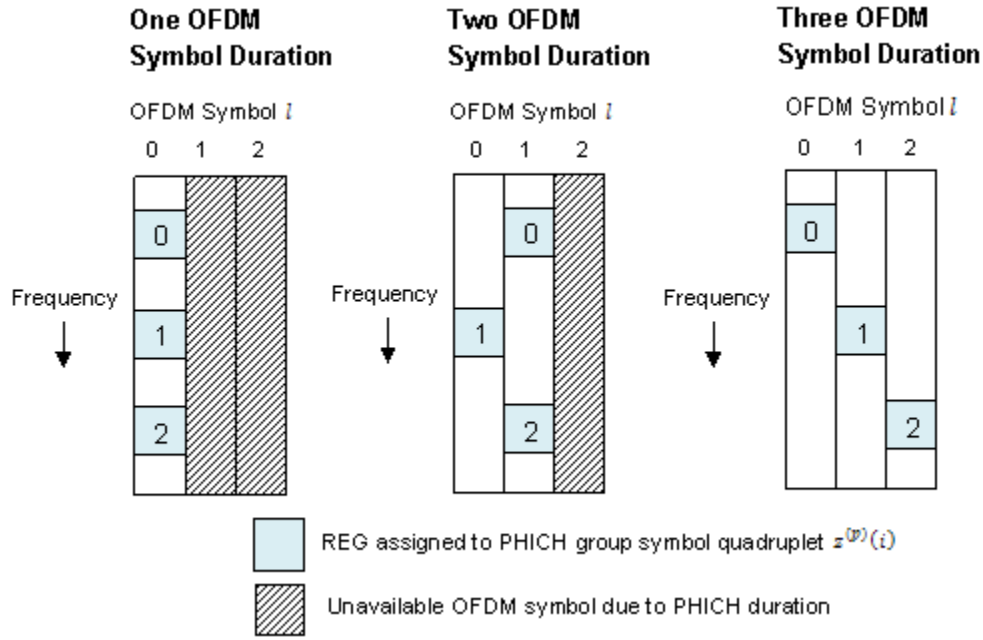
The variable $y_i^{(p)}(n)$ is the n-th element within PHICH i on antenna port p.

PHICH Mapping Units. PHICHs are mapped to REGs using PHICH mapping units, $\tilde{y}_{m'}^{(p)}$, where m' is the index of the mapping unit. For a normal cyclic prefix, each PHICH group is mapped to a PHICH mapping unit. In the case of an extended cyclic prefix, two PHICH groups are mapped to one PHICH mapping unit. Due to the location of the padding zeros added during resource group alignment, when two consecutive groups are added, the zeros of one group overlap the data of the other.

Mapping to REGs. Each mapping unit contains twelve symbols. To map these twelve symbols to REGs, the mapping units are split into three groups of four symbols (quadruplets).

Each of the three symbol quadruplets, $z^{(p)}(i), i = \{0,1,2\}$, is mapped to a REG, (k',l'), so the PHICH is spread over all available OFDM symbols and resource blocks.

The OFDM symbol index, l', is set so adjacent quadruplets are spread amongst the available OFDM symbols, as illustrated in the following figure.



The subcarrier index, k_i', of the REG is based upon the cell ID, N_{ID}^{cell}, and is chosen to spread the three symbol quadruplets over the entire bandwidth.

See Also

`ltePHICH` | `ltePHICHInfo` | `ltePHICHIndices` | `ltePHICHPRBS` | `lteDLResourceGrid` | `lteLayerMap` | `lteLayerDemap` | `lteDLPrecode` |

`lteDLDeprecode` | `lteCRCEncode` | `lteCRCDecode` | `lteSymbolModulate` | `lteSymbolDemodulate`

Related Examples

- "Model HARQ Indicator and PHICH" on page 3-7

Downlink Control Channel

In this section...
"DCI Message Formats" on page 1-57
"PDCCH Restructuring" on page 1-58
"DCI Message Generation" on page 1-58
"DCI Coding" on page 1-58
"The PDCCH" on page 1-63

Control signaling is required to support the transmission of the downlink and uplink transport channels (DL-SCH and UL-SCH). Control information for one or multiple UEs is contained in a Downlink scheduling Control Information (DCI) message and is transmitted through the Physical Downlink Control Channel (PDCCH). DCI messages contain the following information.

- DL-SCH resource allocation (the set of resource blocks containing the DL-SCH) and modulation and coding scheme, which allows the UE to decode the DL-SCH.
- Transmit Power Control (TPC) commands for the Physical Uplink Control Channel (PUCCH) and UL-SCH, which adapt the transmit power of the UE to save power
- Hybrid-Automatic Repeat Request (HARQ) information including the process number and redundancy version for error correction
- MIMO precoding information

DCI Message Formats

Depending on the purpose of DCI message, different DCI formats are defined. The DCI formats are given in the following list.

- **Format 0** — for transmission of uplink shared channel (UL-SCH) allocation
- **Format 1** — for transmission of DL-SCH allocation for Single Input Multiple Output (SIMO) operation

- **Format 1A** — for compact transmission of DL-SCH allocation for SIMO operation or allocating a dedicated preample signature to a UE for random access
- **Format 1B** — for transmission control information of multiple-input multiple-output (MIMO) rank-1 based compact resource assignment
- **Format 1C** — for very compact transmission of PDSCH assignment
- **Format 1D** — same as *Format 1B*, but with additional information of power offset
- **Format 2** and **Format 2A**— for transmission of DL-SCH allocation for closed and open loop MIMO operation, respectively
- **Format 3** and **Format 3A**— for transmission of TPC command for an uplink channel

In one subframe, multiple UE's can be scheduled. Therefore, multiple DCI messages can be sent using multiple PDCCH's.

PDCCH Restructuring

A PDCCH is transmitted on one or an aggregation of several consecutive control channel elements (CCEs). A CCE is a group of nine consecutive resource-element groups (REGs). The number of CCEs used to carry a PDCCH is controlled by the PDCCH format. A PDCCH format of 0, 1, 2, or 3 corresponds to 1, 2, 3, or 4 consecutive CCEs being allocated to one PDCCH.

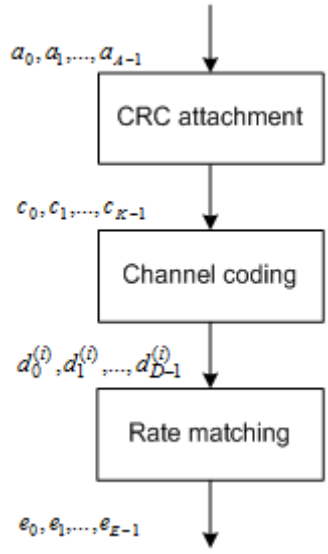
DCI Message Generation

The base station creates a DCI message based on a DCI format given in section 5.3.3.1 of [1]. Each field in a DCI message is mapped in order. Zeros may be appended to a DCI message to avoid ambiguous message lengths.

DCI Coding

- "CRC Attachment" on page 1-59
- "Channel Coding — Tail-Biting Convolutional Coding" on page 1-60
- "Rate Matching" on page 1-62

To form the PDCCH payload, the DCI undergoes coding as shown in the following figure.



CRC Attachment

A cyclic redundancy check (CRC) is used for error detection in DCI messages. The entire PDCCH payload is used to calculate a set of CRC parity bits. The PDCCH payload is divided by a cyclic generator polynomial to generate 16 parity bits. These parity bits are then appended to the end of the PDCCH payload.

As multiple PDCCHs relevant to different UEs can be present in one subframe, the CRC is also used to specify to which UE a PDCCH is relevant. This is done by scrambling the CRC parity bits with the corresponding Radio Network Temporary Identifier (RNTI) of the UE. The scrambled CRC is obtained by performing a bit-wise XOR operation between the 16-bit calculated PDCCH CRC and the 16-bit RNTI.

Different RNTI can be used to scramble the CRC. The following RNTI are some examples.

- A UE unique identifier; for example, a Cell-RNTI
- A paging indication identifier, or Paging-RNTI, if the PDCCH contains paging information
- A system information identifier, or system information-RNTI, if the PDCCH contains system information

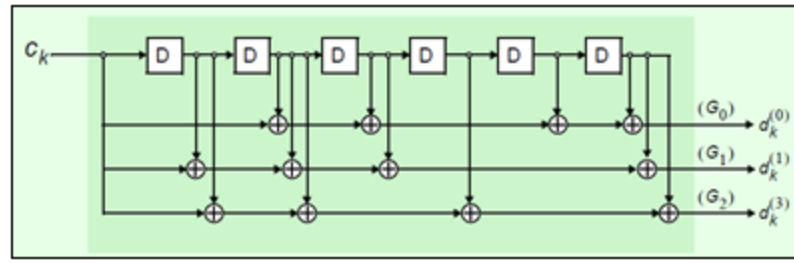
When encoding a format 0 DCI message, which contains the UE UL-SCH resource allocation, and the UE transmit antenna selection is configured and applicable, the RNTI scrambled CRC undergoes a bit-wise XOR operation with an antenna selection mask. This mask informs the UE transmit antenna on which port to transmit. The antenna selection masks are given in the following table.

UE transmit antenna selection	Antenna selection mask, $<x_0^{AS}, ..., x_{15}^{AS}>$
UE Port 0	<0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0>
UE Port 1	<0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1>

Channel Coding — Tail-Biting Convolutional Coding

The DCI message with the CRC attachment undergoes tail biting convolutional coding. Convolutional coding is a form of forward error correction and improves the channel capacity by adding carefully selected redundant information.

LTE uses a rate $1/3$ tail-biting encoder with a constraint length, k, of 7. This means that one in three bits of the output contain useful information while the other two add redundancy. The structure of the convolutional encoder is shown in the following figure.



Each output stream of the coder is obtained by convolving the input with the impulse response of the encoder, as shown in the following equation.

$$d_k^{(i)} \rightarrow C_k * G_i$$

The impulse responses are called the generator sequences of the coder. For LTE, there are the following three generator sequences.

- G_0=133 (octal)
- G_0=177 (octal)
- G_0=165 (octal)

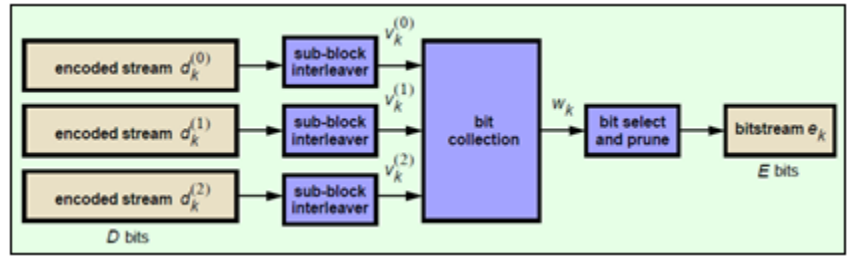
A standard convolutional encoder initializes its internal shift register to the *all zeros* state, and also ensures that the coder finishes in the *all zeros* state by padding the input sequence with k zeros at the end. Knowing the start and end states, which are all zeros, simplifies the design of the decoder, which is typically an implementation of the *Viterbi* algorithm.

A tail biting convolutional coder initializes its internal shift register to the last k bits of the current input block, rather than to the *all zeros* state. Thus, the start and end states are the same, without the need to zero-pad the input block. Since the overhead of terminating the coder has been eliminated, the output block contains fewer bits than a standard convolutional coder. The drawback is that the decoder becomes more complicated because the initial state is unknown; however, the decoder does know the start and end states are the same.

Rate Matching

- "Sub-block Interleaver" on page 1-62
- "Bit Collection, Selection, and Transmission" on page 1-63

The rate matching block creates an output bitstream with a desired code rate. As the number of bits available for transmission depends on the available resources the rate matching algorithm is capable of producing any arbitrary rate. The three bitstreams from the turbo encoder are interleaved followed by bit collection to create a circular buffer. Bits are selected and pruned from the buffer to create an output bitstream with the desired code rate. The process is illustrated in the following figure.



Sub-block Interleaver. The three sub-block interleavers used in the rate matching block are identical. Interleaving is a technique to reduce the impact of burst errors on a signal as consecutive bits of data will not be corrupted.

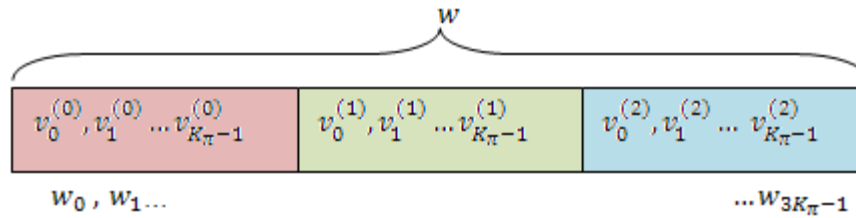
The sub-block interleaver reshapes the encode bit sequence, row-by-row, to form a matrix with $C_{Subblock}^{TC} = 32$ columns and $R_{Subblock}^{TC}$ rows. The variable $R_{Subblock}^{TC}$ is determined by finding the minimum integer such that the number of encoded input bits is $D \le \left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right)$. If $\left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right) > D$, N_D <NULL>'s are appended onto the front of the encoded sequence. In this case, $N_D + D = \left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right)$.

Inter-column permutation is performed on the matrix to reorder the columns as shown in the following pattern.

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

The output of the block interleaver is the bit sequence read out column-by-column from the inter-column permutated matrix to create a stream $K_\pi = \left(R^{TC}_{Subblock} \times C^{TC}_{Subblock}\right)$ bits long.

Bit Collection, Selection, and Transmission. The bit collection stage creates a virtual circular buffer by combining the three interleaved encoded bit streams, as shown in the following figure.



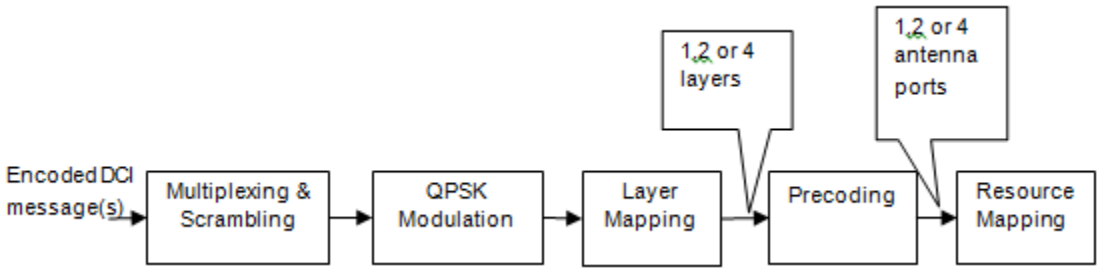
Bits are then selected and pruned from the circular buffer to create an output sequence length which meets the desired code rate. This is achieved by sequentially outputting the bits in the circular buffer from w_0 (looping back to w_0 after $w_{3K_\pi-1}$), discarding <NULL> bits, until the length of the output is x times the length of the input, creating a coding rate of $1/x$.

The PDCCH

- "Multiplexing" on page 1-64
- "Matching PDCCHs to CCE Positions" on page 1-64
- "Scrambling" on page 1-67
- "Modulation" on page 1-67
- "Layer Mapping" on page 1-68

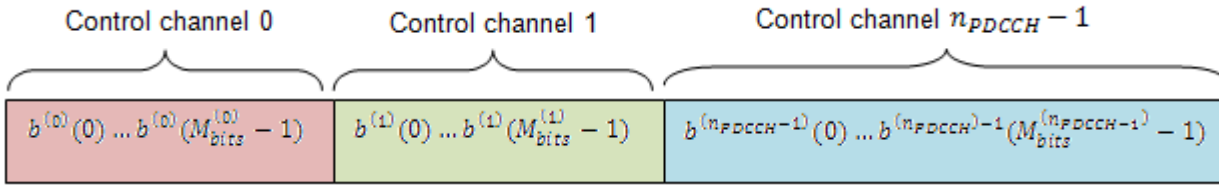
- "Precoding" on page 1-69
- "Mapping to Resource Elements" on page 1-72

The coded DCI messages for each control channel are multiplexed, scrambled, and undergo QPSK modulation, layer mapping, and precoding, as shown in the following figure.



Multiplexing

The blocks of coded bits for each control channel are multiplexed in order to create a block of data, as shown in the following figure.



The variable M_{bits}^{i} is the number of bits in the i^{th} control channel and n_{PDCCH} is the number of control channels.

Matching PDCCHs to CCE Positions

If necessary, `<NIL>` elements are inserted in the block of bits prior to scrambling to ensure PDCCHs start at particular CCE positions and the length of the block of bits matches the amount of REGs not assigned to PCFICH or PHICH.

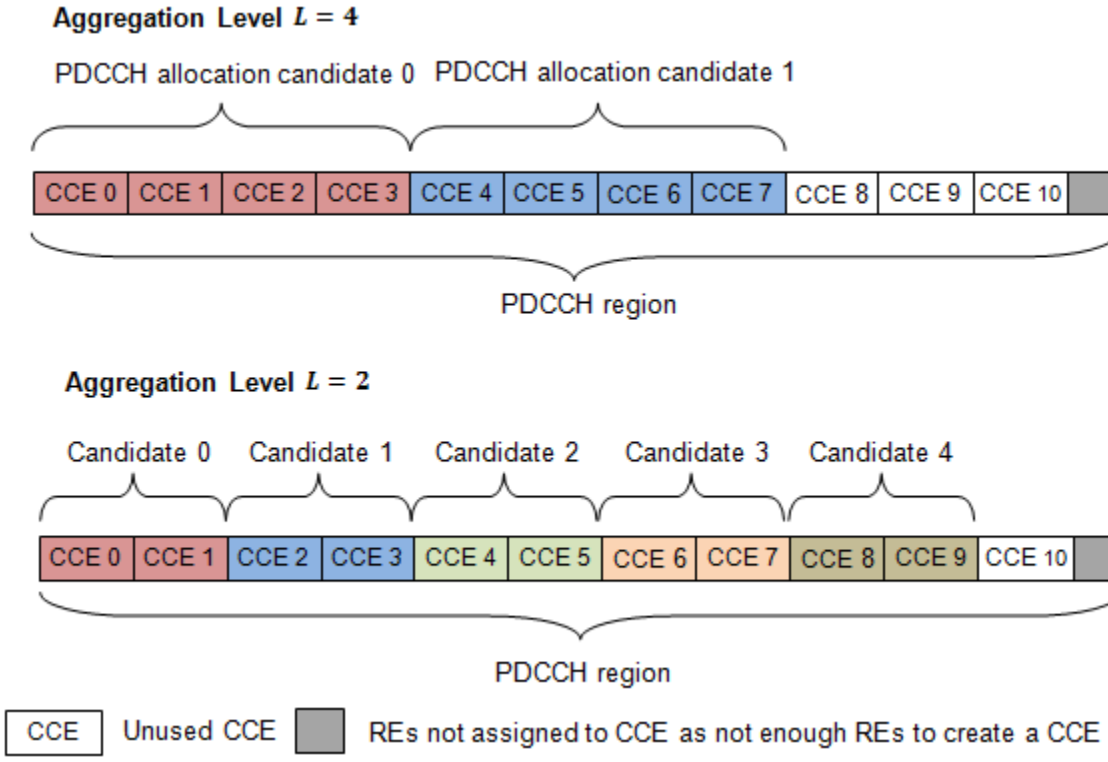
The PDCCH region consists of CCEs which could be allocated to a PDCCH. The configuration of how PDCCHs are mapped to CCEs is flexible.

Common and UE-specific PDCCHs are mapped to CCEs differently; each type has a specific set of search spaces associated with it. Each search space consists of a group of consecutive CCEs which could be allocated to a PDCCH called a PDCCH candidate. The CCE aggregation level is given by the PDCCH format and determines the number of PDCCH candidates in a search space. The number of candidates and size of the search space for each aggregation level is given in the following table.

Search space, $S_k^{(L)}$			**Number of PDCCH**
Type	**Aggregation level**, L	**Size**, in CCEs	
UE-specific	1	6	6
	2	12	6
	4	8	2
	8	16	2
Common	4	16	4
	8	16	2

If the bandwidth is limited, not all candidates may be available because the PDCCH region is truncated.

A PDCCH can be mapped to any candidate within its suitable search space, as long as the allocated CCEs within the candidate do not overlap with a PDCCH already allocated. A simple example that shows the PDCCH candidates of two aggregation levels within a PDCCH region is shown in the following figure.



In this example, only 11 CCEs are available due to bandwidth constraints. The CCEs used to construct each PDCCH candidate are defined by the following equation.

$$L\left\{\left(Y_k+m\right)\bmod\left\lfloor N_{CCE,k}/L\right\rfloor\right\}+i$$

The preceding equation contains the following variables.

- $N_{CCE,k}$ — number of CCEs in a subframe, k
- m — number of PDCCH candidates in a given space, $m = 0,\ldots,M^{(L)}-1$
- L — aggregation level

- i — an integer between 0 and L–1, $i = 0,\ldots,L-1$

When a common search space is used, Y_k is 0. When a UE-specific search space is used, Y_k is given by the following equation.

$$Y_k = (AY_{k-1}) \bmod D$$

In the preceding equation, A is 39,872, D is 65,537, and Y_{-1} is the nonzero UE radio network temporary identifier.

Scrambling

This multiplexed block of bits undergoes a bit-wise exclusive-or (XOR) operation with a cell-specific scrambling sequence.

The scrambling sequence is pseudorandom, created using a length-31 Gold sequence generator and initialized using the slot number within the radio frame, n_s, and the cell ID, N_{ID}^{cell}, at the start of each subframe, as shown in the following equation.

$$c_{init} = \left\lfloor \frac{n_s}{2} \right\rfloor 2^9 + N_{ID}^{cell}$$

Scrambling serves the purpose of intercell interference rejection. When a UE descrambles a received bitstream with a known cell specific scrambling sequence, interference from other cells will be descrambled incorrectly, therefore only appearing as uncorrelated noise.

Modulation

The scrambled bits then undergo QPSK modulation to create a block of complex-valued modulation symbols.

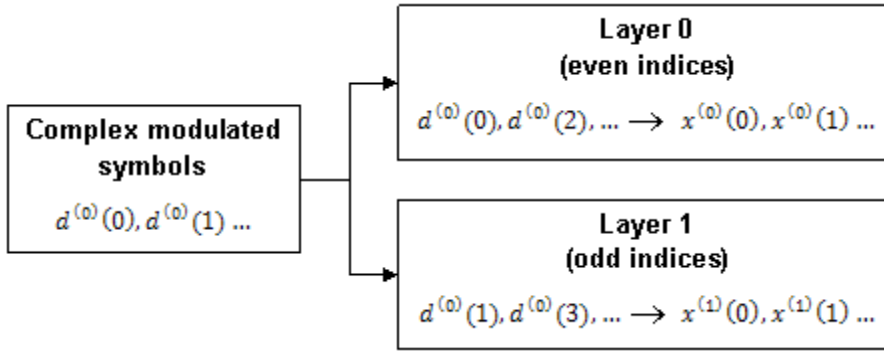
Layer Mapping

The complex symbols are mapped to one, two, or four layers depending on the number of transmit antennas used. The complex modulated input symbols, $d^{(0)}(i)$, are mapped onto v layers, $x^{(0)}(i), x^{(1)}(i), \ldots, x^{(v-1)}(i)$.

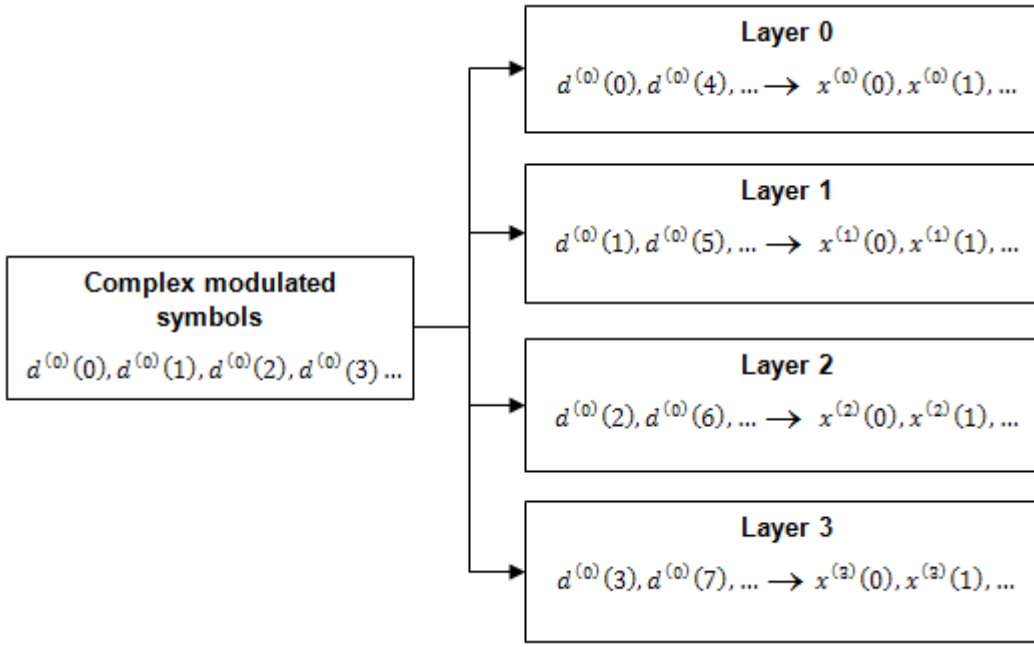
If a single antenna port is used, only one layer is used. Therefore, $x^{(0)}(i) = d^{(0)}(i)$.

If transmitter diversity is used, the input symbols are mapped to layers based on the number of layers.

- **Two Layers** — Even symbols are mapped to layer 0 and odd symbols are mapped to layer 1, as shown in the following figure.



- **Four Layers** — The input symbols are mapped to layers sequentially, as shown in the following figure.



Precoding

- "Two Antenna Port Precoding" on page 1-70
- "Four Antenna Port Precoding" on page 1-70

The precoder takes a block from the layer mapper, $x^{(0)}(i), x^{(1)}(i), \ldots, x^{(\upsilon-1)}(i)$, and generates a sequence for each antenna port, $y^{(p)}(i)$. The variable p is the transmit antenna port number, and can assume values of {0}, {0,1}, or {0,1,2,3}.

For transmission over a single antenna port, no processing is carried out, as shown in the following equation.

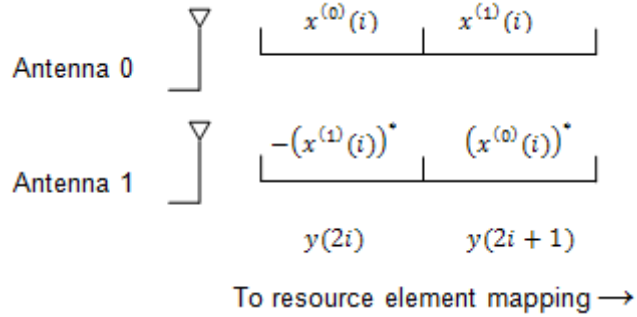
$$y^{(p)}(i) = x^{(0)}(i)$$

Precoding for transmit diversity is available on two or four antenna ports.

Two Antenna Port Precoding. An Alamouti scheme is used for precoding, which defines the relationship between input and output as shown in the following equation.

$$\begin{pmatrix} y^{(0)}(2i) \\ y^{(1)}(2i) \\ y^{(0)}(2i+1) \\ y^{(1)}(2i+1) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & j & 0 \\ 0 & -1 & 0 & j \\ 0 & 1 & 0 & j \\ 1 & 0 & -j & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}\left\{x^{(0)}(i)\right\} \\ \mathrm{Re}\left\{x^{(1)}(i)\right\} \\ \mathrm{Im}\left\{x^{(0)}(i)\right\} \\ \mathrm{Im}\left\{x^{(1)}(i)\right\} \end{pmatrix}$$

In the Alamouti scheme, two consecutive symbols, $x^{(0)}(i)$ and $x^{(1)}(i)$, are transmitted in parallel using two antennas with the following mapping, where the asterisk symbol (*) denotes the complex conjugate operation.

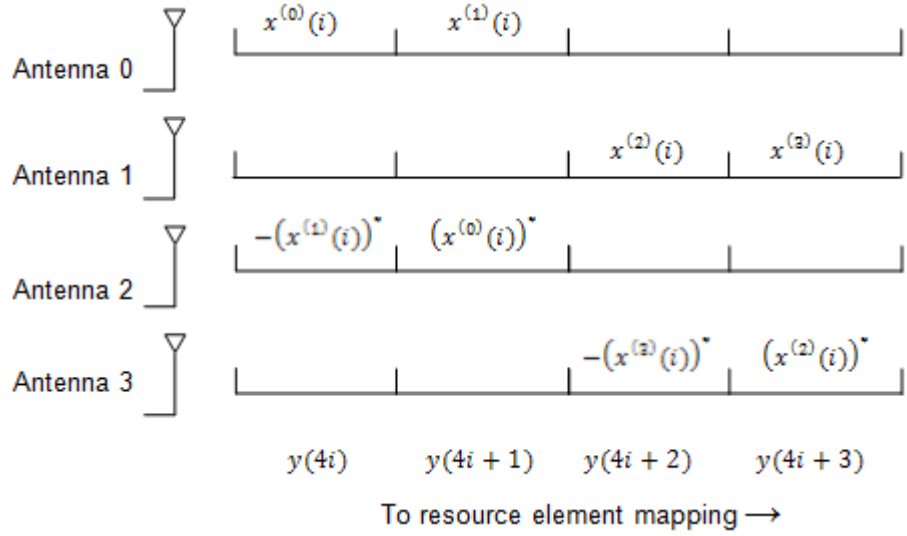


As any two columns in the precoding matrix are orthogonal, the two symbols, $x^{(0)}(i)$ and $x^{(1)}(i)$, can be separated at the UE.

Four Antenna Port Precoding. Precoding for the four antenna port case defines the relationship between the input and output as shown in the following equation.

$$\begin{pmatrix} y^{(0)}(4i) \\ y^{(1)}(4i) \\ y^{(2)}(4i) \\ y^{(3)}(4i) \\ y^{(0)}(4i+1) \\ y^{(1)}(4i+1) \\ y^{(2)}(4i+1) \\ y^{(3)}(4i+1) \\ y^{(0)}(4i+2) \\ y^{(1)}(4i+2) \\ y^{(2)}(4i+2) \\ y^{(3)}(4i+2) \\ y^{(0)}(4i+3) \\ y^{(1)}(4i+3) \\ y^{(2)}(4i+3) \\ y^{(3)}(4i+3) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 0 \\ 0 & -1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & j & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \mathbf{0} & -j & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}\left\{x^{(0)}(i)\right\} \\ \mathrm{Re}\left\{x^{(1)}(i)\right\} \\ \mathrm{Re}\left\{x^{(2)}(i)\right\} \\ \mathrm{Re}\left\{x^{(3)}(i)\right\} \\ \mathrm{Im}\left\{x^{(0)}(i)\right\} \\ \mathrm{Im}\left\{x^{(1)}(i)\right\} \\ \mathrm{Im}\left\{x^{(2)}(i)\right\} \\ \mathrm{Im}\left\{x^{(3)}(i)\right\} \end{pmatrix}$$

In this scheme, two consecutive symbols are transmitted in parallel in two symbol periods using four antennas with the following mapping, where the asterisk symbol (*) denotes the complex conjugate operation.



Mapping to Resource Elements

- "Permutation" on page 1-72
- "Cyclic Shifting" on page 1-73
- "Mapping" on page 1-73

The complex valued symbols for each antenna are divided into quadruplets for mapping to resource elements. The sets of quadruplets then undergo permutation (interleaving) and cyclic shifting before being mapped to resource elements (REs) within resource-element groups (REGs).

Permutation. The blocks of quadruplets are interleaved as discussed in "Sub-block Interleaver" on page 1-62. However, instead of bits being interleaved, blocks of quadruplets are interleaved by substituting the term *bit sequence* with the term *symbol-quadruplet sequence*.

<NULL> symbols from the output of the interleaver are removed to form a sequence of interleaved quadruplets at each antenna, $w^{(p)}(i)$.

Cyclic Shifting. The interleaved sequence of quadruplets at each antenna is cyclically shifted, according to the following equation.

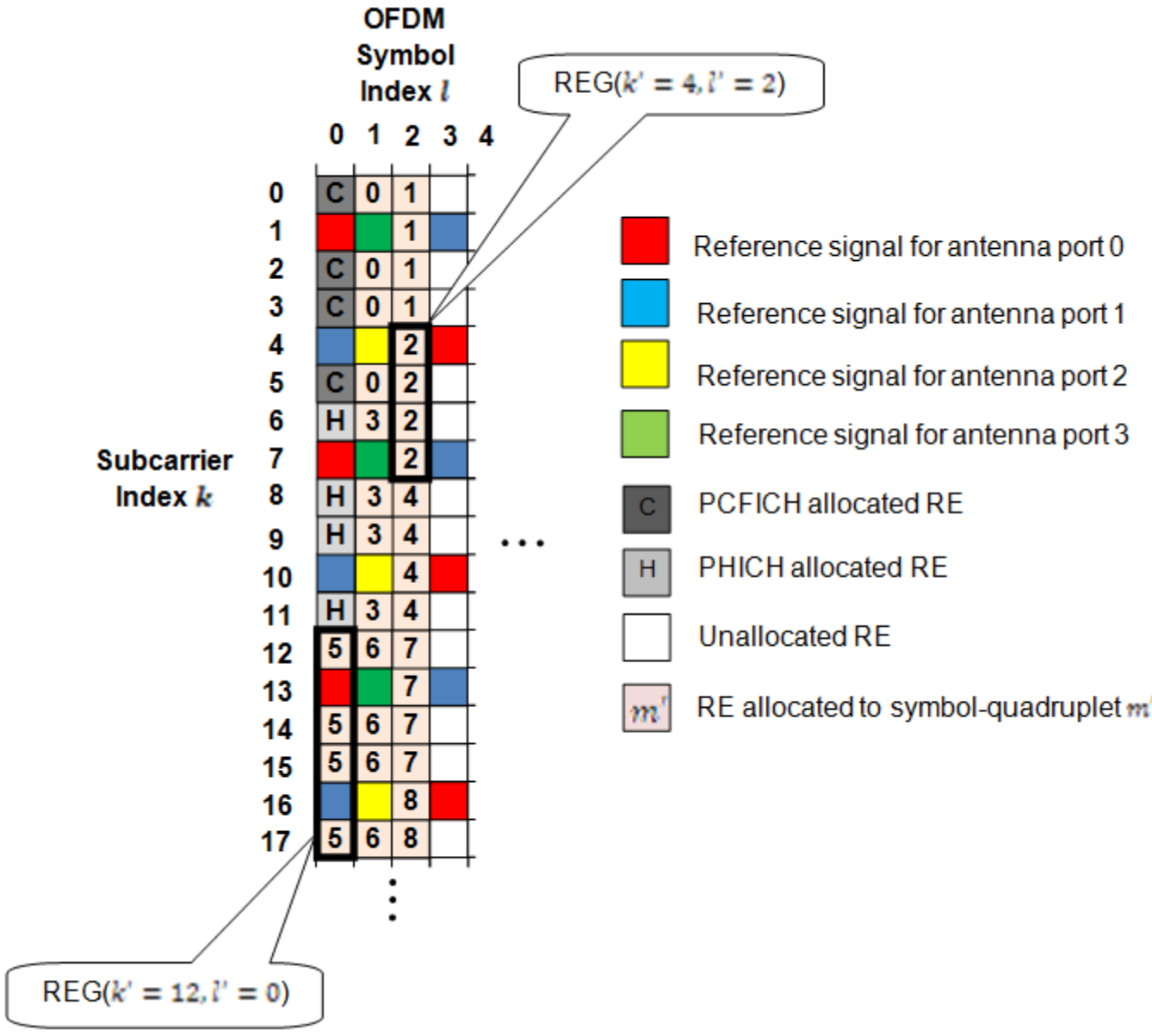
$$\bar{w}^{(p)}(i) = w^{(p)}(i)\left(\left(i + N_{ID}^{cell}\right) \bmod M_{quad}\right)$$

In the preceding equation, the variable M_{quad} is the number of quadruplets, such that $M_{quad} = M_{symb}/4$, and N_{ID}^{cell} is the cell ID.

Mapping. The cyclic shifted symbol quadruplets are mapped to REGs that are not assigned to PCFICH or PHICH.

Each symbol-quadruplet is mapped to an unallocated REG in order, starting with the REG $(k' = 0, l' = 0)$. Symbol quadruplet $\bar{w}^{(p)}(m')$ maps to the REG m'. Then, the REG symbol index, l', is incremented until all the REGs at subcarrier index $k' = 0$ have been allocated. Next, the REG subcarrier index, k', is incremented, and the process repeats. This mapping continues until all symbol quadruplets have been allocated REGs.

The mapping for an example resource grid is shown in the following figure.


OFDM Symbol Index l
0 1 2 3 4
Subcarrier Index k
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
REG($k' = 4, l' = 2$)
REG($k' = 12, l' = 0$)
Reference signal for antenna port 0
Reference signal for antenna port 1
Reference signal for antenna port 2
Reference signal for antenna port 3
C PCFICH allocated RE
H PHICH allocated RE
Unallocated RE
m' RE allocated to symbol-quadruplet m'

Four transmit antenna ports and a control region size of three OFDM symbols are used to create the grid. In this example, the REG $(k'=0,l'=0)$ is allocated to PCFICH, so no symbol quadruplets are allocated to it. The symbol-quadruplets are first mapped to REG $(k'=0,l'=1)$, followed by $(k'=0,l'=2)$. Since there are no further REGs with $k'=0$, the next REG allocated is REG $(k'=4,l'=2)$ because this REG has the lowest value of l' not already allocated. This process repeats to allocate all the symbol quadruplets to REGs.

References

[1] 3GPP TS 36.212. "Multiplexing and channel coding." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`lteDCI` | `lteDCIEncode` | `ltePDCCH` | `ltePDCCHIndices` | `ltePDCCHSpace` | `ltePDCCHInfo` | `ltePDCCHPRBS` | `ltePDCCHInterleave` | `ltePDCCHDeinterleave` | `ltePDCCHDecode` | `lteLayerMap` | `lteLayerDemap` | `lteDLPrecode` | `lteDLDeprecode` | `lteCRCEncode` | `lteCRCDecode` | `lteConvolutionalEncode` | `lteConvolutionalDecode` | `lteRateMatchConvolutional` | `lteRateRecoverConvolutional` | `lteSymbolModulate` | `lteSymbolDemodulate`

Related Examples

- "Model DCI and PDCCH" on page 3-11

Random Access Channel

In this section...
"RACH Coding" on page 1-76
"The PRACH" on page 1-76
"PRACH Conformance Tests" on page 1-79

The Random Access Channel (RACH) is an uplink transmission used by the UE to initiate synchronization with the eNodeB.

RACH Coding

The relationship between RACH, a transport channel, and PRACH, a physical channel, as described by [2], is shown in the following table.

Transport Channel (TrCH)	Physical Channel
RACH	PRACH

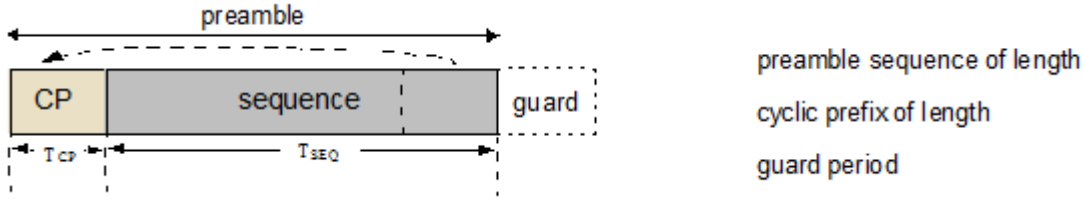
However, there are not actually any coding processes that take place to encode the RACH transport channel onto the input of the PRACH. Also, there is no logical channel which maps into the input of the RACH transport channel; the RACH originates in the MAC layer. The RACH effectively consists of a number of parameters within the MAC layer which ultimately control how and when the PRACH physical channel is generated.

The PRACH

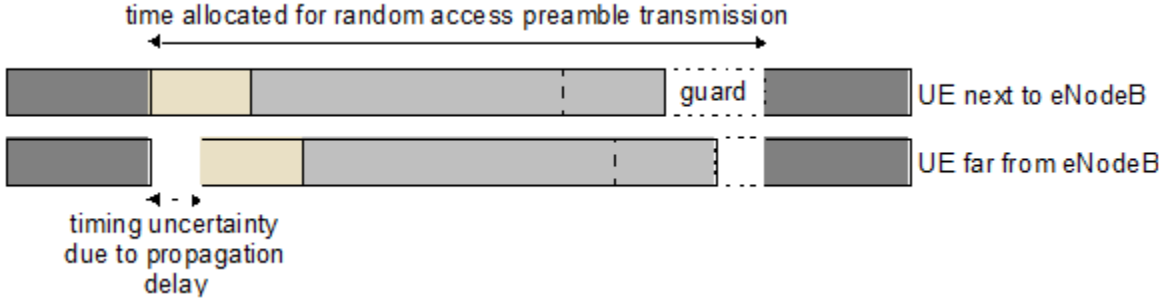
The PRACH transmission (the PRACH preamble) is an OFDM-based signal, but it is generated using a different structure from other uplink transmission; most notably it uses narrower subcarrier spacing and therefore is not orthogonal to the PUSCH, PUCCH and SRS, therefore those channels will suffer from some interference from the PRACH. However, the subcarrier spacing used by the PRACH is an integer submultiple of the spacing used for the other channels and therefore the PUSCH, PUCCH and SRS do not interfere on the PRACH.

PRACH preamble time structure

The PRACH preamble consists of a cyclic prefix, useful part of the sequence and then a guard period which is simply an unused portion of time up to the end of the last subframe occupied by the PRACH.



This guard period allows for timing uncertainty due to the UE to eNodeB distance.



Therefore the size of the guard period determines the cell radius, as any propagation delay exceeding the guard time would cause the random access preamble to overlap the following subframe at the eNodeB receiver.

The use of an OFDM transmission with cyclic prefix allows for an efficient frequency domain based receiver in the eNodeB to perform PRACH detection.

PRACH formats

There are five PRACH preamble formats which have different lengths for the cyclic prefix, useful part of the symbol, and guard period.

Preamble Format	T_{CP}	T_{SEQ}	Guard Period
0	$3{,}168\times T_s$	$24{,}576\times T_s$	$2{,}976\times T_s$
1	$21{,}024\times T_s$	$24{,}576\times T_s$	$15{,}840\times T_s$
2	$6{,}240\times T_s$	$2\times 24{,}576\times T_s$	$6{,}048\times T_s$
3	$21{,}024\times T_s$	$2\times 24{,}576\times T_s$	$21{,}984\times T_s$
4	$448\times T_s$	$4{,}096\times T_s$	$288\times T_s$

Note that Preamble Format 4 is only applicable for TDD in special subframes (subframe 1 or 6) and with Special Subframe Configuration that results in UpPTS with 2 symbols duration i.e. the Preamble Format 4 PRACH sits in UpPTS. Formats 2 and 3 have two repetitions of the nominal PRACH sequence which provides more total transmit energy and therefore allows for detection at lower SNRs. Also, Format 1 versus 0 and Format 3 versus 2 have a longer guard period, allowing for a larger cell size. The downside is that when the cyclic prefix time, sequence time and guard period are totaled up, some of the formats require multiple subframes for transmission.

Preamble Format	Number of Subframes
0	1
1	2
2	2
3	3
4	1

The penalty for using multiple subframes is a reduction in the capacity for normal uplink transmission.

PRACH Preamble Frequency Structure

As already mentioned, the PRACH uses a narrower subcarrier spacing that normal uplink transmission, specifically 1250 Hz for formats 0–3 and 7500 Hz for format 4. The ratio of the normal uplink subcarrier spacing to PRACH subcarrier spacing, K, is K=12 for formats 0–3 and K=2 for format 4.

The PRACH is designed to fit in the same bandwidth as 6 RBs of normal uplink transmission. For example, 72 subcarriers at 15,000 Hz spacing is 1.08 MHz. This makes it easy to schedule gaps in normal uplink transmission to allow for PRACH opportunities.

Therefore, there are $72\times K$ subcarriers for the PRACH, specifically 864 for formats 0–3 and 144 for format 4. As will be explained in the following subsection, the PRACH transmission for formats 0–3 uses 839 active subcarriers, and for format 4 uses 139 active subcarriers; the number of active subcarriers is denoted N_{ZC}.

As with normal uplink SC-FDMA transmission there is a half subcarrier (7500 Hz) shift, which for the PRACH is a $K/2$ subcarrier shift. A further subcarrier offset, φ (7 for formats 0–3 and 2 for format 4), centers the PRACH transmission within the 1.08 MHz bandwidth.

Preamble Format	$+K/2$	N_{ZC}	$72K-N_{ZC}-K/2$
0–3	13	839	12
4	3	139	2

PRACH Subcarrier Content

The actual PRACH transmission is an OFDM-based reconstruction of a Zadoff-Chu sequence in the time domain. The OFDM modulator is used to position the Zadoff-Chu sequence in the frequency domain (i.e. to place the 6RBs of PRACH transmission in the 6 consecutive RBs starting from some particular physical resource block, denoted n_{PRB}^{RA} in the standard). If the output of the OFDM modulator in the time domain is to be a Zadoff-Chu sequence, the input to the OFDM modulator must be a Zadoff-Chu sequence in the frequency domain. Therefore, the active subcarriers, which total N_{ZC} in number, are set to the values of an N_{ZC}-point DFT of an N_{ZC}-sample Zadoff-Chu sequence.

PRACH Conformance Tests

Conformance tests for the PRACH, as defined in section 8.4 of [1], test the false alarm rate and detection rate of the PRACH in various environments.

For a demonstration of how to perform the PRACH false alarm rate test specified in section 8.4.1, see "PRACH False Alarm Probability Conformance Test". For a demonstration of how to perform the PRACH detection rate test specified in section 8.4.2, see "PRACH Detection Conformance Test".

References

[1] 3GPP TS 36.104. "Base Station (BS) radio transmission and reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

[2] 3GPP TS 36.212. "Multiplexing and channel coding." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

See Also

`ltePRACH` | `ltePRACHInfo` | `ltePRACHDetect` | `lteZadoffChuSeq`

Related Examples

- "PRACH False Alarm Probability Conformance Test"
- "PRACH Detection Conformance Test"

Uplink Control Channel Format 1

In this section...
"Uplink Control Information on PUCCH Format 1" on page 1-81
"PUCCH Format 1, 1a, and 1b" on page 1-82
"Demodulation Reference Signals on PUCCH Format 1" on page 1-83
"PUCCH Format 1 Resource Element Mapping" on page 1-87

The physical uplink control channel format 1 is a transmission channel used to carry information regarding scheduling requests in which the UE requests resources to transmit UL-SCH. It is also used to send acknowledgement responses and retransmission requests (ACK and NACK).

Uplink Control Information on PUCCH Format 1

- "Channel Coding for UCI HARQ-ACK" on page 1-81
- "Channel coding for UCI Scheduling Request" on page 1-81

Format 1 uplink control information (UCI) contains scheduling requests and acknowledgement responses or retransmission requests (ACK and NACK).

Channel Coding for UCI HARQ-ACK

The HARQ acknowledgement bits are received from higher layers. Depending on the number of codewords present, a HARQ acknowledgment consists of 1 or 2 information bits. A positive acknowledgement (ACK) is encoded as a binary 1, while a negative acknowledgement (NACK) is encoded as a binary 0. The HARQ-ACK bits are then processed, as required by the PUCCH.

Channel coding for UCI Scheduling Request

The scheduling request indication is received from higher layers. Zero information bits are used to request resources to transmit UL-SCH. However, the eNodeB knows when to expect a scheduling request from each UE within the cell. Therefore, if PUCCH energy is detected, the eNodeB will identify it as a scheduling request from the corresponding UE.

PUCCH Format 1, 1a, and 1b

The various PUCCH Format 1 messages used are identified by the type of control information they carry and the number of control bits they require per subframe. The three PUCCH format 1 types, their modulation schemes, and the number of information bits they use are shown in the following table.

PUCCH format	Modulation scheme	Number of bits per subframe, M_{bit}	Type of control information
1	n/a	n/a	Scheduling request
1a	BPSK	1	HARQ-ACK (1 bit)
1b	QPSK	2	HARQ-ACK (2 bits)

The bandwidth available during one subframe of a single resource block exceeds that needed for the control signaling information of a single user terminal. To make efficient use of the available resources, the resource block can be shared by multiple user terminals. Even though the same RB is used for the PUCCH Formats 1, 1a and 1b, there is no possibility of intra-cell interference if different cyclic shifts of the same base reference sequence are used. Moreover, for PUCCH Formats 1, 1a and 1b, an extra degree of freedom is provided by applying an orthogonal cover code.

Format 1

A request for uplink resources can be made by means of the random access channel. However, due to the probability of collisions during high intensity periods, an alternative method is provided using the PUCCH Format 1.

Each UE in the cell is assigned a specific resource index mapping, a resource which can be used every nth frame to transmit a scheduling request. Therefore, if PUCCH energy is detected, the eNodeB will identify it as a scheduling request from the corresponding UE. Since each UE will have a specific resource allocated, there is no probability of a collision. However, the number of available PUCCH resources is reduced.

Format 1a and 1b

For transmission of the hybrid-ARQ acknowledgement, the HARQ ACK bits are used to generate a BPSK/QPSK symbol, depending on the number of codewords present. The modulated symbol is then used to generate the signal to be transmitted in each of the two PUCCH slots.

Demodulation Reference Signals on PUCCH Format 1

Demodulation reference signals associated with the PUCCH format 1 are used by the base station to perform channel estimation and allow for coherent demodulation of the received signal.

These reference signals are time-multiplexed with data, whereas in the downlink there is both time and frequency multiplexing. This multiplexing is performed to maintain the single-carrier nature of the SC-FDMA signal, which ensures that all data carriers are contiguous.

DRS Generation

- "Base Sequence" on page 1-84
- "DRS Grouping" on page 1-99

The demodulation reference signals are generated using a base sequence denoted by $r_{u,v}(n)$, which is discussed further in "Base Sequence" on page 1-84. More specifically, r^{PUCCH} is used to denote the PUCCH format 1 DRS sequence and is defined by the following equation.

$$r^{PUCCH}\left(m'N_{RS}^{PUCCH}M_{SC}^{RS}+mM_{SC}^{RS}+n\right)=\bar{w}(m)r_{u,v}^{(\alpha)}(n)$$

It is desired that the DRS sequences have small power variations in time and frequency, resulting in high power amplifier efficiency and comparable channel estimation quality for all frequency components. Zadoff-Chu sequences are good candidates, since they exhibit constant power in time and frequency. However, there are a limited number of Zadoff-Chu sequences; therefore, they are not suitable on their own.

The generation and mapping of the DRS associated with the PUCCH format 1 are discussed further in the following sections.

Base Sequence. The demodulation reference signals are defined by a cyclic shift, α, of a base sequence, r.

The base sequence, r, is represented in the following equation.

$$r_{u,v}^{(\alpha)} = e^{j\alpha n} r_{u,v}(n)$$

The preceding equation contains the following variables.

- $n = 0,...,M_{SC}^{RS}$, where M_{SC}^{RS} is the length of the reference signal sequence.
- $U = 0,\ldots,29$ is the base sequence group number.
- $V = 0,1$ is the sequence number within the group and only applies to reference signals of length greater than 6 resource blocks.

A phase rotation in the frequency domain (pre-IFFT in the OFDM modulation) is equivalent to a cyclic shift in the time domain (post IFFT in the OFDM modulation). For frequency non-selective channels over the 12 subcarriers of a resource block, it is possible to achieve orthogonality between DRS generated from the same base sequence if $\alpha = m\pi/6$ for $m = 0,1,\ldots,11$, and assuming the DRS are synchronized in time.

The orthogonality can be exploited to transmit DRS at the same time, using the same frequency resources without mutual interference. In the PUCCH format 1 case, an extra degree of freedom can be achieved by applying an orthogonal cover code, $\bar{w}(m)$. Generally, DRS generated from different base sequences will not be orthogonal; however, they will present low cross-correlation properties.

To maximize the number of available Zadoff-Chu sequences, a prime length sequence is needed. The minimum sequence length in the UL is 12, the number of subcarriers in a resource block, which is not prime.

Therefore, Zadoff-Chu sequences are not suitable by themselves. There are effectively the following two types of base reference sequences.

- those with a sequence length ≥ 36 (spanning 3 or more resource blocks), which use a cyclic extension of Zadoff-Chu sequences
- those with a sequence length ≤ 36 (spanning 2 resource blocks), which use a special QPSK sequence

Base sequences of length • three resource blocks

For sequences of length 3 resource blocks and larger (i.e. $M_{sc}^{RS} \geq 3N_{sc}^{RS}$), the base sequence is a repetition, with a cyclic offset of a Zadoff-Chu sequence of length N_{zc}^{RS}, where N_{zc}^{RS} is the largest prime such that $N_{zc}^{RS} < M_{sc}^{RS}$. Therefore, the base sequence will contain one complete length N_{zc}^{RS} Zadoff-Chu sequence plus a fractional repetition appended on the end. At the receiver the appropriate de-repetition can be done and the zero autocorrelation property will hold across the length N_{zc}^{RS} vector.

Base sequences of length • three resource blocks

For sequences shorter than three resource blocks (i.e. $M_{sc}^{RS} = 12,24$), the sequences are a composition of unity modulus complex numbers drawn from a simulation generated table. These sequences have been found through computer simulation and are specified in the LTE specifications.

DRS Grouping. There are a total of 30 sequence groups, $u \in \{0,1,\ldots,29\}$, each containing one sequence for length less than or equal to 60. This corresponds to transmission bandwidths of 1,2,3,4 and 5 resource blocks. Additionally, there are two sequences (one for v = 0 or 1) for length ≥ 72; corresponding to transmission bandwidths of 6 resource blocks or more.

Note that not all values of m are allowed, where m is the number of resource blocks used for transmission. Only values for m that are the product of powers of 2, 3 and 5 are valid, as shown in the following equation.

$m = 2^{\alpha_0} \times 3^{\alpha_1} \times 5^{\alpha_2}$, where α_i are positive integers

The reason for this restriction is that the DFT sizes of the SC-FDMA precoding operation are limited to values which are the product of powers of 2, 3 and 5. The DFT operation can span more than one resource block, and since each resource block has 12 subcarriers, the total number of subcarriers fed to the DFT will be $12m$. Since the result of $12m$ has to be the product of powers of 2, 3 and 5 this implies that the number of resource blocks must themselves be the product of powers of 2, 3 and 5. Therefore values of m such as 7, 11, 14, 19, etc. are not valid.

For a given time slot, the uplink reference signal sequences to use within a cell are taken from one specific sequence group. If the same group is to be used for all slots then this is known as fixed assignment. On the other hand, if the group number u varies for all slots within a cell this is known as *group hopping*.

Fixed Group Assignment

When fixed group assignment is used, the same group number is used for all slots. For PUCCH, the group number is a function of the cell identity number modulo 30, as shown in the following equation.

$u = N_{ID}^{cell} \bmod 30$, with $N_{ID}^{cell} = 0,1,\ldots,503$

Group Hopping

If group hopping is used, the pattern is applied to the calculation of the sequence group number. This pattern is the same for both the PUCCH and PUSCH.

This pattern is defined as the following equation.

$f_{gh}(n_s) = \sum_{i=0}^{7} c(8n_s + i) \cdot 2^i \bmod 30$

As shown in the preceding equation, this group hopping pattern is a function of the slot number n_s and is calculated making use of a pseudorandom binary sequence $c(k)$, generated using a length-30 *Gold* code. To generate the group

hopping pattern, the PRBS generator is initialized with the following value at the start of each radio frame.

$$c_{init} = \left\lfloor \frac{N_{ID}^{cell}}{30} \right\rfloor$$

For PUCCH with group hopping, the group number, u, is given by the following equation.

$$u = \left(f_{gh}(n_s) + \left(\left(N_{ID}^{cell} \bmod 30 \right) + \Delta_{ss} \right) \right) \bmod 30$$

PUCCH Format 1 Resource Element Mapping

The resource blocks assigned to L1/L2 control information within a subframe are located at the edges of the total available cell bandwidth. A frequency hopping pattern is used where the lower end of the available UL spectrum is used in the first slot of the subframe and the higher end on the second; this adds a level of frequency diversity.

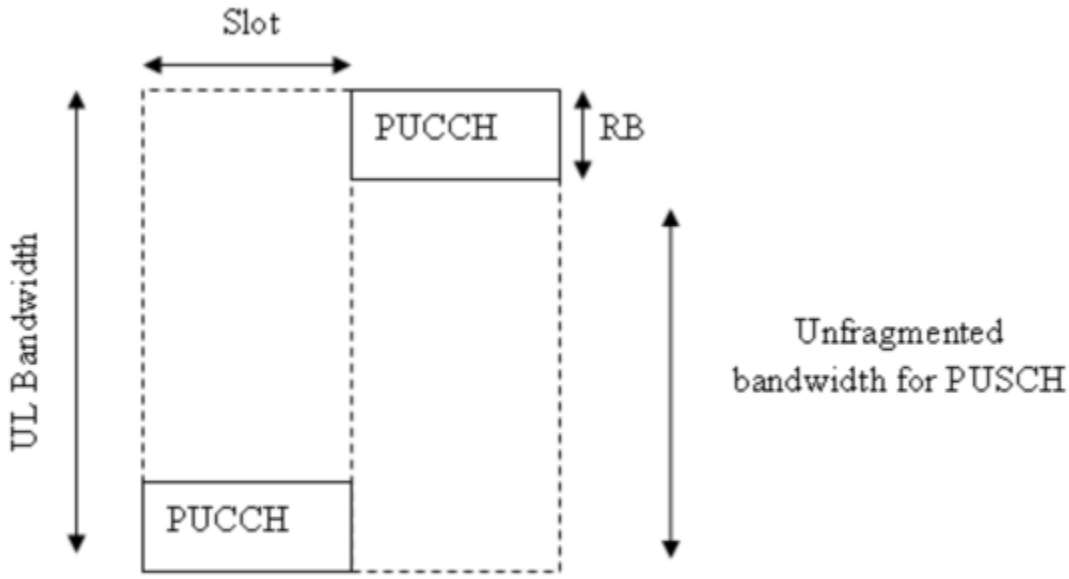
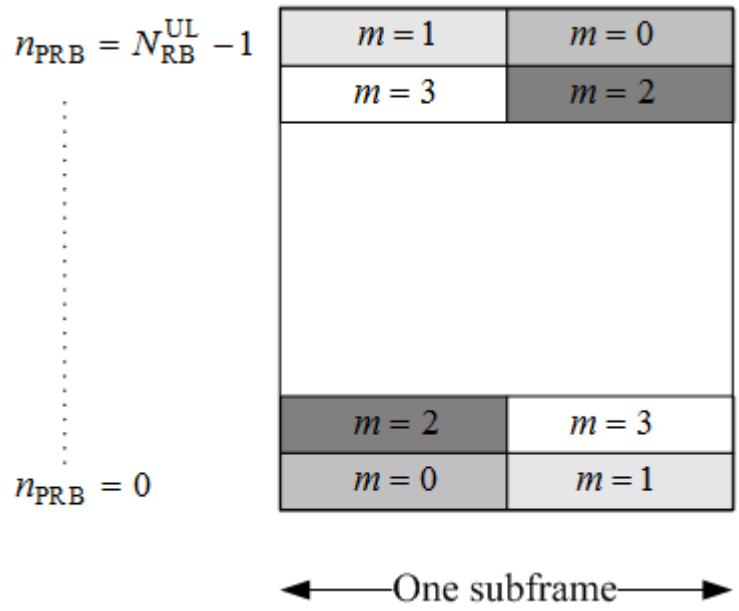


Figure 1: Location of PUCCH RB within a subframe

Bandwidth edges are used so that a large unfragmented portion of the spectrum remains to allocate to the PUSCH. If this spectrum was fragmented by multiple PUCCHs it would not be possible to allocate a number of contiguous RBs to a UE, hence the single carrier nature of SC-FDMA would be lost.

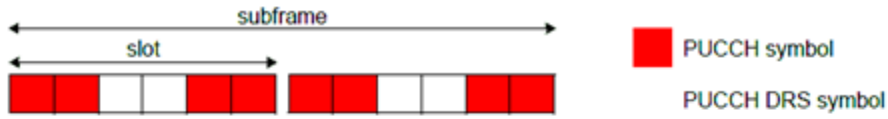
There is a single index, m, derived from the PUCCH resource index and other parameters that specifies the location of the PUCCH in time/frequency. When m is 0, the PUCCH occupies the lowest RB in the first slot and the highest RB in the second slot of a subframe. When m is 1, the opposite corners are used—the highest RB in the first slot and the lowest RB in the second slot. As m increases further, the allocated resource blocks move in towards the band center as shown in the following figure.



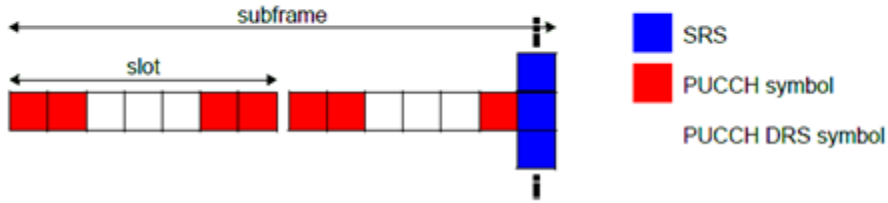
PUCCH formats 1, 1a, and 1b make use of four SC-FDMA symbols per slot. If normal cyclic prefix is used, the remaining 3 symbols, 2 for extended cyclic prefix, are used for PUCCH demodulation reference signal (DRS). If the sounding reference signal (SRS) overlaps the PUCCH symbols, only three symbols are used in the second slot of the subframe. The mapping of the symbols is illustrated in the following figure.

- 4 symbols per slot used for PUCCH formats 1, 1a & 1b
  - 3 remaining symbols (normal CP) used for DRS

  - 2 remaining symbols (extended CP) used for DRS



- If SRS overlaps PUCCH only 3 symbols are used in the 2nd slot



See Also

`ltePUCCH1` | `ltePUCCH1Indices` | `ltePUCCH1DRS` | `ltePUCCH1DRSIndices` | `lteULResourceGrid`

Related Examples

- "Model PUCCH Format 1" on page 3-15
- "PUCCH1a ACK Missed Detection Probability Conformance Test"
- "PUCCH1a Multi User ACK Missed Detection Probability Conformance Test"

Uplink Control Channel Format 2

In this section...
"Uplink Control Information on PUCCH Format 2" on page 1-91
"PUCCH Format 2" on page 1-93
"Demodulation Reference Signals on PUCCH Format 2" on page 1-96
"PUCCH Format 2 Resource Element Mapping" on page 1-100

The Physical Uplink Control Channel (PUCCH) Format 2 is a transmission channel used to carry information regarding channel status reports as well as Hybrid Automatic Repeat request (HARQ) acknowledgements.

Uplink Control Information on PUCCH Format 2

- "Channel Coding for UCI CQI Indication" on page 1-92
- "Channel Coding for UCI CQI and HARQ-ACK" on page 1-93

The UE uses PUCCH format 2 control information to relay an estimate of the channel properties to the base station in order to aid channel dependent scheduling. Channel status reports include channel quality indication (CQI), rank indication (RI), and precoder matrix indication (PMI).

- **CQI** — represents the recommended modulation scheme and coding rate that should be used for the downlink transmission.
- **RI** — provides information about the rank of the channel, which is used to determine the optimal number of layers that should be used for the downlink transmission (only used for spatial multiplexed systems).
- **PMI** — provides information about which precoding matrix to use (only used in closed loop spatial multiplexing systems).

HARQ-ACK can also be transmitted with channel status information. Two forms of channel coding exist—one for the CQI alone and another for the combination of CQI with HARQ-ACK.

Channel Coding for UCI CQI Indication

The CQI codewords are coded using a (20,*A*) block code and are a linear combination of the 13 basis sequences denoted by $M_{i,n}$ and defined by the following equation.

$$b_i = \sum_{n=0}^{A-1} \left(a_n \cdot M_{i,n} \right) \bmod 2, \text{ where } i = 0,1,2,\ldots,B-1$$

The values of the basis sequence, $M_{i,n}$, for a (20,*A*) block code are given in the following table.

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$	$M_{i,11}$	$M_{i,12}$
0	1	1	0	0	0	0	0	0	0	0	1	1	0
1	1	1	1	0	0	0	0	0	0	1	1	1	0
2	1	0	0	1	0	0	1	0	1	1	1	1	1
3	1	0	1	1	0	0	0	0	1	0	1	1	1
4	1	1	1	1	0	0	0	1	0	0	1	1	1
5	1	1	0	0	1	0	1	1	1	0	1	1	1
6	1	0	1	0	1	0	1	0	1	1	1	1	1
7	1	0	0	1	1	0	0	1	1	0	1	1	1
8	1	1	0	1	1	0	0	1	0	1	1	1	1
9	1	0	1	1	1	0	1	0	0	1	1	1	1
10	1	0	1	0	0	1	1	1	0	1	1	1	1
11	1	1	1	0	0	1	1	0	1	0	1	1	1
12	1	0	0	1	0	1	0	1	1	1	1	1	1
13	1	1	0	1	0	1	0	1	0	1	1	1	1
14	1	0	0	0	1	1	0	1	0	0	1	0	1
15	1	1	0	0	1	1	1	1	0	1	1	0	1
16	1	1	1	0	1	1	1	0	0	1	0	1	1
17	1	0	0	1	1	1	0	0	1	0	0	1	1

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$	$M_{i,11}$	$M_{i,12}$
18	1	1	0	1	1	1	1	1	0	0	0	0	0
19	1	0	0	0	0	1	1	0	0	0	0	0	0

Together, the CQI, PMI, and RI form the channel status report. These indications can be brought together in a range of different configurations depending on the transmission mode of the terminal. Therefore, the total number of bits used to report the channel status condition can change, depending on the transmission format. The bit widths for the various wideband and UE selected sub-band reports can be found in sections 5.2.3.3.1 and 5.2.3.3.2 of [1], respectively.

Channel Coding for UCI CQI and HARQ-ACK

When HARQ acknowledgement responses are transmitted with the channel status report in a subframe, a different method is used. Using normal cyclic prefix length, the CQI is block coded as shown in the preceding section with the one or two HARQ-ACK bits appended to the end of the coded CQI sequence. These are coded separately to produce an 11th complex symbol which is transmitted with the PUCCH DRS for formats 2a and 2b.

When extended cyclic prefix is used, the CQI and HARQ bits are coded together. The channel quality indication is multiplexed with the one or two HARQ bits and the bits are block coded as shown in the preceding section. Bits 21 and 22 are coded separately to produce an 11th complex symbol which is transmitted with the PUCCH DRS.

PUCCH Format 2

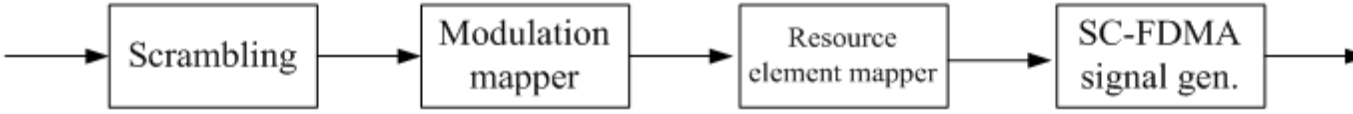
The three PUCCH format 2 types, their modulation schemes, and the number of information bits they use are shown in the following table.

PUCCH format	Modulation scheme	Number of bits per subframe, M_{bit}	Type of control information
2	QPSK	20	Channel status reports
2a	QPSK + BPSK	21	Channel status reports and HARQ-ACK (1 bit)
2b	QPSK + BPSK	22	Channel status reports and HARQ-ACK (2 bits)

Format 2, 2a, and 2b

- "Scrambling" on page 1-94
- "Modulation" on page 1-95
- "Resource Element Mapping" on page 1-95

The block diagram for PUCCH format 2, 2a, and 2b is shown in the following figure.



Scrambling. A block of 20 coded UCI bits undergoes a bit-wise XOR operation with a cell-specific scrambling sequence.

The scrambling sequence is pseudorandom, created using a length-31 Gold sequence generator and initialized using the slot number within the radio frame, n_s, and the cell ID, N_{ID}^{cell}, at the start of each subframe, as shown in the following equation.

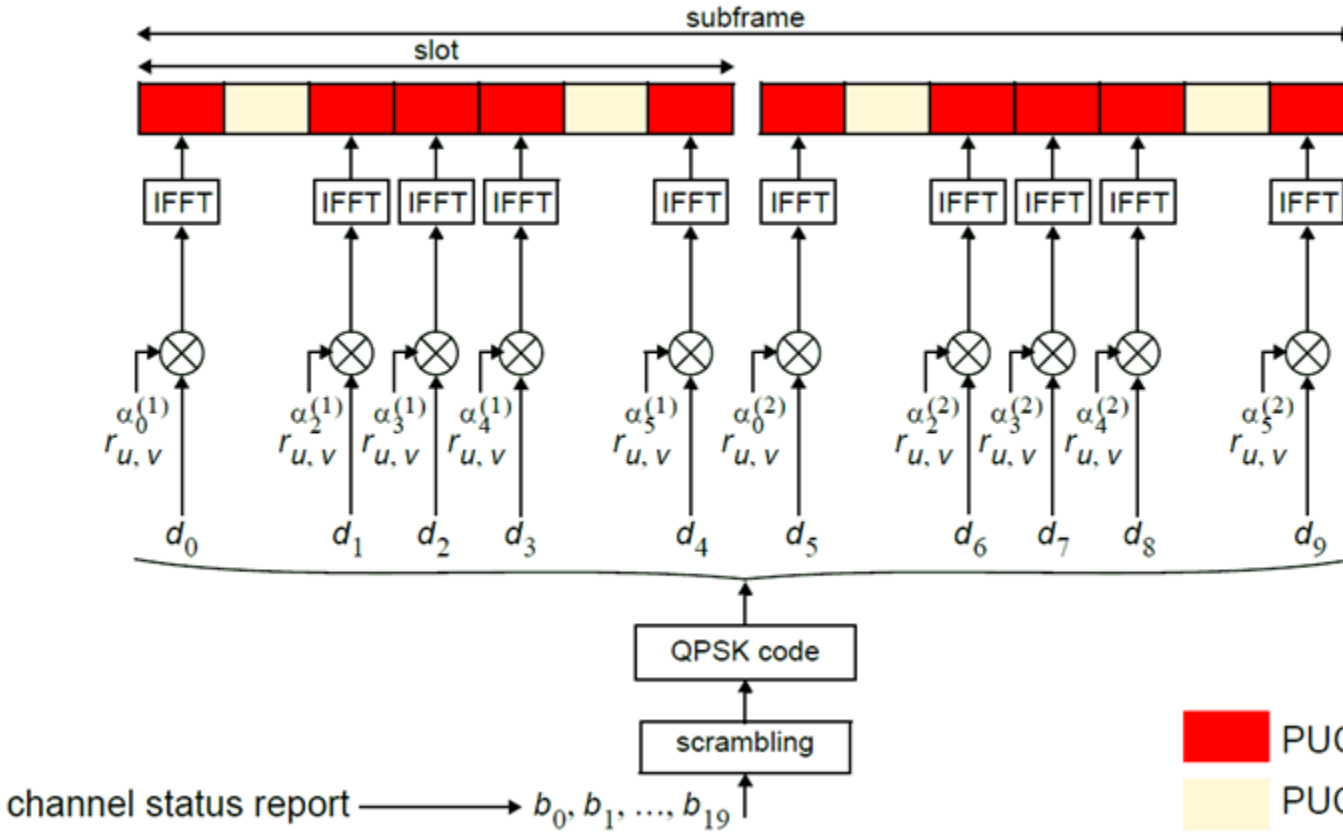
$$c_{init} = \left\lfloor \frac{n_s}{2} \right\rfloor 2^9 + N_{ID}^{cell}$$

Scrambling serves the purpose of intercell interference rejection. When a base station descrambles a received bitstream with a known cell specific scrambling sequence, interference from other cells will be descrambled incorrectly, therefore only appearing as uncorrelated noise.

Modulation. The scrambled bits are then QPSK modulated resulting in a block of complex-valued modulation symbols. Each complex-valued symbol is multiplied with a cyclically shifted length 12 sequence.

Suppose 21 or 22 bits are available. In the case of HARQ-ACK transmission, these are coded separately to produce an 11th complex symbol that is transmitted with the PUCCH DRS for formats 2a and 2b. As in PUCCH Formats 1, 1a, and 1b, a hopping pattern is applied to the cyclic shift to randomize intercell interference. PUCCH Formats 2a and 2b are supported for normal cyclic prefix only.

Resource Element Mapping. The final stage in the PUCCH format 2 processing involves mapping to resource elements. The complete processing chain for normal cyclic prefix, including the position occupied by the PUCCH format 2 in a subframe and in each slot, is shown in the following figure.



The cyclic shifted sequence applied to randomize intercell interference is denoted here by $r_{u,v}$. For extended cyclic prefix, where there are only six SC-FDMA symbols per slot, the mapping to resources changes slightly. In this case, only one reference signal is transmitted per slot, and the signal occupies the third symbol in each slot.

Demodulation Reference Signals on PUCCH Format 2

Demodulation reference signals associated with the PUCCH format 2 are used by the base station to perform channel estimation and allow for coherent demodulation of the received signal.

These reference signals are time-multiplexed with data, whereas in the downlink there is both time and frequency multiplexing. This multiplexing is performed to maintain the single-carrier nature of the SC-FDMA signal, which ensures that all data carriers are contiguous.

DRS Generation

- "Base Sequence" on page 1-97
- "DRS Grouping" on page 1-99

The demodulation reference signals are generated using a base sequence denoted by $r_{u,v}(n)$, which is discussed further in "Base Sequence" on page 1-97. More specifically, r^{PUCCH} is used to denote the PUCCH format 2 DRS sequence and is defined by the following equation.

$$r^{PUCCH}\left(m'N_{RS}^{PUCCH}M_{SC}^{RS}+mM_{SC}^{RS}+n\right)=\bar{w}(m)r_{u,v}^{(\alpha)}(n)$$

It is desired that the DRS sequences have small power variations in time and frequency, resulting in high power amplifier efficiency and comparable channel estimation quality for all frequency components. Zadoff-Chu sequences are good candidates, since they exhibit constant power in time and frequency. However, there are a limited number of Zadoff-Chu sequences; therefore, they are not suitable on their own.

The generation and mapping of the DRS associated with the PUCCH format 2 are discussed further in the following sections.

Base Sequence. The demodulation reference signals are defined by a cyclic shift, α, of a base sequence, r.

The base sequence, r, is represented in the following equation.

$$r_{u,v}^{(\alpha)}=e^{j\alpha n}r_{u,v}(n)$$

The preceding equation contains the following variables.

- $n = 0,...,M_{SC}^{RS}$, where M_{SC}^{RS} is the length of the reference signal sequence.
- $U = 0,\ldots,29$ is the base sequence group number.
- $V = 0,1$ is the sequence number within the group and only applies to reference signals of length greater than 6 resource blocks.

A phase rotation in the frequency domain (pre-IFFT in the OFDM modulation) is equivalent to a cyclic shift in the time domain (post IFFT in the OFDM modulation). For frequency non-selective channels over the 12 subcarriers of a resource block, it is possible to achieve orthogonality between DRS generated from the same base sequence if $\alpha = m\pi/6$ for $m = 0,1,\ldots,11$, and assuming the DRS are synchronized in time.

To maximize the number of available Zadoff-Chu sequences, a prime length sequence is needed. The minimum sequence length in the UL is 12, the number of subcarriers in a resource block, which is not prime.

Therefore, Zadoff-Chu sequences are not suitable by themselves. There are effectively the following two types of base reference sequences.

- those with a sequence length ≥ 36 (spanning 3 or more resource blocks), which use a cyclic extension of Zadoff-Chu sequences
- those with a sequence length ≤ 36 (spanning 2 resource blocks), which use a special QPSK sequence

Base sequences of length • three resource blocks

For sequences of length 3 resource blocks and larger (i.e. $M_{sc}^{RS} \geq 3N_{sc}^{RS}$), the base sequence is a repetition, with a cyclic offset of a Zadoff-Chu sequence of length N_{zc}^{RS}, where N_{zc}^{RS} is the largest prime such that $N_{zc}^{RS} < M_{sc}^{RS}$. Therefore, the base sequence will contain one complete length N_{zc}^{RS} Zadoff-Chu sequence plus a fractional repetition appended on the end. At the receiver the appropriate de-repetition can be done and the zero autocorrelation property will hold across the length N_{zc}^{RS} vector.

Base sequences of length • three resource blocks

For sequences shorter than three resource blocks (i.e. $M_{sc}^{RS} = 12,24$), the sequences are a composition of unity modulus complex numbers drawn from a simulation generated table. These sequences have been found through computer simulation and are specified in the LTE specifications.

DRS Grouping. There are a total of 30 sequence groups, $u \in \{0,1,\ldots,29\}$, each containing one sequence for length less than or equal to 60. This corresponds to transmission bandwidths of 1,2,3,4 and 5 resource blocks. Additionally, there are two sequences (one for v = 0 or 1) for length ≥ 72; corresponding to transmission bandwidths of 6 resource blocks or more.

Note that not all values of m are allowed, where m is the number of resource blocks used for transmission. Only values for m that are the product of powers of 2, 3 and 5 are valid, as shown in the following equation.

$$m = 2^{\alpha_0} \times 3^{\alpha_1} \times 5^{\alpha_2}, \text{ where } \alpha_i \text{ are positive integers}$$

The reason for this restriction is that the DFT sizes of the SC-FDMA precoding operation are limited to values which are the product of powers of 2, 3 and 5. The DFT operation can span more than one resource block, and since each resource block has 12 subcarriers, the total number of subcarriers fed to the DFT will be $12m$. Since the result of $12m$ has to be the product of powers of 2, 3 and 5 this implies that the number of resource blocks must themselves be the product of powers of 2, 3 and 5. Therefore values of m such as 7, 11, 14, 19, etc. are not valid.

For a given time slot, the uplink reference signal sequences to use within a cell are taken from one specific sequence group. If the same group is to be used for all slots then this is known as fixed assignment. On the other hand, if the group number u varies for all slots within a cell this is known as *group hopping*.

Fixed Group Assignment

When fixed group assignment is used, the same group number is used for all slots. For PUCCH, the group number is a function of the cell identity number modulo 30, as shown in the following equation.

$$u = N_{ID}^{cell} \bmod 30, \text{ with } N_{ID}^{cell} = 0,1,\ldots,503$$

Group Hopping

If group hopping is used, the pattern is applied to the calculation of the sequence group number. This pattern is the same for both the PUCCH and PUSCH.

This pattern is defined as the following equation.

$$f_{gh}(n_s) = \sum_{i=0}^{7} c(8n_s + i) \cdot 2^i \bmod 30$$

As shown in the preceding equation, this group hopping pattern is a function of the slot number n_s and is calculated making use of a pseudorandom binary sequence $c(k)$, generated using a length-30 *Gold* code. To generate the group hopping pattern, the PRBS generator is initialized with the following value at the start of each radio frame.

$$c_{init} = \left\lfloor \frac{N_{ID}^{cell}}{30} \right\rfloor$$

For PUCCH with group hopping, the group number, u, is given by the following equation.

$$u = \left(f_{gh}(n_s) + \left(\left(N_{ID}^{cell} \bmod 30 \right) + \Delta_{ss} \right) \right) \bmod 30$$

PUCCH Format 2 Resource Element Mapping

The resource blocks assigned to L1/L2 control information within a subframe are located at the edges of the total available cell bandwidth. A frequency hopping pattern is used where the lower end of the available UL spectrum is used in the first slot of the subframe and the higher end on the second; this adds a level of frequency diversity.

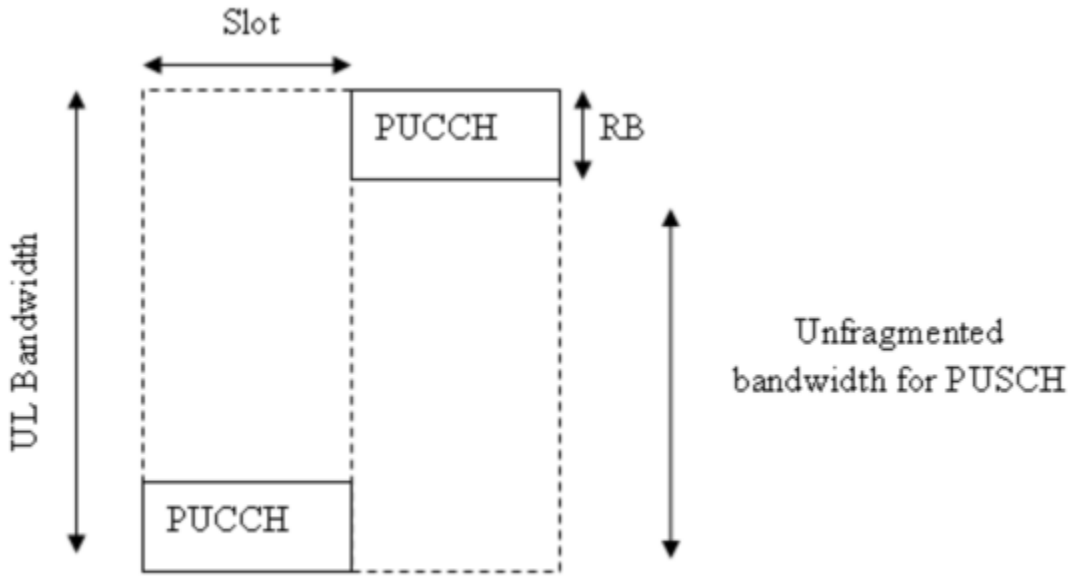
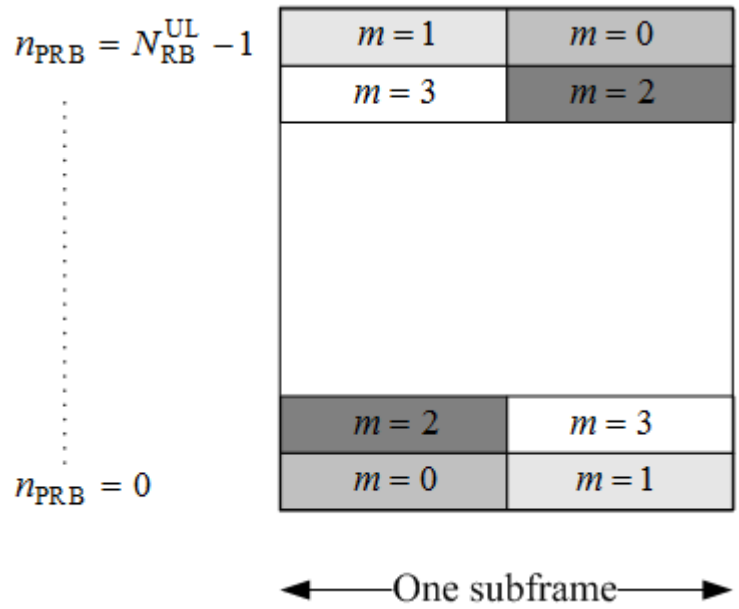


Figure 1: Location of PUCCH RB within a subframe

Bandwidth edges are used so that a large unfragmented portion of the spectrum remains to allocate to the PUSCH. If this spectrum was fragmented by multiple PUCCHs it would not be possible to allocate a number of contiguous RBs to a UE, hence the single carrier nature of SC-FDMA would be lost.

There is a single index, *m*, derived from the PUCCH resource index and other parameters that specifies the location of the PUCCH in time/frequency. When *m* is 0, the PUCCH occupies the lowest RB in the first slot and the highest RB in the second slot of a subframe. When *m* is 1, the opposite corners are used—the highest RB in the first slot and the lowest RB in the second slot. As *m* increases further, the allocated resource blocks move in towards the band center as shown in the following figure.



The resource for PUCCH formats 2, 2a, and 2b is indicated by a single scalar value called resource index. From this value, the cyclic shift can be determined. Time and frequency allocated resources are not derived from this value. However, higher layers are in full control of when and where control information is transmitted.

References

[1] 3GPP TS 36.212. "Multiplexing and channel coding." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`ltePUCCH2` | `ltePUCCH2Indices` | `ltePUCCH2DRS` | `ltePUCCH2DRSIndices` | `lteULResourceGrid`

Related Examples

- “Model PUCCH Format 2” on page 3-17

Downlink Shared Channel

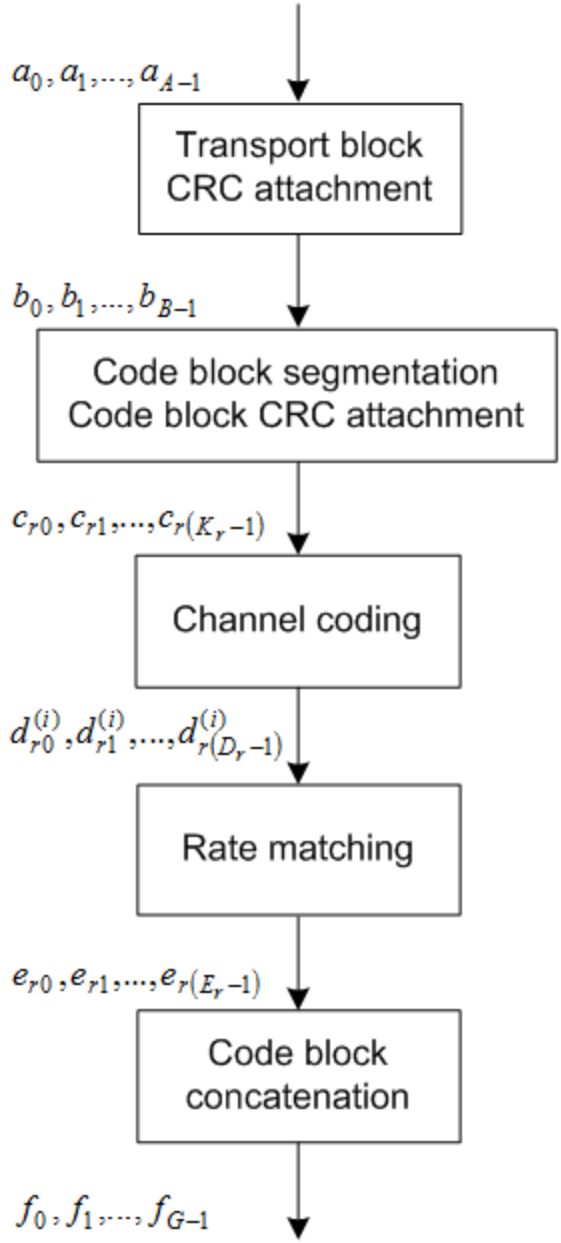
In this section...
"DL-SCH Coding" on page 1-104
"The PDSCH" on page 1-113

The physical downlink shared channel is used to transmit the downlink shared channel (DL-SCH). The DL-SCH is the transport channel used for transmitting downlink data (a transport block).

DL-SCH Coding

- "Transport Block CRC Attachment" on page 1-131
- "Code Block Segmentation and CRC Attachment" on page 1-132
- "Channel Coding" on page 1-107
- "Rate Matching" on page 1-136
- "Code Block Concatenation" on page 1-139

To create the PDSCH payload, a transport block of length A, denoted as $a_0, a_1, \ldots, a_{A-1}$, undergoes transport block CRC attachment, code block segmentation and code block CRC attachment, channel coding, rate matching and code block concatenation. The coding steps are illustrated in the following block diagram.


$a_0, a_1, \ldots, a_{A-1}$
Transport block CRC attachment
$b_0, b_1, \ldots, b_{B-1}$
Code block segmentation Code block CRC attachment
$c_{r0}, c_{r1}, \ldots, c_{r(K_r-1)}$
Channel coding
$d_{r0}^{(i)}, d_{r1}^{(i)}, \ldots, d_{r(D_r-1)}^{(i)}$
Rate matching
$e_{r0}, e_{r1}, \ldots, e_{r(E_r-1)}$
Code block concatenation
$f_0, f_1, \ldots, f_{G-1}$

Transport Block CRC Attachment

A cyclic redundancy check (CRC) is used for error detection in transport blocks. The entire transport block is used to calculate the CRC parity bits. The transport block is divided by a cyclic generator polynomial, described as g_{CRC24A} in section 5.1.1 of "TS 36.212" on page 4-6, to generate 24 parity bits. These parity bits are then appended to the end of transport block.

Code Block Segmentation and CRC Attachment

The input block of bits to the code segmentation block is denoted by $b_0, b_1, \ldots, b_{B-1}$, where $B = A + 24$. In LTE, a minimum and maximum code block size is specified so the block sizes are compatible with the block sizes supported by the turbo interleaver.

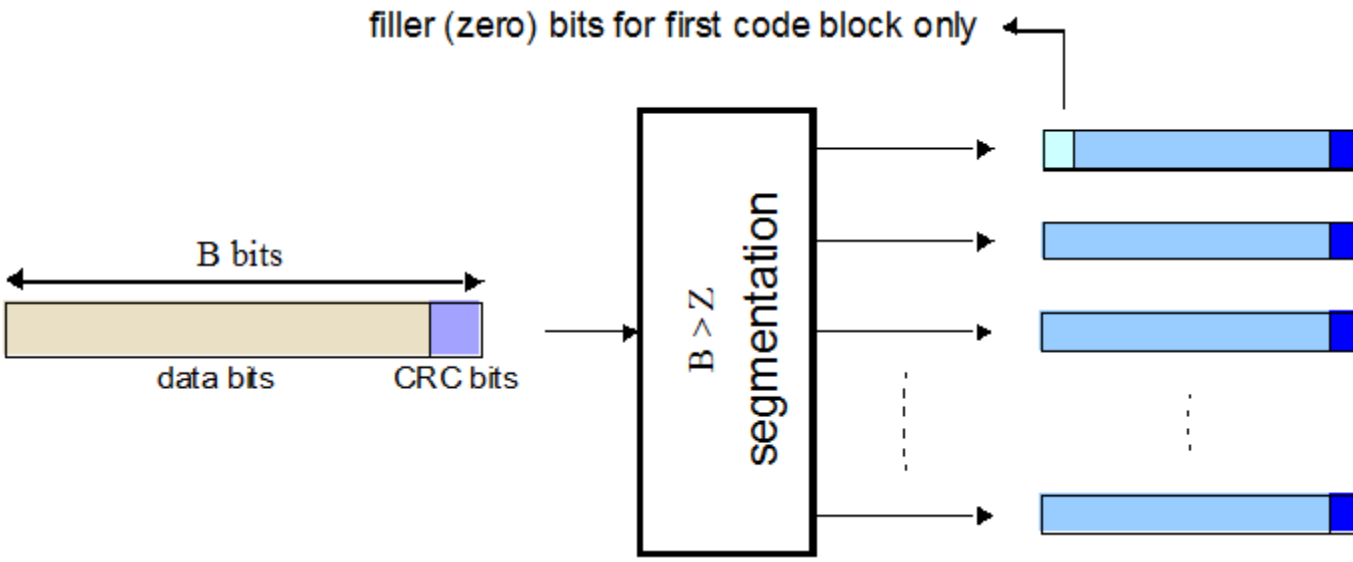
- Minimum code block size = 40 bits
- Maximum code block size, Z = 6144 bits

If the length of the input block, B, is greater than the maximum code block size, the input block is segmented.

When the input block is segmented, it is segmented into $C = \lceil B/(Z-L) \rceil$, where L is 24. Therefore, $C = \lceil B/6120 \rceil$ code blocks.

Each code block has a 24-bit CRC attached to the end, calculated as described in "Transport Block CRC Attachment" on page 1-131, but the generator polynomial, described as g_{CRC24B} in section 5.1.1 of "TS 36.212" on page 4-6 is used.

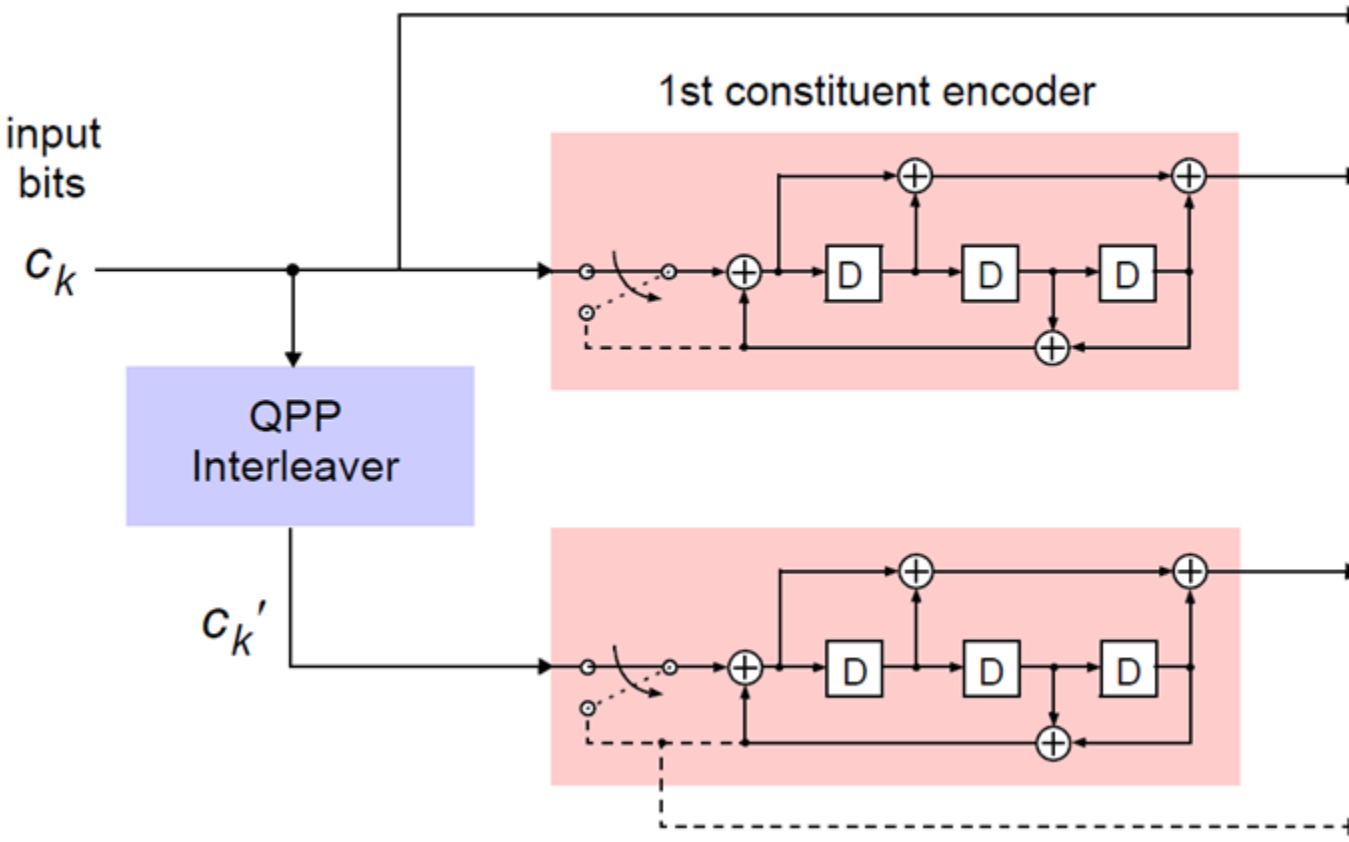
If required, filler bits are appended to the start of the segment so that the code block sizes match a set of valid turbo interleaver block sizes, as shown in the following figure.



If no segmentation is needed, only one code block is produced. If B is less than the minimum size, filler bits (zeros) are added to the beginning of the code block to achieve a total of 40 bits.

Channel Coding

The code blocks undergo turbo coding. Turbo coding is a form of forward error correction that improves the channel capacity by adding redundant information. The turbo encoder scheme used is a Parallel Concatenated Convolutional Code (PCCC) with two recursive convolutional coders and a "contention-free" Quadratic Permutation Polynomial (QPP) interleaver, as shown in the following figure.



The output of the encoder is three streams, $d_k^{(0)}$, $d_k^{(1)}$, and $d_k^{(2)}$, to achieve a code rate of 1/3.

Constituent Encoders. The input to the first constituent encoder is the input bit stream to the turbo coding block. The input to the second constituent encoder is the output of the QPP interleaver, a permutated version of the input sequence.

There are two output sequences from each encoder, a systematic (x_k, x_k') and a parity (z_k, z_k'). Only one of the systematic sequences (x_k) is used as an

output because the other (x'_k) is simply a permutated version of the chosen systematic sequence. The transfer function for each constituent encoder is given by the following equation.

$$G(D) = \left[1, \frac{g_1(D)}{g_0(D)}\right]$$

The first element, 1, represents the systematic output transfer function.

The second element, $\left(\frac{g_1(D)}{g_0(D)}\right)$, represents the recursive convolutional output transfer function.

$$g_0(D) = 1 + D^2 + D^3$$

$$g_1(D) = 1 + D + D^3$$

The output for each sequence can be calculated using the transfer function.

The encoder is initialized with all zeros. If the code block to be encoded is the 0-th and filler bits (F) are used, the input to the encoder (c_k) is set to zero and the output (x_k) and (z_k) set to <NULL> for $k = 0,\ldots,F-1$.

Trellis Termination for Turbo Encoder. In a normal convolutional coder, the coder is driven to an *all zeros* state upon termination by appending zeros onto the end of the input data stream. Since the decoder knows the start and end state of the encoder, it can decode the data. Driving a recursive coder to an all zeros state using this method is not possible. To overcome this problem, trellis termination is used.

Upon termination, the tail bits are fed back to the input of each encoder using a switch. The first three tail bits are used to terminate each encoder.

The QPP Interleaver. The role of the interleaver is to spread the information bits such that in the event of a burst error, the two code streams are affected differently, allowing data to still be recovered.

The output of the interleaver is a permutated version of the input data, as shown in the following equations.

$$c'_i = c_{\Pi(i)}, i = 0,1,\ldots,(K-1)$$

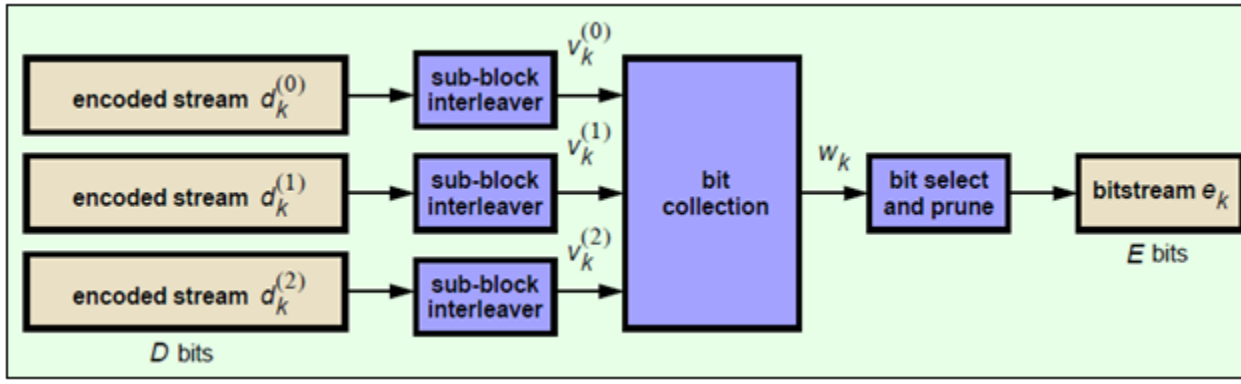
$$\Pi(i) = \left(f_1 i + f_2 i^2\right) \bmod K$$

The variable K is the input length. The variables f_1 and f_2 are coefficients chosen depending on K, in table 5.1.3-3 of "TS 36.212" on page 4-6. For example, K=40, f_1=3, and f_2=10, yields the following sequence.

$$\Pi(i) = 0,13,6,19,12,25,18,31,24,37,30,3,36,9,2,15,8,21,14,27,20,33,26,39,32,5,38,\ldots$$

Rate Matching

The rate matching block creates an output bitstream with a desired code rate. As the number of bits available for transmission depends on the available resources the rate matching algorithm is capable of producing any arbitrary rate. The three bitstreams from the turbo encoder are interleaved followed by bit collection to create a circular buffer. Bits are selected and pruned from the buffer to create an output bitstream with the desired code rate. The process is illustrated in the following figure.



Sub-block Interleaver. The three sub-block interleavers used in the rate matching block are identical. Interleaving is a technique to reduce the impact of burst errors on a signal as consecutive bits of data will not be corrupted.

The sub-block interleaver reshapes the encode bit sequence, row-by-row, to form a matrix with $C_{Subblock}^{TC} = 32$ columns and $R_{Subblock}^{TC}$ rows. The variable $R_{Subblock}^{TC}$ is determined by finding the minimum integer such that the number of encoded input bits is $D \le \left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right)$. If $\left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right) > D$, N_D <NULL>'s are appended onto the front of the encoded sequence. In this case, $N_D + D = \left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right)$.

For blocks $d_k^{(0)}$ and $d_k^{(1)}$, inter-column permutation is performed on the matrix to reorder the columns as shown in the following pattern.

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

The output of the block interleaver for blocks $d_k^{(0)}$ and $d_k^{(1)}$ is the bit sequence read out column-by-column from the inter-column permutated matrix to create a stream $K_\pi = \left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right)$ bits long.

For block $d_k^{(2)}$, the elements within the matrix are permutated separately based on the permutation pattern shown above, but modified to create a permutation which is a function of the variables $R_{Subblock}^{TC}$, $C_{Subblock}^{TC}$, k, and K_π. This process creates three interleaved bitstreams.

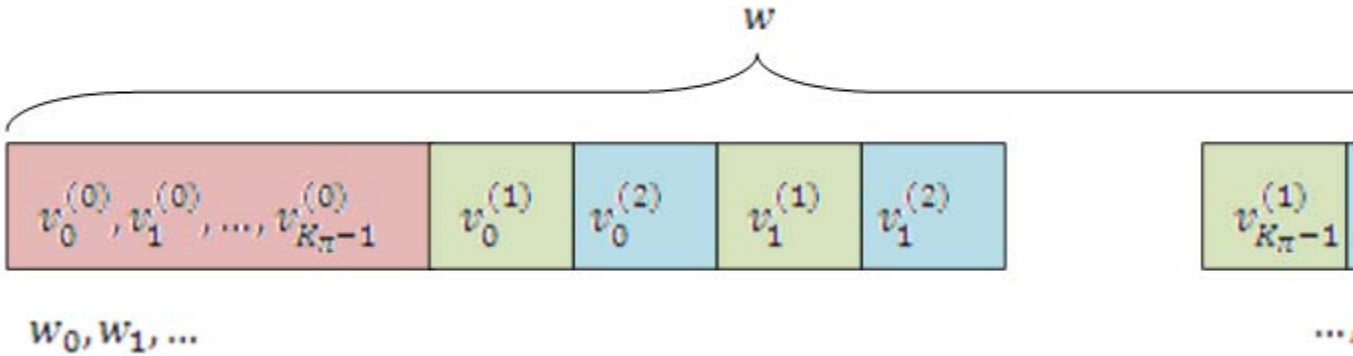
$d_k^{(0)} \rightarrow v_k^{(0)}$

$d_k^{(1)} \rightarrow v_k^{(1)}$

$$d_k^{(2)} \rightarrow v_k^{(2)}$$

Bit Collection, Selection, and Transmission. The bit collection stage creates a virtual circular buffer by combining the three interleaved encoded bit streams.

$v_k^{(1)}$ and $v_k^{(2)}$ are combined by interlacing successive values of $v_k^{(1)}$ and $v_k^{(2)}$. This combination is then appended onto the end of $v_k^{(0)}$ to create the circular buffer w_k shown in the following figure.



Interlacing allows equal levels of protection for each parity sequence.

Bits are then selected and pruned from the circular buffer to create an output sequence length which meets the desired code rate.

The Hybrid Automatic Repeat Request (HARQ) error correction scheme is incorporated into the rate-matching algorithm of LTE. For any desired code rate the coded bits are output serially from the circular buffer from a starting location, given by the redundancy version (RV), wrapping around to the beginning of the buffer if the end of the buffer is reached. `NULL` bits are discarded. Different RVs and hence starting points allow for the retransmission of selected data. Being able to select different starting points enables the following two main methods of recombining data at the receiver in the HARQ process.

- Chase combining — retransmissions contain the same data and parity bit.
- Incremental redundancy — retransmissions contain different information so the receiver gains knowledge upon each retransmission.

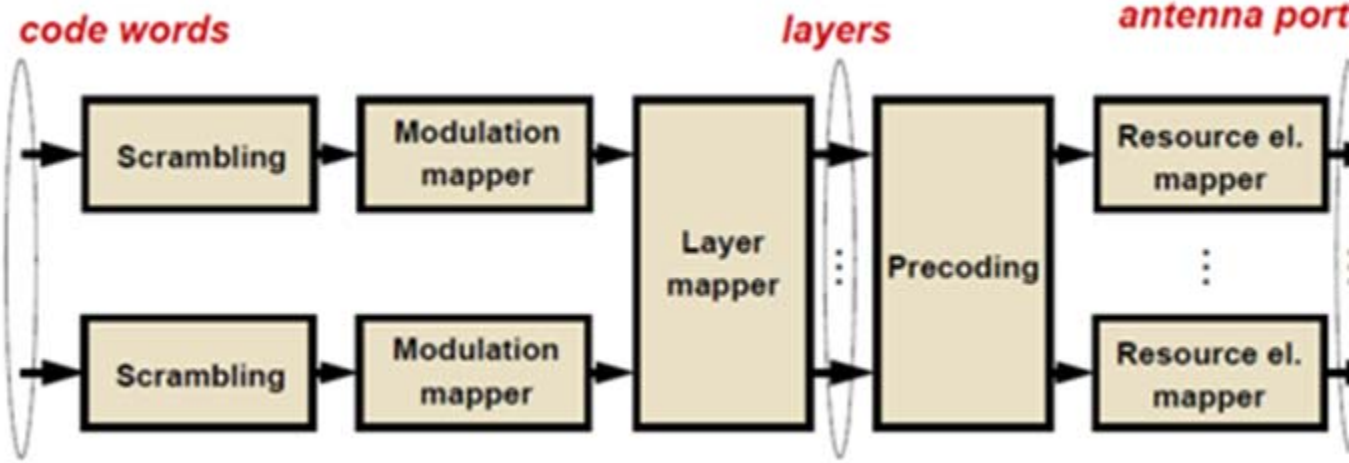
Code Block Concatenation

In this stage, the rate matched code blocks are concatenated back together. This task is done by sequentially concatenating the rate-matched blocks together to create the output of the channel coding, f_k for $k = 0,\ldots,G-1$.

The PDSCH

- "PDSCH Scrambling" on page 1-114
- "Modulation" on page 1-115
- "Layer Mapping" on page 1-115
- "Precoding" on page 1-120
- "Valid Codeword, Layer and Precoding Scheme Combinations" on page 1-127
- "Mapping to Resource Elements" on page 1-128

One or two coded transport blocks (codewords) can be transmitted simultaneously on the PDSCH depending on the precoding scheme used (see section 2.5). The coded DL-SCH code-words undergo scrambling, modulation, layer mapping, precoding and resource element mapping as shown in the following figure.



PDSCH Scrambling

The codewords are bit-wise multiplied with an orthogonal sequence and a UE-specific scrambling sequence to create the following sequence of symbols for each codeword, q.

$$\tilde{b}^{(q)}(0),\ldots,\tilde{b}^{(q)}\left(M_{bit}^{(q)}-1\right)$$

The variable $M_{bit}^{(q)}$ is the number of bits in codeword q.

The scrambling sequence is pseudo-random, created using a length-31 Gold sequence generator and initialized using the slot number within the radio network temporary identifier associated with the PDSCH transmission, n_{RNTI}, the cell ID, N_{ID}^{cell}, the slot number within the radio frame, n_s, and the codeword index, $q=\{0,1\}$, at the start of each subframe.

$$c_{init}=n_{RNTI}\times 2^{14}+q\times 2^{13}+\left\lfloor\frac{n_s}{2}\right\rfloor\times 2^9+N_{ID}^{cell}$$

Scrambling with a cell-specific sequence serves the purpose of intercell interference rejection. When a UE descrambles a received bitstream with a known cell-specific scrambling sequence, interference from other cells will be descrambled incorrectly and therefore only appear as uncorrelated noise.

Modulation

The scrambled codewords undergo QPSK, 16QAM or 64QAM modulation. This choice created flexibility to allow the scheme to maximize the data transmitted depending on the channel conditions.

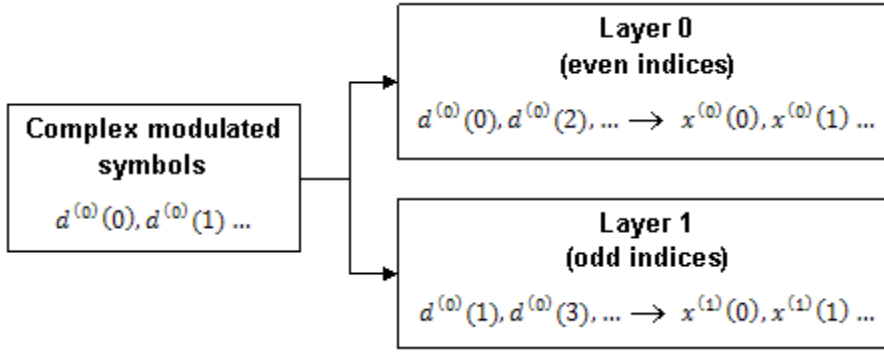
Layer Mapping

The complex symbols are mapped to one, two, or four layers depending on the number of transmit antennas used. The complex modulated input symbols, $d(i)$, are mapped onto v layers, $x^{(0)}(i), x^{(1)}(i), \ldots, x^{(v-1)}(i)$.

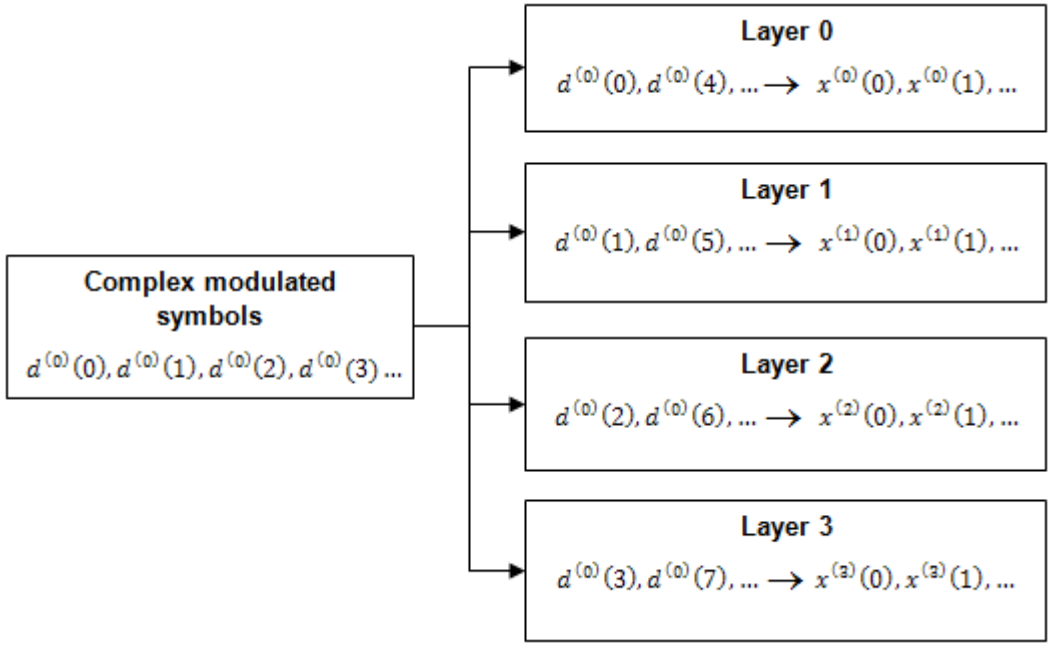
If a single antenna port is used, only one layer is used. Therefore, $x^{(0)}(i) = d^{(0)}(i)$.

Layer Mapping for Transmit Diversity. If transmitter diversity is used, the input symbols are mapped to layers based on the number of layers.

- **Two Layers** — Even symbols are mapped to layer 0 and odd symbols are mapped to layer 1, as shown in the following figure.



- **Four Layers** — The input symbols are mapped to layers sequentially, as shown in the following figure.



If the total number of input symbols is not an integer multiple of four ($M_{symb}^{(0)} \bmod 4 \neq 0$) two null symbols are appended onto the end. This creates a total number of symbols that is an integer multiple of four because the original number of symbols is always an integer multiple of two.

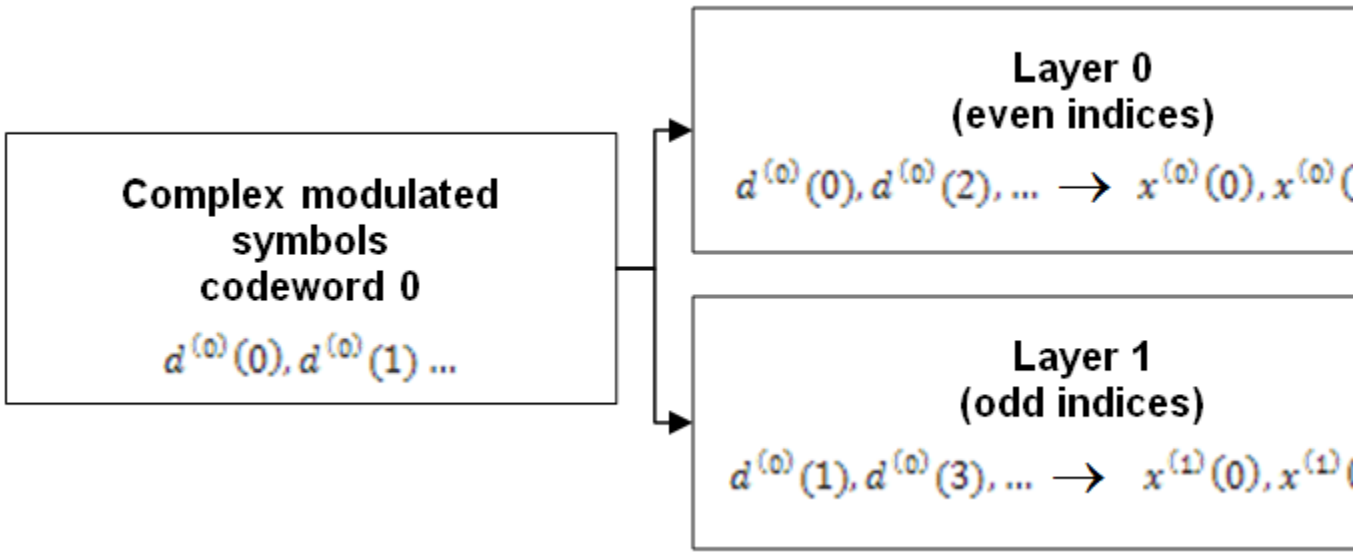
Layer Mapping for Spatial Multiplexing. In the case of spatial multiplexing, the number of layers used is always less or equal to the number of antenna ports used for transmission of the physical channel.

One Layer

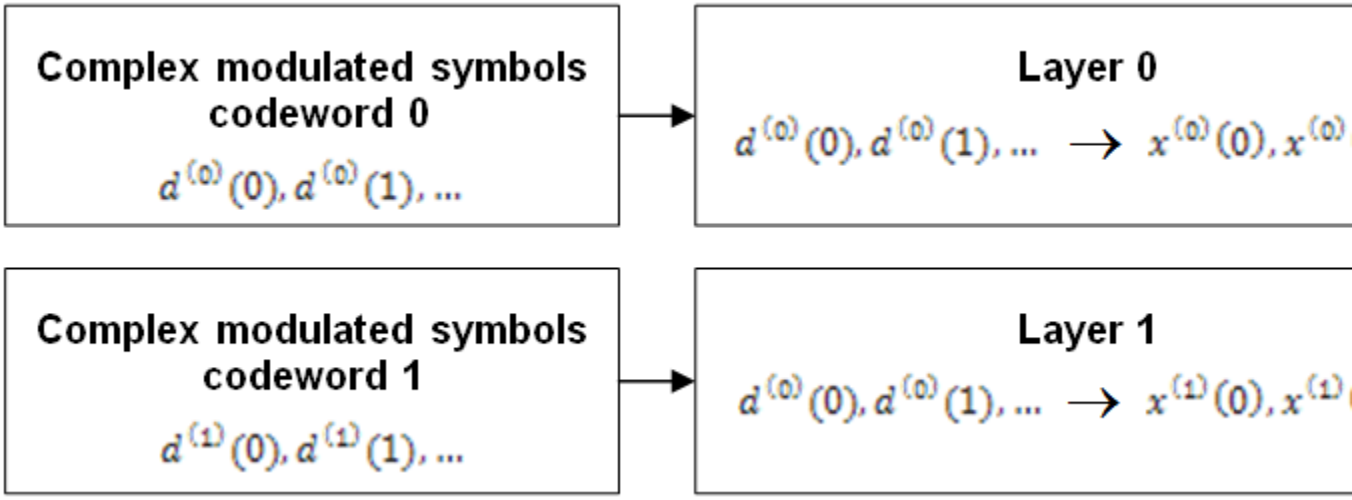
If only one layer is used, $x^{(0)}(i) = d^{(0)}(i)$.

Two Layers

- **One Codeword** — Even symbols are mapped to layer 0 and odd symbols are mapped to layer 1, as shown in the following figure.

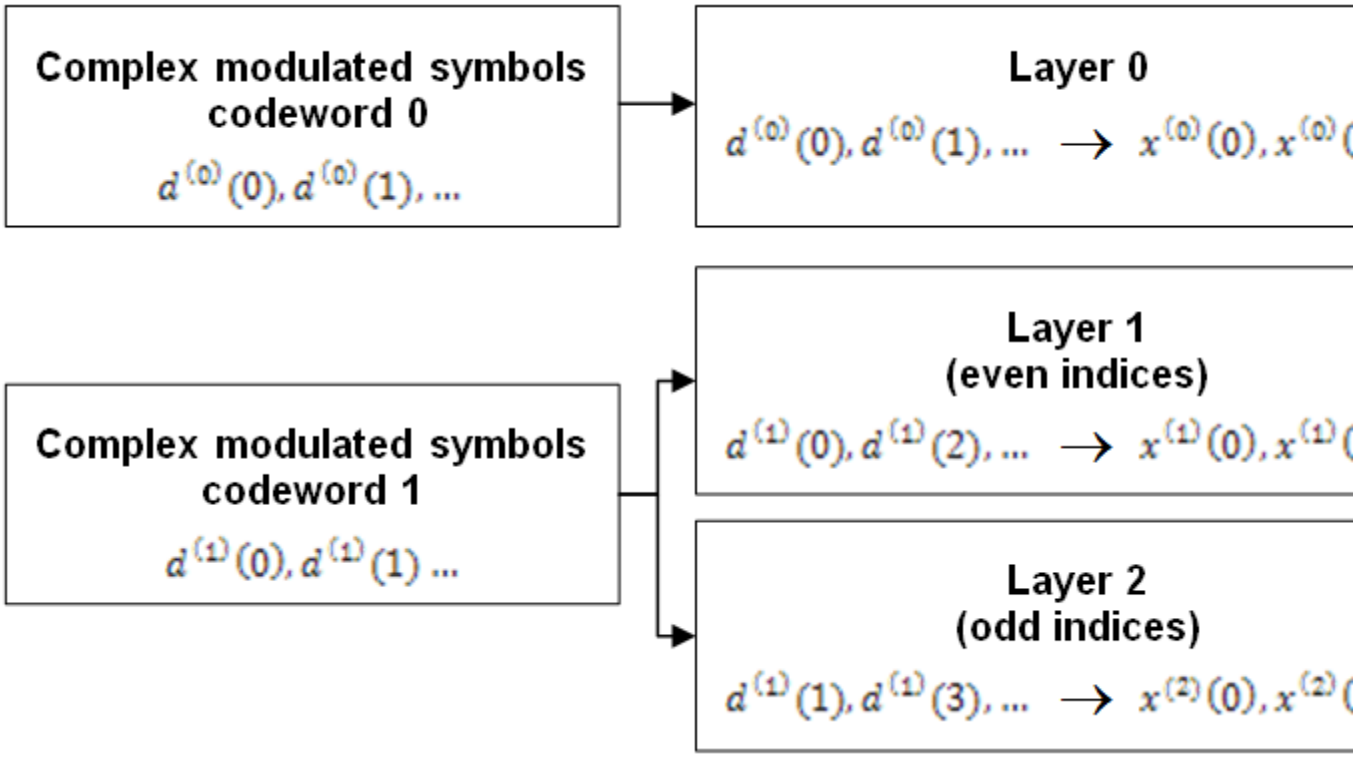


- **Two Codewords** — Codeword 0 is mapped to layer 0 and codeword 1 is mapped to layer 1, as shown in the following figure.



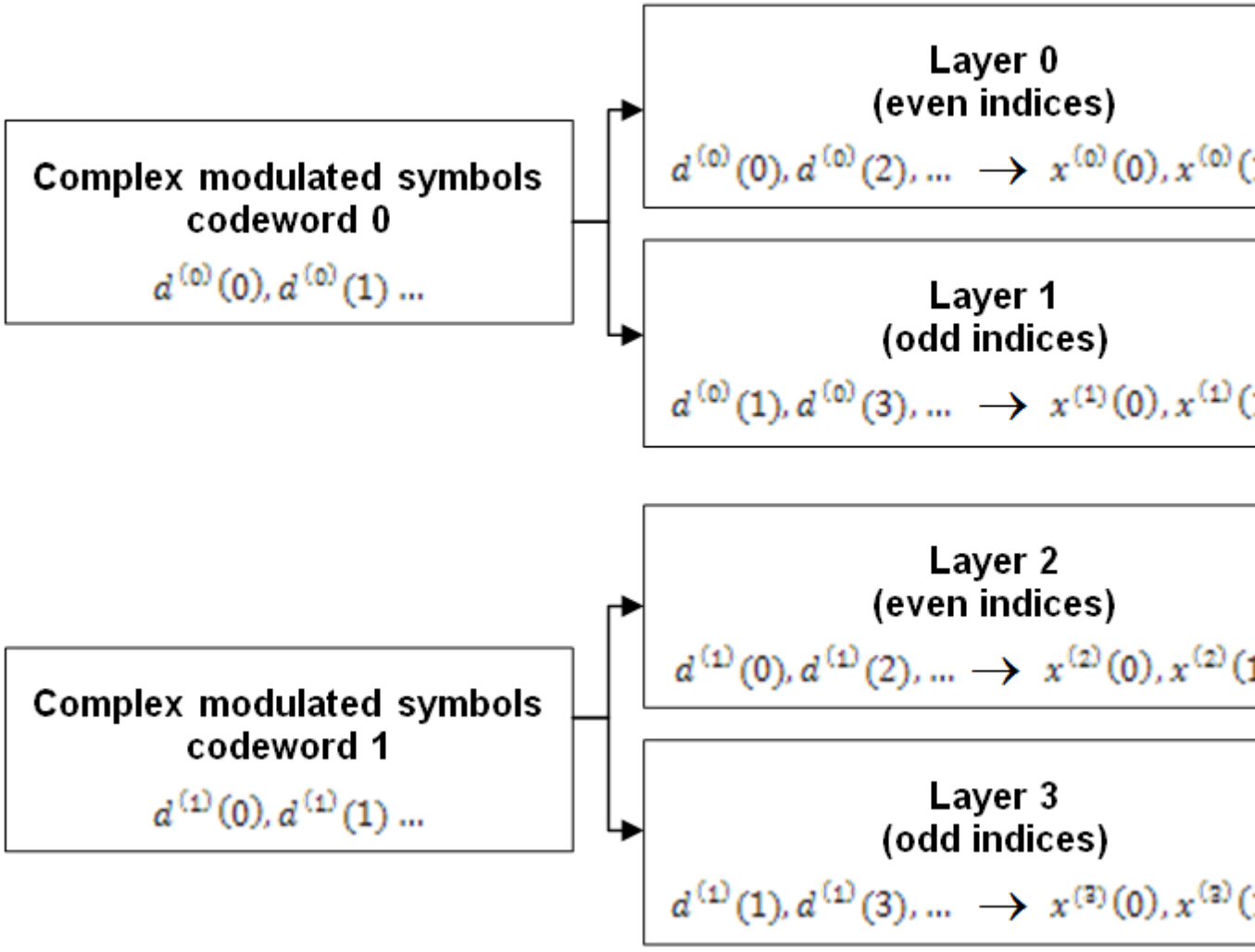
Three Layers

In the case of three layers, codeword 0 is mapped to layer 0 and even symbols within codeword 1 are mapped to layer 1 and odd symbols within codeword 1 are mapped to layer 2.



Four Layers

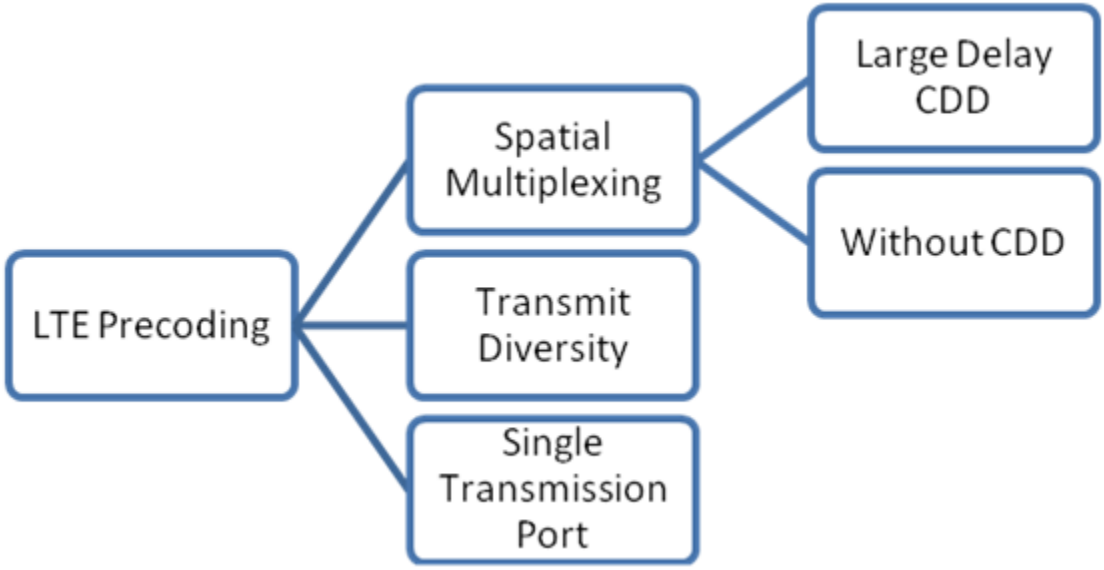
In the case of four layers, two codewords must be used. In this case, the even symbols of codeword 0 are mapped to layer 0, and the odd symbols are mapped to layer 1. The even symbols of codeword 1 are mapped to layer 2, and the odd symbols are mapped to layer 3.



Precoding

Three types of precoding are available in LTE for the PDSCH—spatial multiplexing, transmit diversity, and single antenna port transmission. Within spatial multiplexing, there are two schemes—precoding with large delay cyclic delay diversity (CDD), also known as open loop spatial

multiplexing, and precoding without CDD, also known as closed loop spatial multiplexing. The various types of precoding are illustrated in the following tree diagram.



The precoder takes a block from the layer mapper, $x^{(0)}(i), x^{(1)}(i), \ldots, x^{(v-1)}(i)$, and generates a sequence for each antenna port, $y^{(p)}(i)$. The variable p is the transmit antenna port number, and can assume values of {0}, {0,1}, or {0,1,2,3}.

Single Antenna Port Precoding. For transmission over a single antenna port, no processing is carried out, as shown in the following equation.

$$y^{(p)}(i) = x^{(0)}(i)$$

Precoding for Large Delay CDD Spatial Multiplexing. CDD operation applies a cyclic shift, which is a delay of N_{FFT}/v samples to each antenna, where N_{FFT} is the size of the OFDM FFT. The use of CDD improves the robustness of performance by randomizing the channel frequency response, reducing the probability of deep fading.

Precoding with CDD for spatial multiplexing is defined by the following equation.

$$\begin{pmatrix} y^{(0)}(i) \\ \vdots \\ y^{(p-1)}(i) \end{pmatrix} = W(i) \times D(i) \times U \times \begin{pmatrix} x^{(0)}(i) \\ \vdots \\ x^{(v-1)}(i) \end{pmatrix}$$

Values of the precoding matrix, $W(i)$, of size $P \times v$, are selected from a codebook configured by the eNodeB and user equipment. The precoding codebook is described in "Spatial Multiplexing Precoding Codebook" on page 1-123. Every group of symbols at index *i* across all available layers can use a different precoding matrix, if required. The supporting matrices, $D(i)$ and *U*, are given for a various numbers of layers in the following table.

Number of layers, v	U	$D(i)$
2	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & e^{-j2\pi/2} \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & e^{-j2\pi i/2} \end{pmatrix}$
3	$\frac{1}{\sqrt{3}}\begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3} \\ 1 & e^{-j4\pi/3} & e^{-j8\pi/3} \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-j2\pi i/3} & 0 \end{pmatrix}$
4	$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-j2\pi/4} & e^{-j4\pi/4} & e^{-j6\pi/4} \\ 1 & e^{-j4\pi/4} & e^{-j8\pi/4} & e^{-j12\pi/4} \\ 1 & e^{-j6\pi/4} & e^{-j12\pi/4} & e^{-j18\pi/4} \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-j2\pi i/4} & 0 & 0 \\ 0 & 0 & e^{-j4\pi i/4} & 0 \\ 0 & 0 & 0 & e^{-j6\pi i/4} \end{pmatrix}$

Precoding for Spatial Multiplexing without CDD. Precoding for spatial multiplexing without CDD is defined by the following equation.

$$\begin{pmatrix} y^{(0)}(i) \\ \vdots \\ y^{(p-1)}(i) \end{pmatrix} = W(i) \times \begin{pmatrix} x^{(0)}(i) \\ \vdots \\ x^{(v-1)}(i) \end{pmatrix}$$

Values of the precoding matrix, $W(i)$, of size $P \times \upsilon$, are selected from a codebook configured by the eNodeB and user equipment. Every group of symbols at index i across all available layers can use a different precoding matrix if required. For more information on the precoding codebook, see "Spatial Multiplexing Precoding Codebook" on page 1-123.

Spatial Multiplexing Precoding Codebook. The precoding matrices for antenna ports {0,1} are given in the following table.

Codebook index	Number of layers, υ	
	1	**2**
0	$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$	$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$
1	$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$	$\frac{1}{2}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$
2	$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\j\end{pmatrix}$	$\frac{1}{2}\begin{pmatrix}1 & 1\\j & -j\end{pmatrix}$
3	$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-j\end{pmatrix}$	—

The precoding matrices for antenna ports {0,1,2,3} are given in the following table.

Codebook index	u_n	Number of layers		
		1	2	
0	$u_0 = \begin{bmatrix} 1 & -1 & -1 & -1 \end{bmatrix}^T$	$W_0^{\{1\}}$	$W_0^{\{14\}}/\sqrt{2}$	$W_0^{\{12}$
1	$u_1 = \begin{bmatrix} 1 & -j & 1 & j \end{bmatrix}^T$	$W_1^{\{1\}}$	$W_1^{\{12\}}/\sqrt{2}$	$W_1^{\{12}$
2	$u_2 = \begin{bmatrix} 1 & 1 & -1 & 1 \end{bmatrix}^T$	$W_2^{\{1\}}$	$W_2^{\{12\}}/\sqrt{2}$	$W_2^{\{12}$
3	$u_3 = \begin{bmatrix} 1 & j & 1 & -j \end{bmatrix}^T$	$W_3^{\{1\}}$	$W_3^{\{12\}}/\sqrt{2}$	$W_3^{\{12}$
4	$u_4 = \begin{bmatrix} 1 & (-1-j)/\sqrt{2} & -j & (1-j)/\sqrt{2} \end{bmatrix}^T$	$W_4^{\{1\}}$	$W_4^{\{14\}}/\sqrt{2}$	$W_4^{\{12}$
5	$u_5 = \begin{bmatrix} 1 & (1-j)/\sqrt{2} & j & (-1-j)/\sqrt{2} \end{bmatrix}^T$	$W_5^{\{1\}}$	$W_5^{\{14\}}/\sqrt{2}$	$W_5^{\{12}$
6	$u_6 = \begin{bmatrix} 1 & (1+j)/\sqrt{2} & -j & (-1+j)/\sqrt{2} \end{bmatrix}^T$	$W_6^{\{1\}}$	$W_6^{\{13\}}/\sqrt{2}$	$W_6^{\{13}$
7	$u_7 = \begin{bmatrix} 1 & (-1+j)/\sqrt{2} & j & (1+j)/\sqrt{2} \end{bmatrix}^T$	$W_7^{\{1\}}$	$W_7^{\{13\}}/\sqrt{2}$	$W_7^{\{13}$
8	$u_8 = \begin{bmatrix} 1 & -1 & 1 & 1 \end{bmatrix}^T$	$W_8^{\{1\}}$	$W_8^{\{12\}}/\sqrt{2}$	$W_8^{\{12}$
9	$u_9 = \begin{bmatrix} 1 & -j & -1 & -j \end{bmatrix}^T$	$W_9^{\{1\}}$	$W_9^{\{14\}}/\sqrt{2}$	$W_9^{\{13}$
10	$u_{10} = \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix}^T$	$W_{10}^{\{1\}}$	$W_{10}^{\{13\}}/\sqrt{2}$	$W_{10}^{\{12}$
11	$u_{11} = \begin{bmatrix} 1 & j & -1 & j \end{bmatrix}^T$	$W_{11}^{\{1\}}$	$W_{11}^{\{13\}}/\sqrt{2}$	$W_{11}^{\{13}$
12	$u_{12} = \begin{bmatrix} 1 & -1 & -1 & 1 \end{bmatrix}^T$	$W_{12}^{\{1\}}$	$W_{12}^{\{12\}}/\sqrt{2}$	$W_{12}^{\{12}$
13	$u_{13} = \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}^T$	$W_{13}^{\{1\}}$	$W_{13}^{\{13\}}/\sqrt{2}$	$W_{13}^{\{12}$
14	$u_{14} = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}^T$	$W_{14}^{\{1\}}$	$W_{14}^{\{13\}}/\sqrt{2}$	$W_{14}^{\{12}$
15	$u_{15} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$	$W_{15}^{\{1\}}$	$W_{15}^{\{12\}}/\sqrt{2}$	$W_{15}^{\{12}$

The precoding matrix, $W_n^{\{s\}}$, is the matrix defined by the columns in set $\{s\}$ by the following equation.

$$W_n = I - 2u_n u_n^H / u_n^H u_n$$

In the preceding equation, I is a 4-by-4 identity matrix. The vector u_n is given in the preceding table.

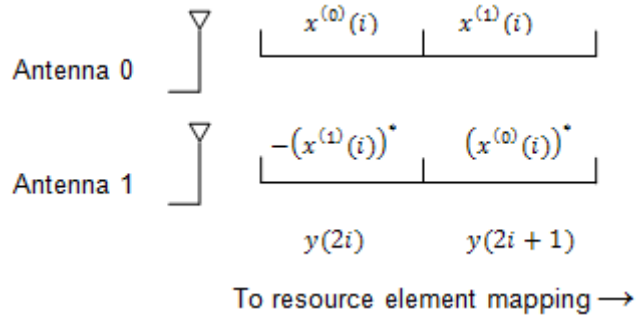
Precoding for Transmit Diversity. Precoding for transmit diversity is available on two or four antenna ports.

Two Antenna Ports

An Alamouti scheme is used for precoding, which defines the relationship between input and output as shown in the following equation.

$$\begin{pmatrix} y^{(0)}(2i) \\ y^{(1)}(2i) \\ y^{(0)}(2i+1) \\ y^{(1)}(2i+1) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & j & 0 \\ 0 & -1 & 0 & j \\ 0 & 1 & 0 & j \\ 1 & 0 & -j & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}\left\{x^{(0)}(i)\right\} \\ \mathrm{Re}\left\{x^{(1)}(i)\right\} \\ \mathrm{Im}\left\{x^{(0)}(i)\right\} \\ \mathrm{Im}\left\{x^{(1)}(i)\right\} \end{pmatrix}$$

In the Alamouti scheme, two consecutive symbols, $x^{(0)}(i)$ and $x^{(1)}(i)$, are transmitted in parallel using two antennas with the following mapping, where the asterisk symbol (*) denotes the complex conjugate operation.



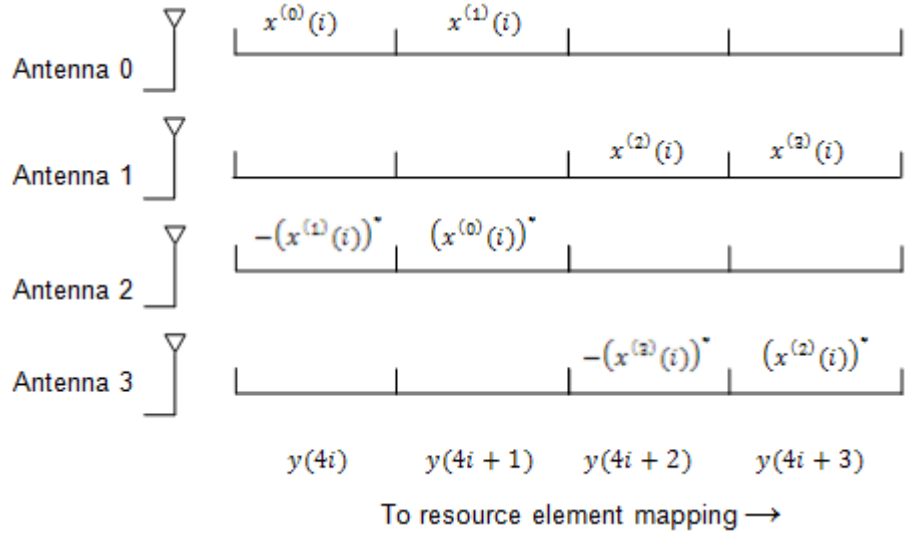
As any two columns in the precoding matrix are orthogonal, the two symbols, $x^{(0)}(i)$ and $x^{(1)}(i)$, can be separated at the UE.

Four Antenna Ports

Precoding for the four antenna port case defines the relationship between the input and output as shown in the following equation.

$$\begin{pmatrix} y^{(0)}(4i) \\ y^{(1)}(4i) \\ y^{(2)}(4i) \\ y^{(3)}(4i) \\ y^{(0)}(4i+1) \\ y^{(1)}(4i+1) \\ y^{(2)}(4i+1) \\ y^{(3)}(4i+1) \\ y^{(0)}(4i+2) \\ y^{(1)}(4i+2) \\ y^{(2)}(4i+2) \\ y^{(3)}(4i+2) \\ y^{(0)}(4i+3) \\ y^{(1)}(4i+3) \\ y^{(2)}(4i+3) \\ y^{(3)}(4i+3) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -j & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & j & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -j & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}\left\{x^{(0)}(i)\right\} \\ \mathrm{Re}\left\{x^{(1)}(i)\right\} \\ \mathrm{Re}\left\{x^{(2)}(i)\right\} \\ \mathrm{Re}\left\{x^{(3)}(i)\right\} \\ \mathrm{Im}\left\{x^{(0)}(i)\right\} \\ \mathrm{Im}\left\{x^{(1)}(i)\right\} \\ \mathrm{Im}\left\{x^{(2)}(i)\right\} \\ \mathrm{Im}\left\{x^{(3)}(i)\right\} \end{pmatrix}$$

In this scheme, two consecutive symbols are transmitted in parallel in two symbol periods using four antennas with the following mapping, where the asterisk symbol (*) denotes the complex conjugate operation.



Valid Codeword, Layer and Precoding Scheme Combinations

As a quick reference of the information described in previous sections, the valid numbers of codewords and layers for each precoding scheme are shown in the following tables..

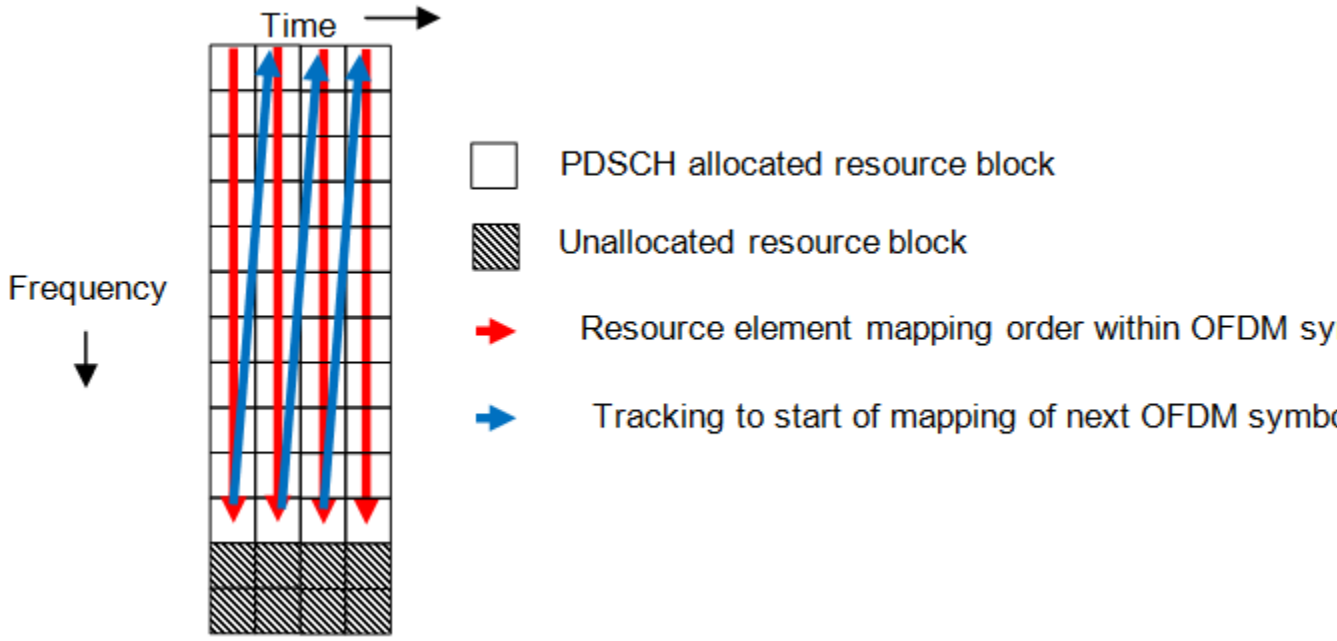
Single Antenna Port	
Codewords	Layers
1	1

Transmit Diversity	
Codewords	Layers
1	2
1	4

Spatial M
Codewords
1
1
2
2
2

Mapping to Resource Elements

For each of the antenna ports used for transmission of the PDSCH, the block of complex valued symbols, $y^{(p)}(i)$, are mapped in sequence to resource elements not occupied by the PCFICH, PHICH, PDCCH, PBCH, or synchronization and reference signals. The number of resource elements mapped to is controlled by the number of resource blocks allocated to the PDSCH. The symbols are mapped by increasing the subcarrier index and mapping all available REs within allocated resource blocks for each OFDM symbol as shown in the following figure.



References

[1] 3GPP TS 36.212. "Multiplexing and channel coding." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

See Also

`lteDLSCH` | `ltePDSCH` | `ltePDSCHIndices` | `lteDLResourceGrid` | `lteDLSCHInfo` | `lteDLSCHDecode` | `lteLayerMap` | `lteLayerDemap` | `lteDLPrecode` | `lteDLDeprecode` | `lteTurboEncode` | `lteTurboDecode` | `ltePDSCHPRBS` | `lteCRCEncode` | `lteCRCDecode` | `lteCodeBlockSegment` | `lteCodeBlockDesegment` | `lteRateMatchTurbo` | `lteRateRecoverTurbo`

Related Examples

- "Model DL-SCH and PDSCH" on page 3-19

Uplink Shared Channel

In this section...
"UL-SCH Coding" on page 1-130
"The PUSCH" on page 1-145
"Demodulation Reference Signals (DRS) on the PUSCH" on page 1-148

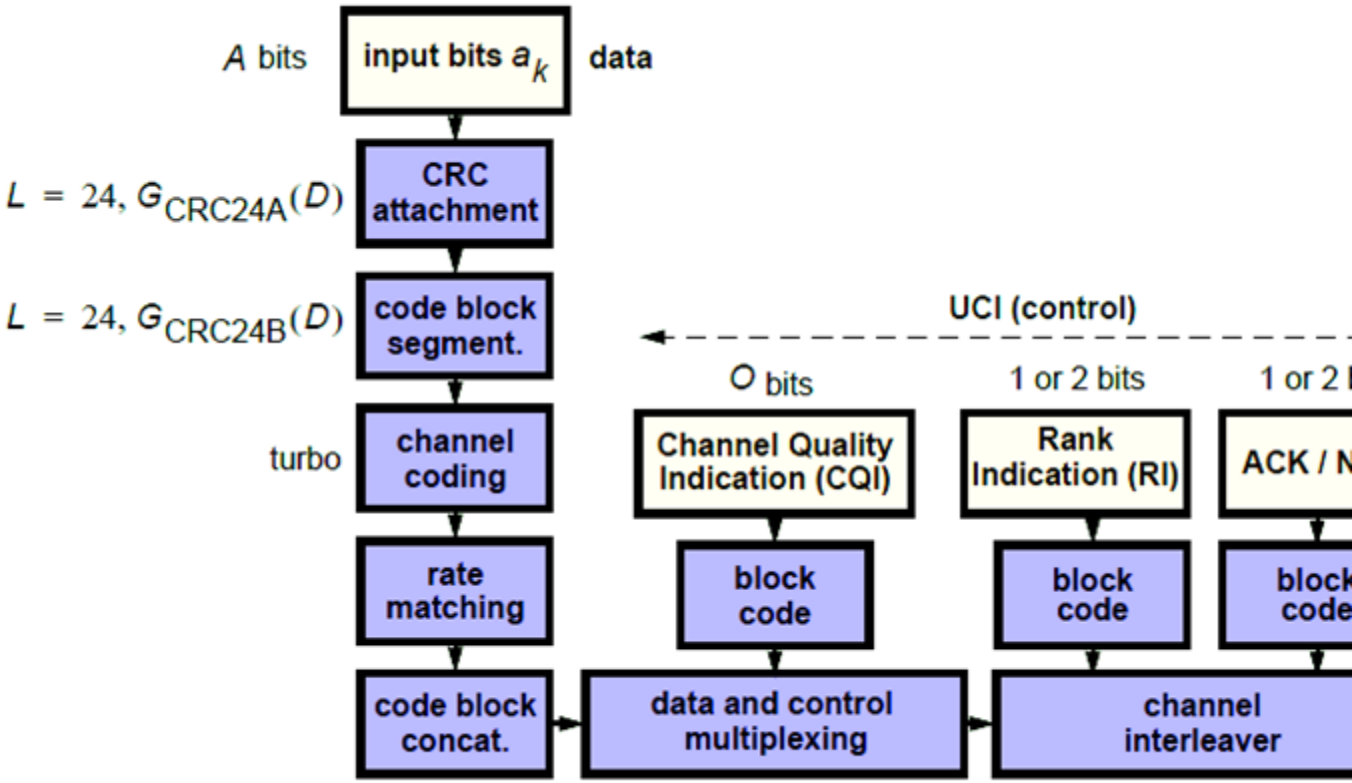
The physical uplink shared channel is used to transmit the uplink shared channel (UL-SCH) and L1 and L2 control information. The UL-SCH is the transport channel used for transmitting uplink data (a transport block). L1 and L2 control signalling can carry the following type of information: HARQ acknowledgements for received DL-SCH blocks, channel quality reports and scheduling requests.

UL-SCH Coding

- "Transport Block CRC Attachment" on page 1-131
- "Code Block Segmentation and CRC Attachment" on page 1-132
- "Channel Coding" on page 1-133
- "Rate Matching" on page 1-136
- "Code Block Concatenation" on page 1-139
- "Channel Coding of Control Information with UL-SCH Data" on page 1-140
- "Data and Control Multiplexing" on page 1-144
- "Channel Interleaver" on page 1-144
- "Channel Coding of Control Information without UL-SCH Data" on page 1-145

To create the PUSCH payload a transport block of length A, denoted as $a_0, a_1, \ldots, a_{A-1}$, undergoes transport block coding. The encoding process includes type-24A CRC calculation, code block segmentation and type-24B CRC attachment if any, turbo encoding, rate matching with RV and code block concatenation. This processing is described in sections 5.2.2.1 to 5.2.2.5 and 5.2.2.8 of [1].

The transport block coding and control information coding and multiplexing steps are illustrated in the following block diagram.



It is possible for the PUSCH to carry only control information and no data. In this case, only the control information coding and multiplexing chain are followed as per the preceding diagram.

Transport Block CRC Attachment

A cyclic redundancy check (CRC) is used for error detection in transport blocks. The entire transport block is used to calculate the CRC parity bits. The transport block is divided by a cyclic generator polynomial, described as

g_{CRC24A} in section 5.1.1 of "TS 36.212" on page 4-6, to generate 24 parity bits. These parity bits are then appended to the end of transport block.

Code Block Segmentation and CRC Attachment

The input block of bits to the code segmentation block is denoted by $b_0, b_1, \ldots, b_{B-1}$, where $B = A + 24$. In LTE, a minimum and maximum code block size is specified so the block sizes are compatible with the block sizes supported by the turbo interleaver.

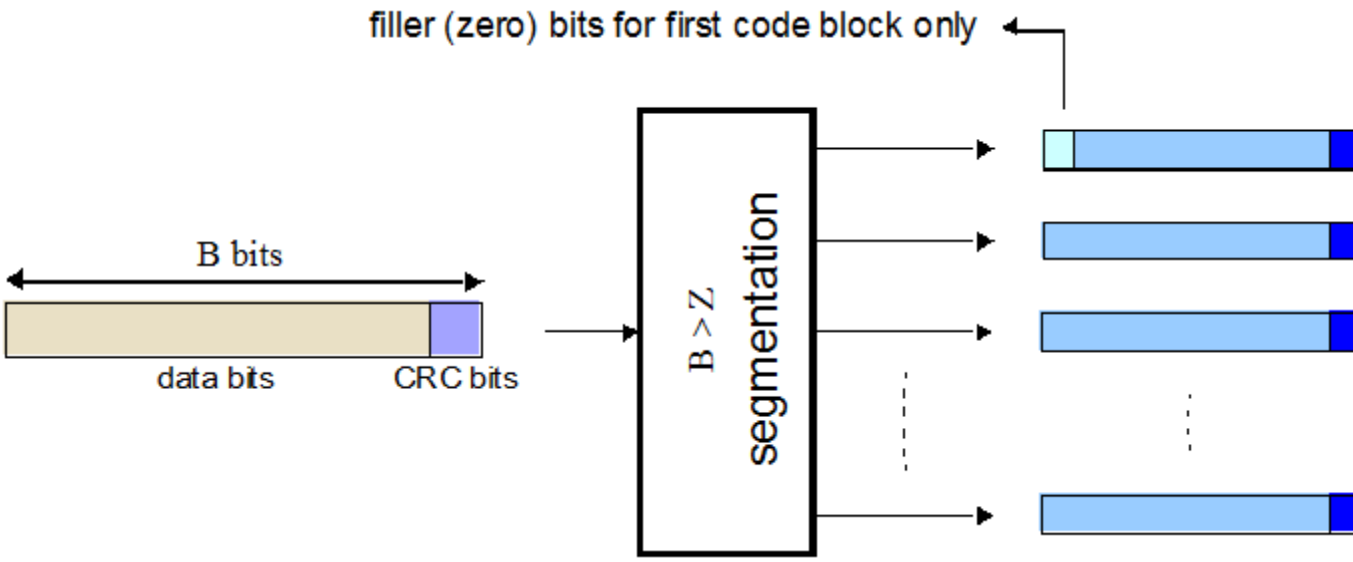
- Minimum code block size = 40 bits
- Maximum code block size, Z = 6144 bits

If the length of the input block, B, is greater than the maximum code block size, the input block is segmented.

When the input block is segmented, it is segmented into $C = \lceil B/(Z-L) \rceil$, where L is 24. Therefore, $C = \lceil B/6120 \rceil$ code blocks.

Each code block has a 24-bit CRC attached to the end, calculated as described in "Transport Block CRC Attachment" on page 1-131, but the generator polynomial, described as g_{CRC24B} in section 5.1.1 of "TS 36.212" on page 4-6 is used.

If required, filler bits are appended to the start of the segment so that the code block sizes match a set of valid turbo interleaver block sizes, as shown in the following figure.

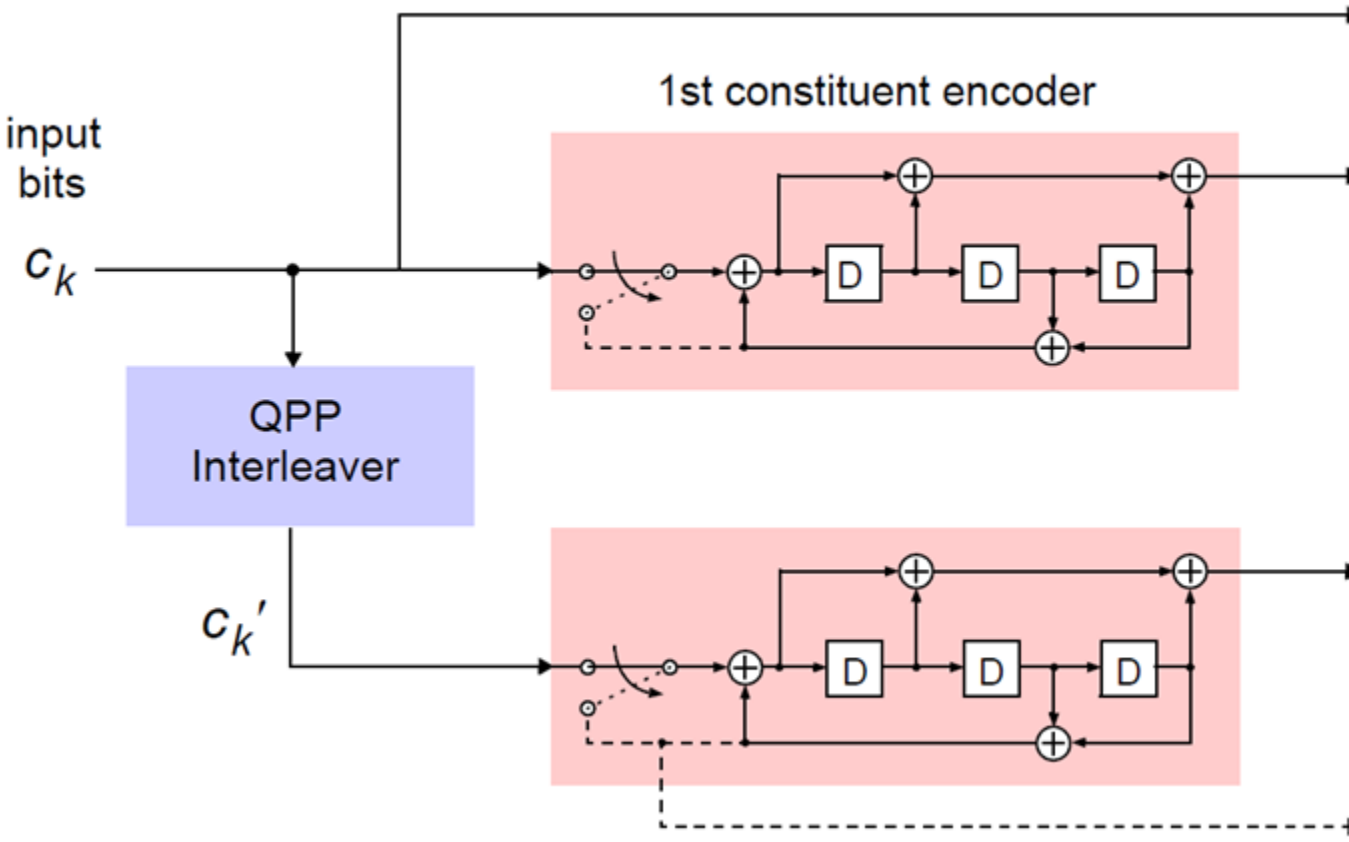


If no segmentation is needed, only one code block is produced. If B is less than the minimum size, filler bits (zeros) are added to the beginning of the code block to achieve a total of 40 bits.

Channel Coding

- "Constituent Encoders" on page 1-134
- "Trellis Termination for Turbo Encoder" on page 1-135
- "The QPP Interleaver" on page 1-136

The code blocks undergo turbo coding. Turbo coding is a form of forward error correction that improves the channel capacity by adding redundant information. The turbo encoder scheme used is a Parallel Concatenated Convolutional Code (PCCC) with two recursive convolutional coders and a "contention-free" Quadratic Permutation Polynomial (QPP) interleaver, as shown in the following figure.



The output of the encoder is three streams, $d_k^{(0)}$, $d_k^{(1)}$, and $d_k^{(2)}$, to achieve a code rate of 1/3.

Constituent Encoders. The input to the first constituent encoder is the input bit stream to the turbo coding block. The input to the second constituent encoder is the output of the QPP interleaver, a permutated version of the input sequence.

There are two output sequences from each encoder, a systematic (x_k, x_k') and a parity (z_k, z_k'). Only one of the systematic sequences (x_k) is used as an

output because the other (x'_k) is simply a permutated version of the chosen systematic sequence. The transfer function for each constituent encoder is given by the following equation.

$$G(D) = \left[1, \frac{g_1(D)}{g_0(D)}\right]$$

The first element, 1, represents the systematic output transfer function.

The second element, $\left(\frac{g_1(D)}{g_0(D)}\right)$, represents the recursive convolutional output transfer function.

$$g_0(D) = 1 + D^2 + D^3$$

$$g_1(D) = 1 + D + D^3$$

The output for each sequence can be calculated using the transfer function.

The encoder is initialized with all zeros. If the code block to be encoded is the 0-th and filler bits (F) are used, the input to the encoder (c_k) is set to zero and the output (x_k) and (z_k) set to <NULL> for $k = 0,\ldots,F-1$.

Trellis Termination for Turbo Encoder. A standard convolutional coder initializes its internal registers to an "all zeros" state and ensures the coder finishes in an "all zeros" state by padding the end of the input sequence with k zeros. Since the decoder knows the start and end state of the encoder, it can decode the data. Driving a recursive coder to an all zeros state using this method is not possible. To overcome this problem, trellis termination is used.

A tail-biting convolutional coder initializes its internal shift registers to the last k bits of the current input block rather than an "all zeros" state. Therefore, the start and end states are the same without the need to append zeros to the input block. The result is the elimination of the necessary overhead to terminate the coder by padding the input with zeros. The drawback to this method is that the decoder becomes more complicated, since the initial state is unknown. However, it is known that the start and end sequence are the same.

The QPP Interleaver. The role of the interleaver is to spread the information bits such that in the event of a burst error, the two code streams are affected differently, allowing data to still be recovered.

The output of the interleaver is a permutated version of the input data, as shown in the following equations.

$$c'_i = c_{\Pi(i)}, i = 0,1,\ldots,(K-1)$$

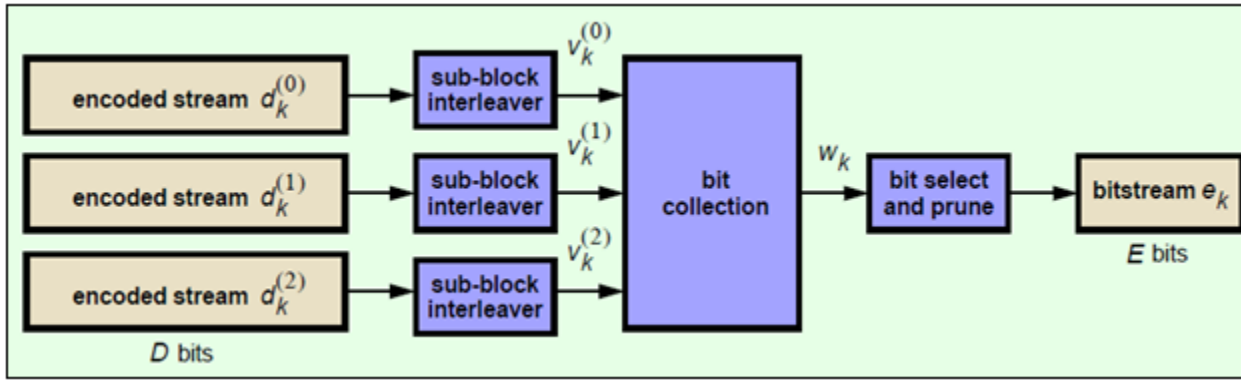
$$\Pi(i) = \left(f_1 i + f_2 i^2\right) \bmod K$$

The variable K is the input length. The variables f_1 and f_2 are coefficients chosen depending on K, in table 5.1.3-3 of "TS 36.212" on page 4-6. For example, K=40, f_1=3, and f_2=10, yields the following sequence.

$$\Pi(i) = 0,13,6,19,12,25,18,31,24,37,30,3,36,9,2,15,8,21,14,27,20,33,26,39,32,5,38,\ldots$$

Rate Matching

The rate matching block creates an output bitstream with a desired code rate. As the number of bits available for transmission depends on the available resources the rate matching algorithm is capable of producing any arbitrary rate. The three bitstreams from the turbo encoder are interleaved followed by bit collection to create a circular buffer. Bits are selected and pruned from the buffer to create an output bitstream with the desired code rate. The process is illustrated in the following figure.



Sub-block Interleaver. The three sub-block interleavers used in the rate matching block are identical. Interleaving is a technique to reduce the impact of burst errors on a signal as consecutive bits of data will not be corrupted.

The sub-block interleaver reshapes the encode bit sequence, row-by-row, to form a matrix with $C_{Subblock}^{TC} = 32$ columns and $R_{Subblock}^{TC}$ rows. The variable $R_{Subblock}^{TC}$ is determined by finding the minimum integer such that the number of encoded input bits is $D \le \left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right)$. If $\left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right) > D$, N_D <NULL>'s are appended onto the front of the encoded sequence. In this case, $N_D + D = \left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right)$.

For blocks $d_k^{(0)}$ and $d_k^{(1)}$, inter-column permutation is performed on the matrix to reorder the columns as shown in the following pattern.

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

The output of the block interleaver for blocks $d_k^{(0)}$ and $d_k^{(1)}$ is the bit sequence read out column-by-column from the inter-column permutated matrix to create a stream $K_\pi = \left(R_{Subblock}^{TC} \times C_{Subblock}^{TC}\right)$ bits long.

For block $d_k^{(2)}$, the elements within the matrix are permutated separately based on the permutation pattern shown above, but modified to create a permutation which is a function of the variables $R_{Subblock}^{TC}$, $C_{Subblock}^{TC}$, k, and K_π. This process creates three interleaved bitstreams.

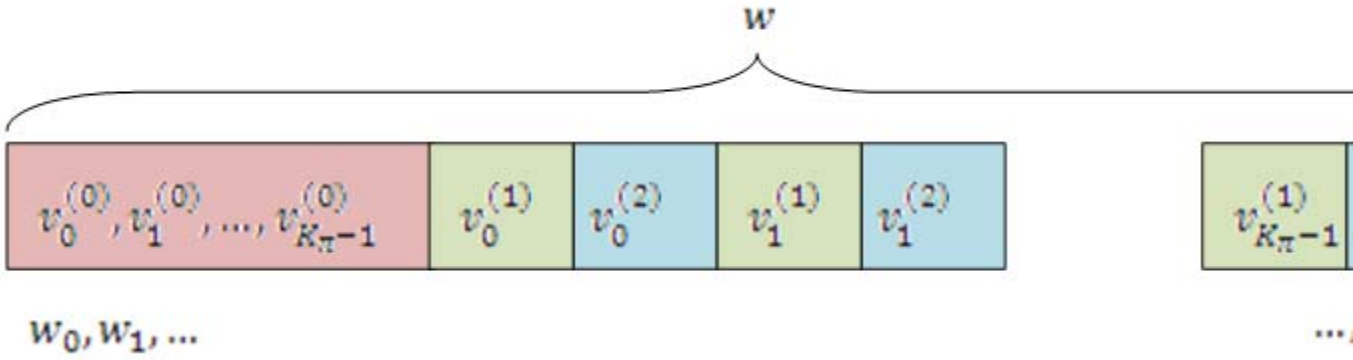
$$d_k^{(0)} \rightarrow v_k^{(0)}$$

$$d_k^{(1)} \rightarrow v_k^{(1)}$$

$$d_k^{(2)} \rightarrow v_k^{(2)}$$

Bit Collection, Selection, and Transmission. The bit collection stage creates a virtual circular buffer by combining the three interleaved encoded bit streams.

$v_k^{(1)}$ and $v_k^{(2)}$ are combined by interlacing successive values of $v_k^{(1)}$ and $v_k^{(2)}$. This combination is then appended onto the end of $v_k^{(0)}$ to create the circular buffer w_k shown in the following figure.



Interlacing allows equal levels of protection for each parity sequence.

Bits are then selected and pruned from the circular buffer to create an output sequence length which meets the desired code rate.

The Hybrid Automatic Repeat Request (HARQ) error correction scheme is incorporated into the rate-matching algorithm of LTE. For any desired code rate the coded bits are output serially from the circular buffer from a starting location, given by the redundancy version (RV), wrapping around to the beginning of the buffer if the end of the buffer is reached. `NULL` bits are discarded. Different RVs and hence starting points allow for the retransmission of selected data. Being able to select different starting points enables the following two main methods of recombining data at the receiver in the HARQ process.

- Chase combining — retransmissions contain the same data and parity bit.
- Incremental redundancy — retransmissions contain different information so the receiver gains knowledge upon each retransmission.

Code Block Concatenation

In this stage, the rate matched code blocks are concatenated back together. This task is done by sequentially concatenating the rate-matched blocks together to create the output of the channel coding, f_k for $k = 0,\ldots,G-1$.

Channel Coding of Control Information with UL-SCH Data

- "HARQ-ACK Information" on page 1-140
- "Rank Indicator" on page 1-141
- "Channel Quality Information and Precoder Matrix Indicator" on page 1-142

Control information arrives at the coder in the form of channel quality information (CQI), precoder matrix indication (PMI), rank indication (RI), and HARQ-Indicator (HI). Different control information coding rates are achieved by allocating a different number of coded symbols for transmission. When control information is transmitted on the PUSCH, the channel coding for HI, RI, and CQI is done independently.

The transmission mode determines the bit widths assigned to the various types control information; the corresponding widths for the transmission modes can be found in section 5.2.2.6.1-4 of [1].

The following sections describe uplink control information on PUSCH with UL-SCH data.

HARQ-ACK Information. The number of coded symbols, Q, used by the UE to transmit the HARQ acknowledgement bits is determined using the number of HARQ bits (1 or 2 depending on the number of codewords present), the scheduled PUSCH bandwidth expressed as a number of subcarriers, the number of SC-FDMA symbols per subframe for the initial PUSCH transmission, and information obtained from the initial PDCCH for the same transport block.

Each positive acknowledgement (ACK) is encoded as a binary `1` and negative acknowledgement (NACK) is encoded as a binary `0`. If the HARQ-ACK

consists of 1-bit of information, $[o_0^{ACK}]$, corresponding to 1 codeword, then it is first encoded according to the following table.

Q_m	Encoded HARQ-ACK
2	$[o_0^{ACK}$ y]
4	$[o_0^{ACK}$ y x x]
6	$[o_0^{ACK}$ y x x x x]

In the preceding table, *x* and *y* are placeholders used to scramble the HARQ-ACK bits in such a way as to maximize the Euclidean distance of the modulation symbols carrying the HARQ information.

If the HARQ-ACK consists of 2 bits of information, $[o_0^{ACK}\ o_1^{ACK}]$, where o_0^{ACK} and o_1^{ACK} correspond to the first and second codeword, respectively, and $o_2^{ACK} = (o_0^{ACK} + o_1^{ACK}) \bmod 2$, then they are encoded according to the following table.

Q_m	Encoded HARQ-ACK
2	$[o_0^{ACK}\ o_1^{ACK}\ o_2^{ACK}\ o_0^{ACK}\ o_1^{ACK}\ o_2^{ACK}]$
4	$[o_0^{ACK}\ o_1^{ACK}$ x x $o_2^{ACK}\ o_0^{ACK}$ x x $o_1^{ACK}\ o_2^{ACK}$ x x]
6	$[o_0^{ACK}\ o_1^{ACK}$ x x x x $o_2^{ACK}\ o_0^{ACK}$ x x x x $o_1^{ACK}\ o_2^{ACK}$ x x x x]

Rank Indicator. The bit widths, *Q*, (1 or 2 information bits) for rank indication feedback for PDSCH transmissions are determined using the maximum number of layers according to the corresponding eNodeB antenna configuration and UE category. If RI consists of 1 information bit, $[o_0^{RI}]$, then it is first encoded according to the following table.

Q_m	Encoded HARQ-ACK
2	$[o_0^{RI}$ y]
4	$[o_0^{RI}$ y x x]
6	$[o_0^{RI}$ y x x x x]

In the preceding table, *x* and *y* are placeholders used to scramble the HARQ-ACK bits in such a way as to maximize the Euclidean distance of the modulation symbols carrying the HARQ information.

If RI consists of 2 information bits, $[o_0^{RI}\ o_1^{RI}]$, then they are first encoded according to the following table.

Q_m	Encoded HARQ-ACK
2	$[o_0^{RI}\ o_1^{RI}\ o_2^{RI}\ o_0^{RI}\ o_1^{RI}\ o_2^{RI}]$
4	$[o_0^{RI}\ o_1^{RI}$ x x $o_2^{RI}\ o_0^{RI}$ x x $o_1^{RI}\ o_2^{RI}$ x x]
6	$[o_0^{RI}\ o_1^{RI}$ x x x x $o_2^{RI}\ o_0^{RI}$ x x x x $o_1^{RI}\ o_2^{RI}$ x x x x]

Channel Quality Information and Precoder Matrix Indicator. The number of coded symbols, *Q*, used for channel quality information is determined from the number of CQI bits present, the number of CRC bits, the scheduled PUSCH bandwidth expressed as a number of subcarriers, and information obtained from the PDCCH for the same transport block.

If the payload size is greater than 11 bits, the CQI bit sequence undergoes CRC attachment, convolution channel coding, and rate matching. If the payload size is less than 11 bits, the channel coding of the CQI is performed using the following steps.

- The CQI bits are coded using a (32,*O*) block code. The codewords of the (32,*O*) block code use the following equation.

$$b_i = \sum_{n=0}^{O-1} \left(o_n \cdot M_{i,n} \right) \bmod 2$$

The codewords of the (32,*O*) block code are a linear combination of the 11 basis sequences denoted in the following table.

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$
0	1	1	0	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0	0	1	1
2	1	0	0	1	0	0	1	0	1	1	1
3	1	0	1	1	0	0	0	0	1	0	1
4	1	1	1	1	0	0	0	1	0	0	1
5	1	1	0	0	1	0	1	1	1	0	1
6	1	0	1	0	1	0	1	0	1	1	1
7	1	0	0	1	1	0	0	1	1	0	1
8	1	1	0	1	1	0	0	1	0	1	1
9	1	0	1	1	1	0	1	0	0	1	1
10	1	0	1	0	0	1	1	1	0	1	1
11	1	1	1	0	0	1	1	0	1	0	1
12	1	0	0	1	0	1	0	1	1	1	1
13	1	1	0	1	0	1	0	1	0	1	1
14	1	0	0	0	1	1	0	1	0	0	1
15	1	1	0	0	1	1	1	1	0	1	1
16	1	1	1	0	1	1	1	0	0	1	0
17	1	0	0	1	1	1	0	0	1	0	0
18	1	1	0	1	1	1	1	1	0	0	0
19	1	0	0	0	0	1	1	0	0	0	0

i	$M_{i,0}$	$M_{i,1}$	$M_{i,2}$	$M_{i,3}$	$M_{i,4}$	$M_{i,5}$	$M_{i,6}$	$M_{i,7}$	$M_{i,8}$	$M_{i,9}$	$M_{i,10}$
20	1	0	1	0	0	0	1	0	0	0	1
21	1	1	0	1	0	0	0	0	0	1	1
22	1	0	0	0	1	0	0	1	1	0	1
23	1	1	1	0	1	0	0	0	1	1	1
24	1	1	1	1	1	0	1	1	1	1	0
25	1	1	0	0	0	1	1	1	0	0	1
26	1	0	1	1	0	1	0	0	1	1	0
27	1	1	1	1	0	1	0	1	1	1	0
28	1	0	1	0	1	1	1	0	1	0	0
29	1	0	1	1	1	1	1	1	1	0	0
30	1	1	1	1	1	1	1	1	1	1	1
31	1	0	0	0	0	0	0	0	0	0	0

The output sequence is obtained by a circular repetition of the CQI/PMI block, as shown in the following equation.

$$q_i = b_{(i \bmod B)}$$

In the preceding equation, the variable B is 32.

Data and Control Multiplexing

The control and transport data multiplexing is performed such that HARQ-ACK information is present in both slots and is mapped to resources around the demodulation reference signals. The mapping is important, as it assumes the channel estimation around the DRS are of better quality. Thus, the integrity of the HARQ information is maintained.

Channel Interleaver

The channel interleaver implements a time-first mapping of modulation symbols onto the transmit waveform while ensuring that HARQ information is present on both slots and is mapped to resources around the DRS.

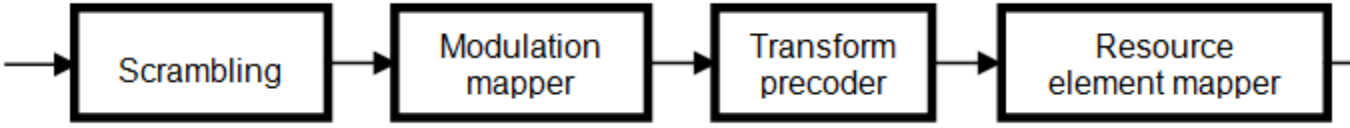
Channel Coding of Control Information without UL-SCH Data

When control data is sent on the PUSCH without UL-SCH data the following coding process can be identified: channel coding of control information, mapping and channel interleaving. The fundamental change is the number of bits used to transmit the control information. After determining the bit widths for the various pieces of coded control information, channel coding and rate matching is then performed as per section 1.6 of this document. The coded channel quality information is remapped into columns of symbols before being interleaved with the coded HI and RI.

The PUSCH

- "Scrambling" on page 1-145
- "Modulation" on page 1-146
- "Precoding" on page 1-146
- "Mapping to Resource Elements" on page 1-147

The Physical Uplink Shared Channel (PUSCH) carries uplink shared channel data and control information. The processing chain for the PUSCH includes scrambling, modulation mapping, precoding, resource element mapping and Single Carrier – Frequency Division Multiple Access (SC-FDMA) modulation. This processing chain is illustrated in the following figure.



Scrambling

The transport codeword is bit-wise multiplied with an orthogonal sequence and a UE-specific scrambling sequence to create the following sequence of symbols for each codeword, q.

$$\tilde{b}^{(q)}(0),\ldots,\tilde{b}^{(q)}\left(M_{bit}^{(q)}-1\right)$$

The variable $M_{bit}^{(q)}$ is the number of bits transmitted on the PUSCH in one subframe q.

The scrambling sequence is pseudorandom, created using a length-31 Gold sequence generator and initialized using the slot number within the radio network temporary identifier associated with the PUSCH transmission, n_{RNTI}, the cell ID, N_{ID}^{cell}, the slot number within the radio frame, n_s, and the codeword index, $q = \{0,1\}$, at the start of each subframe.

$$c_{init} = n_{RNTI} \times 2^{14} + q \times 2^{13} + \left\lfloor \frac{n_s}{2} \right\rfloor \times 2^9 + N_{ID}^{cell}$$

Scrambling with a cell-specific sequence serves the purpose of intercell interference rejection. When a base station descrambles a received bit stream with a known cell specific scrambling sequence, interference from other cells will be descrambled incorrectly and therefore only appear as uncorrelated noise.

Modulation

The scrambled codeword undergoes QPSK, 16QAM, or 64QAM modulation to generate complex valued symbols. This choice provides the flexibility to allow the scheme to maximize the data transmitted depending on the channel conditions.

Precoding

The PUSCH precoding is not the same as the in the downlink (multi-antenna) precoding. The block of complex valued symbols, $d(0),\ldots,d(M_{symb}-1)$, is divided into M_{symb}/M_{sc}^{PUSCH} sets. Each set, which has size M_{sc}^{PUSCH}, corresponds to one SC-FDMA symbol. A Discrete Fourier Transform is then applied to each set, essentially precoding part of the SC-FDMA modulation.

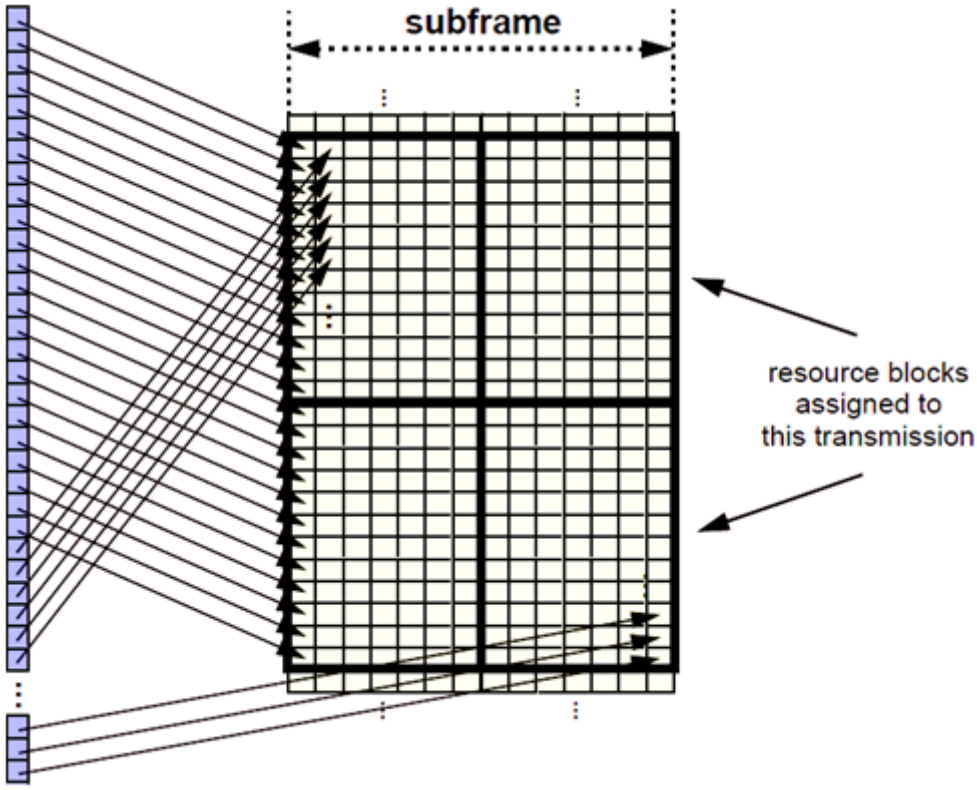
The size of the DFT, which is the value of M_{sc}^{PUSCH}, must have a prime that is a product of 2, 3, or 5, thereby fulfilling the following equation.

$$M_{sc}^{PUSCH} = N_{sc}^{RB} \times 2^{\alpha_2} \times 3^{\alpha_3} \times 5^{\alpha_5} \leq N_{sc}^{RB} N_{RB}^{UL}$$

In the preceding equation, a_2, a_3, and a_5 are a set of nonnegative integers.

Mapping to Resource Elements

The final stage in the PUSCH processing is to map the symbols to the allocated physical resource elements. The allocation sizes are limited to values whose prime factors are 2, 3 and 5; this limit is imposed by the precoding stage. The symbols are mapped in increasing order beginning with subcarriers, then SC-FDMA symbols. SC-FDMA symbols carrying DRS or SRS are avoided during the mapping process. An example of the order of mapping the output of the precoding stage to the allocated resource blocks is shown in the following figure.



Demodulation Reference Signals (DRS) on the PUSCH

- "DRS Generation" on page 1-149
- "DRS Resource Mapping" on page 1-153

Demodulation reference signals associated with the PUSCH are used by the base station to perform channel estimation and allow for coherent demodulation of the received signal.

These reference signals are time-multiplexed with data, whereas in the downlink there is both time and frequency multiplexing. This multiplexing

is performed to maintain the single-carrier nature of the SC-FDMA signal, which ensures that all data carriers are contiguous.

DRS Generation

- "Base Sequence" on page 1-149
- "DRS Grouping" on page 1-151

The demodulation reference signals are generated using a base sequence denoted by $r_{u,v}(n)$, which is discussed further in "Base Sequence" on page 1-149. More specifically, r^{PUSCH} is used to denote the PUSCH DRS sequence and is defined by the following equation.

$$r^{PUSCH}\left(mM_{SC}^{RS}+n\right)=r_{u,v}^{(\alpha)}(n)$$

It is desired that the DRS sequences have small power variations in time and frequency, resulting in high power amplifier efficiency and comparable channel estimation quality for all frequency components. Zadoff-Chu sequences are good candidates, since they exhibit constant power in time and frequency. However, there are a limited number of Zadoff-Chu sequences; therefore, they are not suitable on their own.

The generation and mapping of the DRS associated with the PUSCH are discussed further in the following sections.

Base Sequence. The demodulation reference signals are defined by a cyclic shift, α, of a base sequence, r.

The base sequence, r, is represented in the following equation.

$$r_{u,v}^{(\alpha)}=e^{j\alpha n}r_{u,v}(n)$$

The preceding equation contains the following variables.

- $n=0,\ldots,M_{SC}^{RS}$, where M_{SC}^{RS} is the length of the reference signal sequence.

- $U = 0,\ldots,29$ is the base sequence group number.
- $V = 0,1$ is the sequence number within the group and only applies to reference signals of length greater than 6 resource blocks.

A phase rotation in the frequency domain (pre-IFFT in the OFDM modulation) is equivalent to a cyclic shift in the time domain (post IFFT in the OFDM modulation). For frequency non-selective channels over the 12 subcarriers of a resource block, it is possible to achieve orthogonality between DRS generated from the same base sequence if $\alpha = m\pi/6$ for $m = 0,1,\ldots,11$, and assuming the DRS are synchronized in time.

The orthogonality can be exploited to transmit DRS at the same time, using the same frequency resources without mutual interference. Generally, DRS generated from different base sequences will not be orthogonal; however they will present low cross-correlation properties.

To maximize the number of available Zadoff-Chu sequences, a prime length sequence is needed. The minimum sequence length in the UL is 12, the number of subcarriers in a resource block, which is not prime.

Therefore, Zadoff-Chu sequences are not suitable by themselves. There are effectively the following two types of base reference sequences.

- those with a sequence length ≥ 36 (spanning 3 or more resource blocks), which use a cyclic extension of Zadoff-Chu sequences
- those with a sequence length ≤ 36 (spanning 2 resource blocks), which use a special QPSK sequence

Base sequences of length • three resource blocks

For sequences of length 3 resource blocks and larger (i.e. $M_{sc}^{RS} \geq 3N_{sc}^{RS}$), the base sequence is a repetition, with a cyclic offset of a Zadoff-Chu sequence of length N_{zc}^{RS}, where N_{zc}^{RS} is the largest prime such that $N_{zc}^{RS} < M_{sc}^{RS}$. Therefore, the base sequence will contain one complete length N_{zc}^{RS} Zadoff-Chu sequence plus a fractional repetition appended on the

end. At the receiver the appropriate de-repetition can be done and the zero autocorrelation property will hold across the length N_{zc}^{RS} vector.

Base sequences of length • three resource blocks

For sequences shorter than three resource blocks (i.e. $M_{sc}^{RS} = 12,24$), the sequences are a composition of unity modulus complex numbers drawn from a simulation generated table. These sequences have been found through computer simulation and are specified in the LTE specifications.

DRS Grouping. There are a total of 30 sequence groups, $u \in \{0,1,\ldots,29\}$, each containing one sequence for length less than or equal to 60. This corresponds to transmission bandwidths of 1,2,3,4 and 5 resource blocks. Additionally, there are two sequences (one for v = 0 or 1) for length ≥ 72; corresponding to transmission bandwidths of 6 resource blocks or more.

Note that not all values of m are allowed, where m is the number of resource blocks used for transmission. Only values for m that are the product of powers of 2, 3 and 5 are valid, as shown in the following equation.

$m = 2^{\alpha_0} \times 3^{\alpha_1} \times 5^{\alpha_2}$, where α_i are positive integers

The reason for this restriction is that the DFT sizes of the SC-FDMA precoding operation are limited to values which are the product of powers of 2, 3 and 5. The DFT operation can span more than one resource block, and since each resource block has 12 subcarriers, the total number of subcarriers fed to the DFT will be $12m$. Since the result of $12m$ has to be the product of powers of 2, 3 and 5 this implies that the number of resource blocks must themselves be the product of powers of 2, 3 and 5. Therefore values of m such as 7, 11, 14, 19, etc. are not valid.

For a given time slot, the uplink reference signal sequences to use within a cell are taken from one specific sequence group. If the same group is to be used for all slots then this is known as fixed assignment. On the other hand, if the group number u varies for all slots within a cell this is known as *group hopping*.

Fixed Group Assignment

When fixed group assignment is used, the same group number is used for all slots. For PUSCH, the group number is a function of the cell identity number modulo 30 and is signaled by higher layers through the parameter Δ_{ss}, as shown in the following equation.

$$u = \left(\left(N_{ID}^{cell} \bmod 30\right) + \Delta_{ss}\right) \bmod 30, \text{ with } \Delta_{ss} \in \{0,1,\ldots,29\}$$

This method allows the same u to be used in different cells.

Group Hopping

If group hopping is used, the pattern is applied to the calculation of the sequence group number. This pattern is the same for both the PUCCH and PUSCH.

This pattern is defined as the following equation.

$$f_{gh}(n_s) = \sum_{i=0}^{7} c(8n_s + i) \cdot 2^i \bmod 30$$

As shown in the preceding equation, this group hopping pattern is a function of the slot number n_s and is calculated making use of a pseudorandom binary sequence $c(k)$, generated using a length-30 *Gold* code. To generate the group hopping pattern, the PRBS generator is initialized with the following value at the start of each radio frame.

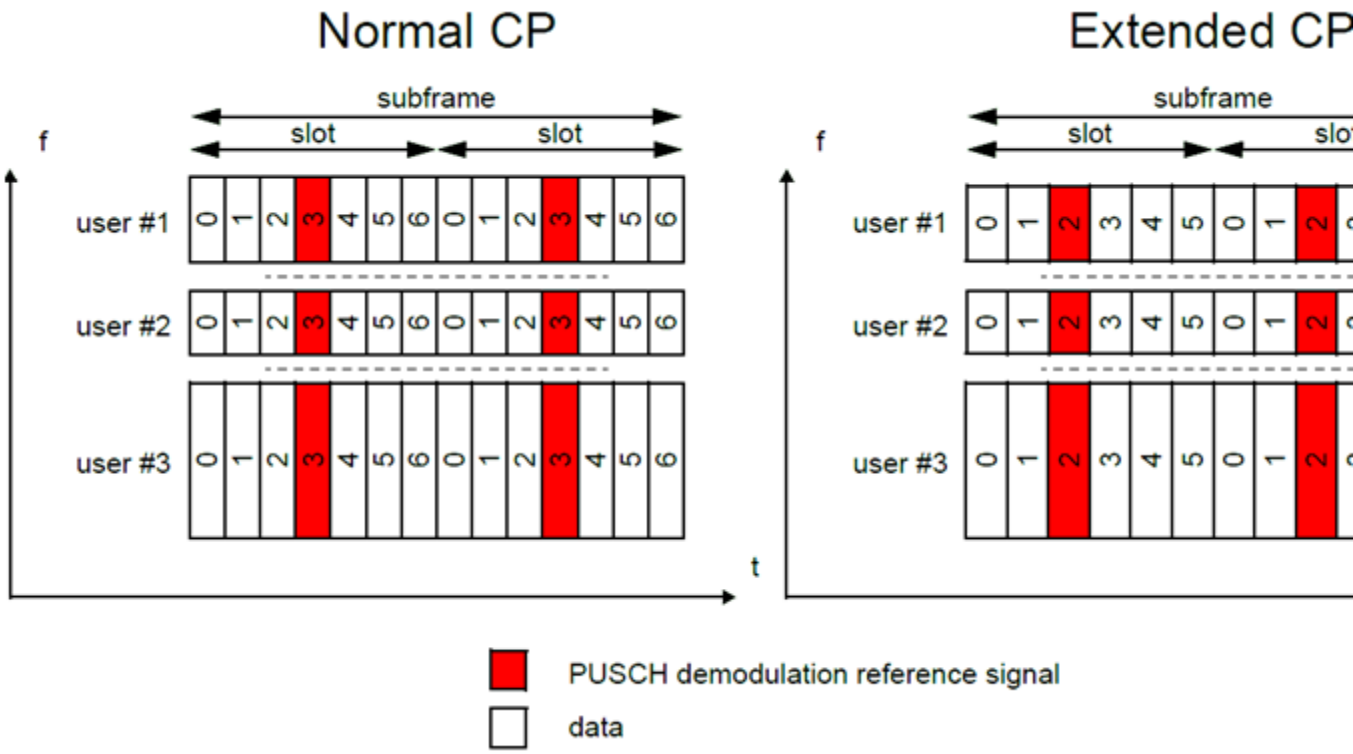
$$c_{init} = \left\lfloor \frac{N_{ID}^{cell}}{30} \right\rfloor$$

For PUSCH with group hopping, the group number, u, is given by the following equation.

$$u = \left(f_{gh}(n_s) + \left(\left(N_{ID}^{cell} \bmod 30\right) + \Delta_{ss}\right)\right) \bmod 30$$

DRS Resource Mapping

The PUSCH demodulation reference signal is mapped to the 4th SC-FDMA symbol of the slot during normal cyclic prefix and to every 3rd SC-FDMA slot during extended cyclic prefix. This resource mapping is shown in the following figure.



References

[1] 3GPP TS 36.212. "Multiplexing and channel coding." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`lteULSCH` | `lteULSCHInfo` | `ltePUSCH` | `ltePUSCHIndices` | `ltePUSCHDRS` | `ltePUSCHDRSIndices` | `lteULResourceGrid`

Related Examples

- “Model UL-SCH and PUSCH” on page 3-25

Propagation Channel Models

In this section...
"Multipath Fading Propagation Conditions" on page 1-155
"High Speed Train Condition" on page 1-157
"Moving Propagation Condition" on page 1-164
"MIMO Channel Correlation Matrices" on page 1-166

The LTE System Toolbox product provides a set of channel models for the test and verification of UE and eNodeB radio transmission and reception as defined in [1] and [2]. The following channel models are available in the LTE System Toolbox product.

- Multipath fading propagation conditions
- High speed train conditions
- Moving propagation conditions

Multipath Fading Propagation Conditions

The multipath fading channel model specifies the following three delay profiles.

- Extended Pedestrian A model (EPA)
- Extended Vehicular A model (EVA)
- Extended Typical Urban model (ETU)

These three delay profiles represent a low, medium, and high delay spread environment, respectively. The multipath delay profiles for these channels are shown in the following tables.

EPA Delay Profile

Excess tap delay (ns)	Relative power (dB)
0	0.0
30	–1.0

EPA Delay Profile (Continued)

Excess tap delay (ns)	Relative power (dB)
70	–2.0
90	–3.0
110	–8.0
190	–17.2
410	–20.8

EVA Delay Profile

Excess tap delay (ns)	Relative power (dB)
0	0.0
30	–1.5
150	–1.4
310	–3.6
370	–0.6
710	–9.1
1090	–7.0
1730	–12.0
2510	–16.9

ETU Delay Profile

Excess tap delay (ns)	Relative power (dB)
0	–1.0
50	–1.0
120	–1.0
200	0.0
230	0.0

ETU Delay Profile (Continued)

Excess tap delay (ns)	Relative power (dB)
500	0.0
1600	–3.0
2300	–5.0
5000	–7.0

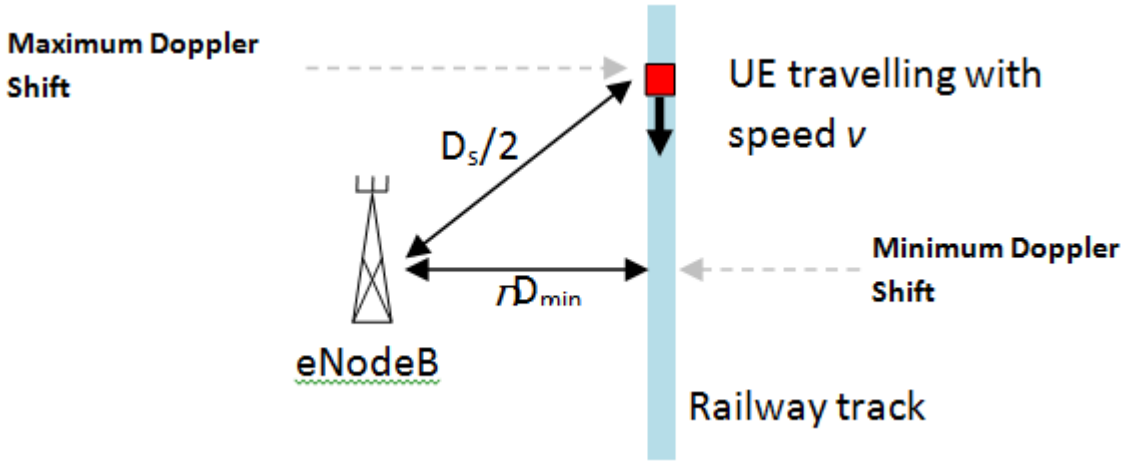
All the taps in the preceding tables have a classical *Doppler* spectrum. In addition to multipath delay profile, a maximum Doppler frequency is specified for each multipath fading propagation condition, as shown in the following table.

Channel model	Maximum Doppler frequency
EPA 5Hz	5 Hz
EVA 5Hz	5 Hz
EVA 70Hz	70 Hz
ETU 70Hz	70 Hz
ETU 300Hz	300 Hz

In the case of MIMO environments, a set of correlation matrices is introduced to model the correlation between UE and eNodeB antennas. These correlation matrices are introduced in “MIMO Channel Correlation Matrices” on page 1-166.

High Speed Train Condition

The high speed train condition defines a non-fading propagation channel with single multipath component, the position of which is fixed in time. This single multipath represents the Doppler shift, which is caused due to a high speed train moving past a base station, as shown in the following figure.



The expression $D_s / 2$ is the initial distance of the train from eNodeB, and $D_{\min}$ is the minimum distance between eNodeB and the railway track. Both variables are measured in meters. The variable *v* is the velocity of the train in meters per second. The Doppler shift due to a moving train is given in the following equation.

$$f_s(t) = f_d \cos\theta(t)$$

The variable $f_s(t)$ is the Doppler shift and f_d is the maximum Doppler frequency. The cosine of angle $\theta(t)$ is given by the following equation.

$$\cos\theta(t) = \frac{D_s/2 - vt}{\sqrt{D_{\min}{}^2 + (D_s/2 - vt)^2}}, 0 \le t \le D_s/v$$

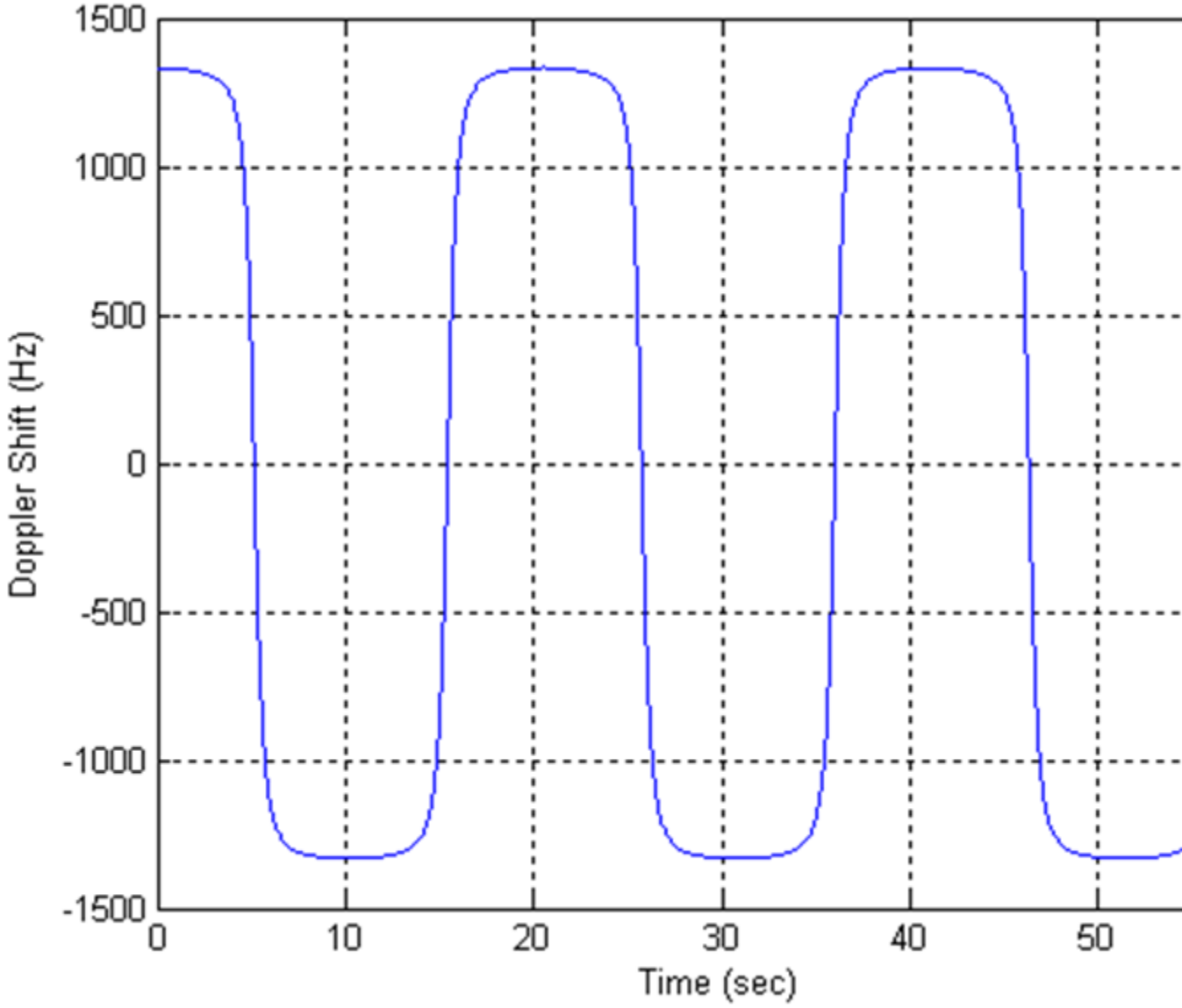
$$\cos\theta(t) = \frac{-1.5D_s + vt}{\sqrt{D_{\min}{}^2 + (-1.5D_s + vt)^2}}, D_s/v < t \le 2D_s/v$$

$$\cos\theta(t) = \cos\theta(t \mod (2D_s/v)), t > 2D_s/v$$

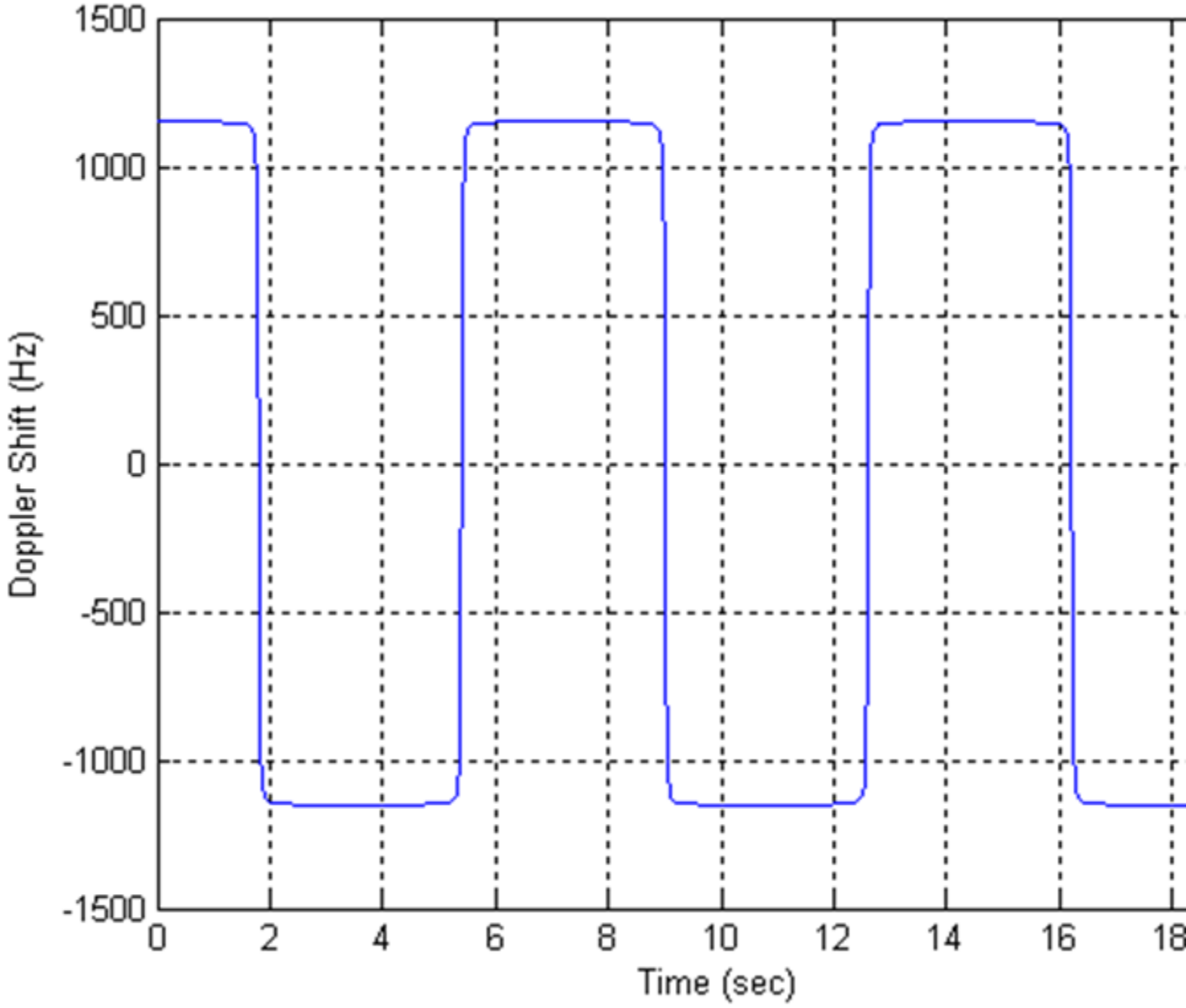
For eNodeB testing, two high speed train scenarios are defined that use the parameters listed in the following table. The Doppler shift, $f_s(t)$, is calculated using the preceding equations and the parameters listed in the following table.

Parameter	**Value**	
	Scenario 1	Scenario 3
D_s	1,000 m	300 m
$D_{\min}$	50 m	2 m
v	350 km/hr	300 km/kr
f_d	1,340 Hz	1,150 Hz

Both of these scenarios result in Doppler shifts that apply to all frequency bands. The Doppler shift trajectory for scenario 1 is shown in the following figure.



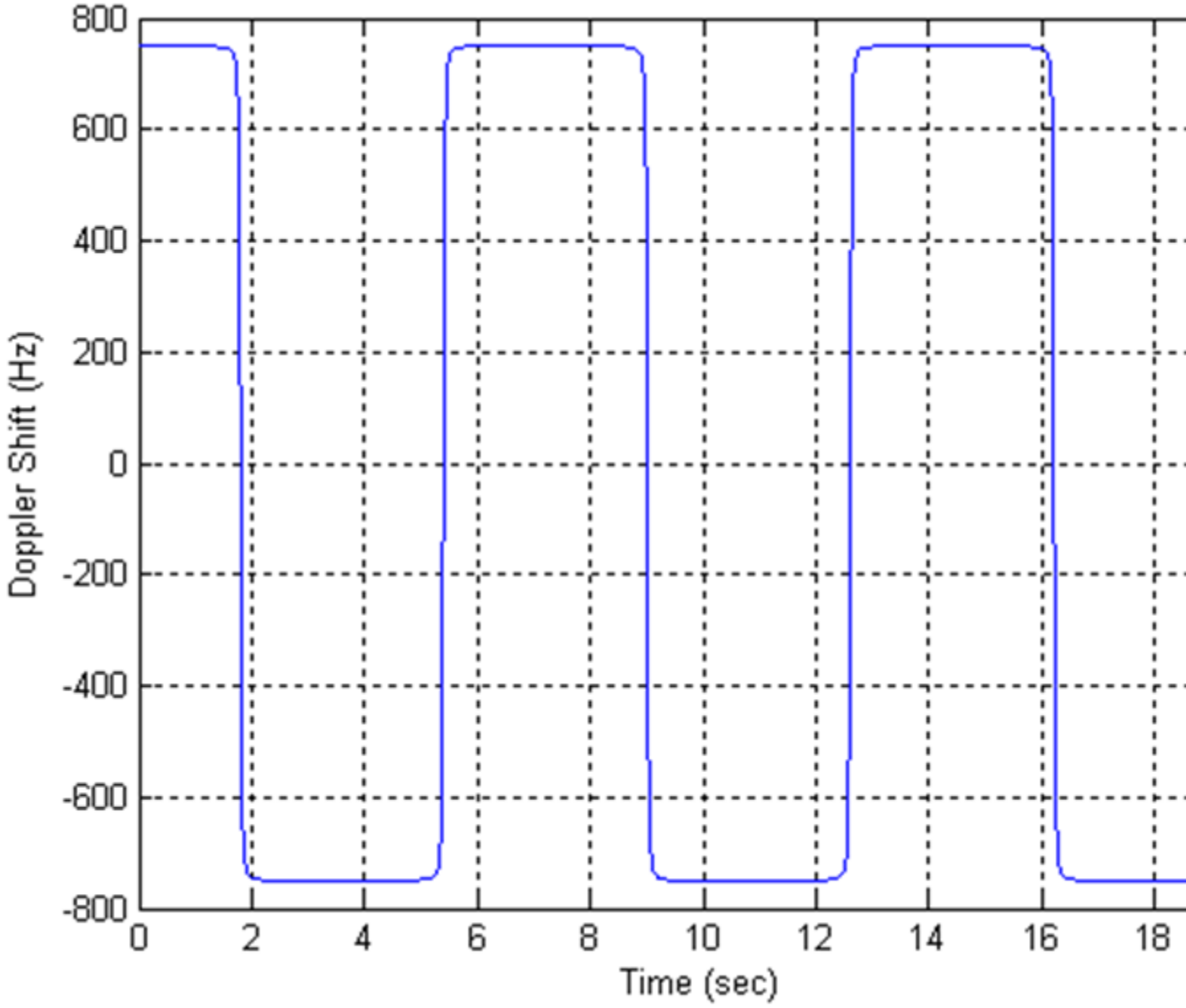
The Doppler shift trajectory for scenario 3 is shown in the following figure.



For UE testing, the Doppler shift, $f_s(t)$, is calculated using the preceding equations and the parameters listed in the following table.

Parameter	Value
D_s	300 m
$D_{\min}$	2 m
v	300 km/hr
f_d	750 Hz

These parameters result in the Doppler shift, applied to all frequency bands, shown in the following figure.


800
600
400
200
0
-200
-400
-600
-800
Doppler Shift (Hz)
0
2
4
6
8
10
12
14
16
18
Time (sec)

Moving Propagation Condition

The moving propagation channel in LTE defines a channel condition where the location of multipath components changes. The time difference between the reference time and the first tap, $\Delta\tau$, is given by the following equation.

$$\Delta\tau = \frac{A}{2} \cdot \sin(\Delta\omega \cdot t)$$

The variable A represents the starting time in seconds and $\Delta\omega$ represents angular rotation in radians per second.

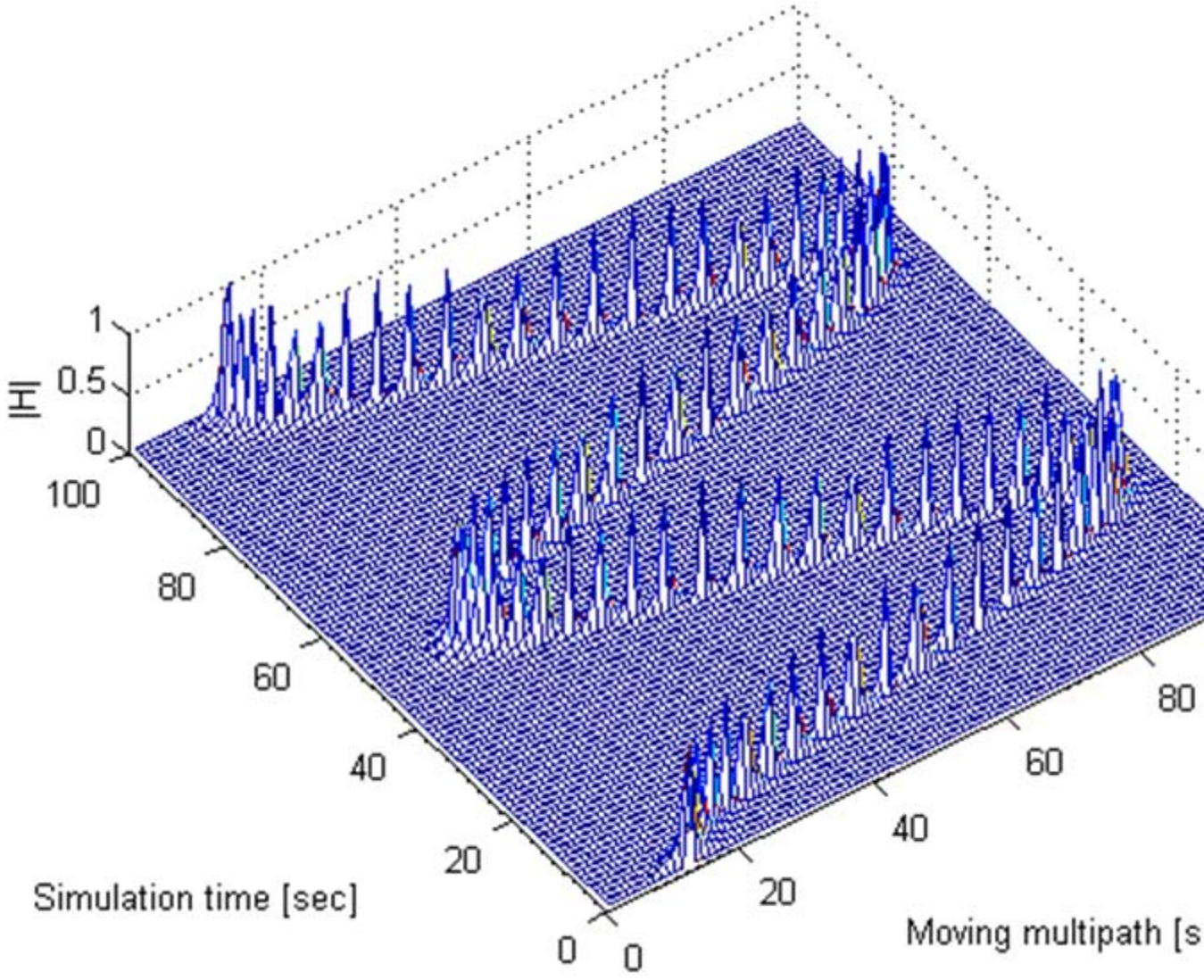
Note Relative time between multipath components stays fixed.

The parameters for the moving propagation conditions are shown in the following table.

Parameter	Scenario 1	Scenario 2
Channel model	ETU200	AWGN
UE speed	120 km/hr	350 km/hr
CP length	Normal	Normal
A	10 μs	10 μs
$\Delta\omega$	0.04 s^{-1}	0.13 s^{-1}

Doppler shift only applies for generating fading samples for scenario 1. In scenario 2, a single non-fading multipath component with additive white gaussian noise (AWGN) is modeled. The location of this multipath component changes with time, according to the preceding equation.

An example of a moving channel with a single non-fading tap is shown in the following figure.



The LTE-specific parameters are scaled up in this plot.

MIMO Channel Correlation Matrices

In MIMO systems, there is correlation between transmit and receive antennas. This depends on a number of factors such as the separation between antenna and the carrier frequency. For maximum capacity, it is desirable to minimize the correlation between transmit and receive antennas.

There are different ways to model antenna correlation. One technique makes use of correlation matrices to describe the correlation between multiple antennas both at the transmitter and the receiver. These matrices are computed independently at both the transmitter-receiver and then combined by means of a Kronecker product in order to generate a channel spatial correlation matrix.

Three different correlation levels are defined in [1].

1 low or no correlation

2 medium correlation

3 high correlation

The parameters α and β are defined for each level of correlation as shown in the following table of correlation values.

Low correlation		**Medium correlation**		**High correlation**	
α	β	α	β	α	β
0	0	0.3	0.9	0.9	0.9

The independent correlation matrices at eNodeB and UE, R_{eNB} and R_{UE}, respectively, are shown for differents set of antennas (1, 2 and 4) in the following table.

Correlation	One antenna	Two antennas	Four antennas
eNodeB	$R_{eNB} = 1$	$R_{eNB} = \begin{pmatrix} 1 & \alpha \\ \alpha^* & 1 \end{pmatrix}$	$R_{eNB} = \begin{pmatrix} 1 & \alpha^{1/9} & \alpha^{4/9} & \alpha \\ \alpha^{1/9*} & 1 & \alpha^{1/9} & \alpha^{4/9} \\ \alpha^{4/9*} & \alpha^{1/9*} & 1 & \alpha^{1/9} \\ \alpha^* & \alpha^{4/9*} & \alpha^{1/9*} & 1 \end{pmatrix}$
UE	$R_{UE} = 1$	$R_{UE} = \begin{pmatrix} 1 & \beta \\ \beta^* & 1 \end{pmatrix}$	$R_{UE} = \begin{pmatrix} 1 & \beta^{1/9} & \beta^{4/9} & \beta \\ \beta^{1/9*} & 1 & \beta^{1/9} & \beta^{4/9} \\ \beta^{4/9*} & \beta^{1/9*} & 1 & \beta^{1/9} \\ \beta^* & \beta^{4/9*} & \beta^{1/9*} & 1 \end{pmatrix}$

The channel spatial correlation matrix, R_{spat}, is given by the following equation.

$$R_{spat} = R_{eNB} \otimes R_{UE}$$

The symbol ⊗ represents the *Kronecker* product. The values of the channel spatial correlation matrix, R_{spat}, for different matrix sizes are defined in the following table.

Matrix size	R_{spat} values
1×2 case	$R_{spat} = R_{UE} = \begin{pmatrix} 1 & \beta \\ \beta^* & 1 \end{pmatrix}$
2×2 case	$R_{spat} = R_{eNB} \otimes R_{UE} = \begin{pmatrix} 1 & \alpha \\ \alpha^* & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & \beta \\ \beta^* & 1 \end{pmatrix} = \begin{pmatrix} 1 & \beta & \alpha & \alpha\beta \\ \beta^* & 1 & \alpha\beta^* & \alpha \\ \alpha^* & \alpha^*\beta & 1 & \beta \\ \alpha^*\beta^* & \alpha^* & \beta^* & 1 \end{pmatrix}$

Matrix size	R_{spat} values
4×2 case	$R_{spat} = R_{eNB} \otimes R_{UE} = \begin{pmatrix} 1 & \alpha^{1/9} & \alpha^{4/9} & \alpha \\ \alpha^{1/9*} & 1 & \alpha^{1/9} & \alpha^{4/9} \\ \alpha^{4/9*} & \alpha^{1/9*} & 1 & \alpha^{1/9} \\ \alpha^{*} & \alpha^{4/9*} & \alpha^{1/9*} & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & \beta \\ \beta^{*} & 1 \end{pmatrix}$
4×4 case	$R_{spat} = R_{eNB} \otimes R_{UE} = \begin{pmatrix} 1 & \alpha^{1/9} & \alpha^{4/9} & \alpha \\ \alpha^{1/9*} & 1 & \alpha^{1/9} & \alpha^{4/9} \\ \alpha^{4/9*} & \alpha^{1/9*} & 1 & \alpha^{1/9} \\ \alpha^{*} & \alpha^{4/9*} & \alpha^{1/9*} & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & \beta^{1/9} & \beta^{4/9} & \beta \\ \beta^{1/9*} & 1 & \beta^{1/9} & \beta^{4/9} \\ \beta^{4/9*} & \beta^{1/9*} & 1 & \beta^{1/9} \\ \beta^{*} & \beta^{4/9*} & \beta^{1/9*} & 1 \end{pmatrix}$

References

[1] 3GPP TS 36.101. "User Equipment (UE) Radio Transmission and Reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

[2] 3GPP TS 36.104. "Base Station (BS) radio transmission and reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`lteFadingChannel` | `lteHSTChannel` | `lteMovingChannel`

Related Examples

- "Simulate Propagation Channels" on page 3-27

Channel Estimation

In this section...

"Channel Estimation Overview" on page 1-169

"Get Pilot Estimates Subsystem" on page 1-173

"Pilot Average Subsystem" on page 1-173

"Create Virtual Pilots Subsystem" on page 1-178

"Interpolation Subsystem" on page 1-181

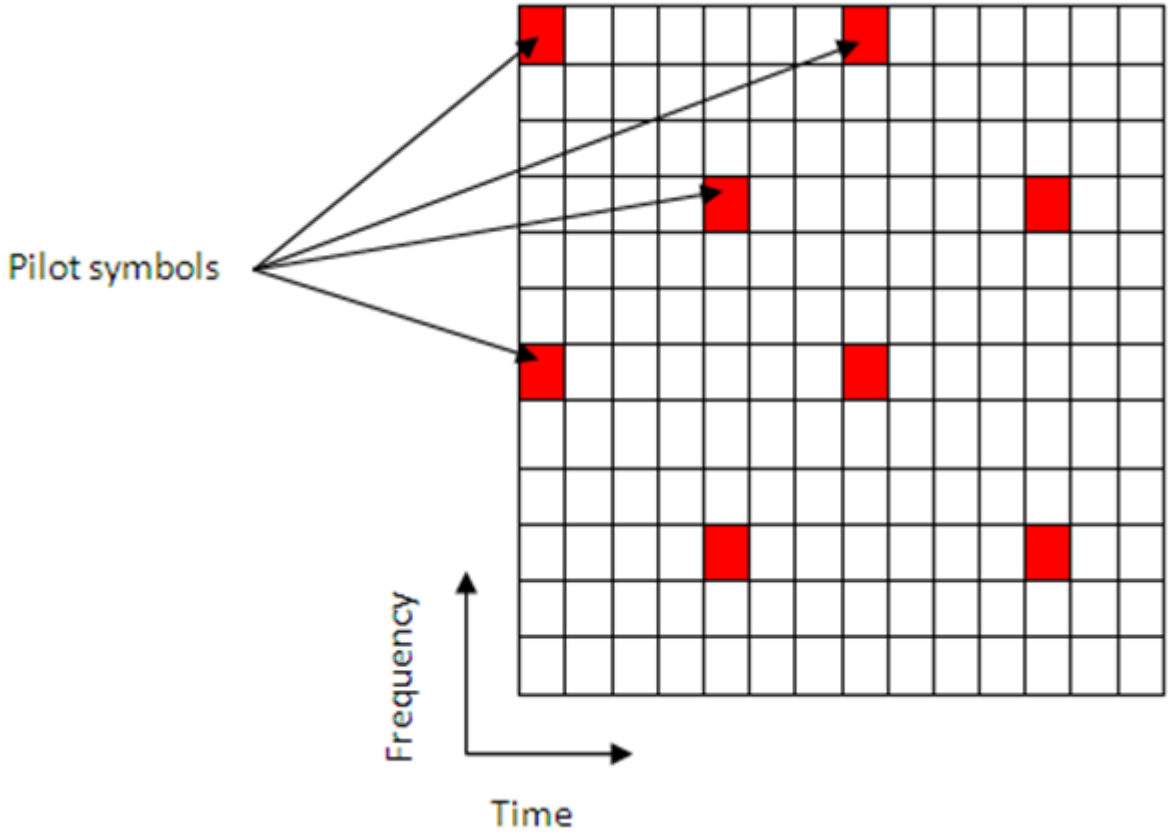
"Noise Estimation" on page 1-181

The LTE System Toolbox product uses orthogonal frequency division multiplexing (OFDM) as its digital multicarrier modulation scheme. Channel estimation plays an important part in an OFDM system. It is used for increasing the capacity of orthogonal frequency division multiple access (OFDMA) systems by improving the system performance in terms of bit error rate.

To facilitate the estimation of the channel characteristics, LTE uses cell-specific reference signals (pilot symbols) inserted in both time and frequency. These pilot symbols provide an estimate of the channel at given locations within a subframe. Through interpolation, it is possible to estimate the channel across an arbitrary number of subframes.

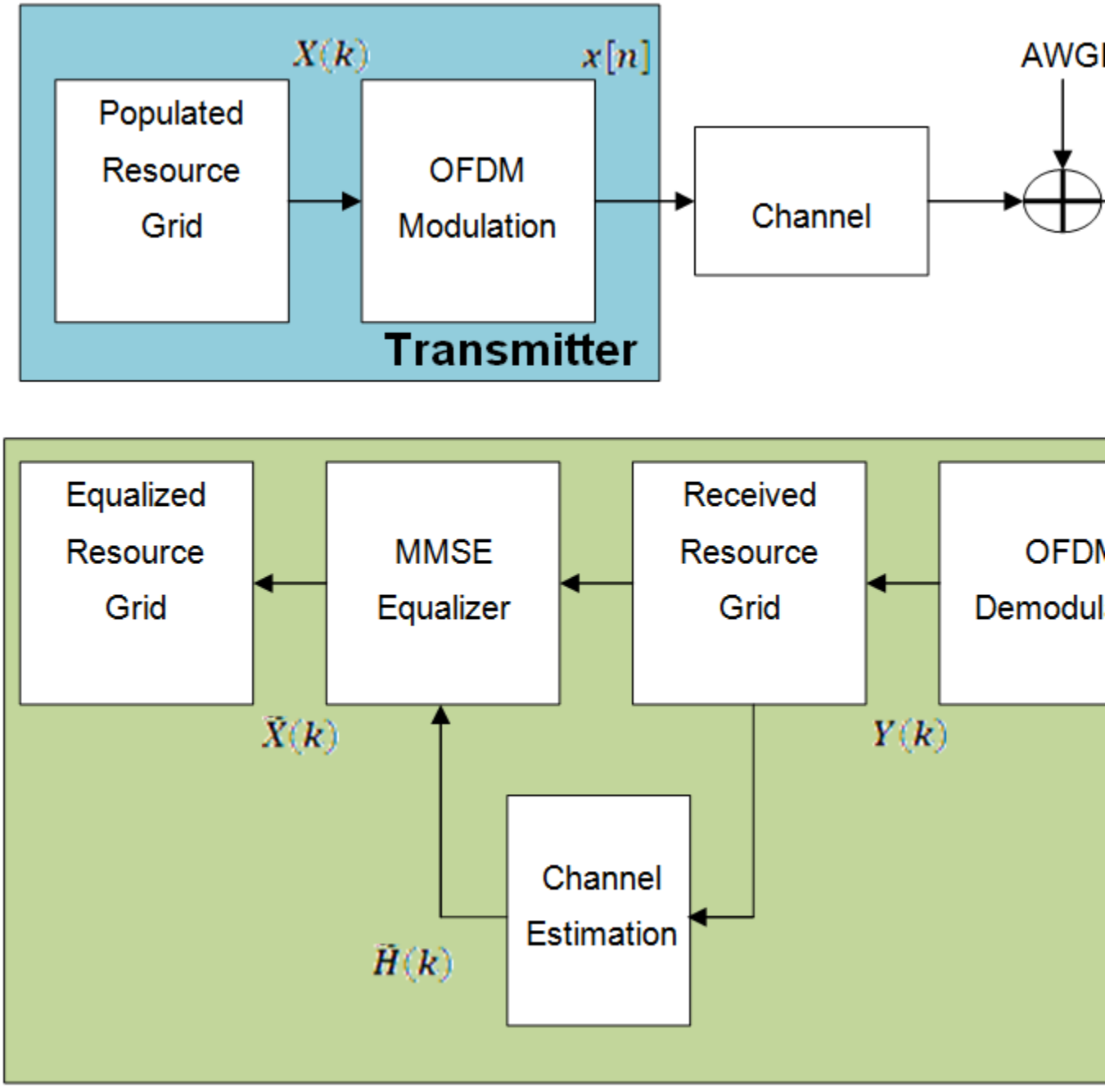
Channel Estimation Overview

The pilot symbols in LTE are assigned positions within a subframe depending on the eNodeB cell identification number and which transmit antenna is being used, as shown in the following figure.



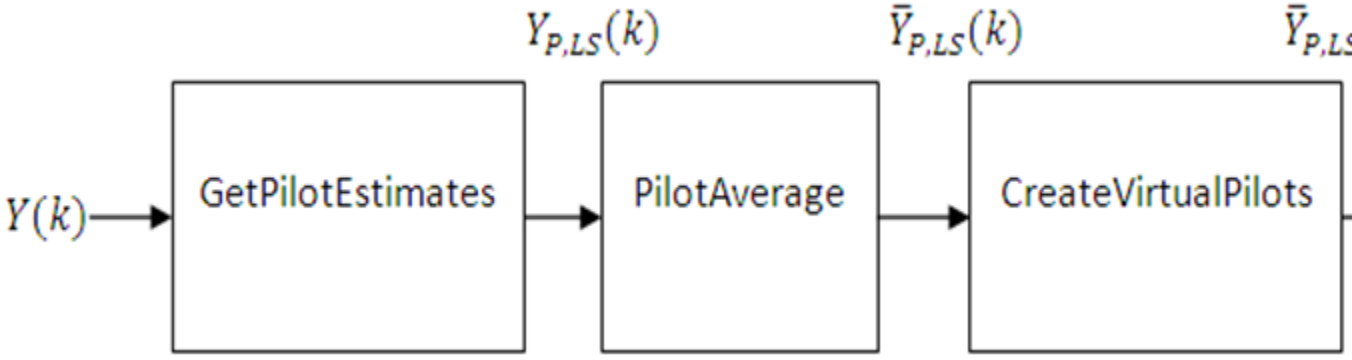
The unique positioning of the pilots ensures that they do not interfere with one another and can be used to provide a reliable estimate of the complex gains imparted onto each resource element within the transmitted grid by the propagation channel.

Both transmit and receive chains and the propagation channel model are shown in the following block diagram.


$X(k)$
$x[n]$
Populated Resource Grid
OFDM Modulation
Transmitter
Channel
AWGN
Equalized Resource Grid
MMSE Equalizer
Received Resource Grid
OFDM Demodul
$\hat{X}(k)$
$Y(k)$
Channel Estimation
$\hat{H}(k)$

The populated resource grid represents a number of subframes containing data. This grid is then OFDM modulated and passed through the model of the propagation channel. Channel noise in the form of additive white Gaussian noise (AWGN) is added before the signal enters the receiver. Once inside the receiver the signal is OFDM demodulated and a received resource grid can be constructed. The received resource grid contains the transmitted resource elements which have been affected by the complex channel gains and the channel noise. Using the known pilot symbols to estimate the channel, it is possible to equalize the effects of the channel and reduce the noise on the received resource grid.

LTE assigns each antenna port a unique set of locations within a subframe to which to map reference signals. This means that no other antenna transmits data at these locations in time and frequency. This allows channel estimation for multi-antenna configurations to be performed. The channel estimation algorithm extracts the reference signals for a transmit/receive antenna pair from the received grid. The least squares estimates of the channel frequency response at the pilot symbols are calculated. The least squares estimates are then averaged to reduce any unwanted noise from the pilot symbols. Virtual pilot symbols are created to aid the interpolation process near the edge of the subframe where no pilot symbols may be located. Using the averaged pilot symbol estimates and the calculated virtual pilot symbols, interpolation is then carried out to estimate the entire subframe. This process is demonstrated in the following block diagram.



Get Pilot Estimates Subsystem

The first step in determining the least squares estimate is to extract the pilot symbols from their known location within the received subframe. The value of these pilot symbols is known and therefore the channel response at these locations can be determined by using the least squares estimate which is obtained by dividing the received pilot symbols by their expected value.

$$Y(k) = X(k)H(k) + noise$$

$$\tilde{H}_{P,LS}(k) = \frac{Y_P(k)}{X_P(k)} + noise$$

Where:

- $Y(k)$ is a received complex symbol value.
- $X(k)$ is a transmitted complex symbol value.
- $H(k)$ is a complex channel gain experienced by a symbol.
- $\tilde{H}_{P,LS}(k)$ is the least squares estimate of the channel at pilot symbol locations.
- $Y_P(k)$ represents the received pilot symbol values.
- $X_P(k)$ represents the known transmitted pilot symbol values.

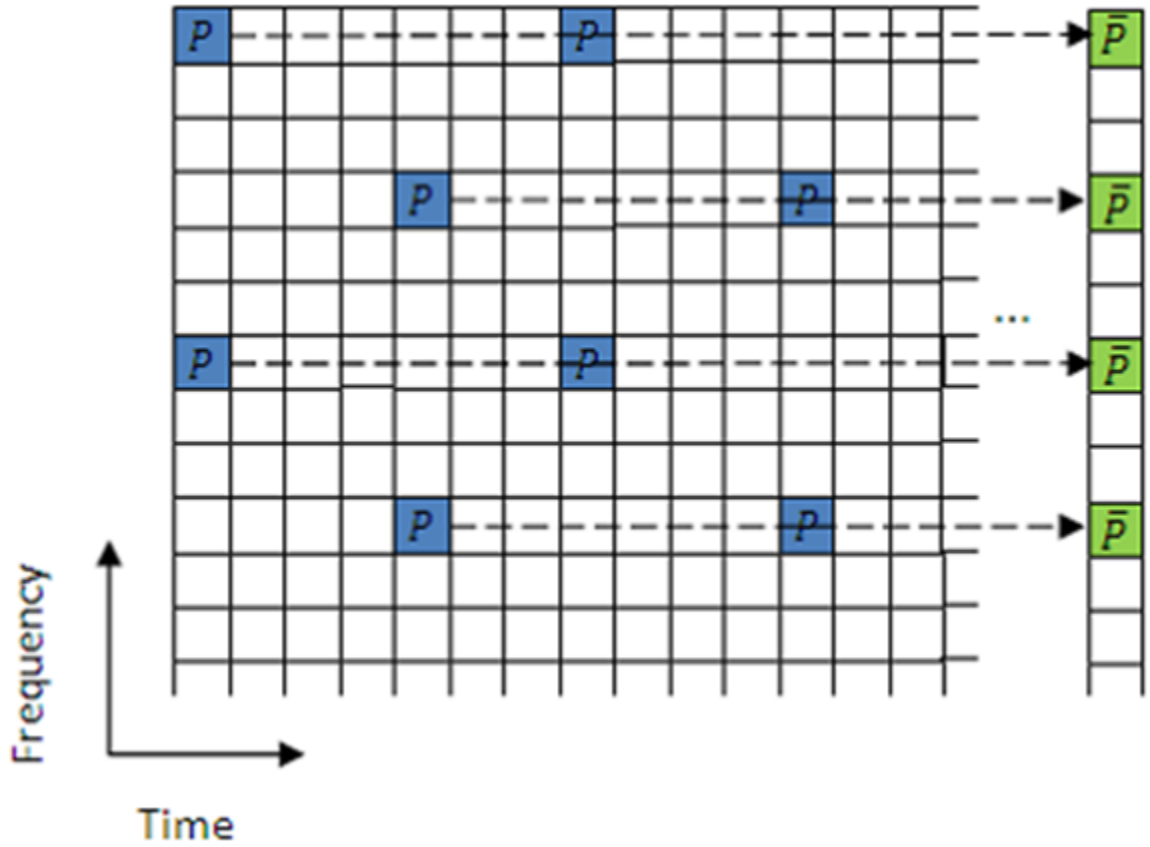
Pilot Average Subsystem

To minimize the effects of noise on the pilot estimates, the least square estimates are averaged using an averaging window. This simple method produces a substantial reduction in the level of noise found on the pilots. The following two pilot symbol averaging methods are available.

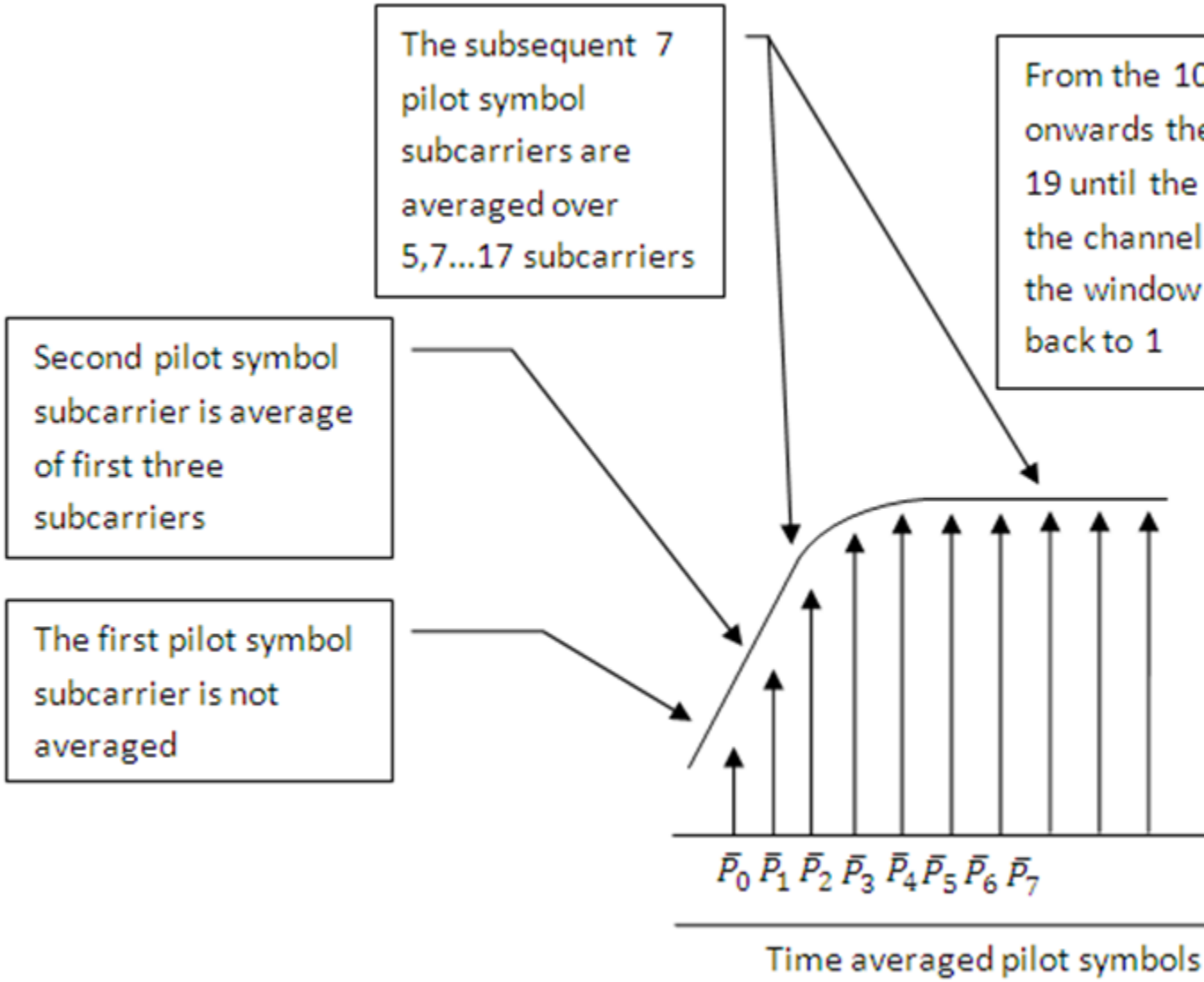
- `'TestEVM'` — follows the method described in annex F.3.4 of [1].
- `'UserDefined'` — allows the user to define the window size and direction of averaging used on the pilot symbols as well as various settings used for the interpolation.

`'TestEVM'`

The first method, `'TestEVM'`, uses the approach described in annex F.3.4 of [1]. Time averaging is performed across each pilot symbol carrying subcarrier, resulting in a column vector containing an average amplitude and phase for each reference signal carrying subcarrier.



The averages of the pilot symbol subcarriers are then frequency averaged using a moving window of maximum size 19.


The subsequent 7 pilot symbol subcarriers are averaged over 5,7...17 subcarriers
From the 10 onwards the 19 until the the channel the window back to 1
Second pilot symbol subcarrier is average of first three subcarriers
The first pilot symbol subcarrier is not averaged
$\bar{P}_0$ $\bar{P}_1$ $\bar{P}_2$ $\bar{P}_3$ $\bar{P}_4$ $\bar{P}_5$ $\bar{P}_6$ $\bar{P}_7$
Time averaged pilot symbols

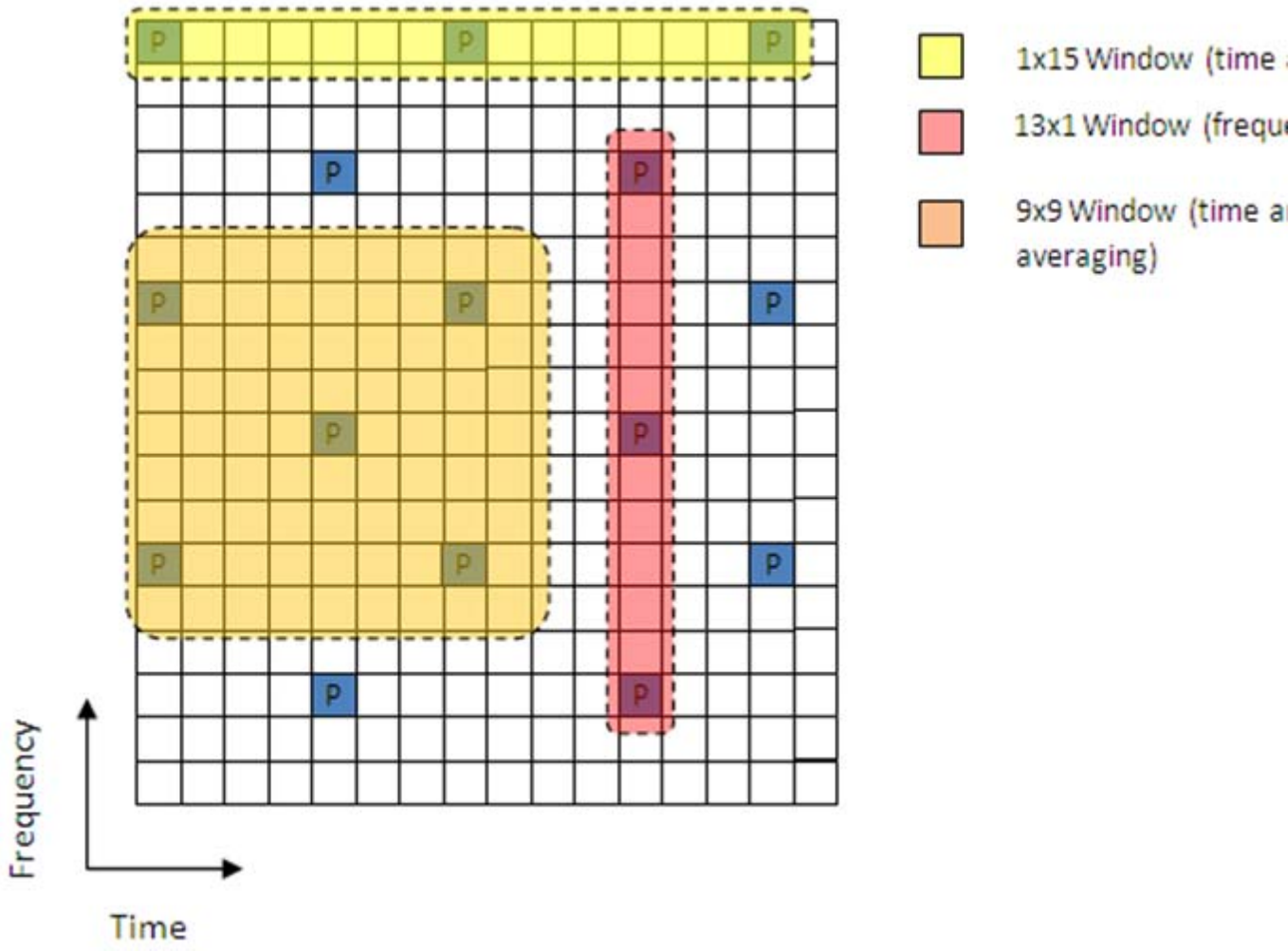
Note There are no user defined parameters with `'TestEVM'`, control of channel estimation parameters is not possible, and the estimation method used is the method described in [1]. The algorithm differs from that described in[1] due to the number of subframes time averaging is performed across. In [1], the method requires 10 subframes to be used. The function `lteDLChannelEstimate` performs time-averaging across the number of subframes passed into the function by `RXGRID`. The greater the number of subframes, the more effective the noise averaging in the time direction.

`'UserDefined'`

The second pilot symbol averaging method, `'UserDefined'`, allows the user to define the size of the averaging window, which direction averaging will be done in (time, frequency or both) and certain aspects of the interpolation that can be adjusted to suit the available data (this will be discussed later).

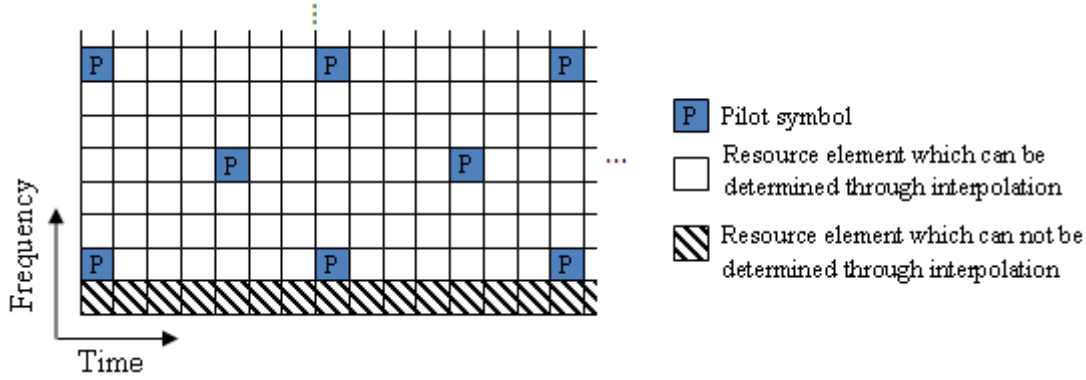
The averaging window size is in terms of resource elements and any pilot symbols located within the window are used to average the value of the pilot symbol found at the centre of the window. The window size must be an odd number ensuring that there is a pilot at the center.

Note Averaging the channel estimates at pilot symbol locations is a simple yet powerful tool, but the window size must be carefully chosen. Using a large window size on a fast fading channel could result in averaging out not only noise, but also channel characteristics. Performing too much averaging on a system with a small amount of noise can have an adverse effect on the quality of the channel estimates. Therefore, using a large averaging window for a fast changing channel could cause the estimate of the channel to appear flat, resulting in a poor estimate of the channel and affecting the quality of the equalization.



Create Virtual Pilots Subsystem

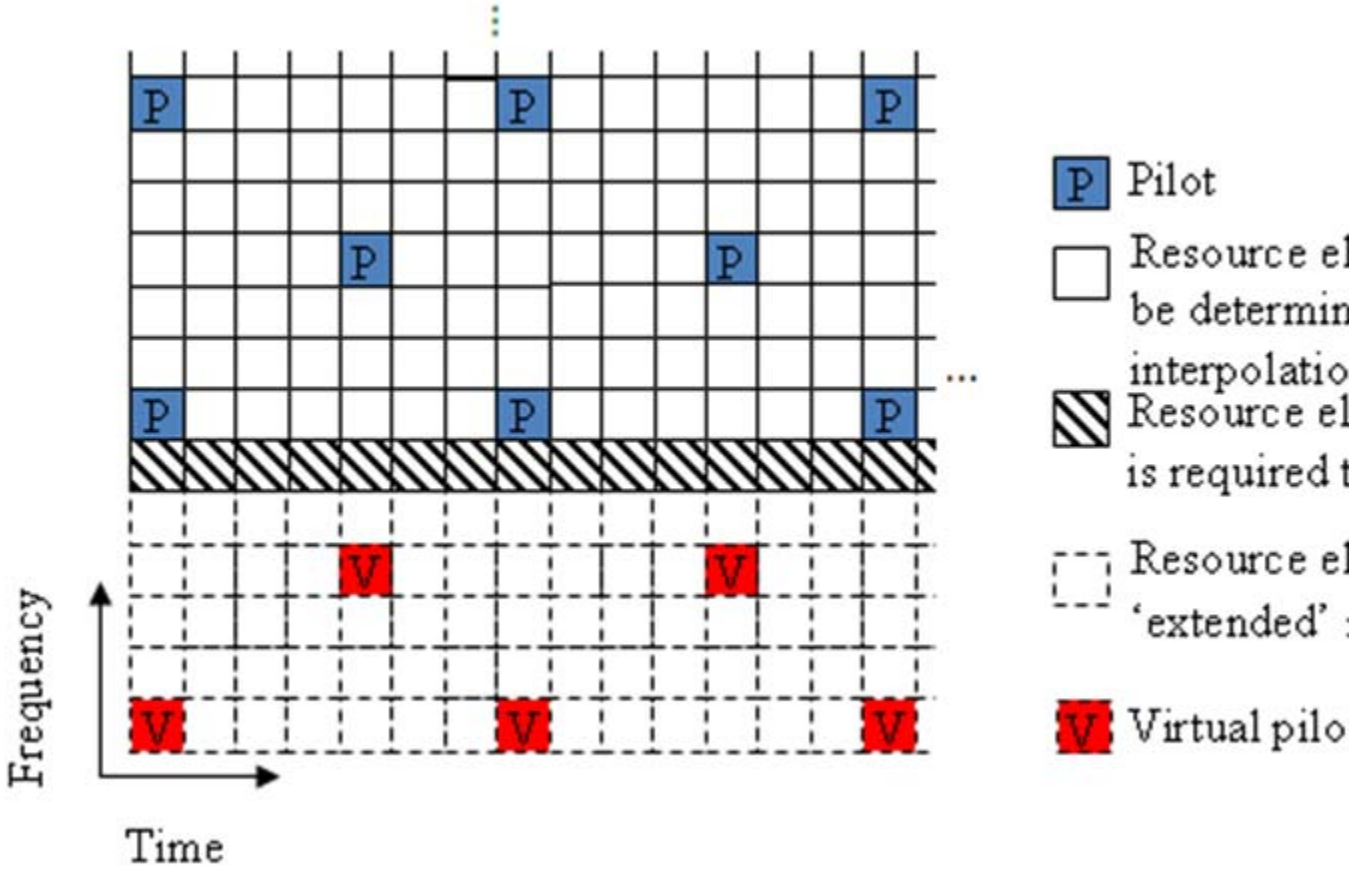
In many instances, edges of the resource grid do not contain any pilot symbols. This effect is shown in the resource grid in the following figure.



In this case, channel estimates at the edges can not be interpolated from the pilot symbols. To overcome this problem, virtual pilot symbols are created. The function `lteDLChannelEstimate` creates virtual pilot symbols on all the edges of the received grid to allow cubic interpolation.

Virtual Pilot Placement

Virtual pilot symbols are created as shown in the following figure.



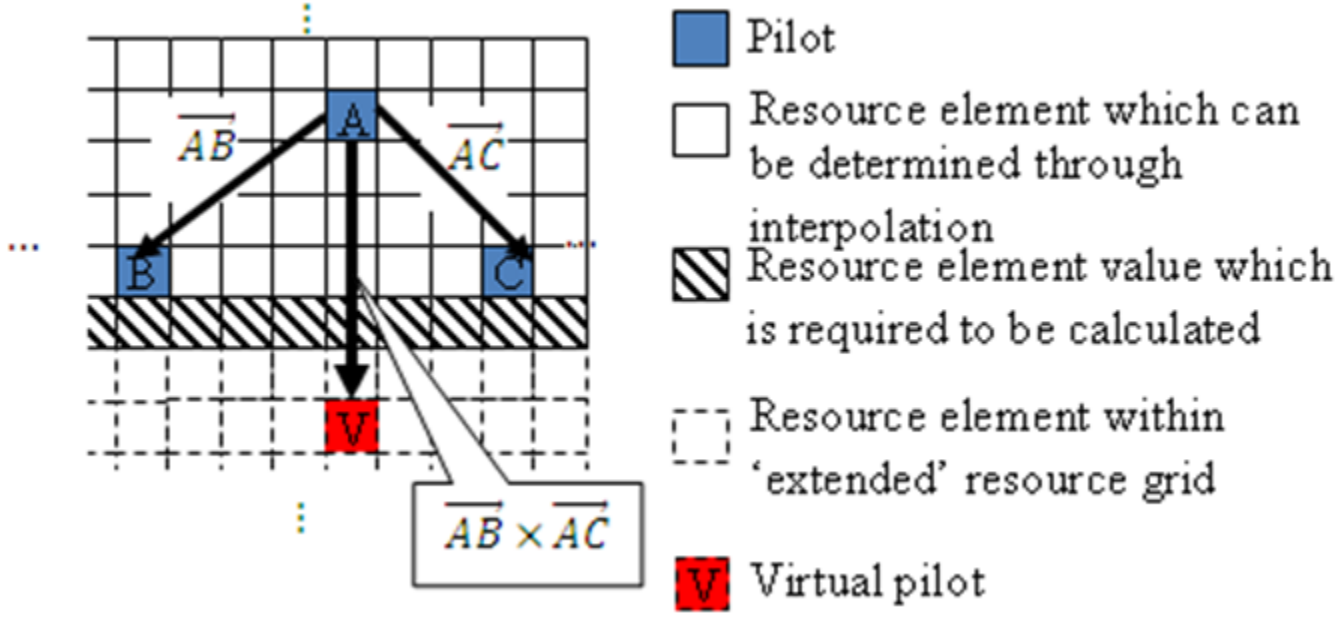
In this system, the resource grid is extended, with virtual pilot symbols created in locations which follow the original reference signal pattern. The presence of virtual pilot symbols allows the channel estimate at the resource elements, which previously could not be calculated by interpolation, to be calculated by interpolation using the original and virtual pilot symbols.

Calculating Virtual Pilot Symbol Values

The virtual pilot symbols are calculated using the original pilot symbols. For each virtual pilot symbol the value is calculated in the following steps.

1. The closest eight ordinary pilots in terms of Euclidian distance in time and frequency are considered.
2. Out of these eight symbols, the closest three pilot symbols are selected. These three symbols must occupy at least two unique subcarriers and two unique OFDM symbols.
3. Two vectors are created—one between the closest and furthest pilot symbols, and one between the second closest and furthest pilot symbols.
4. The cross-product of these two vectors is calculated to create a plane on which the three points reside.
5. The plane is extended to the position of the virtual pilot to calculate the value based on one of the actual pilot values.

This process is shown in the diagram in the following figure.



Note Virtual pilots are only created for the MATLAB® `'linear'` and `'cubic'` interpolation methods.

Interpolation Subsystem

Once the noise has been reduced or removed from the least squares pilot symbol averages and sufficient virtual pilots have been determined, it is possible to use interpolation to estimate the missing values from the channel estimation grid. The pilot averaging method, as described in annex F.3.4 of [1], requires that simple linear interpolation be performed on the time- and frequency-averaged column vector. The interpolation is one-dimensional, since it only estimates the values between the averaged pilot symbol subcarriers in the column vector. The resulting vector is then replicated and used as the channel estimate for the entire resource grid.

The UserDefined pilot-averaging method performs two-dimensional interpolation to estimate the channel response between the available pilot symbols. An interpolation window is used to specify which data is used to perform the interpolation. The pilot symbols used is set by the field `InterpWindow`. This field can take one of the following values.

- `'Causal'` — for past data
- `'Non-causal'` — for future data
- `'Centred'` — for a combination of past, present and future data

The size of this interpolation window can also be adjusted to suit the available data. To specify this window size, set the `InterpWinSize` field.

Noise Estimation

The performance of some receivers can be improved through knowledge of the noise power present on the received signal. The function `lteDLChannelEstimate` provides an estimate of the noise power spectral density (PSD) using the estimated channel response at known reference signal locations. The noise power can be determined by analyzing the noisy least squares estimates and the noise averaged estimates.

The noisy least-squares estimates from the "Get Pilot Estimates Subsystem" on page 1-173 and the noise averaged pilot symbol estimates from the "Pilot Average Subsystem" on page 1-173 provide an indication of the channel noise. The least-squares estimates and the averaged estimates contain the same data, apart from additive noise. Simply taking the difference between the two estimates will result in a value for the noise level on the least squares channel estimates at pilot symbol locations.

$$\tilde{H}_{P,LS}(k) = \frac{Y_P(k)}{X_P(k)} + noise$$

$$\tilde{H}_{P,AVG}(k) = \frac{Y_P(k)}{X_P(k)}$$

$$\tilde{H}_{P,LS}(k) - \tilde{H}_{P,AVG}(k) = noise$$

In practice, it is not possible to completely remove the noise using averaging. It is only possible to reduce the noise and thus only an estimate of the noise power can be made.

Note In the case of a noise free system or system with a high SNR, averaging may have a detrimental effect on the quality of the least squares estimates.

Using the value of the noise power found on the channel response at pilot symbol locations, the noise power per resource element (RE) can be calculated by taking the variance of the resulting noise vector. The noise power per RE for each transmit and receive antenna pair is calculated and stored. The mean of this matrix is returned as the estimate of the noise power per RE.

For a demonstration on how to set up a full transmit and receive chain for channel estimation, see "PDSCH Transmit Diversity Throughput Simulation". In this example, multiple antennas are used and transmission is simulated through a propagation channel model.

References

[1] 3GPP TS 36.141. "Base Station (BS) conformance testing." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`lteDLChannelEstimate` | `lteEqualizeMMSE` | `lteEqualizeZF` | `lteOFDMDemodulate`

Related Examples

- "LTE Downlink Channel Estimation and Equalization"

2

Examples and Demos

- "Create an Empty Resource Grid" on page 2-2
- "Map Reference Signal to Resource Grid" on page 2-3
- "Generate a Test Model" on page 2-5
- "Generate LTE Downlink RMC Waveforms" on page 2-8
- "Generate LTE Uplink RMC Waveforms" on page 2-18
- "Generate LTE Test Model Waveforms" on page 2-27
- "Analyze Throughput for PDSCH Demodulation Performance Test" on page 2-35

Create an Empty Resource Grid

This example shows how to create an empty resource grid of the right dimensions for the cell-wide settings specified in structure `enb`.

Specify the cell-wide settings in a parameter structure.

```
enb.CyclicPrefix = 'Normal';
enb.NDLRB = 9;
enb.CellRefP = 2;
```

Create an empty resource grid. To do so, call the `lteDLResourceGrid` function.

```
resourceGrid = lteDLResourceGrid(enb);
```

The resulting matrix `resourceGrid` is 3-dimensional.

Alternatively, you can use the MATLAB `zeros` function. Specify the parameter structure. For normal cyclic prefix, specify seven symbols per slot.

```
enb.CyclicPrefix = 'Normal';
enb.NDLRB = 9;
enb.CellRefP = 2;
noSymbolsSlot = 7;
```

Create an empty resource grid. This time, call the MATLAB `zeros` function.

```
resourceGrid = zeros(enb.NDLRB*12, noSymbolsSlot*2, enb.CellRefP);
```

See Also

`lteDLResourceGrid` | `zeros`

Map Reference Signal to Resource Grid

This example shows how to map the cell specific reference signals to the resource grid for a subframe in the two antenna case.

Specify the parameter structure. In this scenario, there are 9 resource blocks in the downlink.

```
enb.CyclicPrefix = 'Normal';
enb.NDLRB = 9;
enb.CellRefP = 2;
enb.DuplexMode = 'FDD';
```

Create an empty resource grid for a subframe.

```
resourceGrid = lteDLResourceGrid(enb);
```

Specify `NCellID` and `NSubframe` in the input parameter structure. These fields are required to generate the cell-specific reference signals. In this example, select subframe number 0.

```
enb.NCellID = 10;
enb.NSubframe = 0;
```

Generate the cell-specific reference signals for the two antenna ports. To do so, call the function `lteCellRS`.

```
rsAnt0 = lteCellRS(enb,0);
rsAnt1 = lteCellRS(enb,1);
```

Generate mapping indices for the two antenna ports. You need these mapping indices to map the generated complex symbols to the resource grid.

```
indAnt0 = lteCellRSIndices(enb,0);
indAnt1 = lteCellRSIndices(enb,1);
```

Map the reference signal complex symbols to the resource grid. Use the generated indices in linear form.

```
resourceGrid(indAnt0) = rsAnt0;
resourceGrid(indAnt1) = rsAnt1;
```

The resulting matrix has the complex symbols in `rsAnt0` and `rsAnt1` in the appropriate locations, specified by `indAnt0` and `indAnt1`.

See Also

`lteDLResourceGrid` | `lteCellRS` | `lteCellRSIndices`

Generate a Test Model

In this section...
"E-UTRA Test Models" on page 2-5
"Generate Test Model Waveform" on page 2-6

The LTE specifications define conformance test models for transmitter tests. These tests include transmit signal quality, output power dynamics, EVM for various modulation schemes, BS output power, and RS absolute accuracy. These different test models waveforms can be generated using functions in the LTE System Toolbox product.

E-UTRA Test Models

All E-UTRA test models (E-TMs), as defined in clause 6 of [1], use the following general parameters.

- Single antenna port, 1 code word, 1 layer without any precoding
- Duration is 10 subframes (10 ms)
- Normal cyclic prefix
- Virtual resource blocks of localized type
- UE-specific reference signals are not used

The following physical channels and signals are generated.

- Reference signals (CellRS)
- Primary Synchronization signal (PSS)
- Secondary Synchronization signal (SSS)
- Physical broadcast channel (PBCH)
- Physical control format indicator channel (PCFICH)
- Physical hybrid ARQ indicator channel (PHICH)
- Physical downlink control channel (PDCCH)
- Physical downlink shared channel (PDSCH)

Test models are selected according to the required test case. In this example, the test model under consideration, E-TM1.1, is used to test the following criteria.

- BS output power
- Unwanted emissions
  - Occupied bandwidth
  - ACLR
  - Operating band unwanted emissions
  - Transmitter spurious emissions
- Transmitter intermodulation
- Reference signals absolute accuracy

Generate Test Model Waveform

This example shows how to generate a test model waveform, a time-domain (post-OFDM modulation) signal for a single antenna port and 10 subframes.

Specify, as a string, the test model number. In this example, generate test model 1.1. Test model 1.1 and other test models are defined in section 6 of [1].

```
tm = '1.1';
```

Valid test model values are `'1.1'`, `'1.2'`, `'2'`, `'3.1''`, `'3.2'`, and`'3.3'`.

Specify, as a string, the channel bandwidth. Select a non-standard bandwidth of 9 RB.

```
bw = '9RB';
```

Valid bandwidth values are `'1.4MHz'`, `'3MHz'`, `'5MHz'`, `'10MHz'`, `'15MHz'`, `'20MHz'`, `'9RB'`, `'11RB'`, `'27RB'`, `'45RB'`, `'64RB'`, and `'91RB'`.

Generate the test model waveform.

```
[timeDomainSig,grid] = lteTestModelTool(tm,bw);
```

The channel model number and the bandwidth determine the physical channel and signal parameters, as specified in [1]. The generated waveform, `timeDomainSig`, is a time-domain signal that has undergone OFDM modulation, cyclic prefix insertion, and windowing. The second output, `grid`, is a 2-dimensional array representing the resource grid spanning 10 subframes.

References

[1] 3GPP TS 36.141. "Base Station (BS) conformance testing." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`lteTestModelTool`

Generate LTE Downlink RMC Waveforms

In this section...

"LTE Downlink RMC Generator Overview" on page 2-8

"Open LTE Downlink RMC Generator App" on page 2-8

"Dialog Box Inputs and Outputs" on page 2-10

"Examples" on page 2-14

LTE Downlink RMC Generator Overview

You can use the LTE Downlink RMC Generator app to generate downlink reference measurement channel (RMC) waveforms.

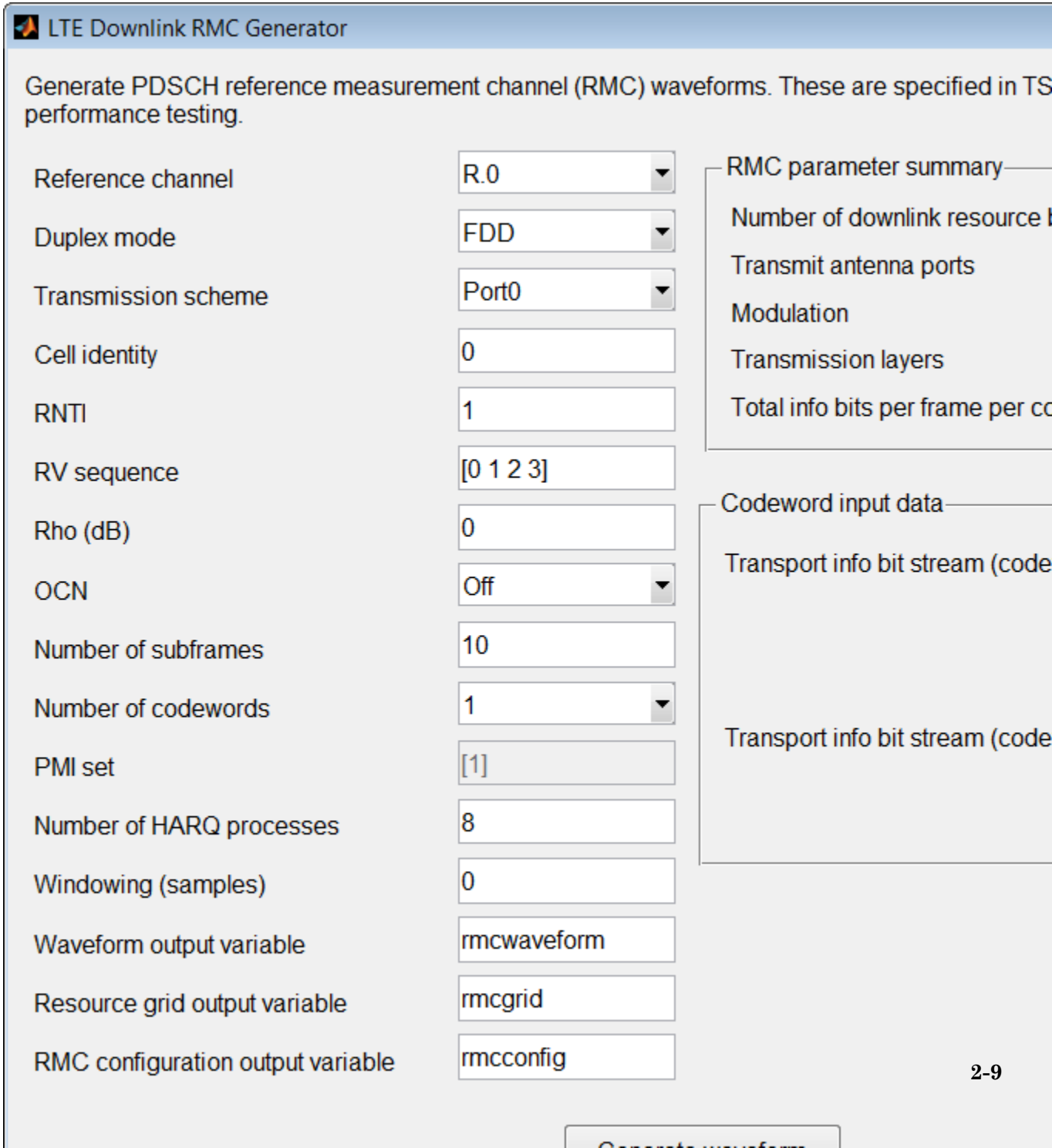
Open LTE Downlink RMC Generator App

To open the LTE Downlink RMC Generator app, select the **APPS** tab on the MATLAB desktop and click the following icon.

Alternatively, you can type the following command in the MATLAB command window.

```
lteRMCDLTool;
```

The LTE Downlink RMC Generator dialog box appears.


LTE Downlink RMC Generator
Generate PDSCH reference measurement channel (RMC) waveforms. These are specified in TS
performance testing.
Reference channel
R.0
Duplex mode
FDD
Transmission scheme
Port0
Cell identity
0
RNTI
1
RV sequence
[0 1 2 3]
Rho (dB)
0
OCN
Off
Number of subframes
10
Number of codewords
1
PMI set
[1]
Number of HARQ processes
8
Windowing (samples)
0
Waveform output variable
rmcwaveform
Resource grid output variable
rmcgrid
RMC configuration output variable
rmcconfig
RMC parameter summary
Number of downlink resource b
Transmit antenna ports
Modulation
Transmission layers
Total info bits per frame per co
Codeword input data
Transport info bit stream (code
Transport info bit stream (code

Dialog Box Inputs and Outputs

Parameters

In the LTE Downlink RMC Generator dialog box, you can set the following parameters.

Parameter (Equivalent Field)	Values	Description
Reference channel (RC)	'R0' (default), 'R1', 'R2', 'R3', 'R4', 'R5', 'R6', 'R7', 'R8', 'R9', 'R10', 'R11', 'R12', 'R13', 'R14', 'R25', 'R26', 'R27', 'R28', 'R6-27RB', 'R12-9RB', 'R11-45RB'	Reference measurement channel (RMC) number or type, as specified in TS36.101 Annex A.3. The 'R6-27RB', 'R12-9RB', and 'R11-45RB' RC values are custom RMC configured for nonstandard bandwidths, but with the same code rate as the standardized versions.
Duplex mode (DuplexMode)	'FDD' (default), 'TDD'	Duplex mode
Transmission scheme (TxScheme)	'Port0', 'TxDiversity', 'CDD', 'SpatialMux', 'MultiUser', 'Port5', 'Port7-8', 'Port8', 'Port7-14'	PDSCH transmission scheme, specified as one of the following options. • 'Port0' — Single-antenna port, port 0 • 'TxDiversity' — Transmit diversity scheme • 'CDD' — Large delay CDD scheme • 'SpatialMux' — Closed-loop spatial multiplexing • 'MultiUser' — Multiuser MIMO scheme

Parameter (Equivalent Field)	Values	Description
		• `'Port5'` — Single-antenna port, port 7, if `NLayers` is 1. Dual layer transmission, port 7 and 8, if `NLayers` is 2. • `'Port7-8'` — Single-antenna port, port 0 • `'Port8'` — Single-antenna port, port 8 • `'Port7-14'` — Up to 8-layer transmission, ports 7–14
Cell identity (`NCellID`)	Nonnegative scalar integer (0,...,503)	Physical layer cell identity
RNTI (`RNTI`)	Scalar integer	Radio network temporary identifier (RNTI) value (16 bits)
RV sequence (`RVSeq`)	Integer vector (0,1,2,3)	Redundancy version (RV) sequence, applied to all subframes
Rho (dB) (`Rho`)	Scalar number	PDSCH resource element power allocation, in dB
OCNG (`OCNG`)	`'Off'`, `'On'`	OFDMA channel noise generator
Number of subframes (`TotSubframes`)	Nonnegative scalar integer	Total number of subframes to generate
Number of codewords (`NCodewords`)	1, 2	Number of PDSCH codewords to use in PDSCH transmission

Parameter (Equivalent Field)	Values	Description
PMI set (`PMISet`)	Integer vector (0...15)	Precoder matrix indication (PMI) set. It can contain either a single value, corresponding to single PMI mode, or multiple values, corresponding to multiple or subband PMI mode.
Number of HARQ processes (`NHARQProcesses`)	Nonnegative scalar integer	Number of HARQ processes
Windowing (samples) (`Windowing`)	Nonnegative scalar integer	Number of time-domain samples over which windowing and overlapping of OFDM symbols is applied
Waveform output variable	String beginning with an alphabetical character and containing alphanumeric characters	Waveform output variable name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.
Resource grid output variable	String beginning with an alphabetical character and containing alphanumeric characters	Resource grid output variable name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.
RMC configuration output variable	String beginning with an alphabetical character and containing alphanumeric characters	RMC configuration output parameter structure name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.

RMC Parameter Summary

The LTE Downlink RMC Generator dialog box displays the following RMC parameters.

Parameter (Equivalent Field)	Values	Description
Number of downlink resource blocks (`NDLRB`)	Scalar integer (6,...,110). Standard bandwidth values are 6, 15, 25, 50, 75, and 100.	Number of downlink (DL) resource blocks (RBs)
Transmit antenna ports (`CellRefP`)	1, 2, 4	Number of cell-specific reference signal (CRS) antenna ports
Modulation (`Modulation`)	`'QPSK'`, `'16QAM'`, `'64QAM'`	Modulation type
Transmission layers (`NLayers`)	1, 2, 3, 4	Number of transmission layers
Total info bits per frame per codeword	Positive scalar integer	Total transport block capacity per frame per codeword

Codeword Input Data

In the LTE Downlink RMC Generator dialog box, you can set the following input data for codewords. These input data are equivalent to elements of the `trdata` cell array in the `lteRMCDLTool` function.

Input Data	Values	Description
Transport info bit stream (codeword 1)	**User defined**, **External file**	Information bits to transmit on PDSCH for codeword 1.
Transport info bit stream (codeword 2)	**User defined**, **External file**	Information bits to transmit on PDSCH for codeword 2.

For each input, you can select one of the following options from the drop-down menu.

- **User defined** — Select this option to specify the information bits as a vector of zeros and ones or a variable name. Either enter in the vector manually, or specify the name of an existing variable in the MATLAB workspace. Each vector contains the information bits stream to be coded across the duration of the generation, which represents multiple concatenated transport blocks. Internally, these vectors are looped if the number of bits required across all subframes of the generation exceeds the length of the vectors provided. This feature allows you to enter a short pattern, such as `[1;0;0;1]`, that is repeated as the input to the transport coding.
- **External file** — Select this option to specify a MAT-file in which the variable that you want to use as input data is stored. When you click **Load File**, the Select Files dialog box opens. Select the file that contains the input data, and click **Open**. The Import Wizard dialog box opens. Select the variable in this file that contains the information bits, and click **Finish**.

Examples

Generate RMC R.12 Waveform

This example shows how to generate a time-domain signal, `waveform`, and a 3-D array of the resource elements, `grid`, for the RMC R.12, as specified in [1].

Open the LTE Downlink RMC Generator app. Select the **APPS** tab on the MATLAB desktop and click the following icon.

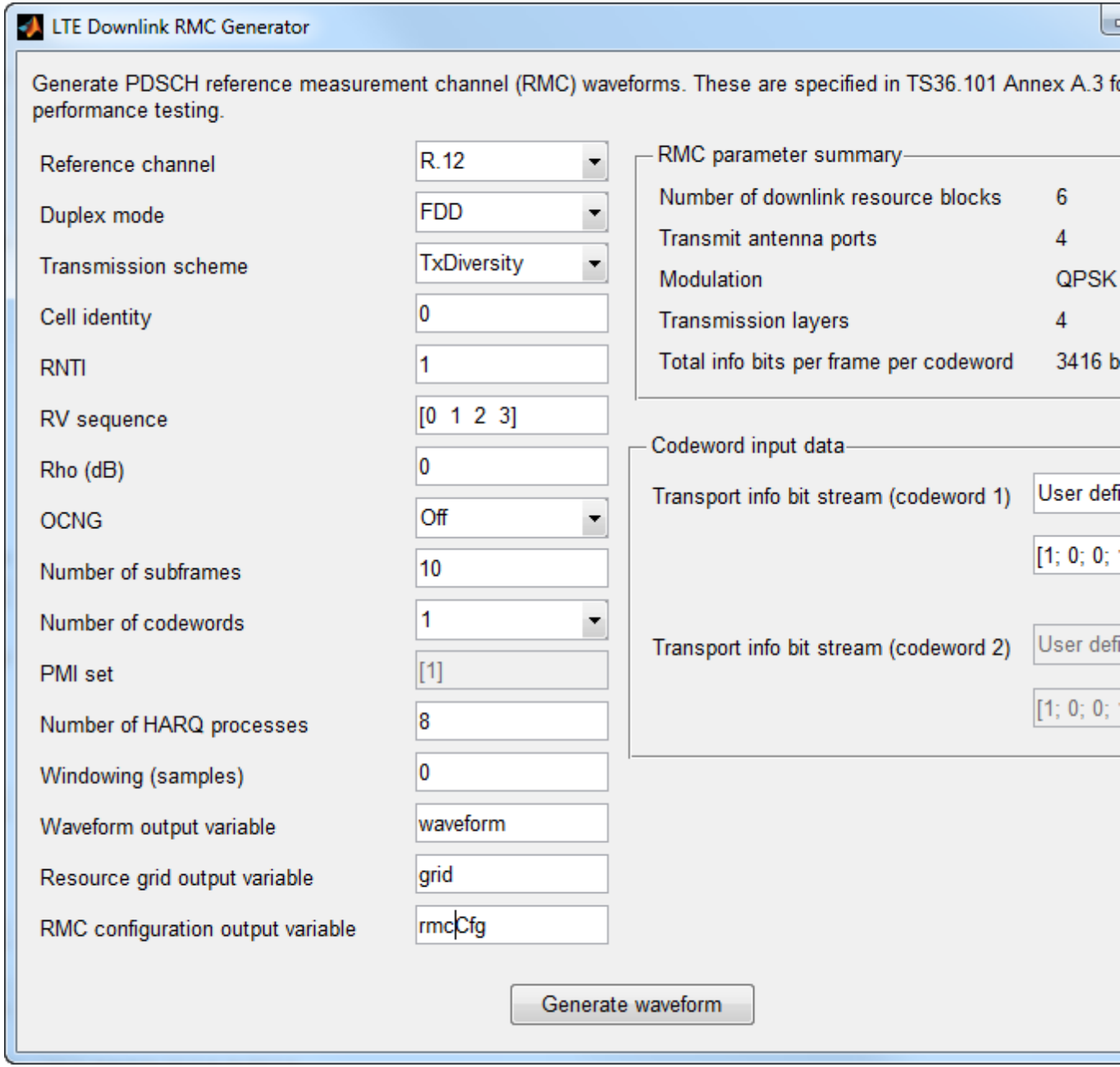
The LTE Downlink RMC Generator dialog box opens.

From the **RC** drop-down list, choose `R.12`.

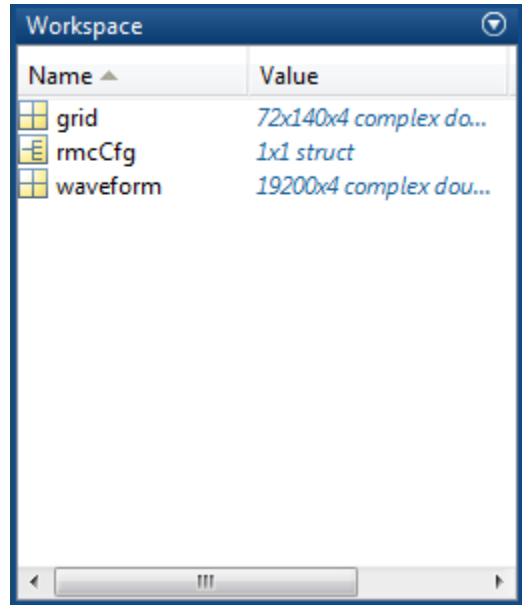
For **Waveform output variable**, enter `waveform`.

For **Resource grid output variable**, enter `grid`.

For **RMC configuration output variable**, enter `rmcCfg`. The dialog box now appears as shown in the following figure.


LTE Downlink RMC Generator
Generate PDSCH reference measurement channel (RMC) waveforms. These are specified in TS36.101 Annex A.3 fo
performance testing.
Reference channel
R.12
Duplex mode
FDD
Transmission scheme
TxDiversity
Cell identity
0
RNTI
1
RV sequence
[0 1 2 3]
Rho (dB)
0
OCNG
Off
Number of subframes
10
Number of codewords
1
PMI set
[1]
Number of HARQ processes
8
Windowing (samples)
0
Waveform output variable
waveform
Resource grid output variable
grid
RMC configuration output variable
rmcCfg
RMC parameter summary
Number of downlink resource blocks
6
Transmit antenna ports
4
Modulation
QPSK
Transmission layers
4
Total info bits per frame per codeword
3416 b
Codeword input data
Transport info bit stream (codeword 1)
User defi
[1; 0; 0;
Transport info bit stream (codeword 2)
User defi
[1; 0; 0;
Generate waveform

Click the **Generate waveform** button. The variables `waveform`, `grid`, and `rmcCfg` now appear in the MATLAB Workspace Browser.



References

[1] 3GPP TS 36.101. "User Equipment (UE) Radio Transmission and Reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`lteRMCDLTool` | `lteRMCDL` | `lteDLConformanceTestTool` | `lteRMCULTool` | `lteTestModelTool`

Concepts

- "Generate LTE Uplink RMC Waveforms" on page 2-18
- "Generate LTE Test Model Waveforms" on page 2-27
- "Analyze Throughput for PDSCH Demodulation Performance Test" on page 2-35

Generate LTE Uplink RMC Waveforms

In this section...

"LTE Uplink RMC Generator Overview" on page 2-18
"Open LTE Uplink RMC Generator App" on page 2-18
"Dialog Box Inputs and Outputs" on page 2-20
"Examples" on page 2-23

LTE Uplink RMC Generator Overview

You can use the LTE Uplink RMC Generator app to generate uplink reference measurement channel (RMC) waveforms.

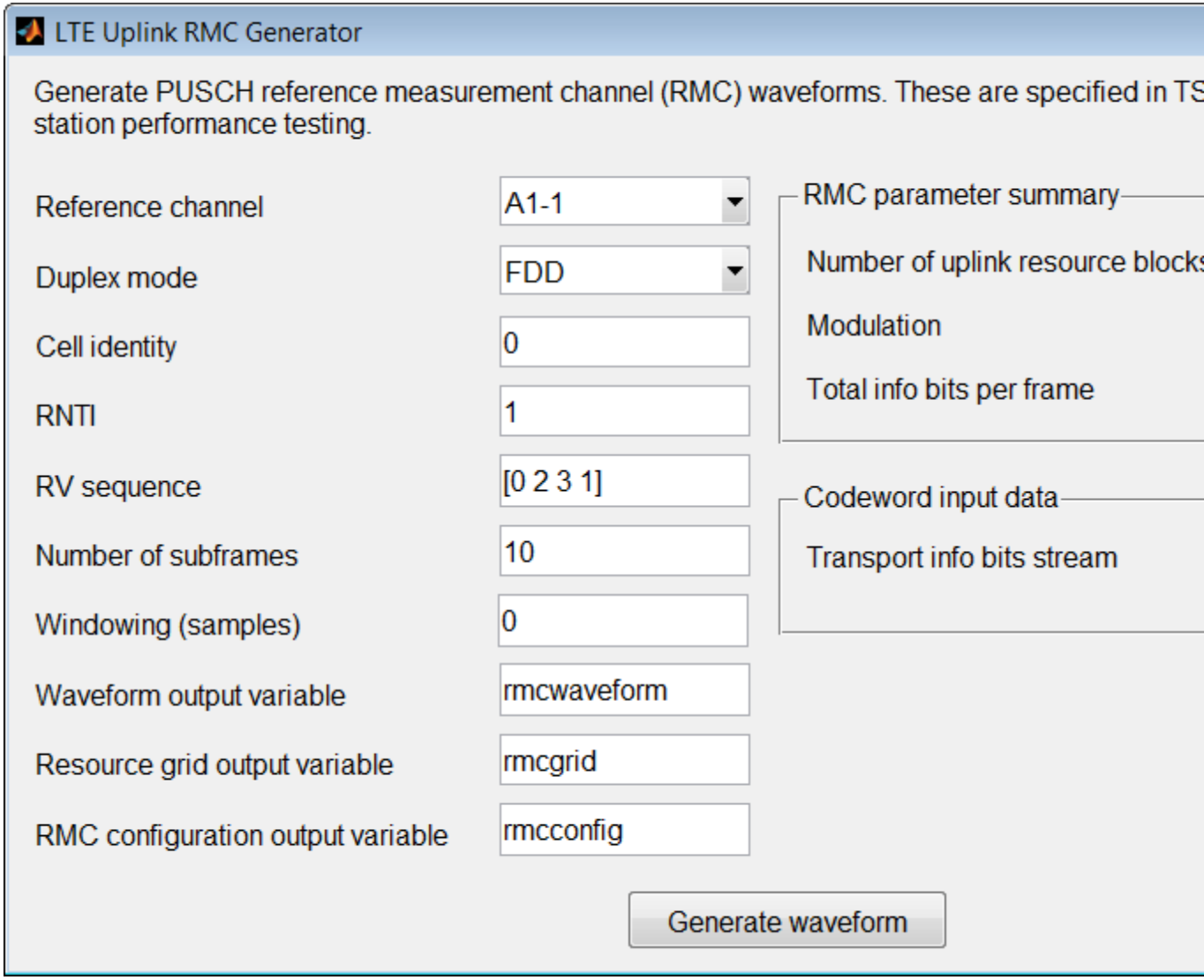
Open LTE Uplink RMC Generator App

To open the LTE Uplink RMC Generator app, select the **APPS** tab on the MATLAB desktop and click the following icon.

Alternatively, you can type the following command in the MATLAB command window.

```
lteRMCULTool;
```

The LTE Uplink RMC Generator dialog box appears.


LTE Uplink RMC Generator
Generate PUSCH reference measurement channel (RMC) waveforms. These are specified in TS
station performance testing.
Reference channel
A1-1
Duplex mode
FDD
Cell identity
0
RNTI
1
RV sequence
[0 2 3 1]
Number of subframes
10
Windowing (samples)
0
Waveform output variable
rmcwaveform
Resource grid output variable
rmcgrid
RMC configuration output variable
rmcconfig
RMC parameter summary
Number of uplink resource blocks
Modulation
Total info bits per frame
Codeword input data
Transport info bits stream
Generate waveform

Dialog Box Inputs and Outputs

Parameters

In the LTE Uplink RMC Generator dialog box, you can set the following parameters.

Parameter (Equivalent Field)	Values	Description
Reference channel (RC)	'A1-1' (default), 'A1-2', 'A1-3', 'A1-4', 'A1-5', 'A2-1', 'A2-2', 'A2-3', 'A3-1', 'A3-2', 'A3-3', 'A3-4', 'A3-5', 'A3-6', 'A3-7', 'A4-1', 'A4-2', 'A4-3', 'A4-4', 'A4-5', 'A4-6', 'A4-7', 'A4-8', 'A5-1', 'A5-2', 'A5-3', 'A5-4', 'A5-5', 'A5-6', 'A5-7', 'A7-1', 'A7-2', 'A7-3', 'A7-4', 'A7-5', 'A7-6', 'A8-1', 'A8-2', 'A8-3', 'A8-4', 'A8-5', 'A8-6', 'A3-2-9RB'	Reference measurement channel (RMC) number or type, as specified in TS36.104 Annex A.
Duplex mode (DuplexMode)	'FDD' (default), 'TDD'	Duplex mode
Cell identity (NCellID)	Nonnegative scalar integer (0,...,503)	Physical layer cell identity
RNTI (RNTI)	Scalar integer	Radio network temporary identifier (RNTI) value (16 bits)
RV sequence (RVSeq)	Integer vector (0,1,2,3)	Redundancy version (RV) sequence, applied to all subframes

Parameter (Equivalent Field)	Values	Description
Number of subframes (`TotSubframes`)	Nonnegative scalar integer	Total number of subframes to generate
Windowing (samples) (`Windowing`)	Nonnegative scalar integer	Number of time-domain samples over which windowing and overlapping of OFDM symbols is
Waveform output variable	String beginning with an alphabetical character and containing alphanumeric characters	Waveform output variable name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.
Resource grid output variable	String beginning with an alphabetical character and containing alphanumeric characters	Resource grid output variable name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.
RMC configuration output variable	String beginning with an alphabetical character and containing alphanumeric characters	RMC configuration output parameter structure name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.

RMC Parameter Summary

The LTE Uplink RMC Generator dialog box displays the following RMC parameters.

Parameter (Equivalent Field)	Values	Description
Number of uplink resource blocks (`NULRB`)	Positive scalar integer	Number of uplink (UL) resource blocks (RBs)
Modulation (`Modulation`)	`'QPSK'`, `'16QAM'`, `'64QAM'`	Modulation type
Total info bits per frame	Positive scalar integer	Total transport block capacity per frame

Codeword Input Data

In the LTE Downlink RMC Generator dialog box, you can set the following input data for codewords. These input data are equivalent to elements of the `trdata` cell array in the `lteRMCDLTool` function.

Input Data	Values	Description
Transport info bit stream	Logical column vector, variable name	Information bits to transmit on PUSCH

Either enter in the vector manually, or specify the name of an existing variable in the MATLAB workspace. Each vector contains the information bits stream to be coded across the duration of the generation, which represents multiple concatenated transport blocks. Internally, these vectors are looped if the number of bits required across all subframes of the generation exceeds the length of the vectors provided. This feature allows you to enter a short pattern, such as `[1;0;0;1]`, that is repeated as the input to the transport coding.

Examples

Generate RMC A1-1 Waveform

This example shows how to generate a time-domain signal, `txWaveform`, and a 3-D array of the resource elements, `txGrid`, for the RMC A1-1, as specified in [1].

Open the LTE Uplink RMC Generator app. Select the **APPS** tab on the MATLAB desktop and click the following icon.

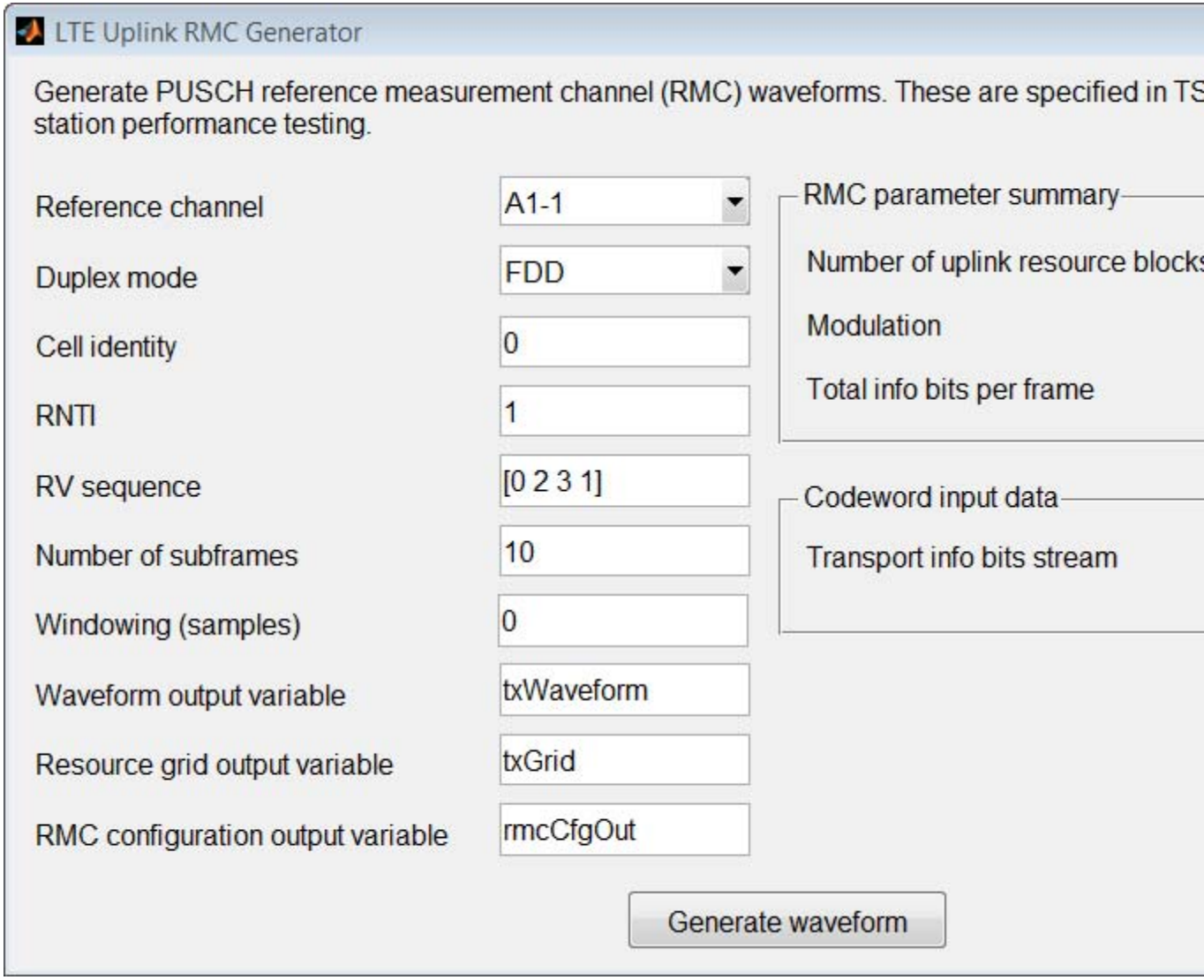
The LTE Uplink RMC Generator dialog box opens.

From the **RC** drop-down list, choose `A1-1`.

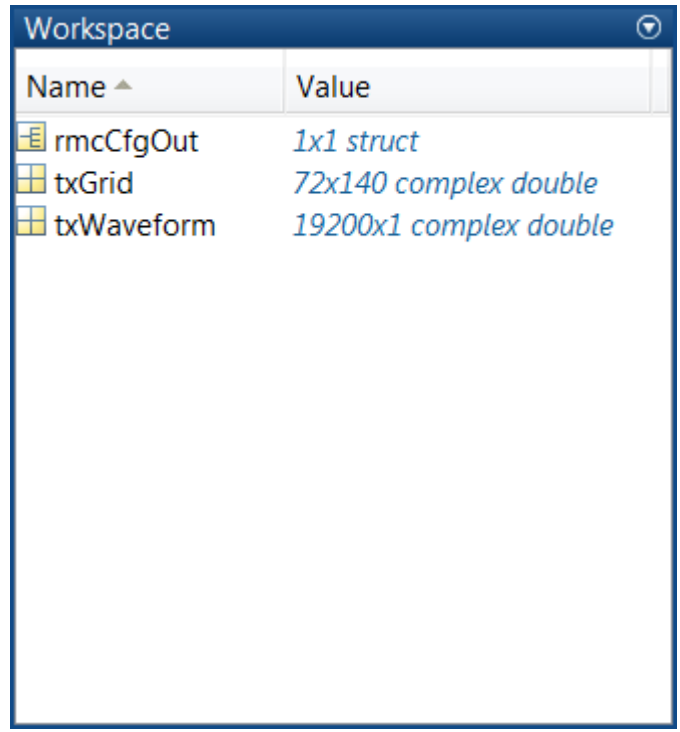
For **Waveform output variable**, enter `txWaveform`.

For **Resource grid output variable**, enter `txGrid`.

For **RMC configuration output variable**, enter `rmcCfgOut`. The dialog box now appears as shown in the following figure.



Click the **Generate waveform** button. The variables `txWaveform`, `txGrid`, and `rmcCfgOut` now appear in the MATLAB Workspace Browser.



References

[1] 3GPP TS 36.104. "Base Station (BS) radio transmission and reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

See Also

`lteRMCULTool` | `lteRMCUL` | `lteRMCDLTool` | `lteDLConformanceTestTool` | `lteTestModelTool`

Concepts

- "Generate LTE Downlink RMC Waveforms" on page 2-8
- "Generate LTE Test Model Waveforms" on page 2-27

- "Analyze Throughput for PDSCH Demodulation Performance Test" on page 2-35

Generate LTE Test Model Waveforms

In this section...

"LTE Test Model Generator Overview" on page 2-27

"Open the LTE Test Model Generator App" on page 2-27

"Dialog Box Inputs and Outputs" on page 2-29

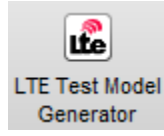
"Examples" on page 2-30

LTE Test Model Generator Overview

You can use the LTE Test Model Generator app to generate E-UTRA Test Model (E-TM) waveforms.

Open the LTE Test Model Generator App

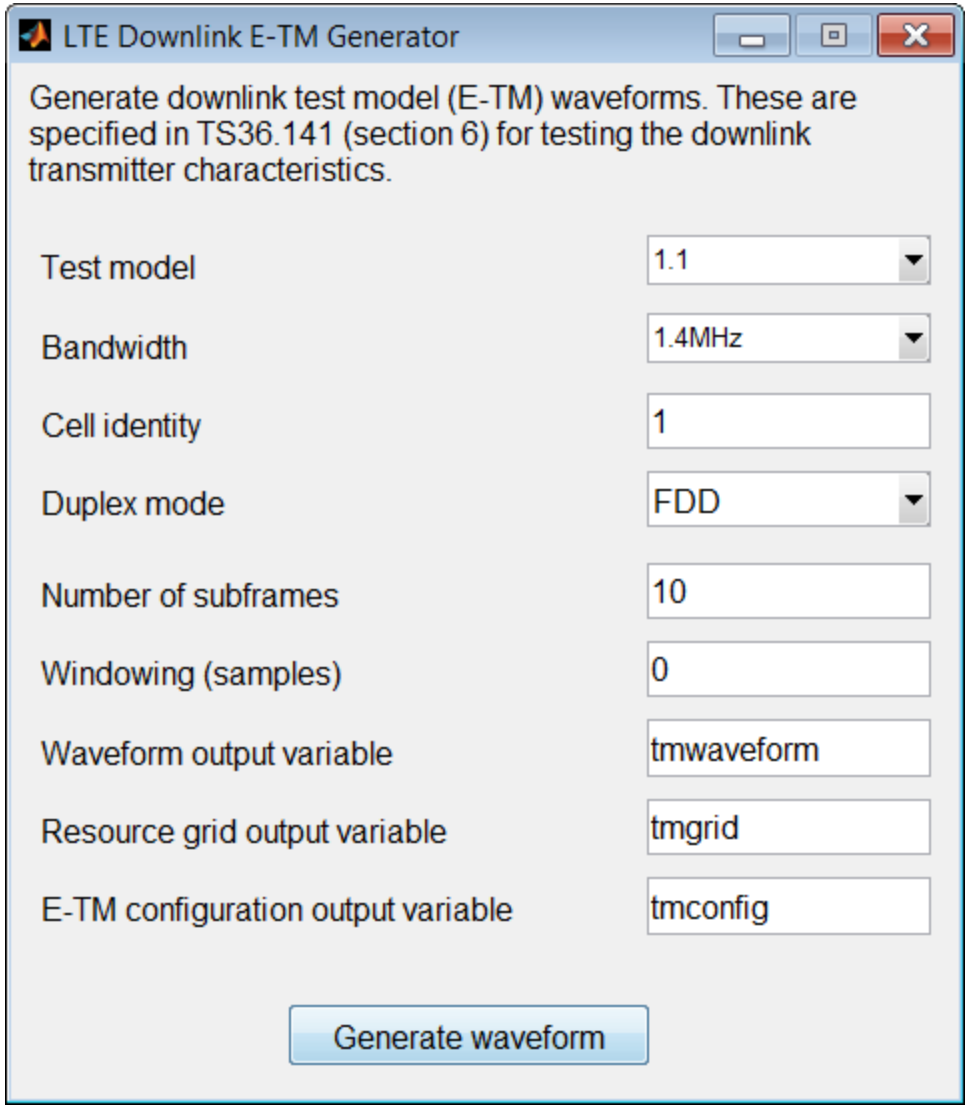
To open the LTE Test Model Generator app, select the **APPS** tab on the MATLAB desktop and click the following icon.



Alternatively, you can type the following command in the MATLAB command window.

```
lteTestModelTool;
```

The LTE Downlink E-TM Generator dialog box appears.


LTE Downlink E-TM Generator
Generate downlink test model (E-TM) waveforms. These are specified in TS36.141 (section 6) for testing the downlink transmitter characteristics.
Test model
1.1
Bandwidth
1.4MHz
Cell identity
1
Duplex mode
FDD
Number of subframes
10
Windowing (samples)
0
Waveform output variable
tmwaveform
Resource grid output variable
tmgrid
E-TM configuration output variable
tmconfig
Generate waveform

Dialog Box Inputs and Outputs

Parameters

In the LTE Downlink E-TM Generator dialog box, you can set the following parameters.

Parameter (Equivalent Field)	Values	Description
Test model (`TMN`)	`'1.1'`, `'1.2'`, `'2'`, `'3.1'`, `'3.2'`, `'3.3'`	Test model number, as specified in TS 36.141, a type string used when generating test model waveforms
Bandwidth (`BW`)	`'1.4MHz'`, `'3MHz'`, `'5MHz'`, `'10MHz'`, `'15MHz'`, `'20MHz'`, `'9RB'`, `'11RB'`, `'27RB'`, `'45RB'`, `'64RB'`, `'91RB'`,	Channel bandwidth type string when generating test model waveforms
Cell identity (`NCellID`)	Nonnegative scalar integer (0,...,503)	Physical layer cell identity
Duplex mode (`DuplexMode`)	`'FDD'` (default), `'TDD'`	Duplex mode
Number of subframes (`TotSubframes`)	Nonnegative scalar integer	Total number of subframes to generate
Windowing (samples) (`Windowing`)	Nonnegative scalar integer	Number of time-domain samples over which windowing and overlapping of OFDM symbols is applied

Parameter (Equivalent Field)	Values	Description
Waveform output variable	String beginning with an alphabetical character and containing alphanumeric characters	Waveform output variable name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.
Resource grid output variable	String beginning with an alphabetical character and containing alphanumeric characters	Resource grid output variable name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.
E-TM configuration output variable	String beginning with an alphabetical character and containing alphanumeric characters	E-TM configuration output parameter structure name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.

Examples

Generate E-TM 3.2 Waveform

This example shows how to generate a time-domain signal, `waveform`, and a 3-D array of the resource elements, `grid`, for the E-TM 3.2, as specified in [1].

Open the LTE Test Model Generator app. Select the **APPS** tab on the MATLAB desktop and click the following icon.

The LTE Downlink E-TM Generator dialog box opens.

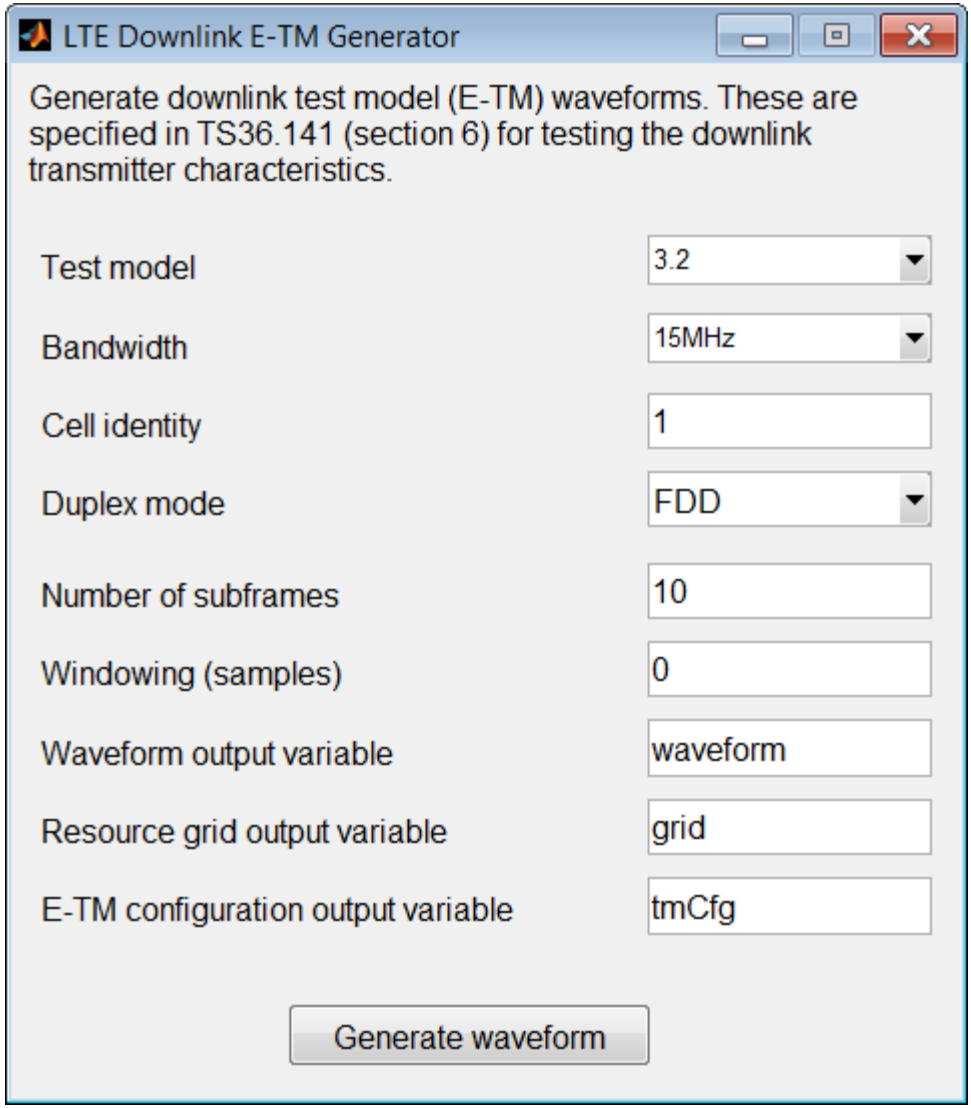
From the **Test model** drop-down list, choose `3.2`.

From the **Bandwidth** drop-down list, choose `15MHz`.

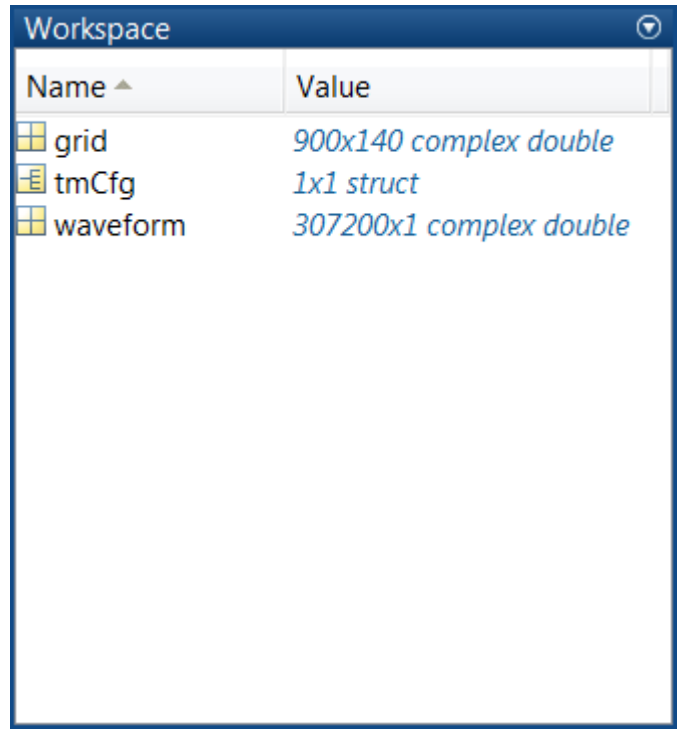
For **Waveform output variable**, enter `waveform`.

For **Resource grid output variable**, enter `grid`.

For **E-TM configuration output variable**, enter `tmCfg`. The dialog box now appears as shown in the following figure.



Click the **Generate waveform** button. The variables `waveform`, `grid`, and `tmCfg` now appear in the MATLAB Workspace Browser.



References

[1] 3GPP TS 36.141. "Base Station (BS) conformance testing." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

See Also

`lteTestModelTool` | `lteTestModel` | `lteRMCDLTool` | `lteRMCULTool` | `lteDLConformanceTestTool`

Concepts

- "Generate LTE Downlink RMC Waveforms" on page 2-8
- "Generate LTE Uplink RMC Waveforms" on page 2-18

- "Analyze Throughput for PDSCH Demodulation Performance Test" on page 2-35

Analyze Throughput for PDSCH Demodulation Performance Test

In this section...
"LTE Throughput Analyzer Overview" on page 2-35
"Open LTE Throughput Analyzer App" on page 2-35
"Dialog Box Inputs and Outputs" on page 2-37
"Examples" on page 2-39

LTE Throughput Analyzer Overview

You can use the LTE Throughput Analyzer app to execute a physical downlink shared channel (PDSCH) demodulation performance test.

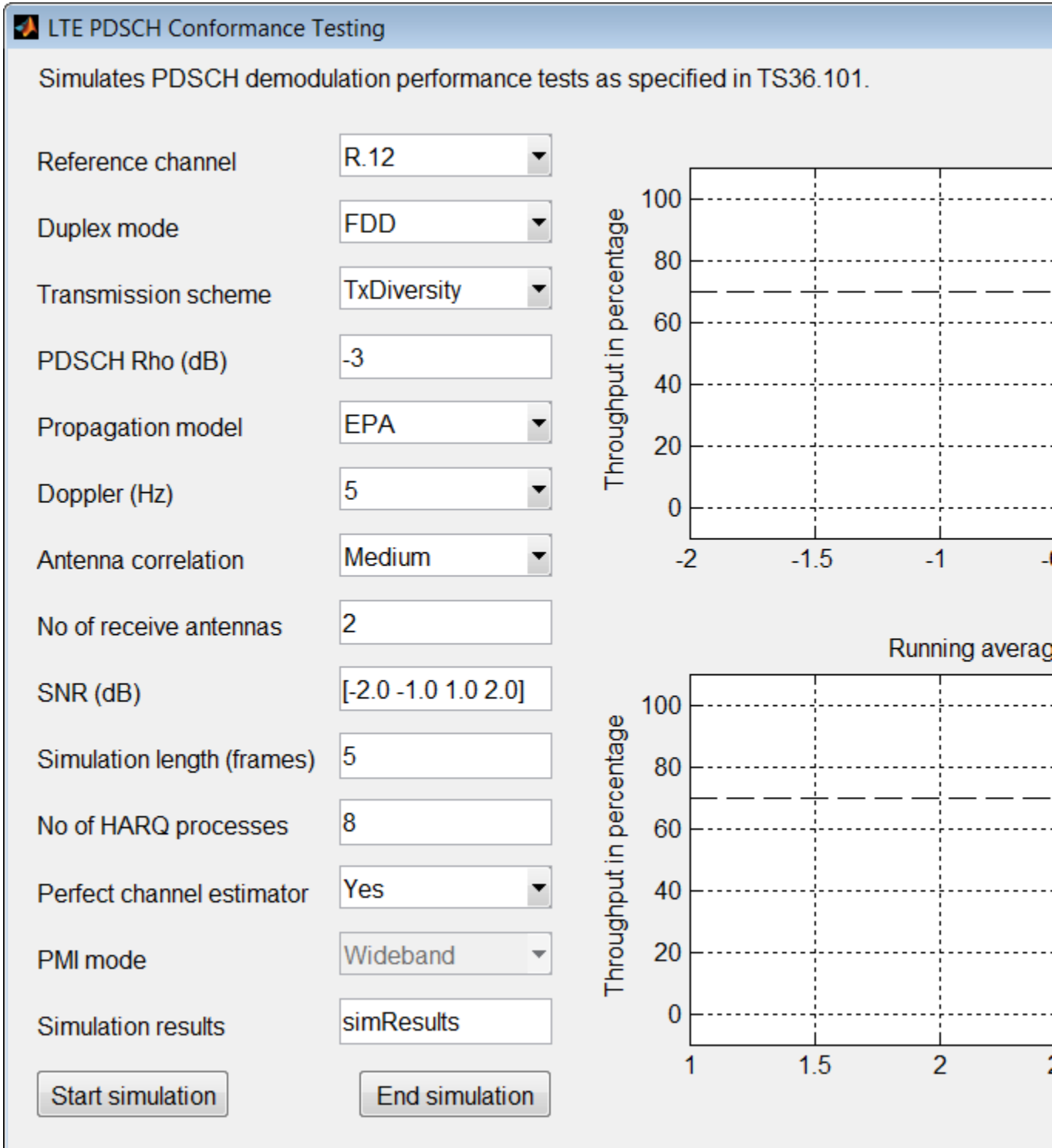
Open LTE Throughput Analyzer App

To open the LTE Throughput Analyzer app, select the **APPS** tab on the MATLAB desktop and click the following icon.

Alternatively, you can type the following command in the MATLAB command window.

```
lteDLConformanceTestTool;
```

The LTE PDSCH Conformance Testing dialog box appears.


LTE PDSCH Conformance Testing
Simulates PDSCH demodulation performance tests as specified in TS36.101.
Reference channel
R.12
Duplex mode
FDD
Transmission scheme
TxDiversity
PDSCH Rho (dB)
-3
Propagation model
EPA
Doppler (Hz)
5
Antenna correlation
Medium
No of receive antennas
2
SNR (dB)
[-2.0 -1.0 1.0 2.0]
Simulation length (frames)
5
No of HARQ processes
8
Perfect channel estimator
Yes
PMI mode
Wideband
Simulation results
simResults
Start simulation
End simulation
Throughput in percentage
100
80
60
40
20
0
-2
-1.5
-1
Running averag
Throughput in percentage
100
80
60
40
20
0
1
1.5
2

Dialog Box Inputs and Outputs

Parameters

In the The LTE PDSCH Conformance Testing dialog box, you can set the following parameters.

Parameter (Equivalent Field)	Values	Description
Reference channel (RC)	`'R0'` (default), `'R1'`, `'R2'`, `'R3'`, `'R4'`, `'R5'`, `'R6'`, `'R7'`, `'R8'`, `'R9'`, `'R10'`, `'R11'`, `'R12'`, `'R13'`, `'R14'`, `'R6-27RB'`, `'R12-9RB'`, `'R11-45RB'`, `User defined`	Reference measurement channel (RMC) number or type, as specified in TS36.101 Annex A.3. To define your own reference channel, select `User defined`. The User Defined Configuration dialog box opens. For **Configuration structure variable name**, type the name of an RC parameter structure variable in the MATLAB workspace.
Duplex mode (DuplexMode)	`'FDD'` (default), `'TDD'`	Duplex mode
Transmission scheme (TxScheme)	`'Port0'`, `'TxDiversity'`, `'CDD'`, `'SpatialMux'`, `'MultiUser'`, `'Port5'`, `'Port7-8'`, `'Port8'`, `'Port7-14'`	PDSCH transmission scheme, specified as one of the following options. • `'Port0'` — Single-antenna port, port 0 • `'TxDiversity'` — Transmit diversity scheme • `'CDD'` — Large delay CDD scheme • `'SpatialMux'` — Closed-loop spatial multiplexing

Parameter (Equivalent Field)	Values	Description
		• 'MultiUser' — Multiuser MIMO scheme • 'Port5' — Single-antenna port, port 7, if NLayers is 1. Dual layer transmission, port 7 and 8, if NLayers is 2. • 'Port7-8' — Single-antenna port, port 0 • 'Port8' — Single-antenna port, port 8 • 'Port7-14' — Up to 8-layer transmission, ports 7–14
PDSCH Rho (dB) (Rho)	Scalar number	PDSCH resource element power allocation, in dB
Propagation Model (DelayProfile)	'Off', 'EPA' (default), 'EVA', 'ETU', 'HST'	Delay profile model
Doppler (Hz) (DopplerFreq)	'5', '70', '300', '750'	*Doppler* frequency, in hertz
Antenna Correlation (MIMOCorrelation)	'Low', 'Medium', 'High'	Correlation between UE and eNodeB antennas
No of receive antennas (NRxAnts)	Nonnegative scalar integer	Number of receive antennas
SNR (dB)	Numeric vector	SNR values, in dB

Parameter (Equivalent Field)	Values	Description
Simulation length (frames)	Positive scalar integer	Simulation length, in frames
Number of HARQ processes (`NHARQProcesses`)	Nonnegative scalar integer	Number of HARQ processes
Perfect channel estimator	`'Yes'`, `'No'`	Channel estimator
PMI mode (`PMIMode`)	`'Wideband'`, `'Subband'`	PMI reporting mode
Simulation results	String beginning with an alphabetical character and containing alphanumeric characters	Simulation results output variable name. When you click **Generate waveform**, a new variable with this name is created in the MATLAB workspace.

Examples

Perform 4-by-2 Transmit Diversity Conformance Test

This example shows how to run a conformance test for a single codeword RMC R.12-9RB for the transmit diversity transmission scheme with EPA-5 fading.

Open the LTE Throughput Analyzer app. Select the **APPS** tab on the MATLAB desktop and click the following icon.

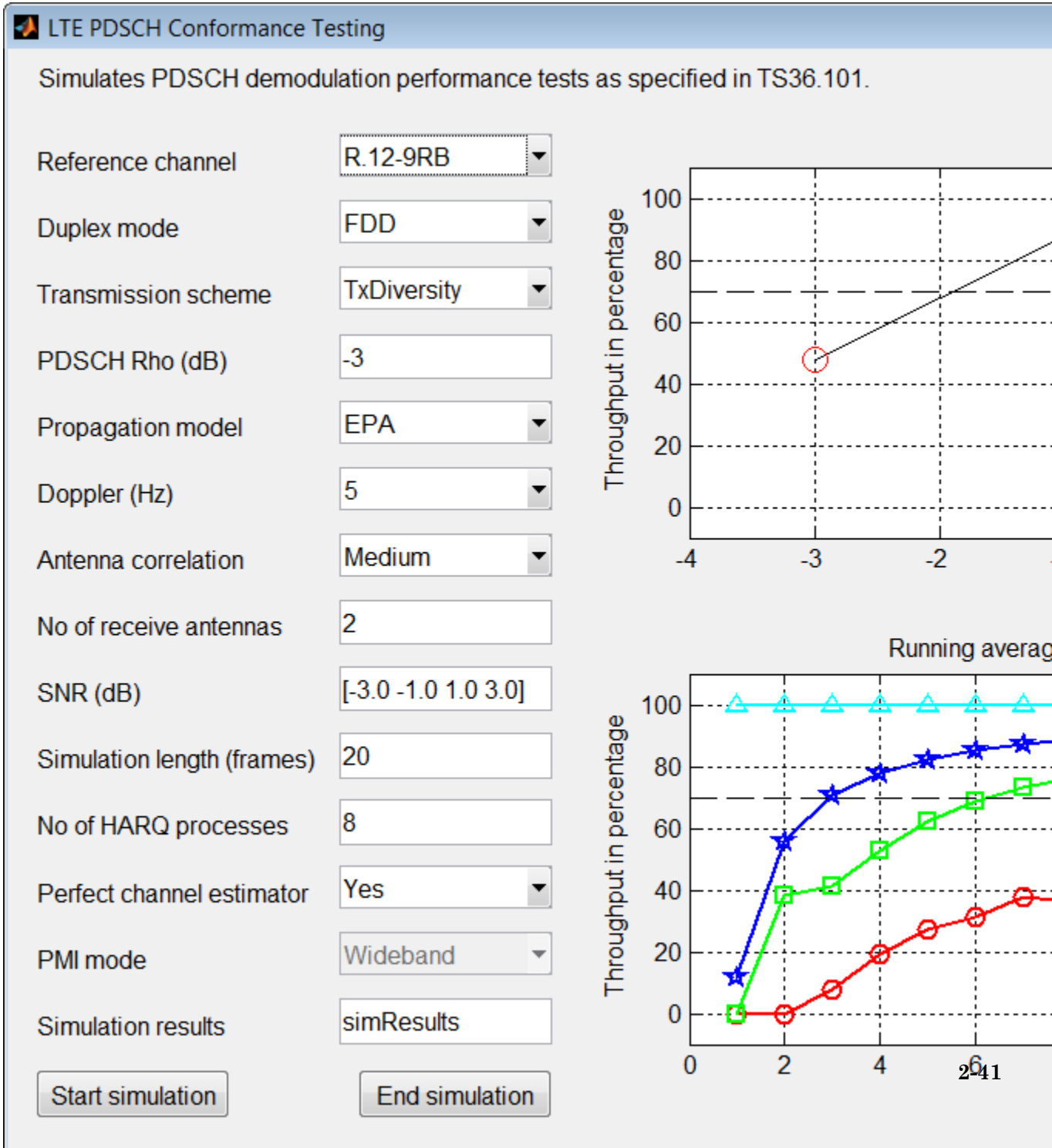
The LTE PDSCH Conformance Testing dialog box opens.

From the **Reference channel** drop-down list, choose `R.12-9RB`.

For **SNR**, enter `[-3.0 -1.0 1.0 3.0]`.

For **Simulation length (frames)**, enter `20`.

Click the **Start simulation** button. Wait a few minutes for the simulation to run. In the bottom-right corner of the window, next to **Estimated time remaining**, the tool displays an approximation of how long the simulation still needs to run. When the simulation finishes, the dialog box appears as shown in the following figure.


LTE PDSCH Conformance Testing
Simulates PDSCH demodulation performance tests as specified in TS36.101.
Reference channel
R.12-9RB
Duplex mode
FDD
Transmission scheme
TxDiversity
PDSCH Rho (dB)
-3
Propagation model
EPA
Doppler (Hz)
5
Antenna correlation
Medium
No of receive antennas
2
SNR (dB)
[-3.0 -1.0 1.0 3.0]
Simulation length (frames)
20
No of HARQ processes
8
Perfect channel estimator
Yes
PMI mode
Wideband
Simulation results
simResults
Start simulation
End simulation
Throughput in percentage
100
80
60
40
20
0
-4
-3
-2
Running averag
Throughput in percentage
100
80
60
40
20
0
0
2
4
6

The simulation result for a 20-frame run is displayed in the MATLAB Command Window.

```
Result for -3 dB SNR
Throughput: 47.65%

Result for -1 dB SNR
Throughput: 87.65%

Result for 1 dB SNR
Throughput: 95.59%

Result for 3 dB SNR
Throughput: 100.00%
```

In addition, the `simResults` variable now appears in the MATLAB workspace. Enter `simResults` to see its contents.

```
simResults

simResults =

1x4 struct array with fields:

    throughput
    tpPerFrame
    rawBER
```

Perform Customized Conformance Test with User-Defined Configuration

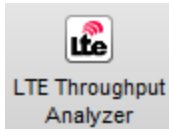
This example shows how to run a conformance test for a user-defined configuration structure. You can carry out performance analysis and testing under user-defined settings. To do so, select **'User defined'** from the "Reference channel" popup menu, which will then prompt for the configuration structure variable name. The test bench will expect this variable to be present (already defined by the user) in the 'base' workspace.

Perform the single physical resource block (PRB) RMC R.0 conformance test, except with the allocated resource block moved to the upper band edge rather than lower band edge. First, create the basic configuration structure with the

function `lteRMCDL`. Choose the most closely-matched RMC. Then, modify it with this the `PRBSet` requirement.

```
rmc = lteRMCDL('R.0');
rmc.PDSCH.PRBSet = rmc.NDLRB-1;
```

Open the LTE Throughput Analyzer app. Select the **APPS** tab on the MATLAB desktop and click the following icon.

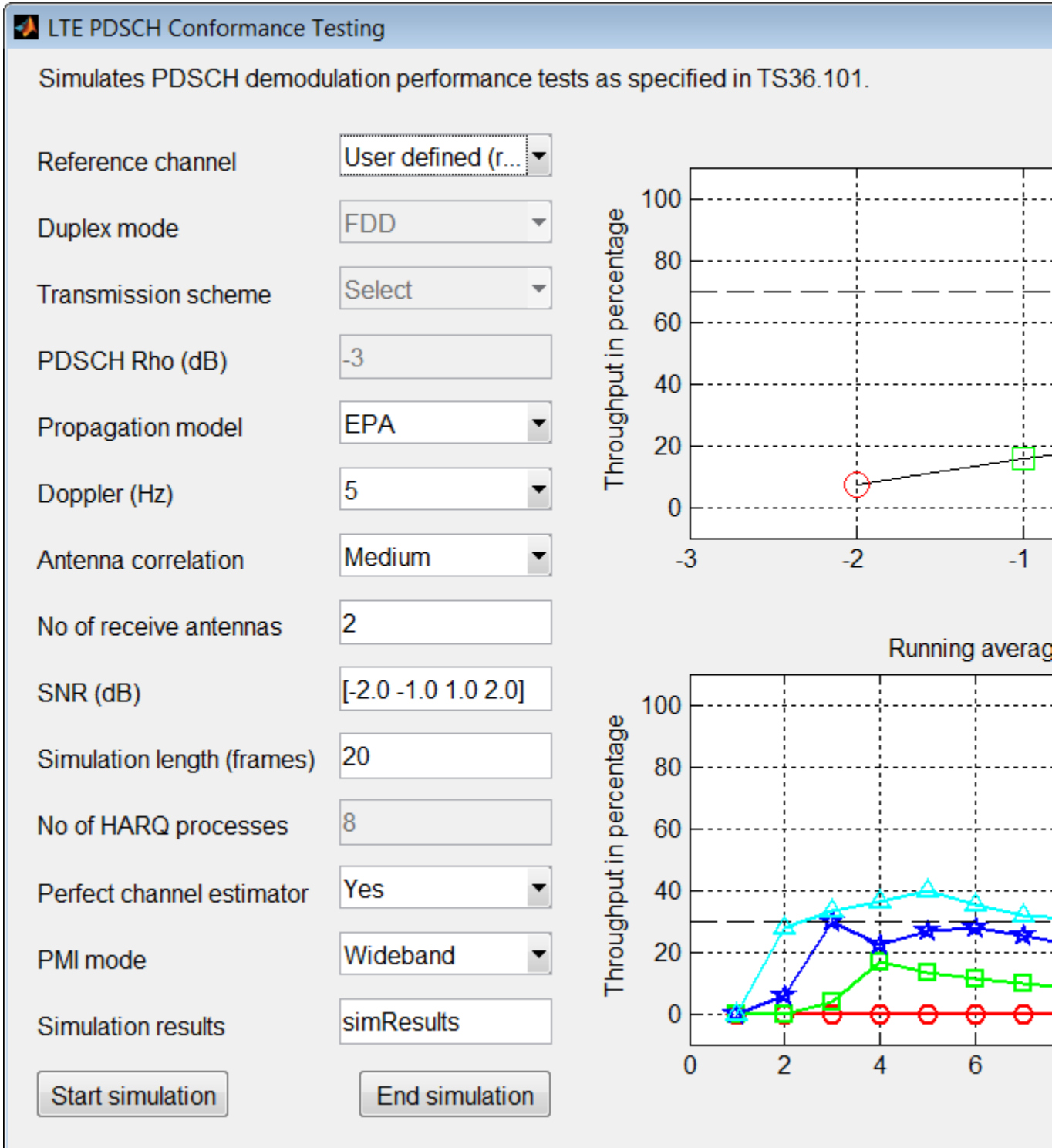


The LTE PDSCH Conformance Testing dialog box opens.

From the **Reference channel** drop-down list, choose `User defined`. The User Defined Configuration dialog box opens.

For **Configuration structure variable name**, enter `rmc`. Click **OK**.

Click the **Start simulation** button. Wait a few minutes for the simulation to run. In the bottom-right corner of the window, next to **Estimated time remaining**, the tool displays an approximation of how long the simulation still needs to run. When the simulation finishes, the dialog box appears as shown in the following figure.


LTE PDSCH Conformance Testing
Simulates PDSCH demodulation performance tests as specified in TS36.101.
Reference channel
User defined (r...
Duplex mode
FDD
Transmission scheme
Select
PDSCH Rho (dB)
-3
Propagation model
EPA
Doppler (Hz)
5
Antenna correlation
Medium
No of receive antennas
2
SNR (dB)
[-2.0 -1.0 1.0 2.0]
Simulation length (frames)
20
No of HARQ processes
8
Perfect channel estimator
Yes
PMI mode
Wideband
Simulation results
simResults
Start simulation
End simulation
Throughput in percentage
Running averag

The simulation result for a 20-frame run is displayed in the MATLAB Command Window.

```
Result for -2 dB SNR
Throughput: 7.22%

Result for -1 dB SNR
Throughput: 15.56%

Result for 1 dB SNR
Throughput: 28.33%

Result for 2 dB SNR
Throughput: 33.89%
```

In addition, the `simResults` variable now appears in the MATLAB workspace. Enter `simResults` to see its contents.

```
simResults

simResults =

1x4 struct array with fields:

    throughput
    tpPerFrame
    rawBER
```

References

[1] 3GPP TS 36.101. "User Equipment (UE) Radio Transmission and Reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`lteDLConformanceTestTool` | `lteRMCDLTool` | `lteRMCULTool` | `lteTestModelTool`

Concepts

- "Generate LTE Downlink RMC Waveforms" on page 2-8
- "Generate LTE Uplink RMC Waveforms" on page 2-18

- “Generate LTE Test Model Waveforms” on page 2-27

3

LTE Physical Layer Examples

- "Create Synchronization Signals" on page 3-2
- "Model CFI and PCFICH" on page 3-4
- "Model HARQ Indicator and PHICH" on page 3-7
- "Model DCI and PDCCH" on page 3-11
- "Model PUCCH Format 1" on page 3-15
- "Model PUCCH Format 2" on page 3-17
- "Model DL-SCH and PDSCH" on page 3-19
- "Model UL-SCH and PUSCH" on page 3-25
- "Simulate Propagation Channels" on page 3-27
- "Find Channel Impulse Response" on page 3-30

Create Synchronization Signals

This example shows how to construct synchronization signals using the LTE System Toolbox product. In this example, you create the primary and secondary synchronization signals and map them to a resource grid.

Set up the cell-wide settings. Create a structure and specify the cell-wide settings as its fields.

```
enb.NDLRB = 9;
enb.CyclicPrefix = 'Normal';
enb.CellRefP = 1;
enb.NCellID = 1;
enb.NSubframe = 0;
enb.DuplexMode = 'FDD';
```

Many of the functions used in this example require a subset of the preceding settings specified.

Generate the PSS symbols. Use the function `ltePSS` with the *eNodeB* configuration, `enb`.

```
pss = ltePSS(enb);
```

When a PSS signal is not located in `enb.NSubframe`, the function does not generate PSS symbols and returns an empty vector.

Next, generate the PSS indices. These indices map the PSS complex symbols to the subframe resource grid. Use the function `ltePSSIndices`, the eNodeB configuration, and the antenna number. In this case, since only one antenna port is used, `antenna` is set to 0.

```
antenna = 0;
pssIndices = ltePSSIndices(enb, antenna);
```

In this example, you generate subframe 0. Since subframe 0 contains a PSS signal, the function generates PSS indices. If `enb.NSubframe` is a subframe that does not contain a PSS signal, the function would return an empty vector.

Generate the SSS symbols. Use the function `lteSSS` with the eNodeB configuration, `enb`.

```
sss = lteSSS(enb);
```

When an SSS signal is not located in `enb.NSubframe`, the function does not generate SSS symbols. It returns an empty vector.

Next, generate the SSS indices. These indices map the SSS complex symbols to the subframe resource grid. Call the function `lteSSSIndices`, providing the eNodeB configuration and the antenna port number.

```
antenna = 0;
sssIndices = lteSSSIndices(enb, antenna);
```

In this example, you generate subframe 0. Since subframe 0 contains an SSS signal, the function generates SSS indices. If `enb.NSubframe` is a subframe that does not contain an SSS signal, the function returns empty indices.

Generate the subframe resource grid. Use the function `lteDLResourceGrid`. You create an empty resource grid for one subframe.

```
subframe = lteDLResourceGrid(enb);
```

Finally, map the PSS and SSS symbols directly to the resource grid using the generated indices.

```
subframe(pssIndices) = pss;
subframe(sssIndices) = sss;
```

See Also

`ltePSS` | `ltePSSIndices` | `lteSSS` | `lteSSSIndices` | `lteDLResourceGrid` | `lteZadoffChuSeq`

Concepts

- “Synchronization Signals (PSS and SSS)” on page 1-11

Model CFI and PCFICH

This example shows how to generate a PCFICH with the LTE System Toolbox product. First, create a CFI based on the eNodeB configuration and code it. Then, generate a PCFICH using the coded CFI and map it to a resource grid.

Set up the cell-wide settings. Create a structure and specify the cell-wide settings as its fields.

```
enb.NDLRB = 9;
enb.CyclicPrefix = 'Normal';
enb.PHICHDuration = 'Normal';
enb.CFI = 3;
enb.CellRefP = 4;
enb.NCellID = 1;
enb.NSubframe = 0;
```

Many of the functions used in this example require a subset of the preceding settings specified.

Create a resource grid. Call the `lteDLResourceGrid` function. This function creates an empty resource grid for one subframe.

```
subframe = lteDLResourceGrid(enb);
```

The resulting subframe is a 3 dimensional matrix. The number of rows represents the number of subcarriers available, 12×`enb.NDLRB`, since there are 12 subcarriers per resource block. The number of columns corresponds to the number of OFDM symbols in a subframe, 7×2, since there are 7 OFDM symbols per slot for normal cyclic prefix and there are 2 slots in a subframe. The third dimension of the matrix corresponds to the number of transmit antenna ports used. There are four specified in the example, so `enb.CellRefP` is 4.

Code the CFI channel. Call the `lteCFI` function. The result, `cfiCodedBits`, is a 32 bit long set of coded bits.

```
cfiCodedBits = lteCFI(enb);
```

As described earlier, the number of OFDM symbols used to transmit the control information in a subframe is defined by the CFI value. The `eNodeB`

configuration structure assigns the CFI a value of 3. Thus, 4 OFDM symbols are used for the control region because the number of resource blocks used is less than 11, since `enb.NDLRB` is 9.

Generate PCFICH complex symbols. Call the `ltePCFICH` function to scramble the CFI coded bits, modulate using QPSK, map to layers, and precode to form the PCFICH complex symbols.

```
pcfichSymbols = ltePCFICH(enb,cfiCodedBits);
```

The resulting matrix, `pcfichSymbols`, has 4 columns. Each column contains the PCFICH complex symbols that map to each of the antenna ports.

Generate the PCFICH mapping indices. These indices map the PCFICH complex symbols to the subframe resource grid. Call the function `ltePCFICHIndices`.

```
pcfichIndices = ltePCFICHIndices(enb,'1based');
```

The resulting matrix, `pcfichIndices`, has 4 columns. Each column contains the indices in linear form for each antenna port. These indices are 1-based, since MATLAB uses 1-based indices. However, you can also generate 0-based indices.

Map the PCFICH complex symbols to the subframe resource grid using the appropriate mapping indices. The linear indexing style used makes the mapping process straightforward.

```
subframe(pcfichIndices) = pcfichSymbols;
```

The resulting matrix, `subframe`, contains the complex symbols in `pcfichSymbols` in the locations specified by `pcfichIndices`.

View the resource usage with the `ltePCFICHInfo` function. This function returns the number of resource elements, `NRE`, and the number of resource element groups, `NREG`, used by the PCFICH within the structure, `info`.

```
info = ltePCFICHInfo;
```

The resulting structure, `info`, contains the fields `NRE`, the number of resource elements, and `NREG`, the number of resource element groups, used by the PHICH.

See Also

`lteCFI` | `ltePCFICH` | `ltePCFICHInfo` | `ltePCFICHIndices` | `lteDLResourceGrid` | `ltePCFICHPRBS` | `lteSymbolModulate` | `lteSymbolDemodulate` | `lteLayerMap` | `lteLayerDemap` | `lteDLPrecode` | `lteDLDeprecode`

Concepts

- “Control Format Indicator (CFI) Channel” on page 1-32

Model HARQ Indicator and PHICH

This example shows how to implement the HARQ Indicator (HI) and physical HI channel (PHICH). You create the processing chain of coding hybrid indicator values, create the PHICH, and map it to a resource grid.

Set up the cell-wide settings. Create a structure and specify the cell-wide settings as its fields.

```
enb.NDLRB = 9;
enb.CyclicPrefix = 'Normal';
enb.PHICHDuration = 'Normal';
enb.Ng = 'Sixth';
enb.CellRefP = 4;
enb.NCellID = 1;
enb.NSubframe = 0;
enb.DuplexMode = 'FDD';
```

Many of the functions used in this example require a subset of the preceding settings specified.

Generate PHICH resource information. To do so, call the function `ltePHICHInfo`.

```
phichInfo = ltePHICHInfo(enb);
```

The function returns `phichInfo`, a structure containing the relevant data required to define PHICH sets. The elements and values of the structure are shown in the following table.

Structure Element	Description	Value
`NREG`	The number of resource element groups used to map the PHICHs	3
`NRE`	The number of resource elements used to map the PHICHs	12
`NPHICH`	The maximum number of PHICH that can be used	8

Structure Element	Description	Value
`NGroups`	The maximum number of PHICH groups that can be used	1
`NMappingUnits`	The number of PHICH mapping units used to map the maximum number of PHICH groups	1
`NSequences`	The maximum number of orthogonal sequences that can be used within each group	8
`PHICHDuration`	The number of OFDM symbols used to map the PHICH	1

Generate the HARQ indicator (HI) set. An HI set consists of a HARQ indicator value, 1 for ACK and 0 for NACK, and a PHICH index pair that contains the PHICH group index, (n_{PHICH}^{group}), and the orthogonal sequence index, (n_{PHICH}^{seq}), for the PHICH containing HI. The values of n_{PHICH}^{group} and n_{PHICH}^{seq} can be determined using the PHICH resource dimension information returned by `ltePHICHInfo`. The number of groups determines acceptable values of the PHICH group index and the number of sequences determines acceptable values of sequence indexes.

```
HISet = [ [0 0 1]; ...
          [0 1 0]; ...
          [0 4 0]; ...
          [0 7 1]; ];
```

In this example, you create one PHICH group containing four PHICHs with the indices shown in the following table.

PHICH Group Index	PHICH Sequence Index	HARQ Indicator Value
0	0	1 — ACK
0	1	0 — NACK

PHICH Group Index	PHICH Sequence Index	HARQ Indicator Value
0	4	0 — NACK
0	7	1 — ACK

In the LTE System Toolbox product, a HI set matrix is used to define the HI and PHICH index pair for each HI within the subframe. The HI set matrix defines a single PHICH in terms of $\left(n_{PHICH}^{group} \quad n_{PHICH}^{seq} \quad \text{HARQ Indicator}\right)$.

Generate the PHICH complex symbols. You generate them from the eNodeB configuration and HARQ indicator matrix. Call the function `ltePHICH` to perform the channel coding, modulation, scrambling, layer mapping, precoding, and combining of the PHICH groups. The result is the complex symbols that you can map to the resource grid.

```
phichSymbols = ltePHICH(enb,HISet);
```

The resulting vector, `phichSymbols`, has 12 rows and 4 columns. Each column contains the complex symbols to be mapped to the resource grids for each of the four antenna ports.

Generate the PHICH mapping indices. To do so, call the function `ltePHICHIndices`. These mapping indices are required to map the complex PHICH symbols to the subframe resource grid.

```
phichIndices = ltePHICHIndices(enb);
```

This function returns a matrix with 4 columns, one column for each antenna port. The rows contain the indices in linear form for mapping the PHICH symbols to the subframe resource grid.

Generate a subframe resource grid. To do so, call the function `lteDLResourceGrid`. This function creates an empty resource grid for one subframe.

```
subframe = lteDLResourceGrid(enb);
```

Map the PHICH to the resource grid. You can easily map the complex PHICH symbols to each of the resource grids for each antenna port.

```
subframe(phichIndices) = phichSymbols;
```

See Also

`ltePHICH` | `ltePHICHInfo` | `ltePHICHIndices` | `ltePHICHPRBS` | `lteDLResourceGrid` | `lteLayerMap` | `lteLayerDemap` | `lteDLPrecode` | `lteDLDeprecode` | `lteCRCEncode` | `lteCRCDecode` | `lteSymbolModulate` | `lteSymbolDemodulate`

Concepts

- "HARQ Indicator (HI) Channel" on page 1-41

Model DCI and PDCCH

This example shows how to model the control region used in an LTE downlink subframe and its channel structure. It demonstrates how you create a DCI message, encode it, create the PDCCH, and map it to a resource grid.

Set up the cell-wide settings. Create a structure and specify the cell-wide settings as its fields.

```
enb.NDLRB = 9;
enb.CyclicPrefix = 'Normal';
enb.PHICHDuration = 'Normal';
enb.CFI = 3;
enb.Ng = 'Sixth';
enb.CellRefP = 1;
enb.NCellID = 1;
enb.NSubframe = 0;
enb.DuplexMode = 'FDD';
```

Many of the functions used in this example require a subset of the preceding settings specified.

Set up the DCI message structure.

```
dci.NDLRB = enb.NDLRB;
dci.DCIFormat = 'Format1A';
dci.Allocation.RIV = 26;
dci.DuplexMode = 'FDD';
dci.NTxAnts = 1;
```

The DCI message contains the following parameters.

- `NDLRB` — number of downlink resource blocks (RBs)
- `DCIFormat` — DCI format, selected from those discussed in "DCI Message Formats" on page 1-57
- `Allocation.RIV` — resource indication value (RIV)
- `DuplexMode` — transmission frame structure type — `'FDD'` for frame structure type 1 or `'TDD'` for frame structure type 2

- `NTxAnts` — number of transmit antennas

The RIV indicates the contiguous RB allocations for a UE. The UE uses the RIV to determine the first virtual RB and the length of contiguous allocation of RBs. In this example, an RIV setting of 26 corresponds to full bandwidth assignment.

Generate a DCI message. To do so, call the function `lteDCI`. You can map this generated message to the PDCCH.

```
[dciMessage,dciMessageBits] = lteDCI(enb,dci);
```

The function `lteDCI` returns a structure, `dciMessage`, and a vector containing the DCI message bits, `dciMessageBits`. Both outputs contain the same information, but are ideal for different purposes. The output structure is more readable, while the serialized DCI message is a more suitable format to send to the channel coding stage.

Set up the PDCCH configuration structure. The channel coding stages require the parameters shown in the following list.

- number of downlink resource blocks (RBs)
- UE-specific mask (16-bit C-RNTI value)
- PDCCH format

```
pdcch.RNTI = 100;
pdcch.PDCCHFormat = 0;
```

Channel encode the DCI message bits. This process consists of the addition of a CRC attachment, convolutional coding, and rate matching according to the PDCCH format capacity.

```
codedDciBits = lteDCIEncode(pdcch,dciMessageBits);
```

The resulting vector, `codedDciBits`, has 72 elements.

Generate PDCCH bits. The encoded DCI messages are then assigned to CCEs, as discussed in “Matching PDCCHs to CCE Positions” on page 1-64. The capacity of the control region depends on the bandwidth, the CFI, the

number of antenna ports and the HICH groups. The total number of resources available for PDCCH can be calculated using the function `ltePDCCHInfo`.

```
pdcchInfo = ltePDCCHInfo(enb);
```

This function returns a structure, `pdcchInfo`, which contains the resources available to the PDCCH in different units (one per field): bits, CCEs, REs and REGs.

The total number of bits available in the PDCCH region can be found in the field `pdcchInfo.MTot`. This allows a vector to be built with the appropriate number of elements.

```
pdcchBits = -1*ones(1,pdcchInfo.MTot);
```

Not all the available bits in the PDCCH region are necessarily used. Therefore, following the convention in the LTE System Toolbox product, set unused bits to –1. Since all elements have been initialized in `pdcchBits` to –1, this indicates that initially all the bits are unused. Now elements of `codedDciBits` can be mapped to the appropriate locations in `pdcchBits`.

Calculate indices of candidate bits. To do so, call the function `ltePDCCHSpace`. Only a subset of all the bits in `pdcchBits` may be used, which are called the candidate bits.

```
candidates = ltePDCCHSpace(enb,pdcch,{'bits','1based'});
```

This function returns a two-column matrix. Each row contains an available candidate location for the cell-wide settings provided by `enb` and the PDCCH configuration structure `pdcch`. The first and second columns contain the indices of the first and last locations of each candidate. In this example, the indices are 1-based and refer to bits. Hence, they can be used to access locations in `pdcchBits`.

Use the first available candidate to map the coded DCI bits.

```
pdcchBits (candidates(1,1):candidates(1,2)) = codedDciBits;
```

The vector `pdcchBits` has 736 elements. The 72 bits of `codedDciBits` are mapped to the chosen candidate in `pdcchBits`. Therefore, out of 736 elements, 72 will take 0 and 1 values, while the rest remain set to –1. The function `ltePDCCH`, which is used to generate complex-modulated symbols, will

interpret these locations as unused and will only consider those containing 1s and 0s.

Generate PDCCH complex symbols. To do so, call the function `ltePDCCH`. You can generate the PDCCH complex symbols from the set of bits used in `pdcchBits`, values not set to –1. The function performs the required scrambling, QPSK modulation, layer mapping, and precoding operations.

```
pdcchSymbols = ltePDCCH(enb, pdcchBits);
```

Generate PDCCH mapping indices. To do so, call the function `ltePDCCHIndices`. You can use these indices to map the complex values in `pdcchSymbols` to the subframe resource grid.

```
pdcchIndices = ltePDCCHIndices(enb,{'1based'});
```

This function returns a column vector. The rows contain the indices in linear form for mapping the PDCCH symbols to the subframe resource grid.

Map the PDCCH to the resource grid. You can easily map the complex PDCCH symbols to each of the resource grids for each antenna port.

```
subframe = lteDLResourceGrid(enb);
subframe(pdcchIndices) = pdcchSymbols;
```

See Also

`lteDCI` | `lteDCIEncode` | `ltePDCCH` | `ltePDCCHIndices` | `ltePDCCHSpace` | `ltePDCCHInfo` | `ltePDCCHPRBS` | `ltePDCCHInterleave` | `ltePDCCHDeinterleave` | `ltePDCCHDecode` | `lteLayerMap` | `lteLayerDemap` | `lteDLPrecode` | `lteDLDeprecode` | `lteCRCEncode` | `lteCRCDecode` | `lteConvolutionalEncode` | `lteConvolutionalDecode` | `lteRateMatchConvolutional` | `lteRateRecoverConvolutional` | `lteSymbolModulate` | `lteSymbolDemodulate`

Concepts

- “Downlink Control Channel” on page 1-57

Model PUCCH Format 1

This example shows how to model the control region used in an LTE uplink subframe and its channel structure. It demonstrates how you create the physical uplink control channel (PUCCH) format 1 structures and map the generated symbols to a resource grid.

Setup user-equipment (UE) settings. Define UE settings in a structure.

```
ue1.NCellID = 10;
ue1.CyclicPrefixUL = 'Normal';
ue1.NSubframe = 0;
ue1.Hopping = 'Off';
ue1.NULRB = 9;
ue1.Shortened = 0;
```

Many of the functions used in this example require a subset of the preceding settings specified.

Configure the PUCCH. In addition to the UE settings specified in `ue`, you must define parameters related to the physical channel to generate the PUCCH Format 1.

```
pucch1.ResourceIdx = 0;
pucch1.DeltaShift = 1;
pucch1.CyclicShifts = 6;
```

Generate the PUCCH Format 1 symbols. To do so, call the function `ltePUCCH1`, providing the UE and PUCCH configuration structures, `ue` and `pucch1`, as input arguments.

```
hi = [0 0];
pucch1symbols = ltePUCCH1(ue1,pucch1,hi);
```

The variable `hi` specifies the HARQ Indicator bits.

Generate the PUCCH Format 1 indices. To do so, call the function `ltePUCCH1Indices`. You can use these generated indices to map the PUCCH complex symbols to the subframe resource grid. This function requires the same input argument structures as `ltePUCCH1`.

```
pucch1indices = ltePUCCH1Indices(ue1,pucch1);
```

Generate the PUCCH Format 1 demodulation reference signals (DRS). To do so, call the function `ltePUCCH1DRS`. This function requires the same input argument structures as `ltePUCCH1` and `ltePUCCH1Indices`.

```
drs1 = ltePUCCH1DRS(ue1,pucch1);
```

Generate the PUCCH Format 1 DRS indices. To do so, call the function `ltePUCCH1DRSIndices`. These indices map the DRS to the subframe resource grid.

```
drs1indices = ltePUCCH1DRSIndices(ue1,pucch1);
```

Generate the subframe resource grid. To do so, call the function `lteULResourceGrid`. This function creates an empty resource grid for one subframe.

```
subframe = lteULResourceGrid(ue1);
```

Map the PUCCH Format 1 symbols and DRS to the resource grid, using the generated indices.

```
subframe(pucch1indices) = pucch1symbols;
subframe(drs1indices) = drs1;
```

See Also

`ltePUCCH1` | `ltePUCCH1Indices` | `ltePUCCH1DRS` | `ltePUCCH1DRSIndices` | `lteULResourceGrid`

Concepts

- "Uplink Control Channel Format 1" on page 1-81

Model PUCCH Format 2

This example shows how to model the control region used in an LTE uplink subframe and its channel structure. It demonstrates how you create the physical uplink control channel (PUCCH) format 2 structures and map the generated symbols to a resource grid.

Setup user-equipment (UE) settings. Define UE settings in a structure.

```
ue.NCellID = 10;
ue.CyclicPrefixUL = 'Normal';
ue.NSubframe = 0;
ue.Hopping = 'Off';
ue.NULRB = 9;
ue.RNTI = 77;
```

Many of the functions used in this example require a subset of the preceding settings specified.

Configure the PUCCH Format 2. In addition to the UE settings specified in `ue`, you must define parameters related to the physical channel to generate the PUCCH Format 2.

```
pucch2.ResourceIdx = 36;
pucch2.ResourceSize = 3;
pucch2.CyclicShifts = 6;
```

Generate the UCI message from the CQI bits.

```
cqi = [0 1 1 0 0 1];
codedcqi = lteUCIEncode(cqi);
```

Generate the PUCCH Format 2 symbols. To do so, call the function `ltePUCCH2`, providing the UE and PUCCH configuration structures, `ue` and `pucch2`, as input arguments.

```
pucch2Sym = ltePUCCH2(ue,pucch2,codedcqi);
```

Generate the PUCCH Format 2 indices. To do so, call the function `ltePUCCH2Indices`. You can use these generated indices to map the PUCCH

complex symbols to the subframe resource grid. This function requires the same input argument structures as `ltePUCCH2`.

```
pucch2Indices = ltePUCCH2Indices(ue,pucch2);
```

Generate the PUCCH Format 2 demodulation reference signals (DRS). To do so, call the function `ltePUCCH2DRS`. This function requires the same input argument structures as `ltePUCCH2` and `ltePUCCH2Indices`. Since 0 HARQ bits are transmitted, pass an empty vector into the 3rd input argument of the function.

```
pucch2DRSSym = ltePUCCH2DRS(ue,pucch2,[]);
```

Generate the PUCCH Format 2 DRS indices. To do so, call the function `ltePUCCH2DRSIndices`. You can use these indices to map the DRS to the subframe resource grid.

```
pucch2DRSIndices = ltePUCCH2DRSIndices(ue,pucch2);
```

Generate the subframe resource grid. To do so, call the function `lteULResourceGrid`. This function creates an empty resource grid for one subframe.

```
subframe = lteULResourceGrid(ue);
```

Map the PUCCH Format 2 symbols and DRS to the resource grid, using the generated indices.

```
subframe(pucch2Indices) = pucch2Sym;
subframe(pucch2DRSIndices) = pucch2DRSSym;
```

See Also

`ltePUCCH2` | `ltePUCCH2Indices` | `ltePUCCH2DRS` | `ltePUCCH2DRSIndices` | `lteULResourceGrid`

Concepts

- “Uplink Control Channel Format 2” on page 1-91

Model DL-SCH and PDSCH

This example shows how to construct the physical downlink shared channel (PDSCH). It demonstrates how to generate a transport block, perform downlink shared channel (DL-SCH) coding to create a codeword, perform physical channel coding to create the physical channel, and map the complex symbols to the resource grid.

Set up the cell-wide settings. Create a structure and specify the cell-wide settings as its fields.

```
enb.NDLRB = 9;
enb.CyclicPrefix = 'Normal';
enb.PHICHDuration = 'Normal';
enb.CFI = 3;
enb.Ng = 'Sixth';
enb.CellRefP = 4;
enb.NCellID = 1;
enb.NSubframe = 0;
enb.DuplexMode = 'FDD';
```

Many of the functions used in this example require a subset of the preceding settings specified.

Configure the PDSCH. In addition to the cell-wide settings specified in `enb`, you must define other parameters related to the modulation and channel transmission configuration, `pdsch`, such as the radio network temporary identifier (RNTI), to generate the PDSCH.

```
pdsch.NTxAnts = 4;
pdsch.NLayers = 4;
pdsch.TxScheme = 'TxDiversity';
pdsch.Modulation = {'QPSK'};
pdsch.RV = 0;
pdsch.RNTI = 1;
```

In this example, you use a single codeword to form the PDSCH symbols. However, in LTE, up to two codewords can be combined to form the PDSCH. Each codeword can be modulated with a different scheme. You use a cell array to indicate the modulation scheme for each codeword.

Allocate the physical resource blocks (PRBs). This allocation is required to determine to which resource elements the PDSCH is mapped. A column vector containing the indices of PDSCH allocated PRBs is required. In this example, assume full allocation; all resource blocks are allocated to the PDSCH. Specify this full subframe resource allocation using a column vector.

```
prbs = (0:enb.NDLRB-1).';
```

The allocation specified in `prbs` is 0-based. In this case, assume that both slots in the subframe share the same resource allocation. To have different allocations for each slot, specify a two column matrix where each column refers to each slot in the subframe.

Generate the PDSCH indices. To do so, call the function `ltePDSCHIndices`. Provide the cell-wide settings structure, `enb`, the channel transmission configuration, `pdsch`, and the physical resource block allocation, `prbs`, as input arguments.

```
[pdschIndices,pdschIndInfo] = ltePDSCHIndices(enb,pdsch,prbs,{'1based'});
```

The first output, `pdschIndices`, specifies the PDSCH indices. The second output, `pdschIndInfo`, provides additional information related to the PDSCH capacity.

Determine DL-SCH payload and coded transport block size. These items are required for creating the PDSCH payload due to the rate matching portion of the DL-SCH transport block coding. There are the following two methods of determining coded transport block size and the DL-SCH payload size.

- Using the PDSCH indices information output, as shown in this example
- Using the reference measurement channel (RMC) transport block sizes as a guide

The coded transport block size is one of the fields of the PDSCH indices information output, `pdschIndInfo`.

```
codedTrBlkSize = pdschIndInfo.G;
```

In this example, `codedTrBlkSize` is 480. Alternatively, you could read the coded transport block size for a given modulation scheme, PRB allocation, and number of antennas from the RMC tables in annex A.3.3 and A.3.4 of [1].

Once you know the coded transport block size, calculate the DL-SCH payload using the rules in annex A.2.1.2 of [1], titled, "Determination of payload size," with target code rate, R, equal to 1/3, and the number of bits per subframe given by `codedTrBlkSize`. Determine the payload size, A, such that the resulting coding rate is as close as possible to the desired coding rate, R, for a given coded transport block size, N_{ch}, as shown in the following equation.

$$\min\left|R-(A+24)/N_{ch}\right|$$

In this example, the payload size for 6 RBs calculated using the preceding equation is A=152. This is the value at which the error between the desired code rate and actual code rate is minimized.

The payload size, A, must be one of a specific set for a specific number of resource blocks given by table 7.1.7.2.1–1 or 7.1.7.2.2–1 in section 7.1.7.2 of [2]. These tables are represented by the `lteTBS` function. In this example, the payload size for 6 RBs that minimizes the error between the desired code rate and the actual code rate is A=152. This value was selected from table 7.1.7.2.2–1. Therefore, the payload size, `transportBlkSize`, is 152.

Alternate Method — Use RMCs to Determine Transport Block Sizes

Alternatively, you could determine suitable payload size and coded transport block size from the tables in annex A.3.3 and A.3.4 of [1], titled "Reference Measurement Channels for PDSCH performance requirements." Despite the advantage of simply being able to read values from the tables, the channel bandwidths and PDSCH allocations are restricted to the RMCs available. For example, you can use Table A.3.3.2.2-1, titled "Fixed Reference Channel four antenna ports."

To generate a PDSCH for a 1.4MHz channel bandwidth, 4-antenna transmission with QPSK modulation, and a coding rate of 1/3, use the highlighted rows titled "Information Bit Payload" to find the DL-SCH payload size for each subframe, and "Binary Channel Bits," to find the coded transport block size for each subframe.

Parameter	Unit	Value					
Reference Channel		R.12 FDD	R.13 FDD	R.14 FDD	R.14-1 FDD	R.14-2 FDD	R.36 FDD
Channel bandwidth	MHz	1.4	10	10	10	10	10
Allocated resource blocks		6	50	50	6	3	50
Allocated subframes per Radio Frame		9	9	9	8	8	9
Modulation		QPSK	QPSK	16QAM	16QAM	16QAM	64QAM
Target Coding Rate		1/3	1/3	1/2	1/2	1/2	1/2
Information Bit Payload							
For Sub-Frames 1, 2, 3, 4, 6, 7, 8, and 9	Bits	408	4392	12960	1544	744	18336
For Sub-Frame 5	Bits	n/a	n/a	n/a	n/a	n/a	n/a
For Sub-Frame 0	Bits	152	3264	11448	n/a	n/a	18336
Number of Code Blocks							
For Sub-Frames 1, 2, 3, 4, 6, 7, 8, and 9		1	1	3	1	1	3
For Sub-Frame 5		n/a	n/a	n/a	n/a	n/a	n/a
For Sub-Frame 0		1	1	2	n/a	n/a	3
Binary Channel Bits per Sub-Frame							
For Sub-Frames 1, 2, 3, 4, 6, 7, 8, and 9	Bits	1248	12800	25600	3072	1536	38400
For Sub-Frame 5	Bits	n/a	n/a	n/a	n/a	n/a	n/a
For Sub-Frame 0	Bits	480	12032	24064	n/a	n/a	36096
Max. Throughput averaged over 1 frame	Mbps	0.342	3.876	11.513	1.235	0.595	16.502
UE Category		≥ 1	≥ 1	≥ 2	≥ 1	≥ 1	≥ 2

Define a transport block of information bits, using the payload size, `transportBlkSize`, calculated in the last step.

```
dlschTransportBlk = round(rand(1,152));
```

Create the PDSCH payload. To encode the transport block bits into a single codeword, call the function `lteDLSCH` . This process includes CRC calculation, code block segmentation and CRC insertion, turbo coding, rate matching, and code block concatenation. Use the cell-wide settings, `enb`, and PDSCH parameters, `pdsch`, defined in previous steps.

```
codeword = lteDLSCH(enb,pdsch,codedTrBlkSize,dlschTransportBlk);
```

Generate the PDSCH complex symbols. To do so, call the `ltePDSCH` function. This process applies scrambling, modulation, layer mapping, and precoding operations to the coded transport block. Use the cell-wide settings, `enb`, and PDSCH parameters, `pdsch`, defined in previous steps.

```
pdschSymbols = ltePDSCH(enb,pdsch,codeword);
```

The resulting matrix, `pdschSymbols`, has 4 columns. Each column contains the complex symbols to map to each antenna port.

Generate the subframe resource grid. To do so, call the function `lteDLResourceGrid`. This function creates an empty resource grid for one subframe.

```
subframe = lteDLResourceGrid(enb);
```

Map the PDSCH symbols to the resource grid, using the generated indices.

```
subframe(pdschIndices) = pdschSymbols;
```

References

[1] 3GPP TS 36.101. "User Equipment (UE) Radio Transmission and Reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

[2] 3GPP TS 36.213. "Physical layer procedures." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

See Also

`lteDLSCH` | `ltePDSCH` | `ltePDSCHIndices` | `lteDLResourceGrid` | `lteDLSCHInfo` | `lteDLSCHDecode` | `lteLayerMap` | `lteLayerDemap` | `lteDLPrecode` | `lteDLDeprecode` | `lteTurboEncode` | `lteTurboDecode` | `ltePDSCHPRBS` | `lteCRCEncode` | `lteCRCDecode` | `lteCodeBlockSegment` | `lteCodeBlockDesegment` | `lteRateMatchTurbo` | `lteRateRecoverTurbo`

Concepts

- "Downlink Shared Channel" on page 1-104

Model UL-SCH and PUSCH

This example shows how to construct the physical uplink shared channel (PUSCH). It demonstrates how to generate a transport block, perform uplink shared channel (UL-SCH) coding to create a codeword, perform physical channel coding to create the physical channel, and map the complex symbols to the resource grid.

Setup user-equipment (UE) settings. Define UE settings in a structure.

```
ue.NULRB = 9;
ue.NSubframe = 0;
ue.NCellID = 10;
ue.RNTI = 1;
ue.CyclicPrefixUL = 'Normal';
ue.Hopping = 'Off';
ue.SeqGroup = 0;
ue.CyclicShift = 0;
ue.Shortened = 0;
```

Many of the functions used in this example require a subset of the preceding settings specified.

Configure the PUSCH. In addition to the UE settings specified in `ue`, you must define parameters related to the physical channel to generate the PUSCH.

```
pusch.PRBSet = (0:5).';
pusch.Modulation = 'QPSK';
pusch.RV = 0;
pusch.DynCyclicShift = 0;
```

Generate the subframe resource grid. To do so, call the function `lteULResourceGrid`. This function creates an empty resource grid for one subframe.

```
subframe = lteULResourceGrid(ue);
```

Generate the UL-SCH message. To do so, call the function `lteULSCH`, providing the transport block data, `trblk`, the UE-specific structure, `ue`, and the channel-specific structure, `pusch`, as input arguments.

```
trblk = round(rand(1,504));
cw = lteULSCH(ue,pusch,trblk);
```

Generate the PUSCH symbols. To do so, call the function `ltePUSCH`, providing the UE and PUSCH configuration structures, `ue` and `pusch`, as input arguments.

```
puschSymbols = ltePUSCH(ue,pusch,cw);
```

Generate the PUSCH indices. To do so, call the function `ltePUSCHIndices`. You can use these generated indices to map the PUSCH complex symbols to the subframe resource grid. This function requires the same input argument structures as `ltePUSCH`.

```
puschIndices = ltePUSCHIndices(ue,pusch);
```

Generate the PUSCH DRS symbols. To do so, call the function `ltePUSCHDRS`. This function requires the same input argument structures as `ltePUSCH`.

```
drsSymbols = ltePUSCHDRS(ue,pusch);
```

Generate the PUSCH DRS indices. To do so, call the function `ltePUSCHDRSIndices`. You can use these indices to map the DRS to the subframe resource grid.

```
drsIndices = ltePUSCHDRSIndices(ue,pusch);
```

Map the PUSCH symbols and DRS to the resource grid, using the generated indices.

```
subframe(puschIndices) = puschSymbols;
subframe(drsIndices) = drsSymbols;
```

See Also

`lteULSCH` | `lteULSCHInfo` | `ltePUSCH` | `ltePUSCHIndices` | `ltePUSCHDRS` | `ltePUSCHDRSIndices` | `lteULResourceGrid`

Concepts

- “Uplink Shared Channel” on page 1-130

Simulate Propagation Channels

This example shows how to simulate propagation channels. It demonstrates how to generate cell-specific reference signals, map them onto a resource grid, perform OFDM modulation, and pass the result through a fading channel.

Set up the cell-wide settings. Create a structure and specify the cell-wide settings as its fields.

```
enb.NDLRB = 9;
enb.CyclicPrefix = 'Normal';
enb.PHICHDuration = 'Normal';
enb.CFI = 3;
enb.Ng = 'Sixth';
enb.CellRefP = 1;
enb.NCellID = 10;
enb.NSubframe = 0;
enb.DuplexMode = 'FDD';
antennaPort = 0;
```

Many of the functions used in this example require a subset of the preceding settings specified.

Generate a subframe resource grid. To create the resource grid, call the function `lteDLResourceGrid`. This function creates an empty resource grid for one subframe.

```
subframe = lteDLResourceGrid(enb);
```

Generate cell-specific reference symbols (CellRS). Then, map them onto the resource elements (REs) of a resource grid using linear indices.

```
cellRSsymbols = lteCellRS(enb,antennaPort);
cellRSindices = lteCellRSIndices(enb,antennaPort,{'1based'});
subframe(cellRSindices) = cellRSsymbols;
```

Perform OFDM modulation of the complex symbols in a subframe, `subframe`, using cell-wide settings structure, `enb`.

```
[txWaveform,info] = lteOFDMModulate(enb,subframe);
```

The first output argument, `txWaveform`, contains the transmitted OFDM modulated symbols. The second output argument, `info`, is a structure that contains details about the modulation process. The field `info.SamplingRate` provides the sampling rate of the time domain waveform, as given by the following equation.

$$R_{Sampling} = 30.72\text{MHz}/2048 \times N_{fft}$$

In the preceding equation, N_{fft} is the size of the OFDM inverse Fourier transform (IFT).

Construct the LTE multipath fading channel. First, set up the channel parameters by creating a structure, `channel`.

```
channel.Seed = 1;
channel.NRxAnts = 1;
channel.DelayProfile = 'EVA';
channel.DopplerFreq = 5;
channel.MIMOCorrelation = 'Low';
channel.SamplingRate = info.SamplingRate;
channel.InitTime = 0;
```

The sampling rate in the channel model, `channel.SamplingRate`, must be set to the value created by `lteOFDMModulate`, `info.SamplingRate`.

Pass data through the LTE fading channel. To do so, call the function `lteFadingChannel`. This function generates an LTE multipath fading channel, as specified in [1]. The first input argument, `txWaveform`, is an array of LTE transmitted samples. Each row contains the waveform samples for each of the transmit antennas. These waveforms are filtered with the delay profiles as specified in the parameter structure, `channel`.

```
rxWaveform = lteFadingChannel(channel,txWaveform);
```

The output argument, `rxWaveform`, is the channel output signal matrix. Each row corresponds to the waveform at each of the receive antennas. Since you have defined one receive antenna, the number of rows in the `rxWaveform` matrix is one.

References

[1] 3GPP TS 36.101. "User Equipment (UE) Radio Transmission and Reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`lteFadingChannel` | `lteHSTChannel` | `lteMovingChannel`

Related Examples

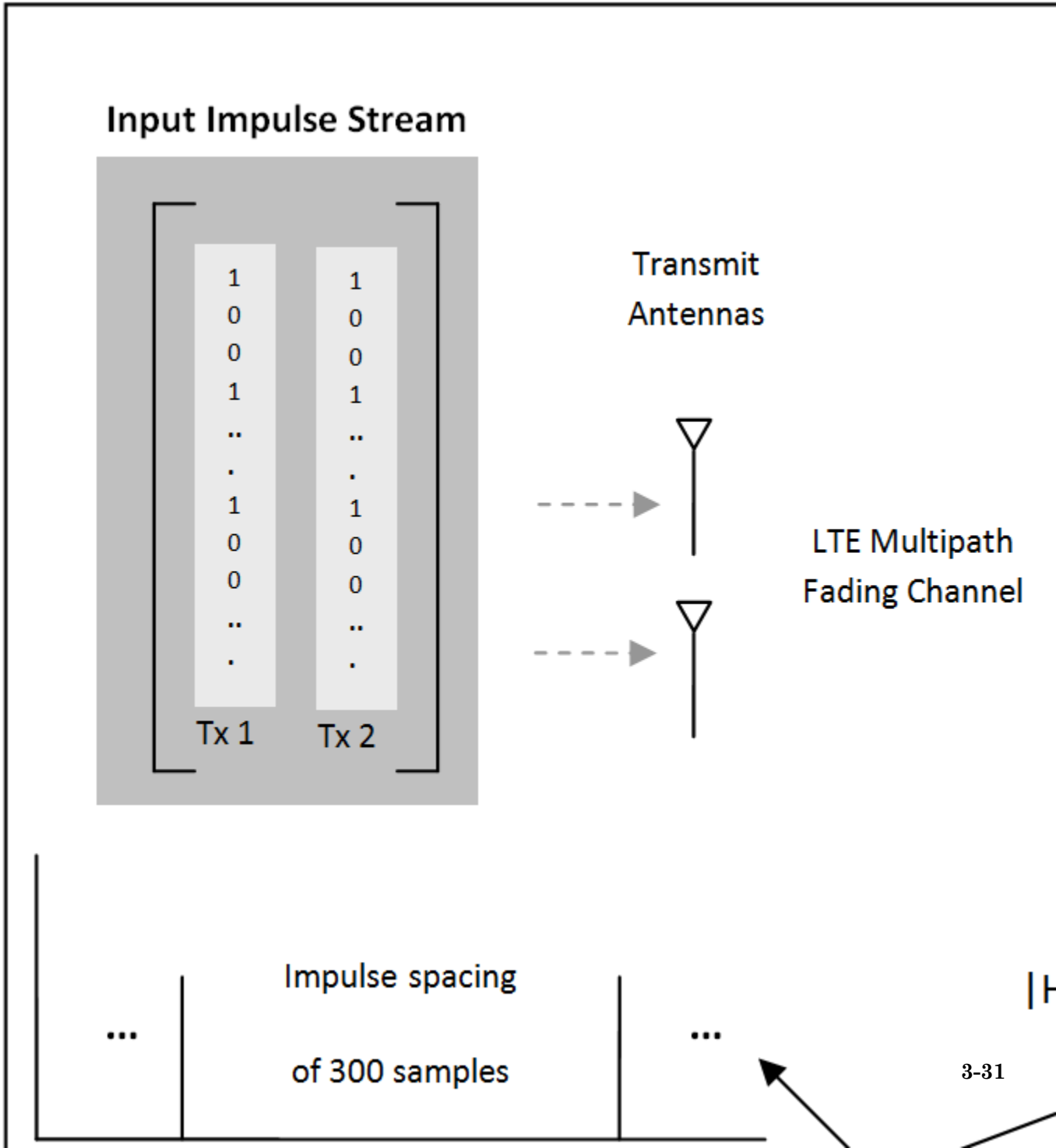
- "Find Channel Impulse Response" on page 3-30

Concepts

- "Propagation Channel Models" on page 1-155

Find Channel Impulse Response

This example shows how to find the channel impulse response of a 2-by-2 MIMO system. The input is a matrix of impulses where each impulse is separated by 300 samples. Each column in the matrix, the size of which is the number of transmit antennas, is the input waveform to the channel model function and is therefore a series of impulses. This series of impulses allows the changing impulse response of the channel to be viewed over time. For clear visualization, the impulse spacing should be greater than maximum delay spread of the channel. The input waveform is passed through the LTE multipath fading channel model. The output matrix has complex samples corresponding to each receive antenna. This process is shown in the following figure.


Input Impulse Stream
1
0
0
1
..
.
1
0
0
..
.
Tx 1
1
0
0
1
..
.
1
0
0
..
.
Tx 2
Transmit
Antennas
LTE Multipath
Fading Channel
...
Impulse spacing
of 300 samples
...

Pre-configure the LTE multipath fading channel. To do so, set up a simple structure and specify the fading channel parameters.

```
channel.Seed = 1;
channel.NRxAnts = 2;
channel.DelayProfile = 'EVA';
channel.DopplerFreq = 300;
channel.CarrierFreq = 2e9;
channel.MIMOCorrelation = 'Low';
channel.SamplingRate = 1/10e-9;
channel.InitTime = 0;
```

The structure, `channel`, contains the following fields.

- `Seed` — the channel seed
- `NRxAnts` — the number of receive antennas
- `DelayProfile` — the delay profile
- `DopplerFreq` — the *Doppler* frequency
- `CarrierFreq` — the carrier frequency
- `MIMOCorrelation` — the MIMO correlation
- `SamplingRate` — the channel sampling rate
- `InitTime` — the initial channel time

Create two identical input streams of data. These input streams are passed through two transmit antennas, as shown in the preceding figure.

```
nAntIn = 2;
impulseSpacing = 300;
noImpResponse = 150;
nInputSamples = impulseSpacing * noImpResponse;
in = zeros(nInputSamples, nAntIn);
onesLocations = 1:impulseSpacing:nInputSamples;
in(onesLocations,1) = 1;
```

The variable `nAntIn` is the number of transmit antennas. The variable `impulseSpacing` is greater than the maximum channel delay spread. The variable `noImpResponse` is the number of impulse responses to calculate.

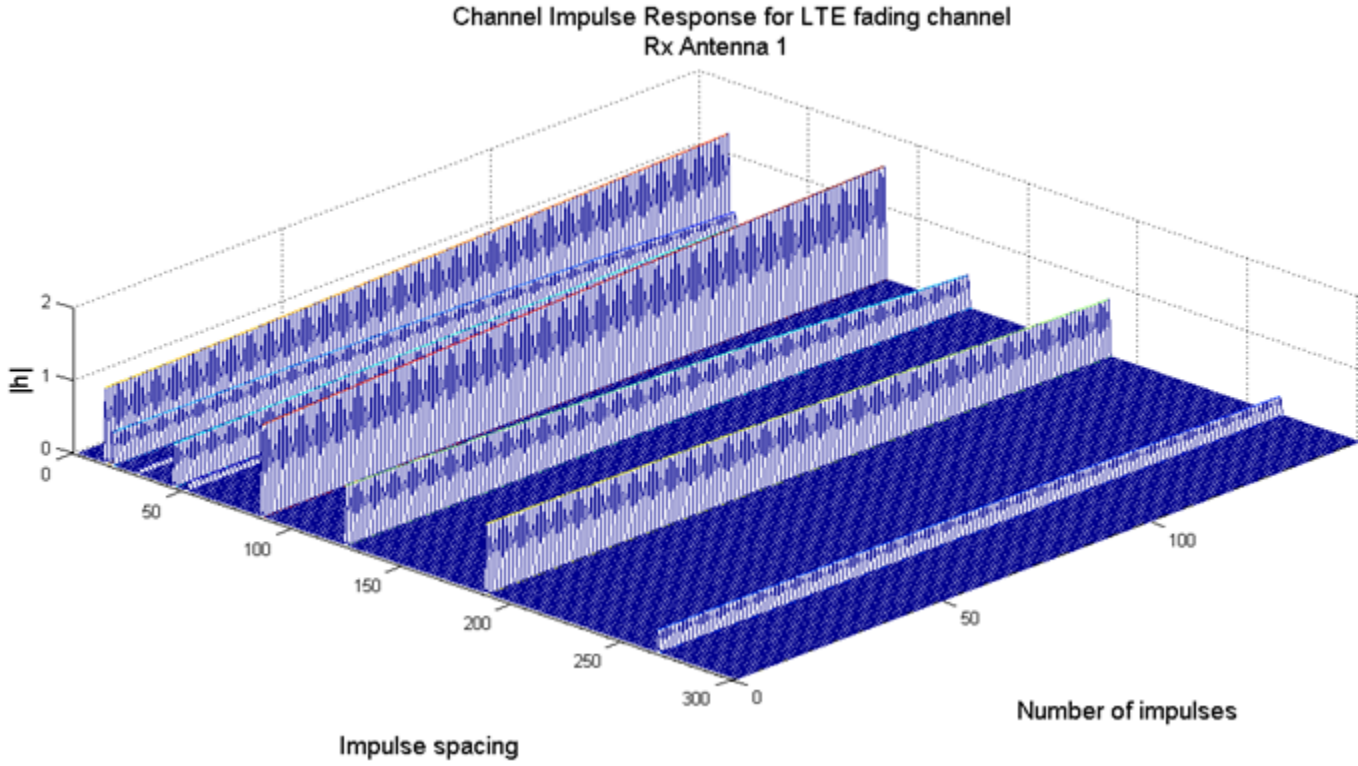
Filter with the LTE fading channel. To do so, call the function `lteFadingChannel`. This function generates an LTE multi-path fading channel, as specified in [1]. The first input argument, `in`, is an array of LTE transmitted samples. Each row contains the waveform samples for each of the transmit antennas. These waveforms are filtered with the delay profiles as specified in the parameter structure, `channel`.

```
out = lteFadingChannel(channel, in);
```

Finally, plot the receive waveform.

```
for antNo = 1:channel.NRxAnts
    figure
    mesh(squeeze(abs(reshape(out(:,antNo), ...
        impulseSpacing,noImpResponse).')));
    titleStr = ['Rx Antenna' num2str(antNo)];
    title({'Channel Impulse Response for LTE fading channel',titleStr});
    ylabel('number of impulses');
    xlabel('Impulse spacing [no of samples]');
    zlabel('|H|');
end;
```

The channel impulse response at receive antenna 1 is shown in the following figure.



References

[1] 3GPP TS 36.101. "User Equipment (UE) Radio Transmission and Reception." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

See Also

`lteFadingChannel` | `lteHSTChannel` | `lteMovingChannel`

Related Examples

- "Simulate Propagation Channels" on page 3-27

Concepts

- “Propagation Channel Models” on page 1-155

4

Selected Bibliography

- "TS 36.101" on page 4-2
- "TS 36.104" on page 4-3
- "TS 36.141" on page 4-4
- "TS 36.211" on page 4-5
- "TS 36.212" on page 4-6
- "TS 36.213" on page 4-7
- "TS 36.214" on page 4-8
- "TS 36.321" on page 4-9

TS 36.101

3GPP TS 36.101. “User Equipment (UE) Radio Transmission and Reception.” *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

TS 36.104

3GPP TS 36.104. “Base Station (BS) radio transmission and reception.” *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

TS 36.141

3GPP TS 36.141. "Base Station (BS) conformance testing." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

TS 36.211

3GPP TS 36.211. “Physical Channels and Modulation.” *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

TS 36.212

3GPP TS 36.212. "Multiplexing and channel coding." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

TS 36.213

3GPP TS 36.213. “Physical layer procedures.” *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.

TS 36.214

3GPP TS 36.214. "Physical layer — Measurements." *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).* URL: http://www.3gpp.org.

TS 36.321

3GPP TS 36.321. “Medium Access Control (MAC) protocol specification.” *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA)*. URL: http://www.3gpp.org.