US 20210336635A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2021/0336635 A1

(43) Pub. Date: Oct. 28, 2021

MOUHOUCHE et al.

(54) **TRANSMITTING APPARATUS AND MAPPING METHOD THEREOF**

(71) Applicant: **SAMSUNG ELECTRONICS CO., LTD.**, Suwon-si (KR)

(72) Inventors: **Belkacem MOUHOUCHE**, Staines (GB); **Daniel Ansorregui Lobete**, Staines (GB); **Hong-sil Jeong**, Suwon-si (KR)

(73) Assignee: **SAMSUNG ELECTRONICS CO., LTD.**, Suwon-si (KR)

(21) Appl. No.: **17/366,790**

(22) Filed: **Jul. 2, 2021**

Related U.S. Application Data

(63) Continuation of application No. 16/905,980, filed on Jun. 19, 2020, now Pat. No. 11,082,064, which is a continuation of application No. 16/538,132, filed on Aug. 12, 2019, now Pat. No. 10,727,871, which is a continuation of application No. 14/715,747, filed on May 19, 2015, now Pat. No. 10,432,222.

Publication Classification

(51) **Int. Cl.**

H03M 13/11	(2006.01)
H03M 13/25	(2006.01)
H03M 13/27	(2006.01)
H03M 13/29	(2006.01)
H04L 1/00	(2006.01)

(52) **U.S. Cl.**

CPC ***H03M 13/1165*** (2013.01); ***H03M 13/255*** (2013.01); ***H03M 13/2707*** (2013.01); ***H03M 13/2732*** (2013.01); *H03M 13/152* (2013.01); ***H03M 13/2906*** (2013.01); ***H04L 1/0042*** (2013.01); ***H04L 1/0057*** (2013.01); ***H04L 1/0071*** (2013.01); ***H03M 13/2778*** (2013.01)

(57) **ABSTRACT**

A transmitting apparatus is disclosed. The transmitting apparatus includes an encoder to perform channel encoding with respect to bits and generate a codeword, an interleaver to interleave the codeword, and a modulator to map the interleaved codeword onto a non-uniform constellation according to a modulation scheme, and the constellation may include constellation points defined based on various tables according to the modulation scheme.

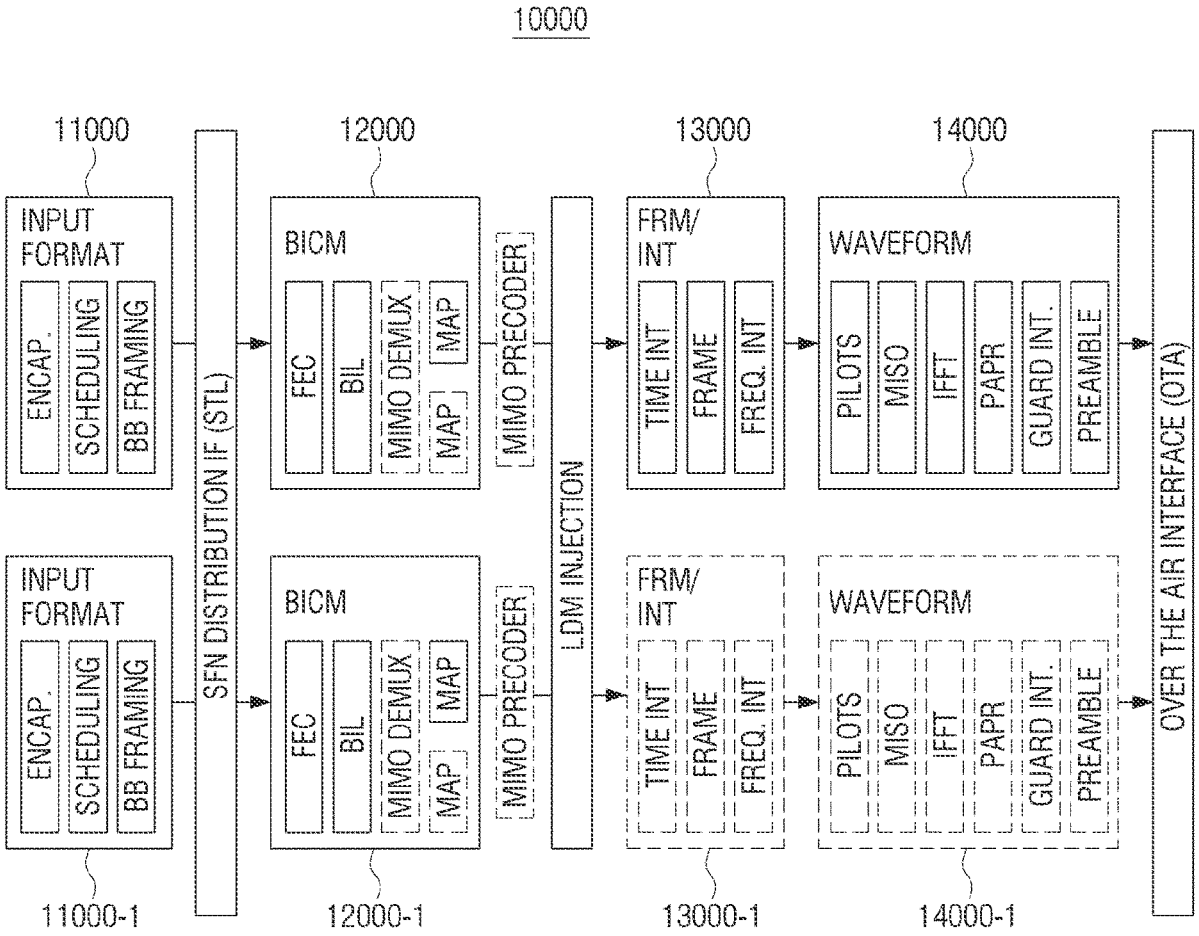


FIG. 1A

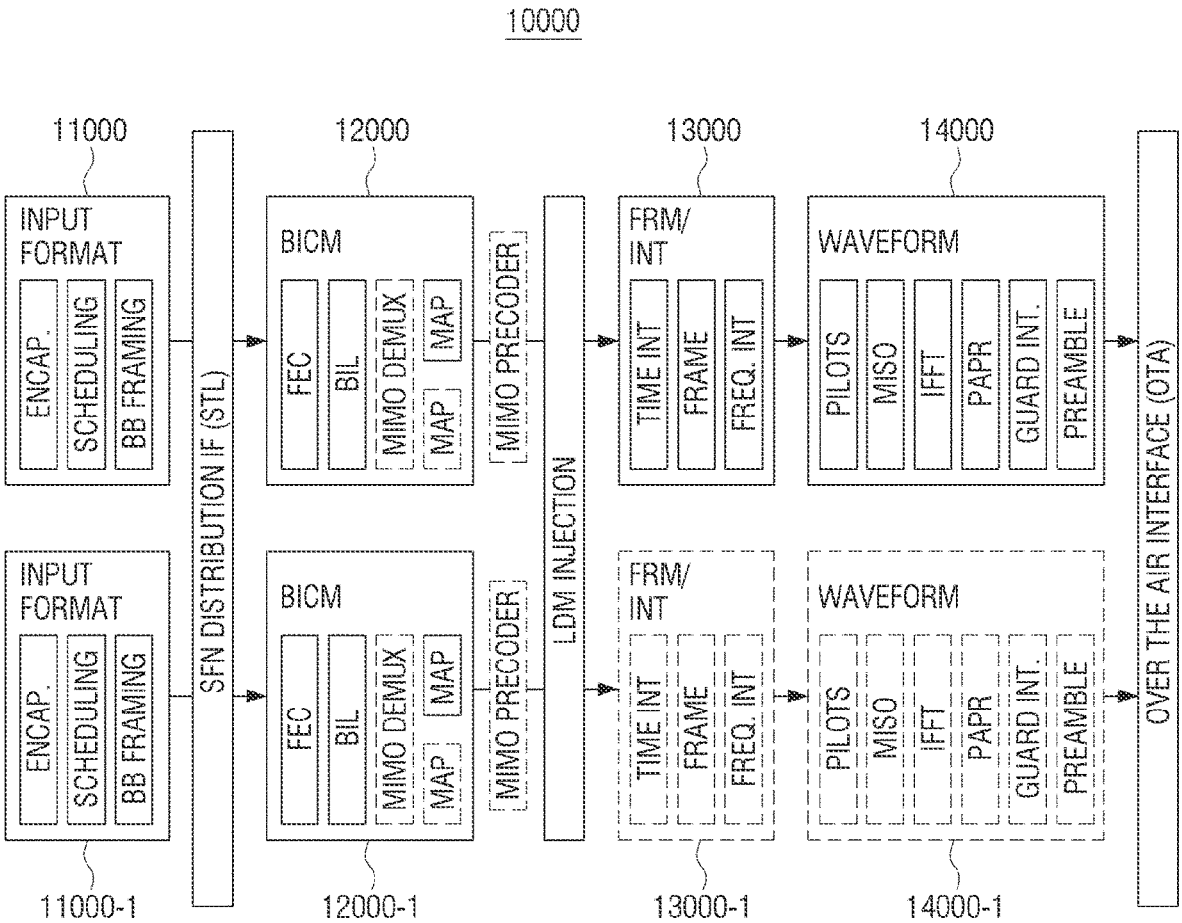

10000
11000
INPUT FORMAT
ENCAP.
SCHEDULING
BB FRAMING
11000-1
INPUT FORMAT
ENCAP.
SCHEDULING
BB FRAMING
SFN DISTRIBUTION IF (STL)
12000
BICM
FEC
BIL
MIMO DEMUX
MAP
MAP
12000-1
BICM
FEC
BIL
MIMO DEMUX
MAP
MAP
MIMO PRECODER
MIMO PRECODER
LDM INJECTION
13000
FRM/ INT
TIME INT
FRAME
FREQ. INT
13000-1
FRM/ INT
TIME INT
FRAME
FREQ. INT
14000
WAVEFORM
PILOTS
MISO
IFFT
PAPR
GUARD INT.
PREAMBLE
14000-1
WAVEFORM
PILOTS
MISO
IFFT
PAPR
GUARD INT.
PREAMBLE
OVER THE AIR INTERFACE (OTA)

FIG. 1B

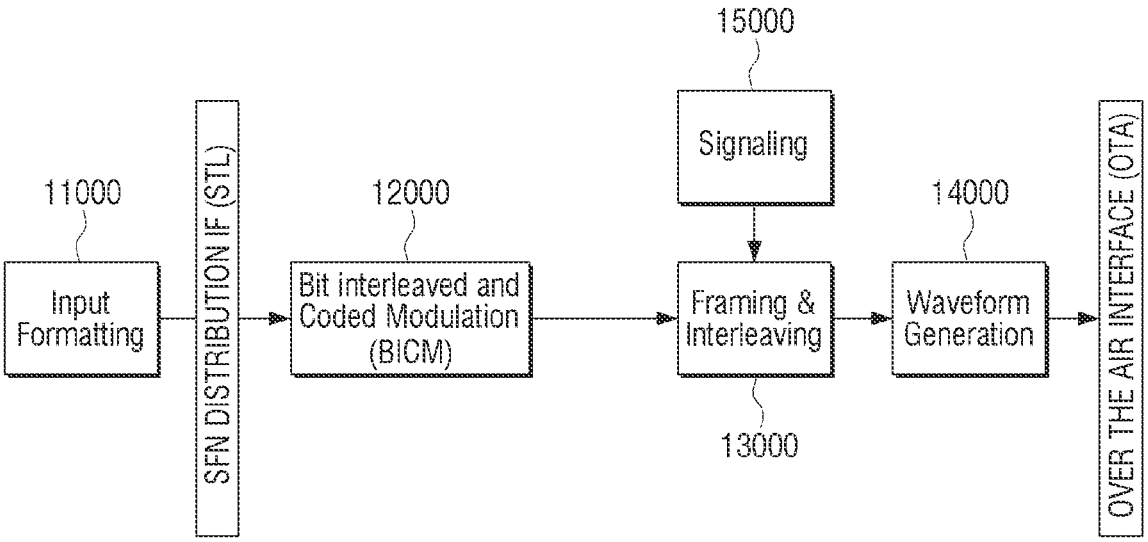

11000
Input Formatting
SFN DISTRIBUTION IF (STL)
12000
Bit interleaved and Coded Modulation (BICM)
15000
Signaling
Framing & Interleaving
13000
14000
Waveform Generation
OVER THE AIR INTERFACE (OTA)

FIG. 1C

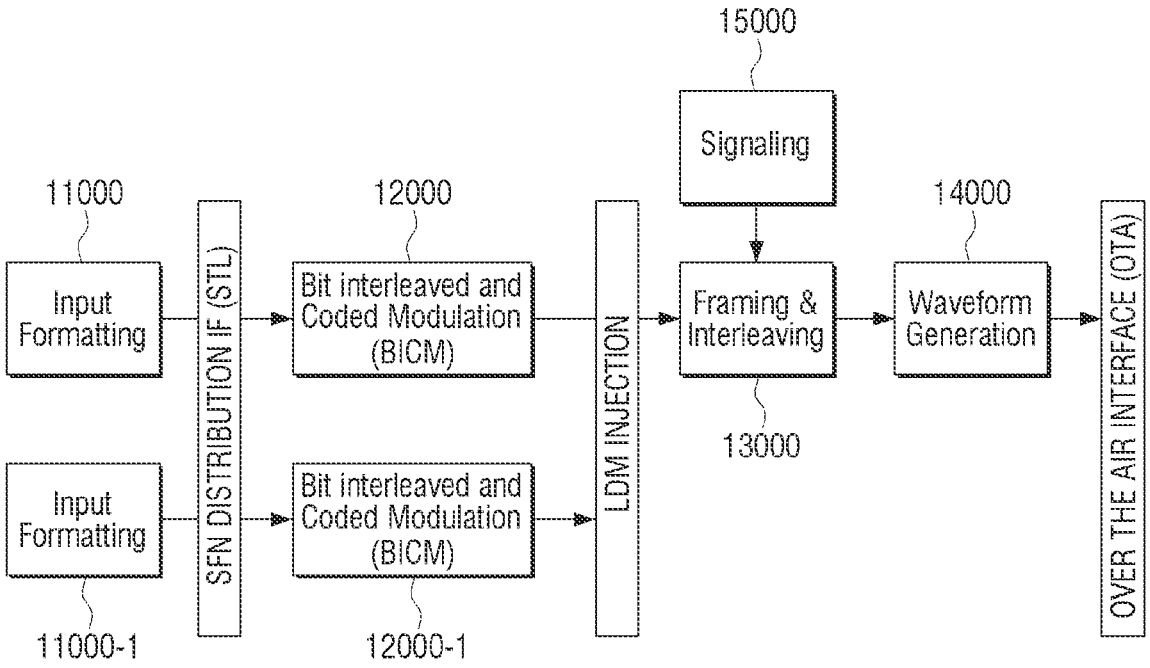

15000
Signaling
11000
12000
14000
Input Formatting
SFN DISTRIBUTION IF (STL)
Bit interleaved and Coded Modulation (BICM)
LDM INJECTION
Framing & Interleaving
Waveform Generation
OVER THE AIR INTERFACE (OTA)
13000
Input Formatting
Bit interleaved and Coded Modulation (BICM)
11000-1
12000-1

FIG. 2

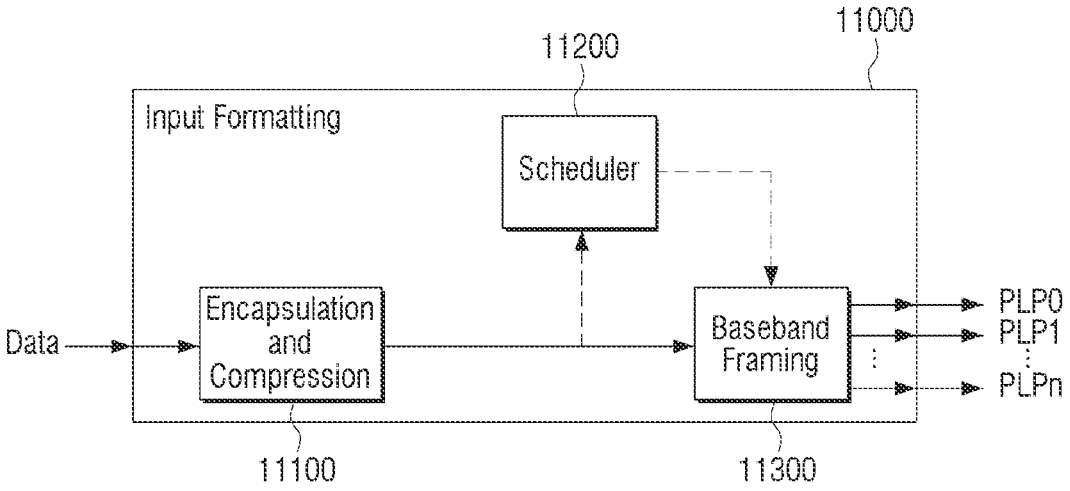

11000
11200
Input Formatting
Scheduler
Data
Encapsulation and Compression
Baseband Framing
PLP0
PLP1
PLPn
11100
11300

FIG. 3A

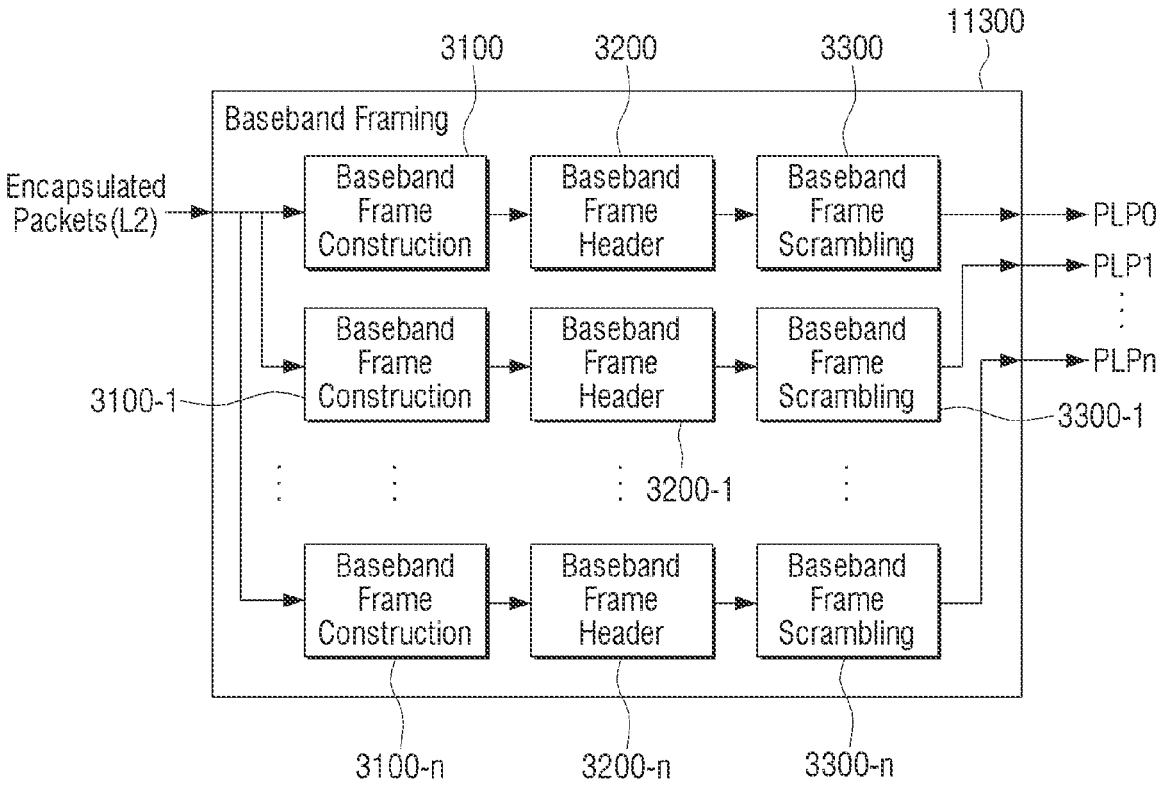

11300
3100
3200
3300
Baseband Framing
Encapsulated Packets(L2)
Baseband Frame Construction
Baseband Frame Header
Baseband Frame Scrambling
PLP0
PLP1
PLPn
3100-1
3200-1
3300-1
3100-n
3200-n
3300-n

FIG. 3B

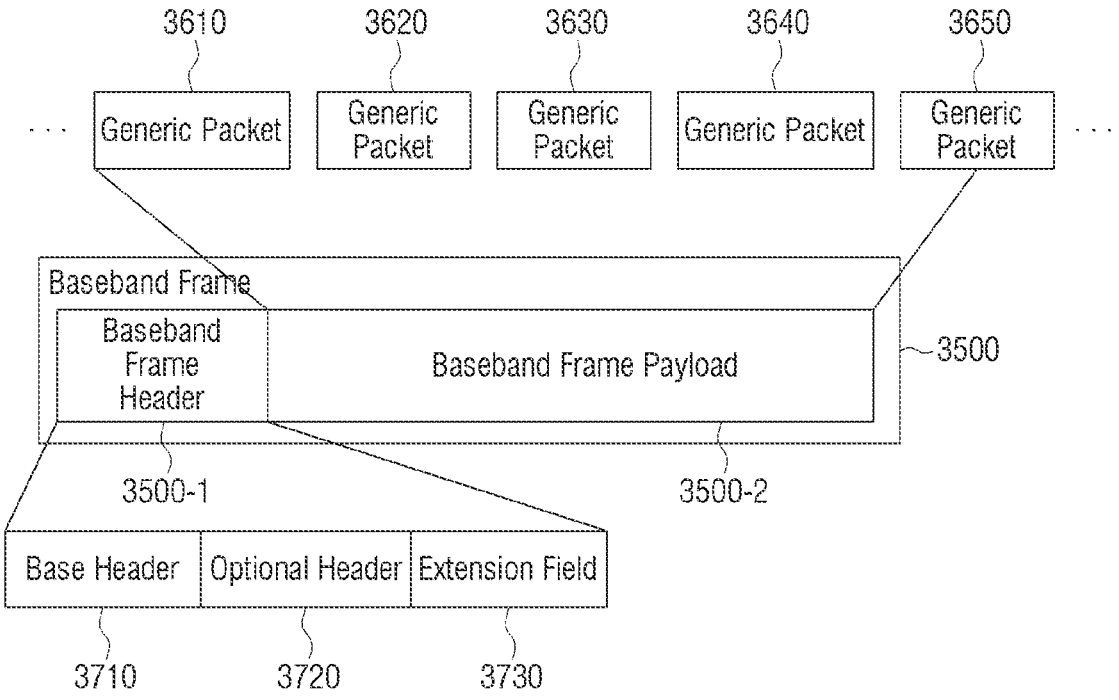

3610
3620
3630
3640
3650
Generic Packet
Generic Packet
Generic Packet
Generic Packet
Generic Packet
Baseband Frame
Baseband Frame Header
Baseband Frame Payload
3500
3500-1
3500-2
Base Header
Optional Header
Extension Field
3710
3720
3730

FIG. 4

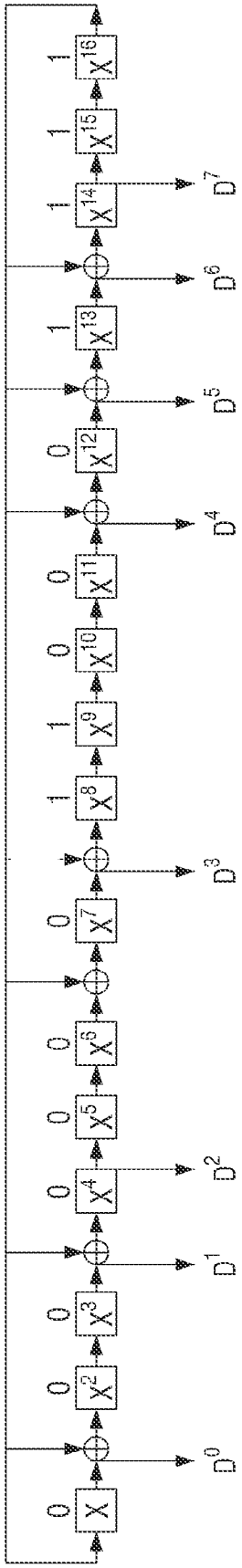

0
X
D0
0
X2
0
X3
D1
0
X4
D2
0
X5
0
X6
0
X7
D3
1
X8
1
X9
0
X10
0
X11
D4
0
X12
D5
1
X13
D6
1
X14
D7
1
X15
1
X16

FIG. 5

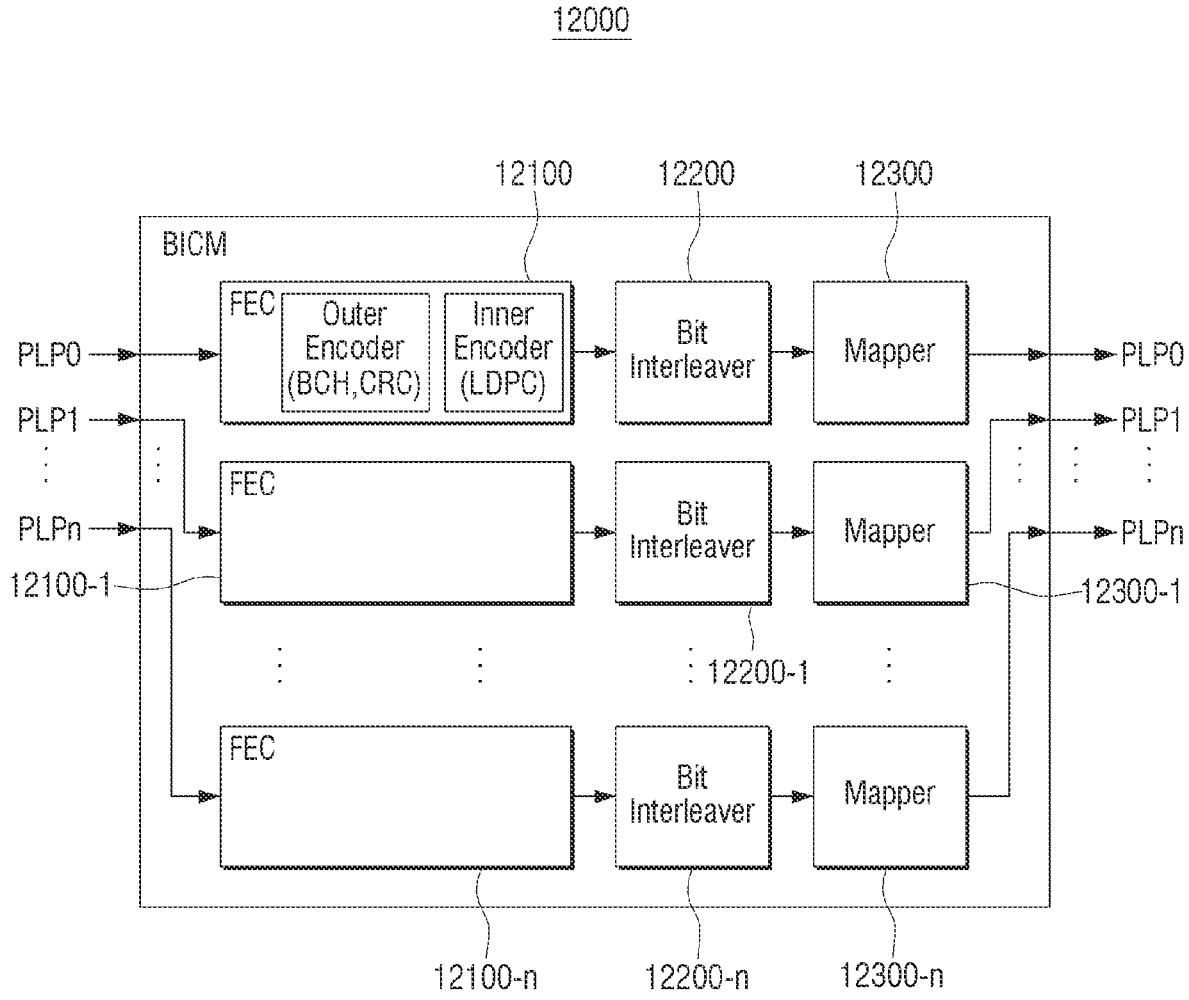

12000
12100
12200
12300
BICM
FEC
Outer Encoder (BCH,CRC)
Inner Encoder (LDPC)
Bit Interleaver
Mapper
PLP0
PLP1
PLPn
12100-1
12200-1
12300-1
12100-n
12200-n
12300-n

FIG. 6A

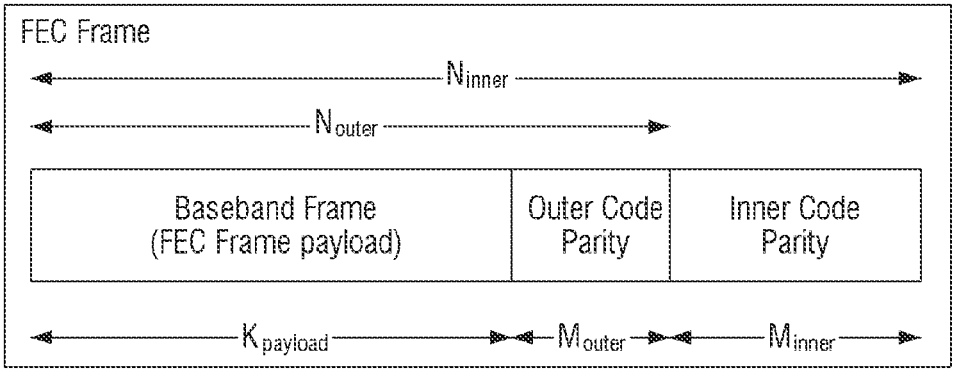

FEC Frame
N_{inner}
N_{outer}
Baseband Frame
(FEC Frame payload)
Outer Code
Parity
Inner Code
Parity
$K_{payload}$
M_{outer}
M_{inner}

FIG. 6B

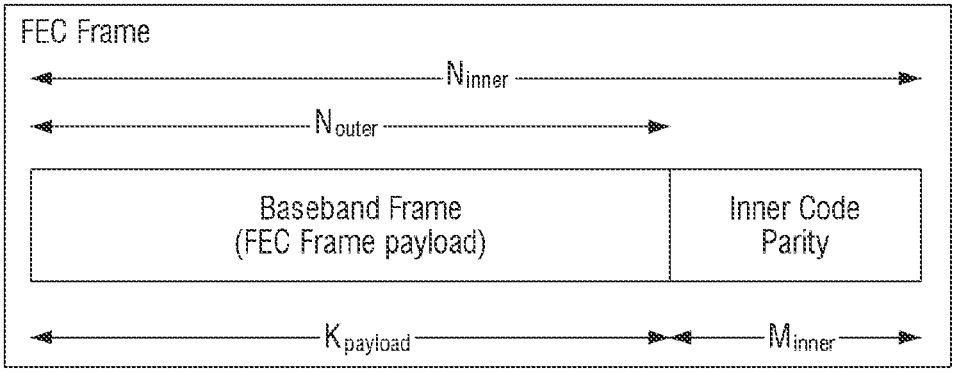

FEC Frame
N_{inner}
N_{outer}
Baseband Frame
(FEC Frame payload)
Inner Code
Parity
$K_{payload}$
M_{inner}

FIG. 7

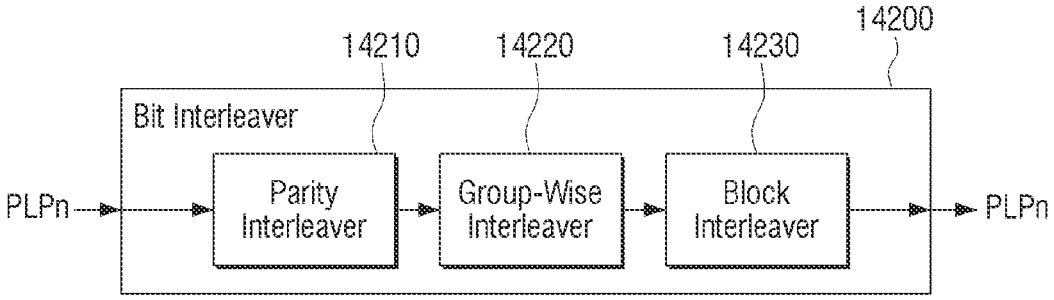

14200
14210
14220
14230
Bit Interleaver
PLPn
Parity Interleaver
Group-Wise Interleaver
Block Interleaver
PLPn

FIG. 8

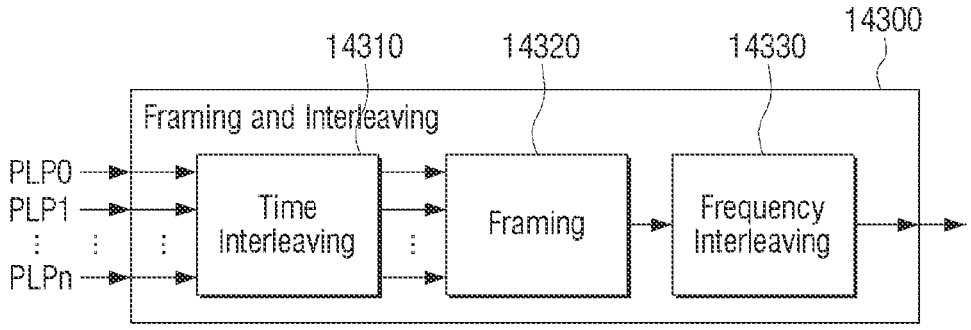

14300
14310
14320
14330
Framing and Interleaving
PLP0
PLP1
PLPn
Time Interleaving
Framing
Frequency Interleaving

FIG. 9A

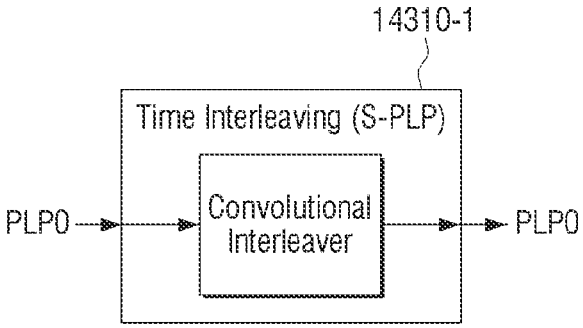

14310-1
Time Interleaving (S-PLP)
Convolutional Interleaver
PLP0
PLP0

FIG. 9B

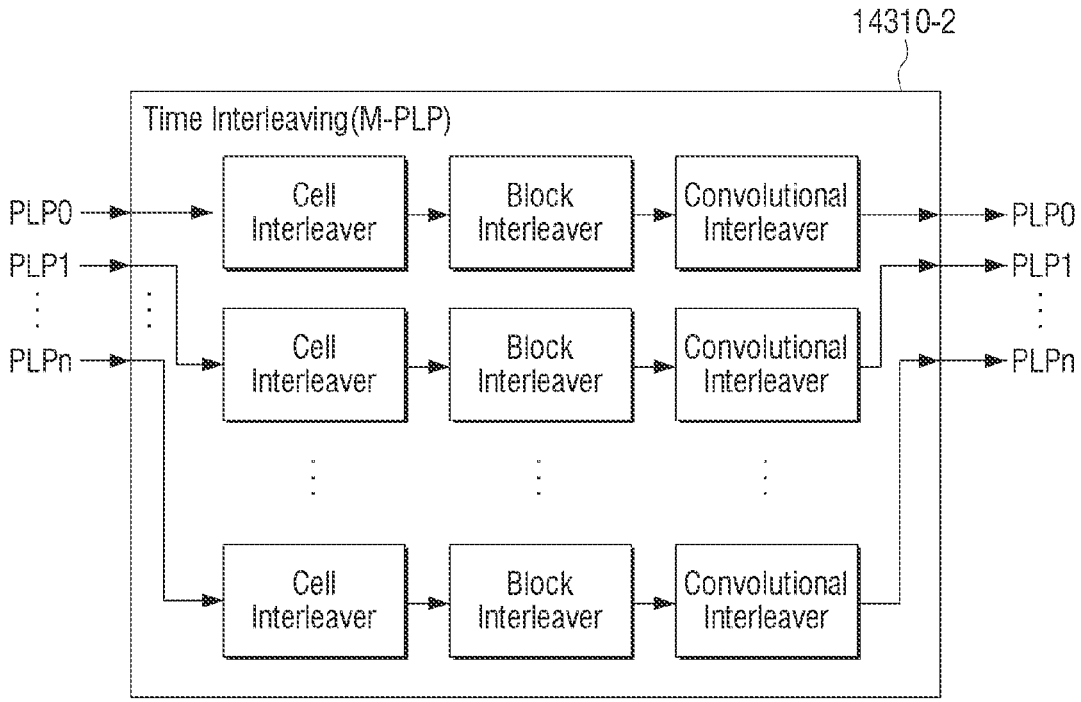

14310-2
Time Interleaving(M-PLP)
PLP0
PLP1
PLPn
Cell Interleaver
Block Interleaver
Convolutional Interleaver
Cell Interleaver
Block Interleaver
Convolutional Interleaver
Cell Interleaver
Block Interleaver
Convolutional Interleaver
PLP0
PLP1
PLPn

FIG. 10

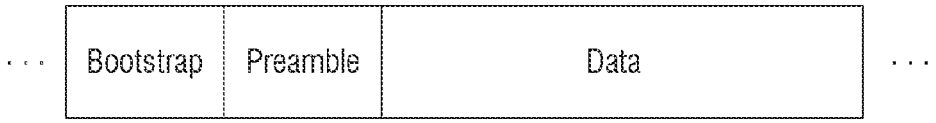


FIG. 11

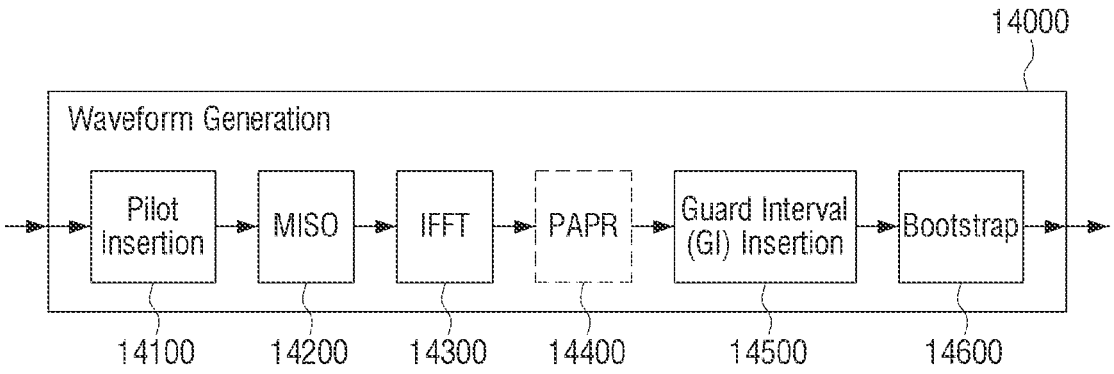

14000
Waveform Generation
Pilot Insertion
MISO
IFFT
PAPR
Guard Interval (GI) Insertion
Bootstrap
14100
14200
14300
14400
14500
14600

FIG. 12

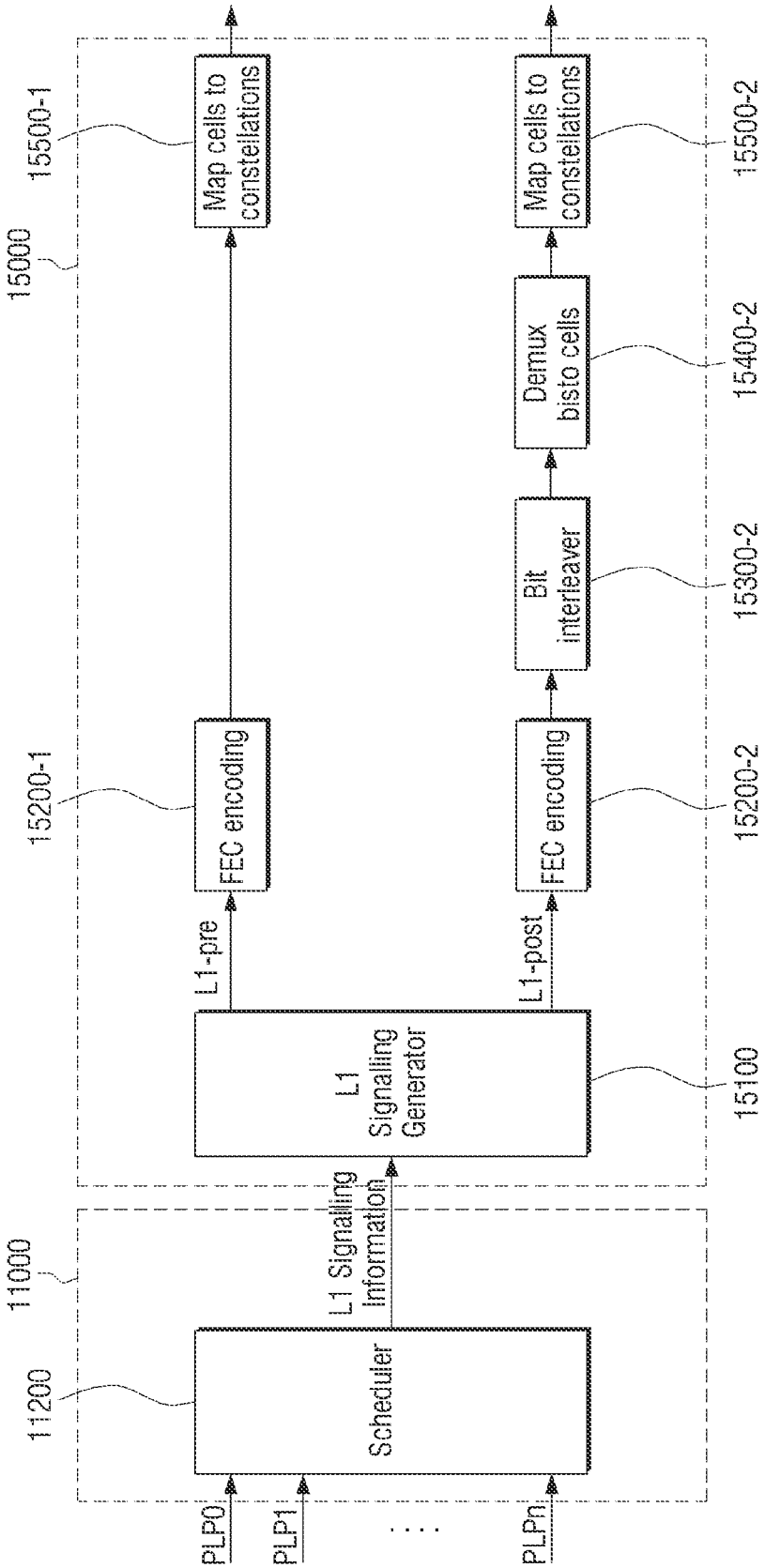

PLP0
PLP1
PLPn
11000
11200
Scheduler
L1 Signalling Information
15000
15100
L1 Signalling Generator
L1-pre
L1-post
15200-1
FEC encoding
15200-2
FEC encoding
15300-2
Bit interleaver
15400-2
Demux bisto cells
15500-1
Map cells to constellations
15500-2
Map cells to constellations

FIG. 13

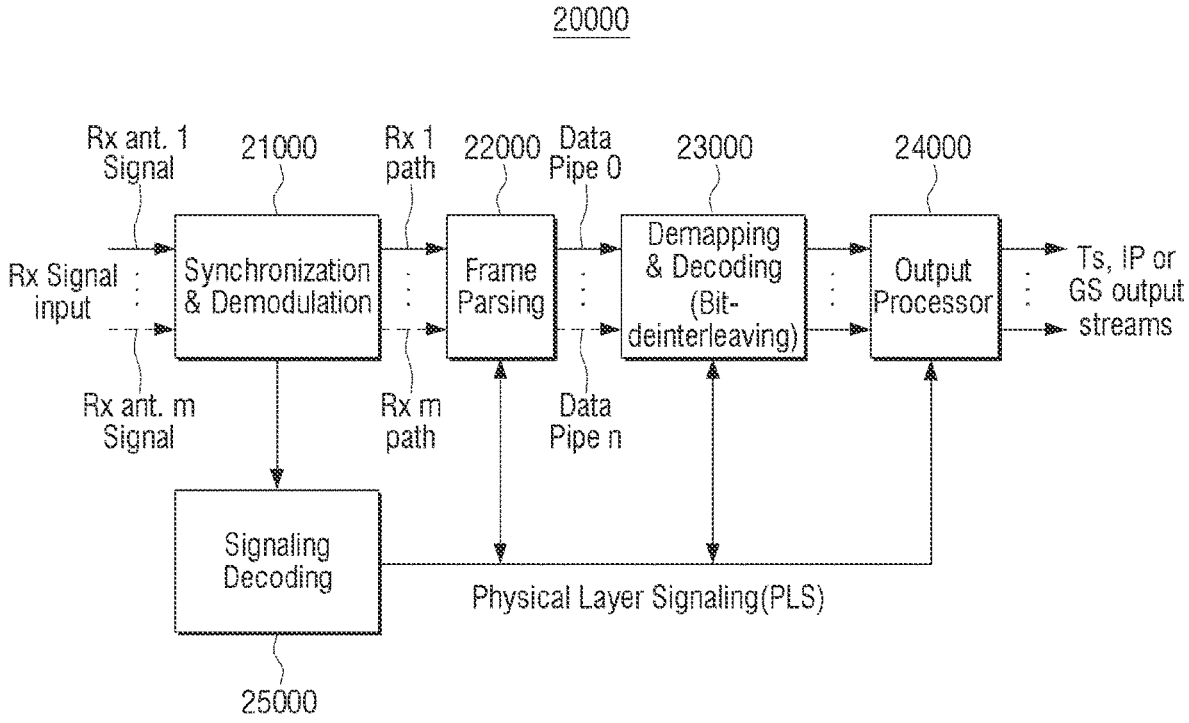

20000
Rx ant. 1 Signal
21000
Rx 1 path
22000
Data Pipe 0
23000
24000
Rx Signal input
Synchronization & Demodulation
Frame Parsing
Demapping & Decoding (Bit-deinterleaving)
Output Processor
Ts, IP or GS output streams
Rx ant. m Signal
Rx m path
Data Pipe n
Signaling Decoding
Physical Layer Signaling(PLS)
25000

FIG. 14

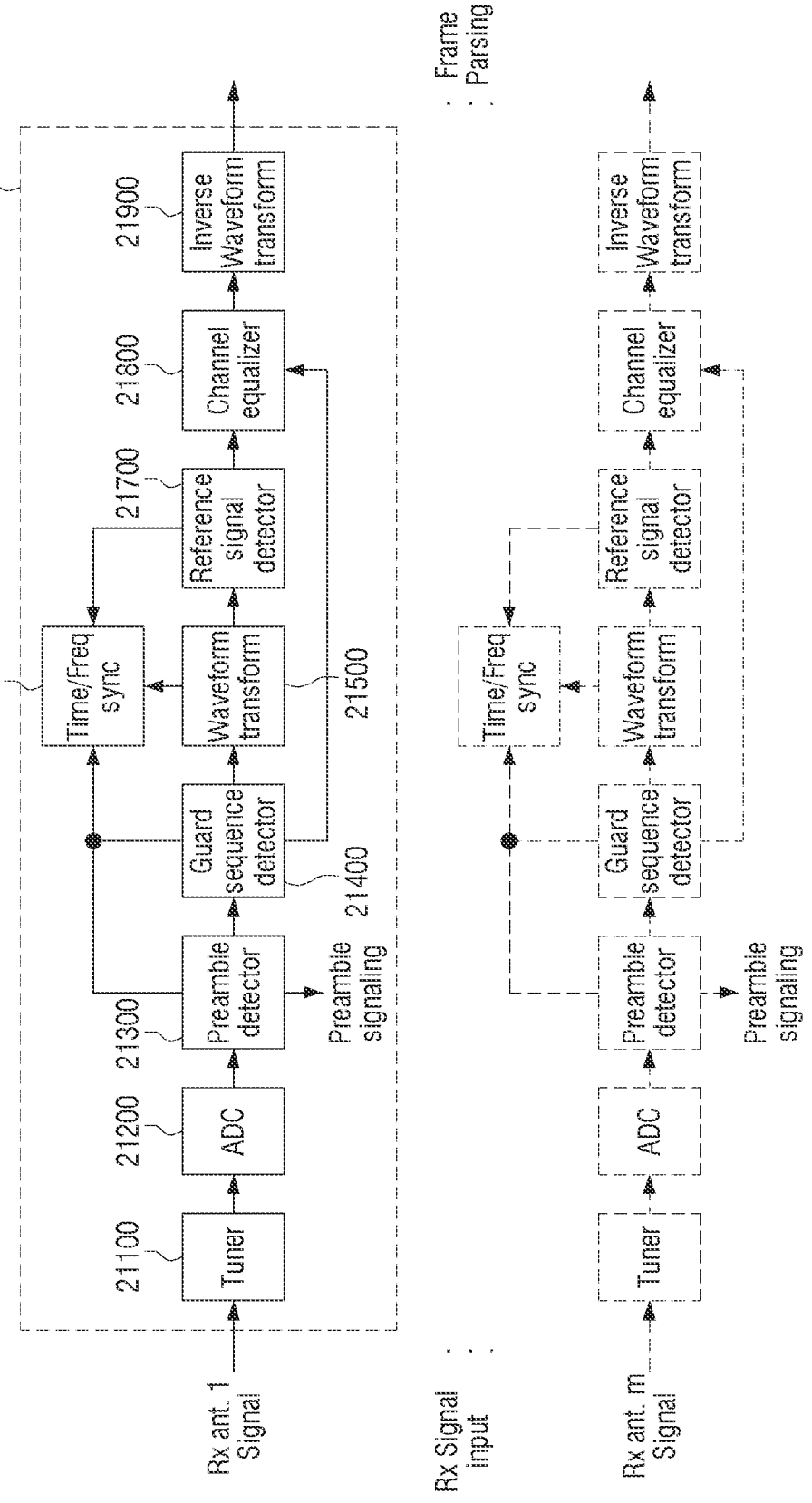

21000
21100
Tuner
21200
ADC
21300
Preamble detector
21400
Guard sequence detector
21500
Waveform transform
21600
Time/Freq sync
21700
Reference signal detector
21800
Channel equalizer
21900
Inverse Waveform transform
Rx ant. 1 Signal
Preamble signaling
Rx Signal input
Rx ant. m Signal
Frame Parsing

FIG. 15

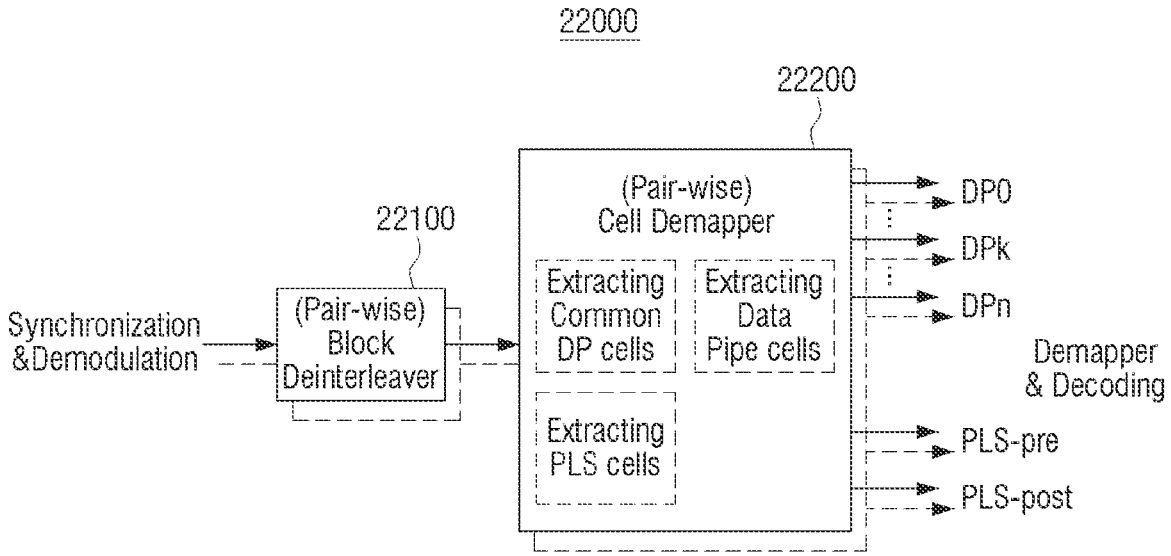

22000
22200
(Pair-wise)
Cell Demapper
22100
Synchronization
&Demodulation
(Pair-wise)
Block
Deinterleaver
Extracting
Common
DP cells
Extracting
Data
Pipe cells
Extracting
PLS cells
DP0
DPk
DPn
Demapper
& Decoding
PLS-pre
PLS-post

FIG. 16

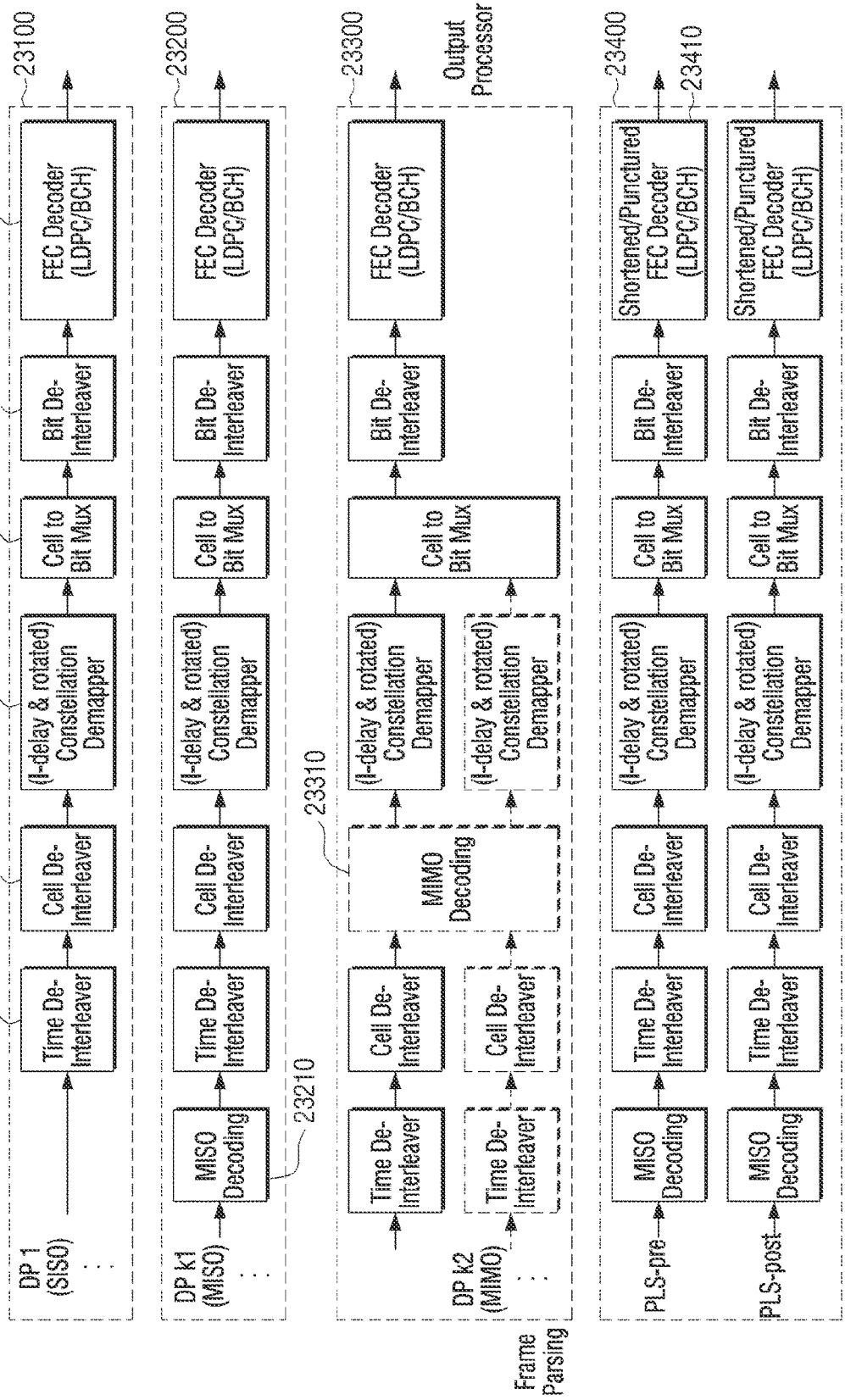

23000
23100
23110
23120
23130
23140
23150
23160
23200
23210
23300
23310
23400
23410
Output Processor
Frame Parsing
DP 1 (SISO)
DP k1 (MISO)
DP k2 (MIMO)
PLS-pre
PLS-post
Time De-Interleaver
Cell De-Interleaver
MISO Decoding
MIMO Decoding
(I-delay & rotated) Constellation Demapper
Cell to Bit Mux
Bit De-Interleaver
FEC Decoder (LDPC/BCH)
Shortened/Punctured FEC Decoder (LDPC/BCH)

FIG. 17

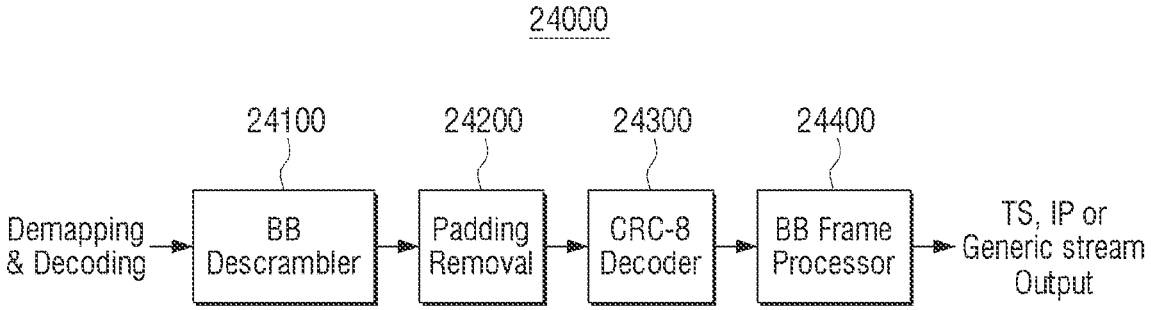

24000
24100
24200
24300
24400
Demapping & Decoding
BB Descrambler
Padding Removal
CRC-8 Decoder
BB Frame Processor
TS, IP or Generic stream Output

FIG. 18

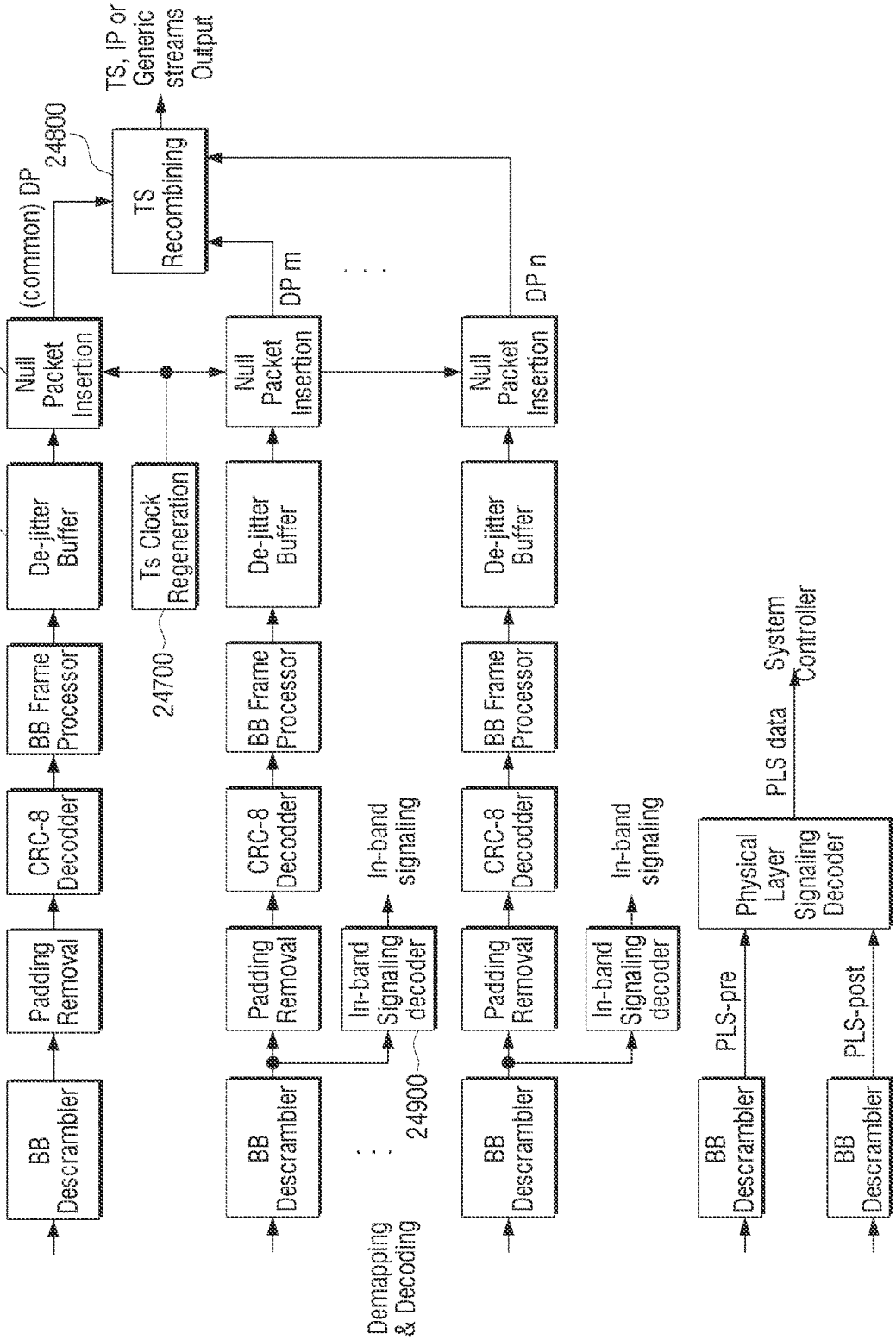
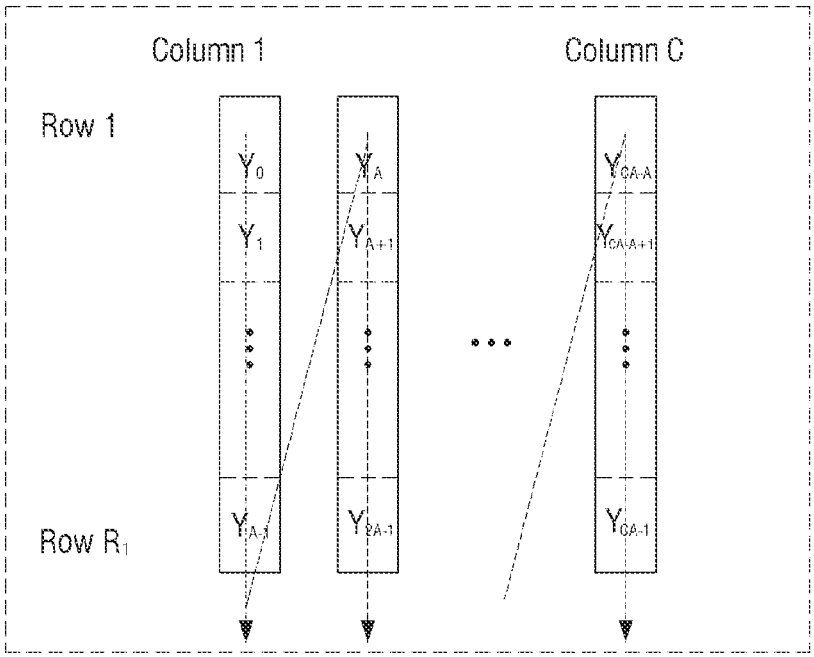

TS, IP or Generic streams Output
24800
TS Recombining
(common) DP
24600
Null Packet Insertion
DP m
DP n
24500
De-jitter Buffer
Ts Clock Regeneration
24700
BB Frame Processor
CRC-8 Decodder
Padding Removal
In-band Signaling decoder
In-band signaling
24900
BB Descrambler
Demapping & Decoding
PLS-pre
PLS-post
Physical Layer Signaling Decoder
PLS data
System Controller

FIG. 19


Column 1
Column C
Row 1
Row R1
Y0
Y1
YA
YA+1
YCA-A
YCA-A+1
YA-1
Y2A-1
YCA-1

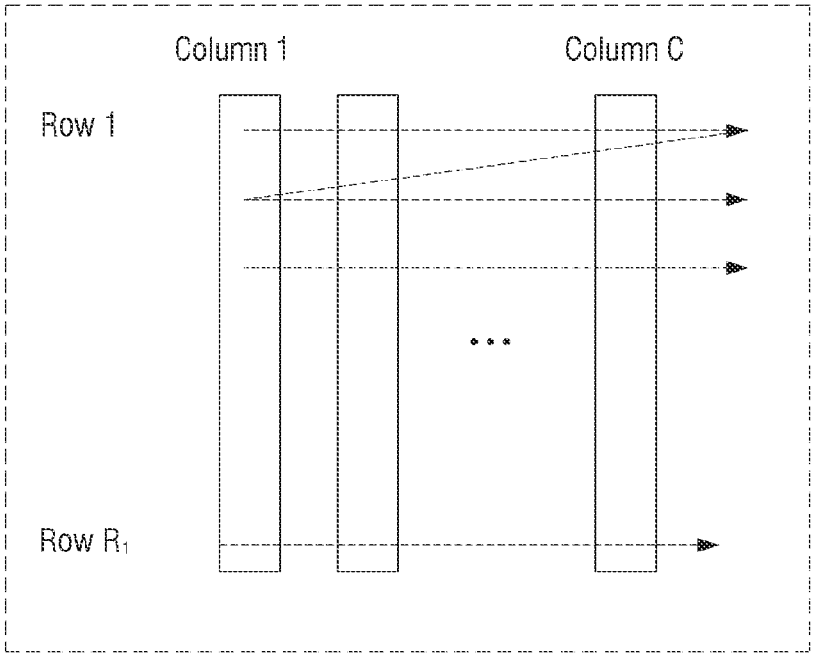

Column 1
Column C
Row 1
Row R1

FIG. 20

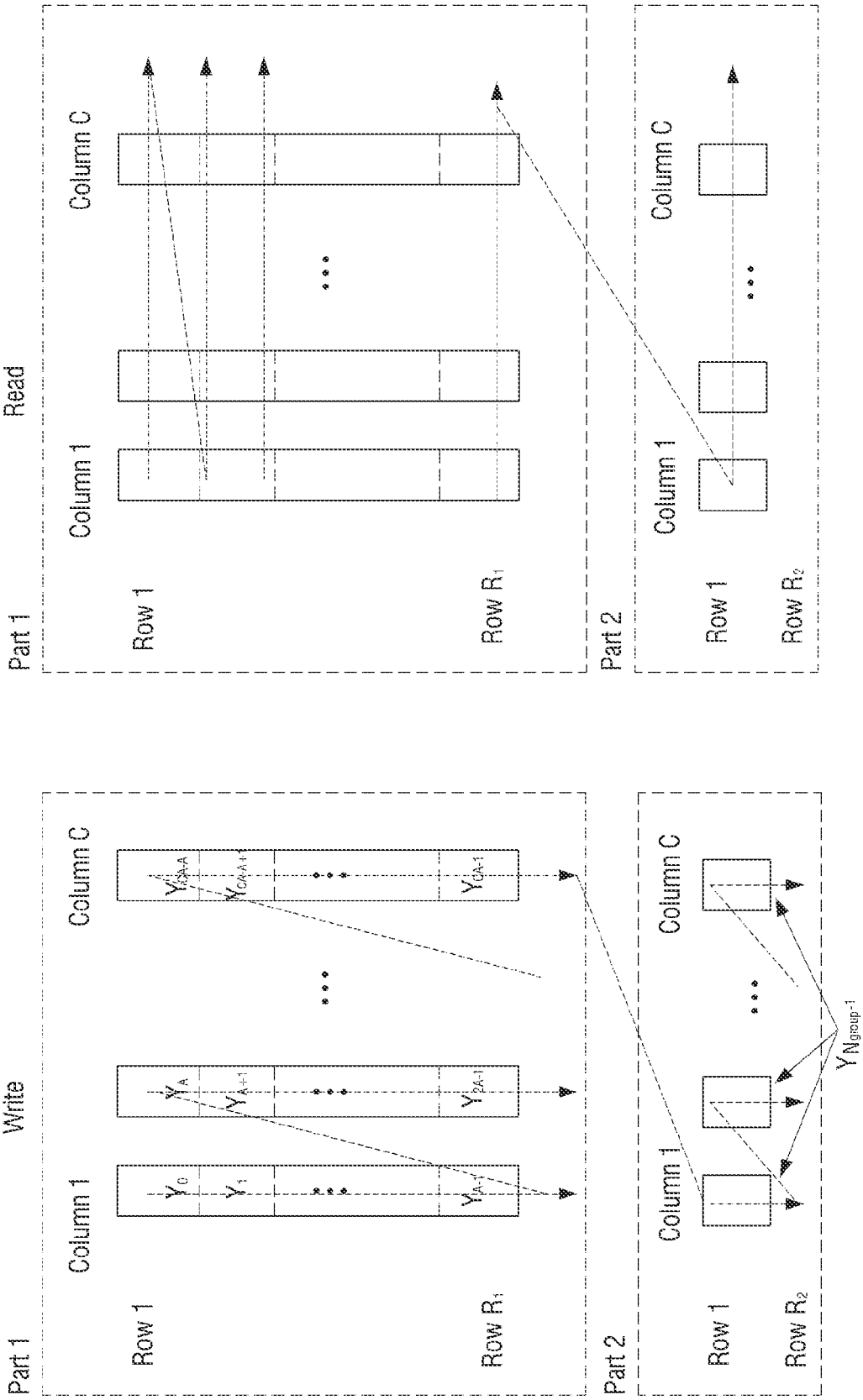

Read
Part 1
Column C
Column 1
Row 1
Row R_1
Part 2
Column C
Column 1
Row 1
Row R_2
Write
Part 1
Column C
Column 1
Row 1
Row R_1
Y_0
Y_1
Y_A
Y_{A+1}
$Y_{(A-1)A}$
$Y_{(A-1)A+1}$
$Y_{(A-1)A-A}$
$Y_{(A-1)A-1}$
Y_{A-1}
Part 2
Column C
Column 1
Row 1
Row R_2
$Y_{N_{group}-1}$

FIG. 21

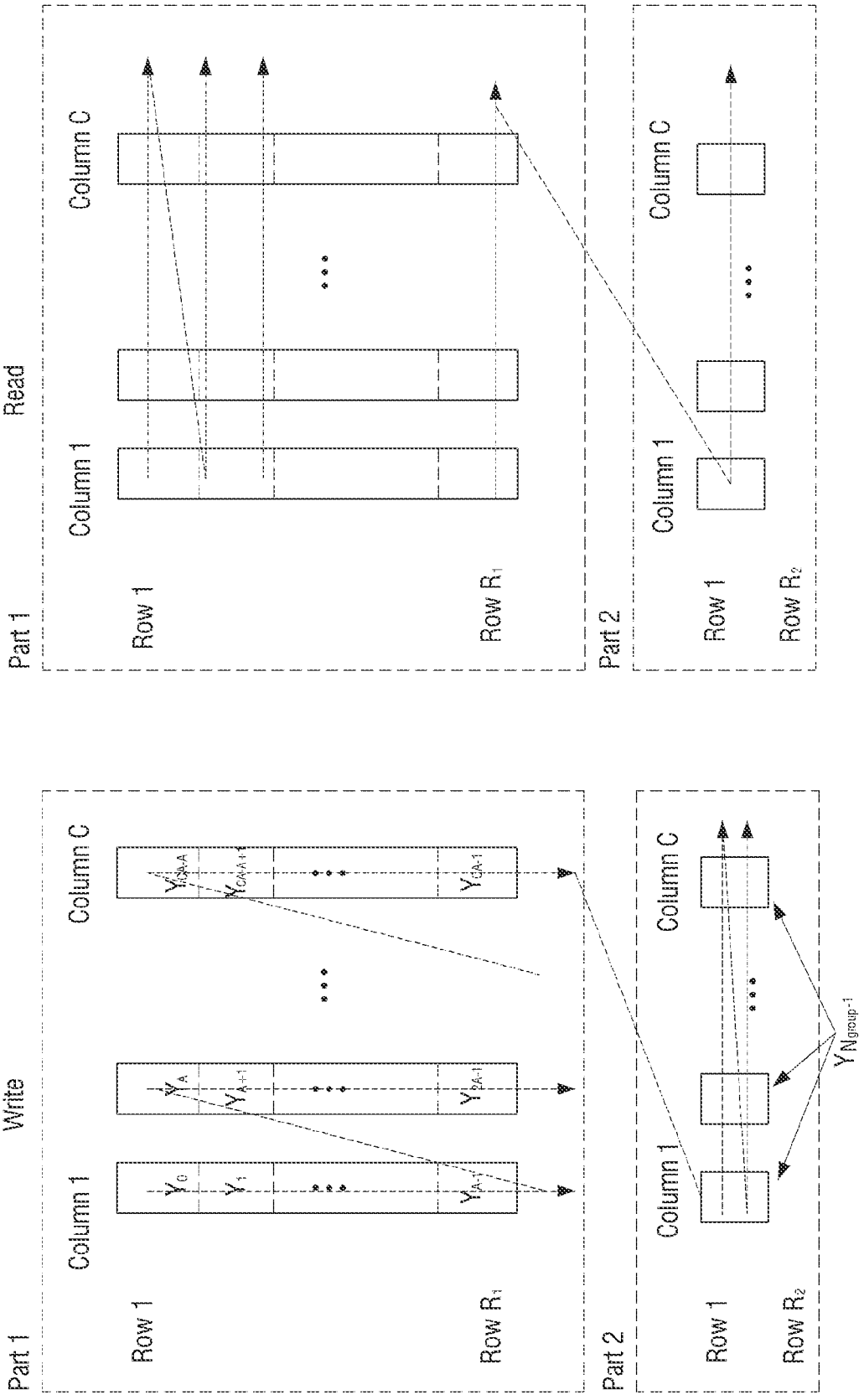

Read
Part 1
Column C
Column 1
Row 1
Row R_1
Part 2
Column C
Column 1
Row 1
Row R_2
Write
Part 1
Column C
Column 1
Row 1
Row R_1
Y_0
Y_1
Y_A
Y_{A+1}
$Y_{(A-1)*A}$
$Y_{(A-1)*A+1}$
Y_{A-1}
$Y_{(A-1)^2}$
Part 2
Column C
Column 1
Row 1
Row R_2
$Y_{N_{group}-1}$

FIG. 22

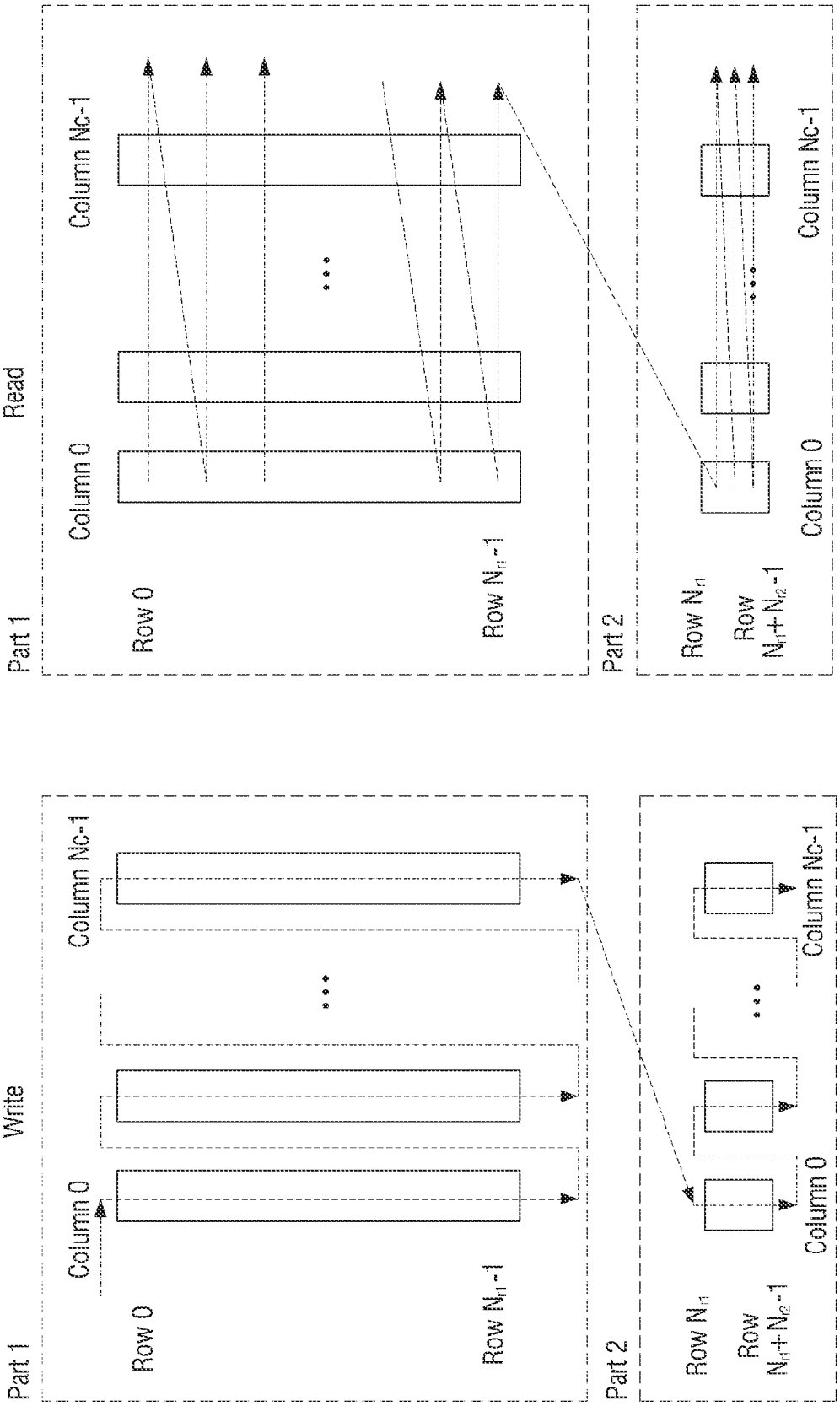

Read
Part 1
Column 0
Column Nc-1
Row 0
Row N_{r1}-1
Part 2
Row N_{r1}
Row N_{r1}+N_{r2}-1
Column 0
Column Nc-1
Write
Part 1
Column 0
Column Nc-1
Row 0
Row N_{r1}-1
Part 2
Row N_{r1}
Row N_{r1}+N_{r2}-1
Column 0
Column Nc-1

FIG. 23

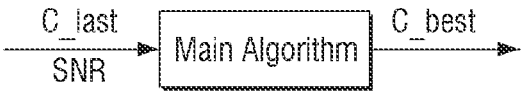


FIG. 24

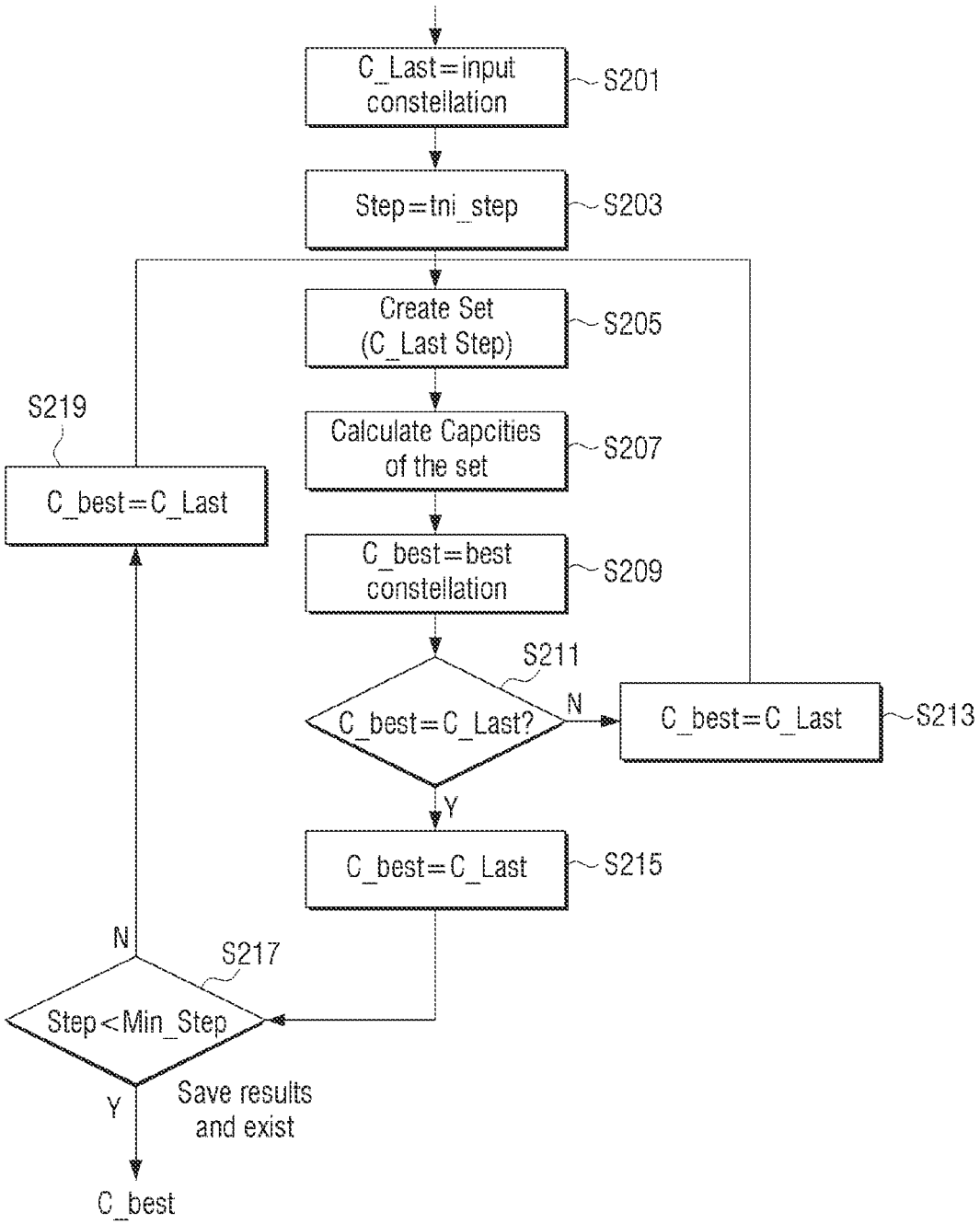

C_Last=input constellation
S201
Step=tni_step
S203
Create Set (C_Last Step)
S205
Calculate Capcities of the set
S207
C_best=best constellation
S209
S211
C_best=C_Last?
N
C_best=C_Last
S213
Y
C_best=C_Last
S215
S217
N
Step<Min_Step
Y
Save results and exist
C_best
S219
C_best=C_Last

FIG. 25

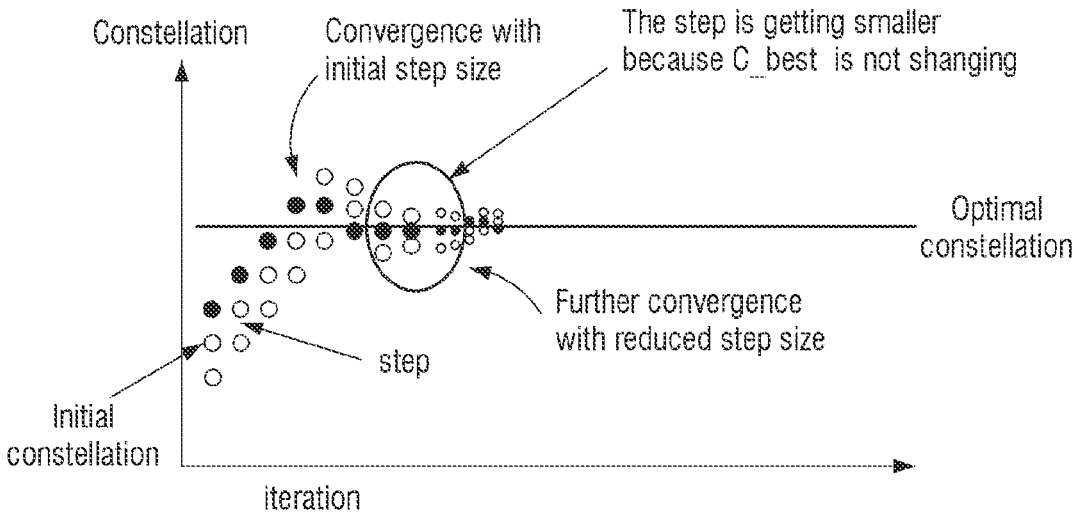

Constellation
Convergence with initial step size
The step is getting smaller because C_best is not shanging
Optimal constellation
Further convergence with reduced step size
step
Initial constellation
iteration

FIG. 26

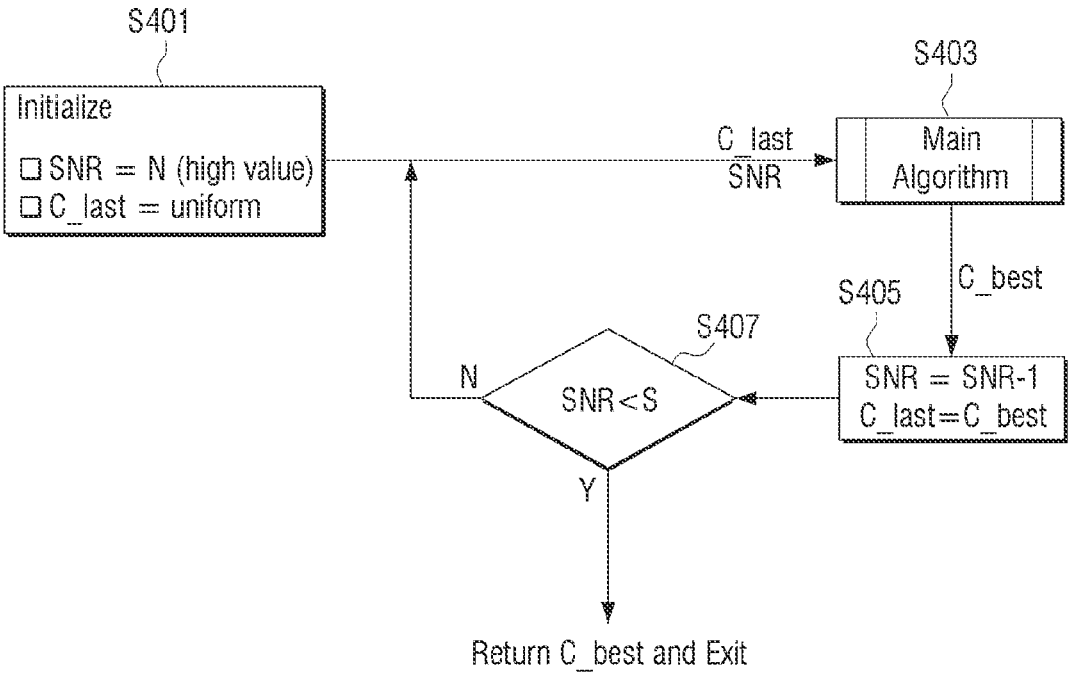

S401
Initialize
SNR = N (high value)
C_last = uniform
C_last
SNR
S403
Main Algorithm
C_best
S405
SNR = SNR-1
C_last=C_best
S407
SNR<S
N
Y
Return C_best and Exit

FIG. 27

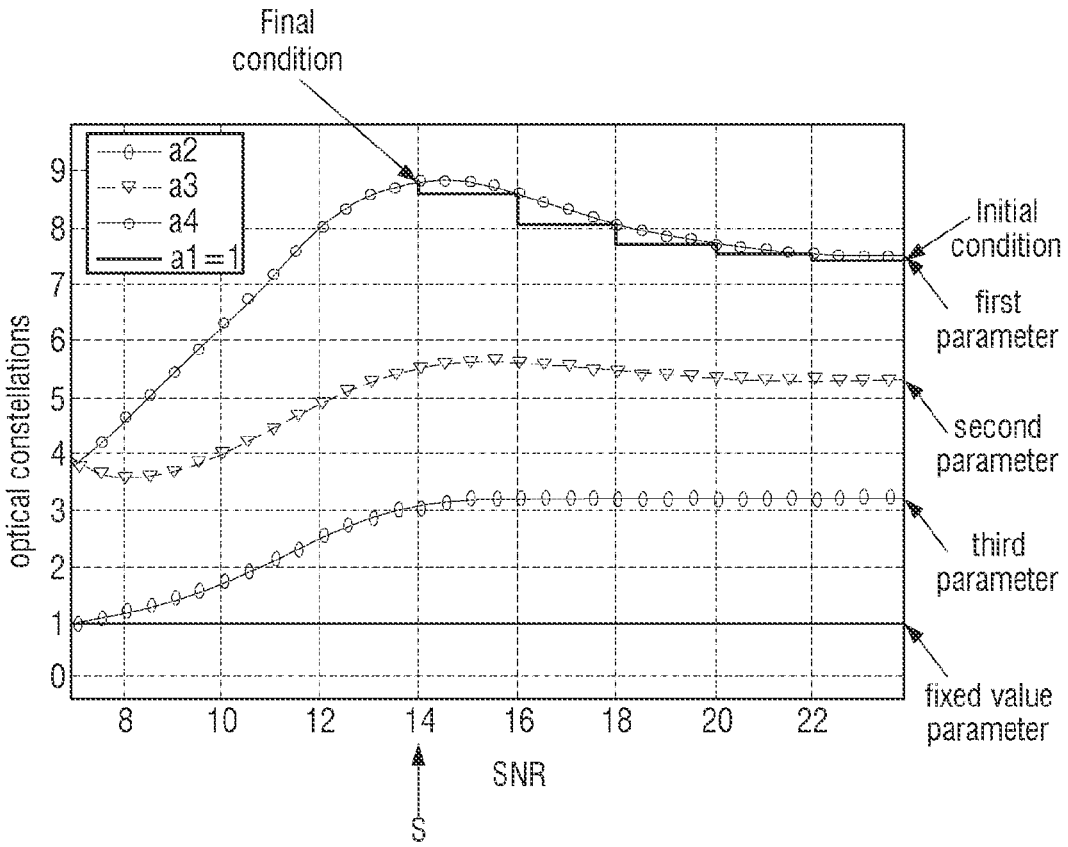

Final condition
a2
a3
a4
a1=1
optical constellations
0
1
2
3
4
5
6
7
8
9
8
10
12
14
16
18
20
22
SNR
S
Initial condition
first parameter
second parameter
third parameter
fixed value parameter

FIG. 28

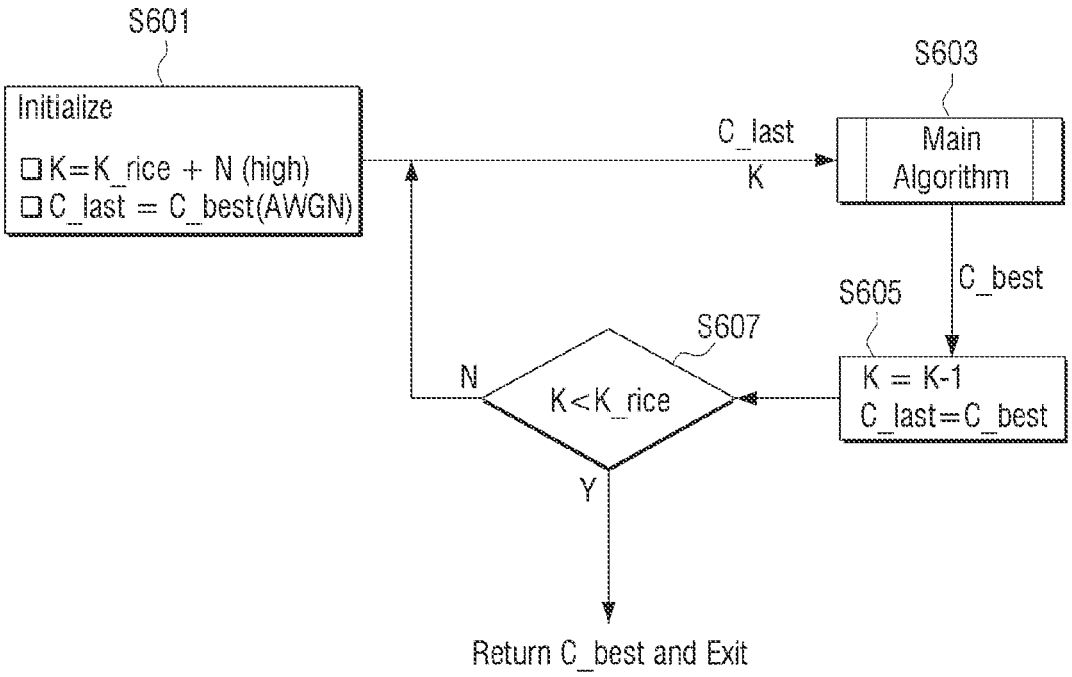

S601
Initialize
K=K_rice + N (high)
C_last = C_best(AWGN)
C_last
K
S603
Main Algorithm
C_best
S605
K = K-1
C_last=C_best
S607
K<K_rice
N
Y
Return C_best and Exit

FIG. 29

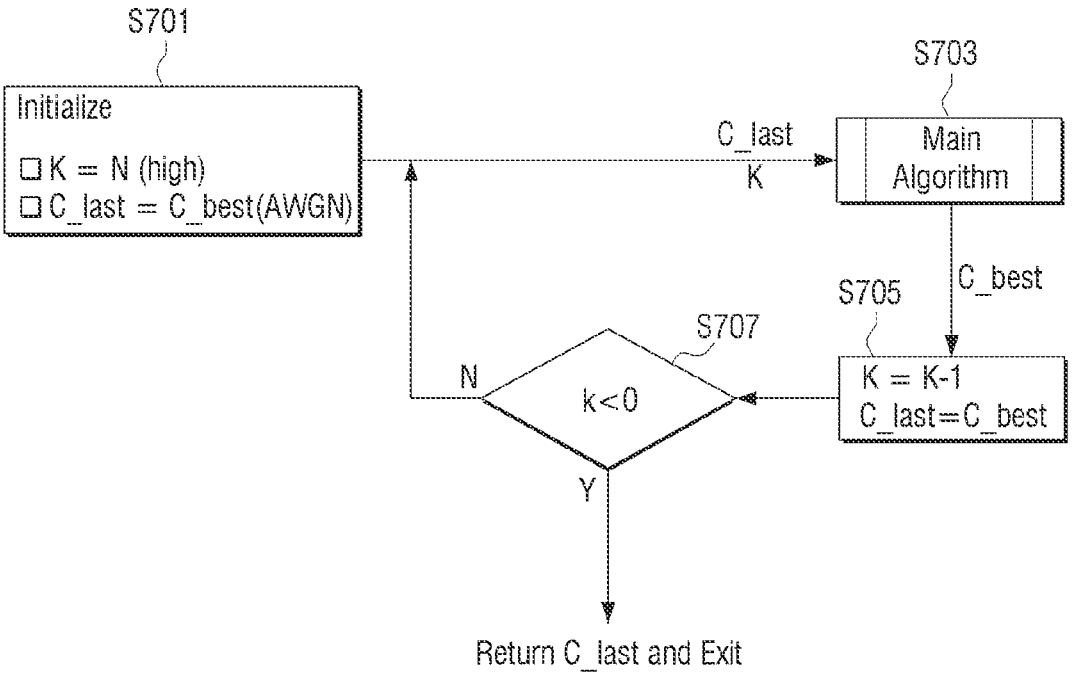

S701
Initialize
K = N (high)
C_last = C_best(AWGN)
C_last
K
S703
Main Algorithm
C_best
S705
K = K-1
C_last=C_best
S707
k<0
N
Y
Return C_last and Exit

FIG. 30

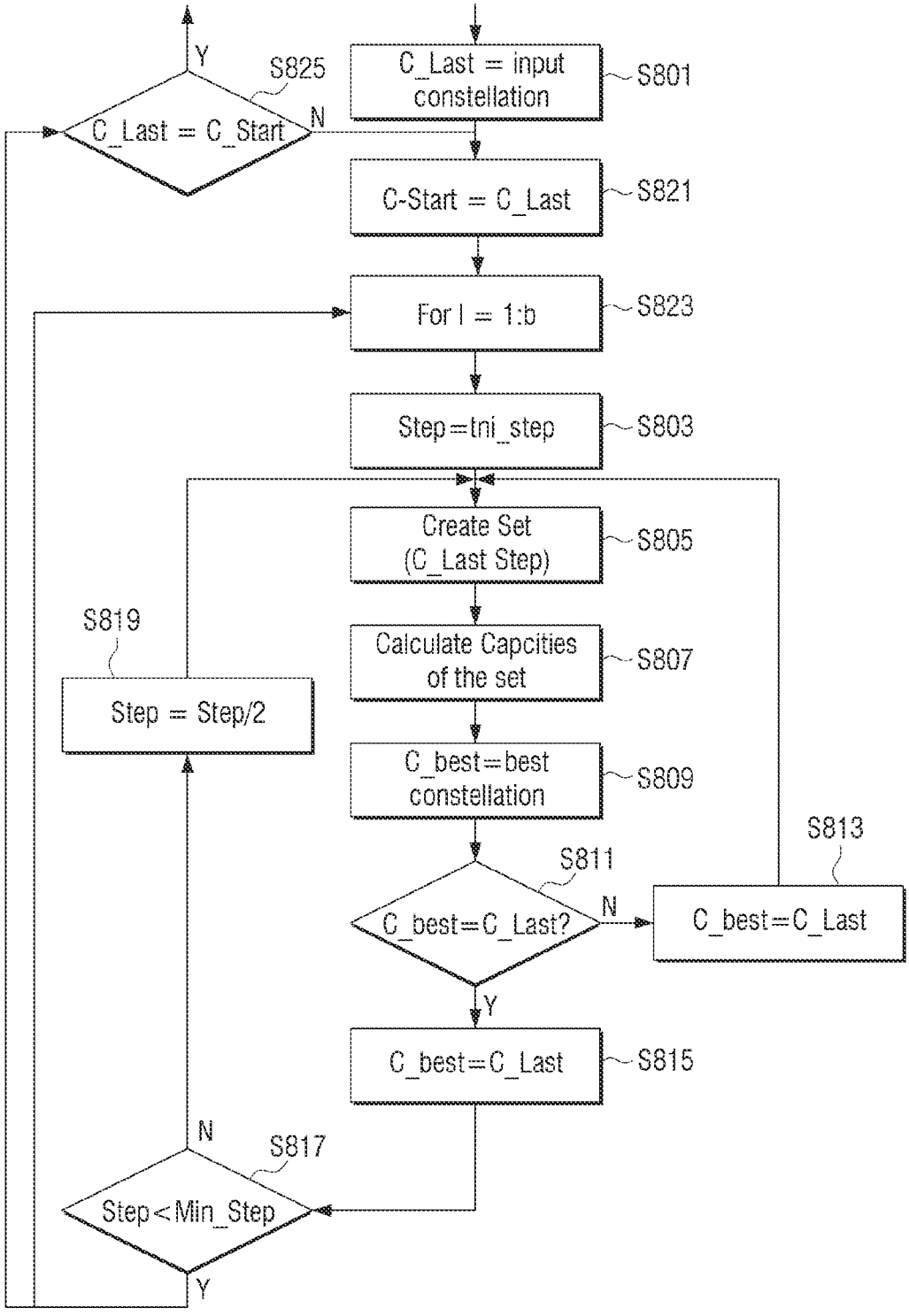

C_Last = input constellation
S801
C-Start = C_Last
S821
For l = 1:b
S823
Step=tni_step
S803
Create Set (C_Last Step)
S805
Calculate Capcities of the set
S807
C_best=best constellation
S809
C_best=C_Last?
S811
N
Y
C_best=C_Last
S813
C_best=C_Last
S815
Step<Min_Step
S817
N
Y
Step = Step/2
S819
C_Last = C_Start
S825
Y
N

FIG. 31

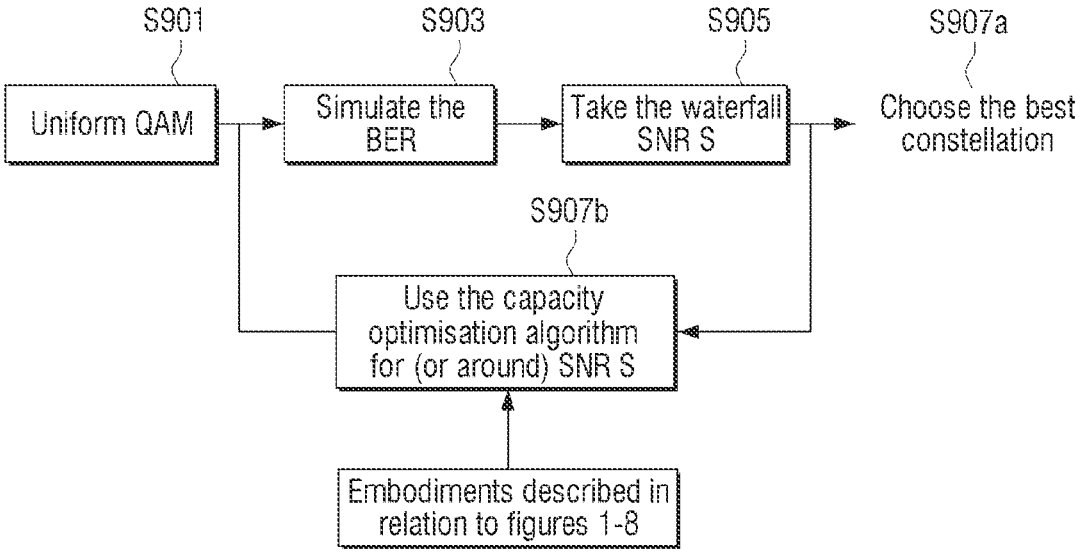

S901
S903
S905
S907a
Uniform QAM
Simulate the BER
Take the waterfall SNR S
Choose the best constellation
S907b
Use the capacity optimisation algorithm for (or around) SNR S
Embodiments described in relation to figures 1-8

FIG. 32

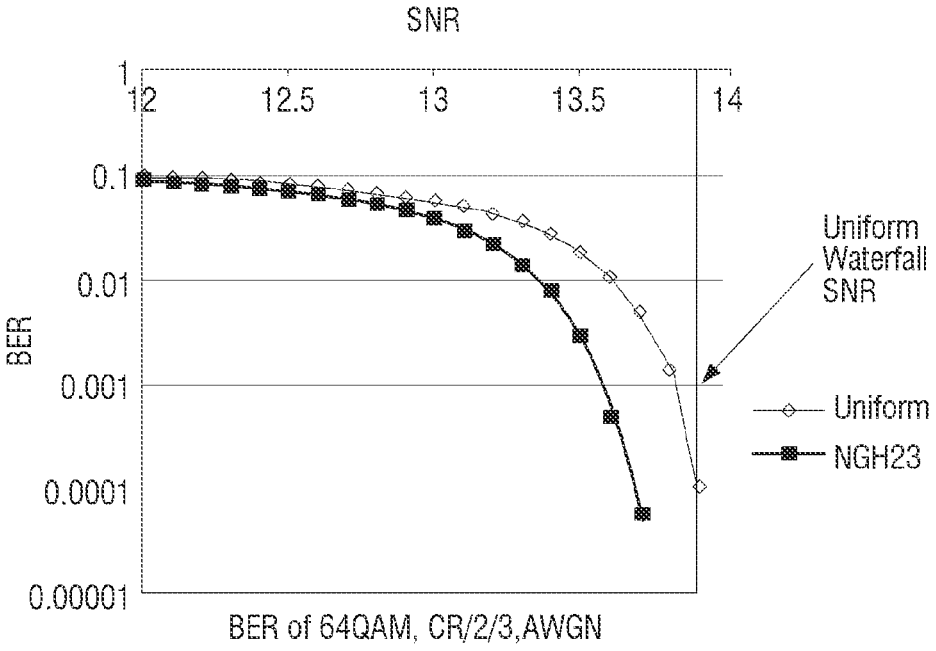

SNR
1
12
12.5
13
13.5
14
0.1
0.01
0.001
0.0001
0.00001
BER
Uniform
Waterfall
SNR
Uniform
NGH23
BER of 64QAM, CR/2/3,AWGN

FIG. 33

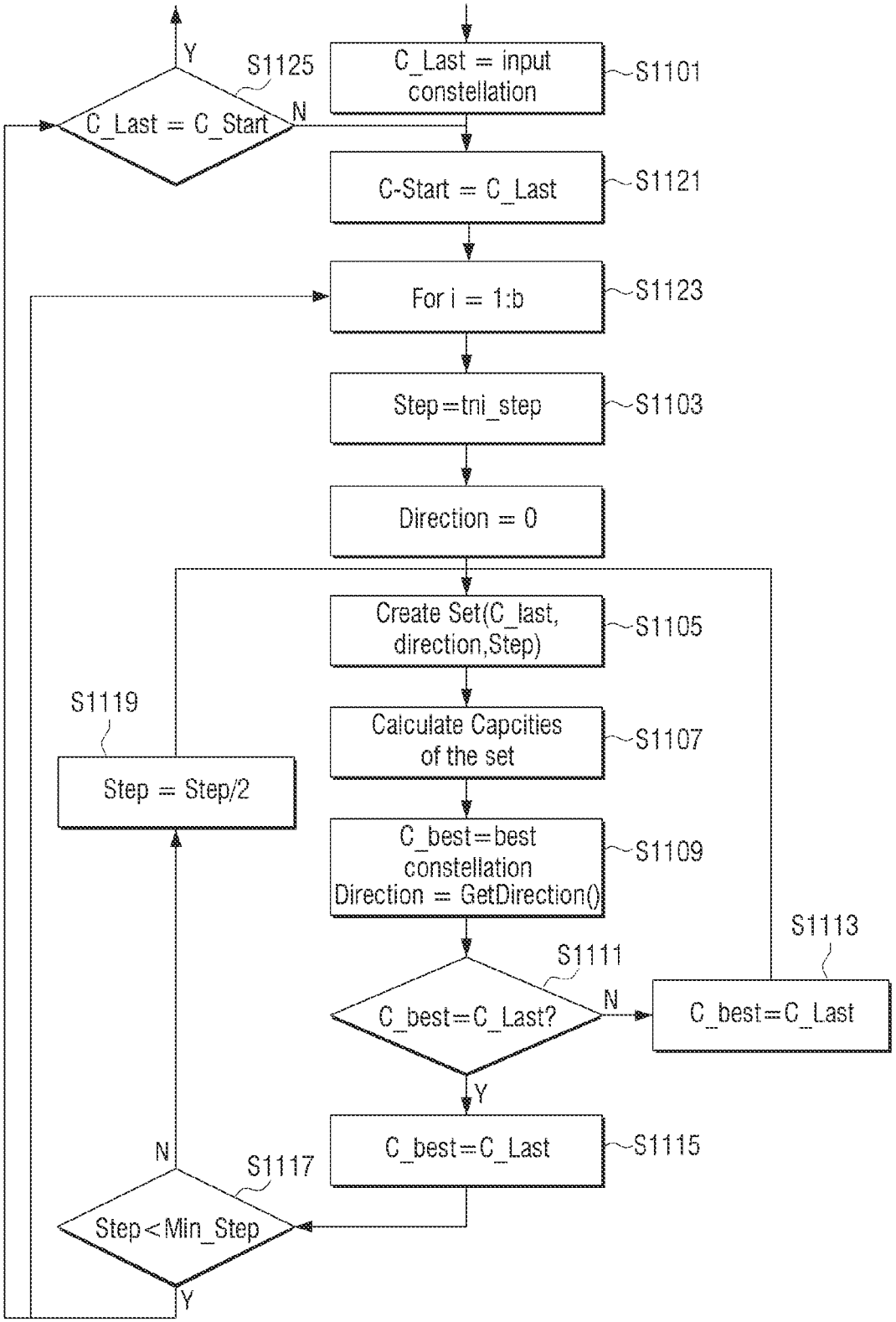
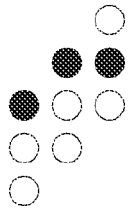

C_Last = input constellation
S1101
C-Start = C_Last
S1121
For i = 1:b
S1123
Step=tni_step
S1103
Direction = 0
Create Set(C_last, direction,Step)
S1105
Calculate Capcities of the set
S1107
C_best=best constellation Direction = GetDirection()
S1109
C_best=C_Last?
S1111
N
C_best=C_Last
S1113
Y
C_best=C_Last
S1115
Step<Min_Step
S1117
N
Y
Step = Step/2
S1119
C_Last = C_Start
S1125
Y
N

FIG. 34

Example1: (1D -> NUQAM)

- Direction is up or down



» First iteration all the 3 points are computed, the upper one is the best one.

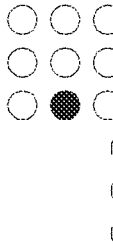
» second iteration only the top one is computed, the upper is the best one.

» Third iteration only the top one is computed, the middle is the best one.

» Instead of 9 points, We needed only 5.

Example2: (2D -> NUC)

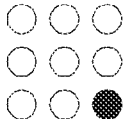
- Direction are 9 values.



» First iteration all the 9 points are computed, the bottom one is the best one.

» second iteration only the 3 bottom ones is computed, the conner is the best one.

» Third iteration only the 5 borders are computed, the middle is the best one.



» Instead of 9*3=27 points, we needed only 9+3+5=17.

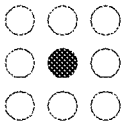


FIG. 35

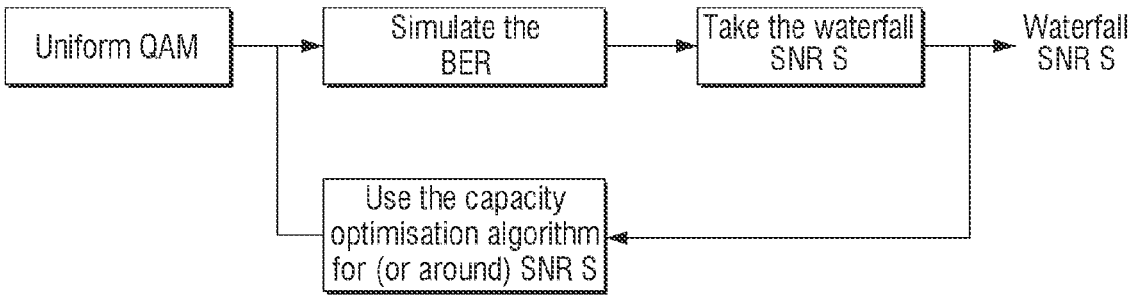

Uniform QAM
Simulate the BER
Take the waterfall SNR S
Waterfall SNR S
Use the capacity optimisation algorithm for (or around) SNR S

FIG. 36

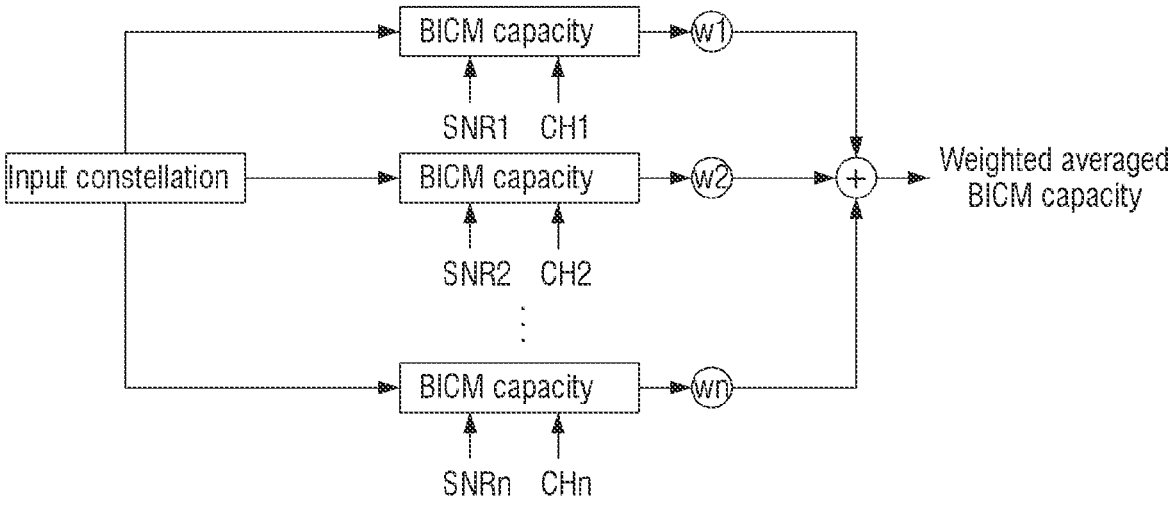

Input constellation
BICM capacity
BICM capacity
BICM capacity
SNR1
CH1
SNR2
CH2
SNRn
CHn
w1
w2
wn
+
Weighted averaged
BICM capacity

FIG. 37

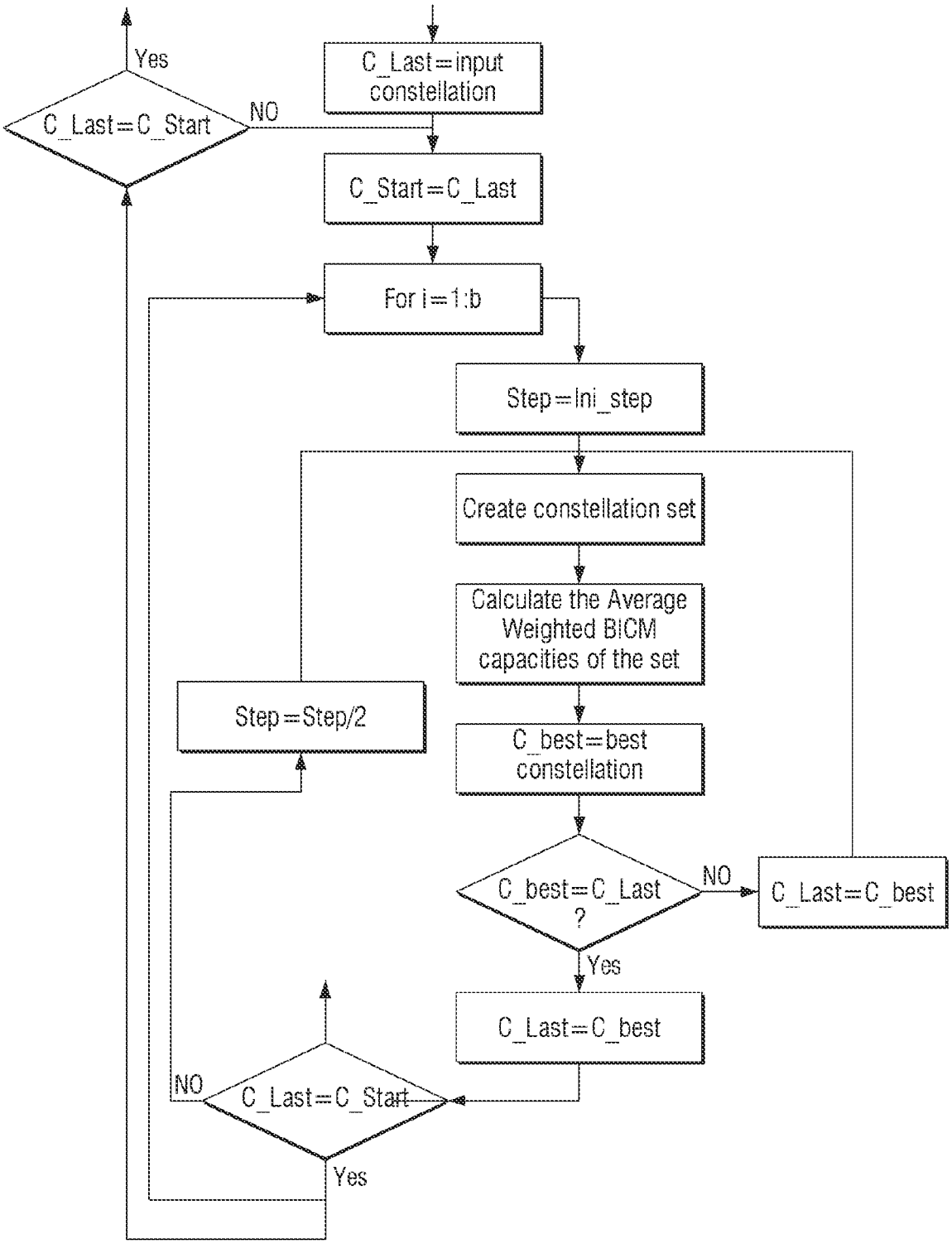

C_Last=input constellation
C_Last=C_Start
Yes
NO
C_Start=C_Last
For i=1:b
Step=Ini_step
Create constellation set
Calculate the Average Weighted BICM capacities of the set
C_best=best constellation
C_best=C_Last ?
NO
C_Last=C_best
Yes
C_Last=C_best
Step=Step/2
C_Last=C_Start
NO
Yes

FIG. 38A

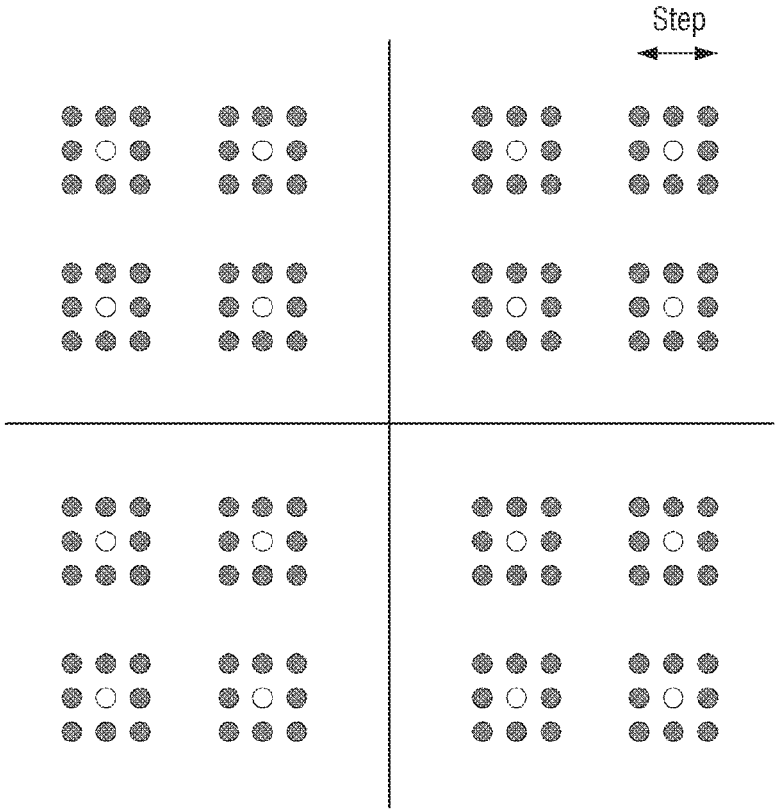


FIG. 38B

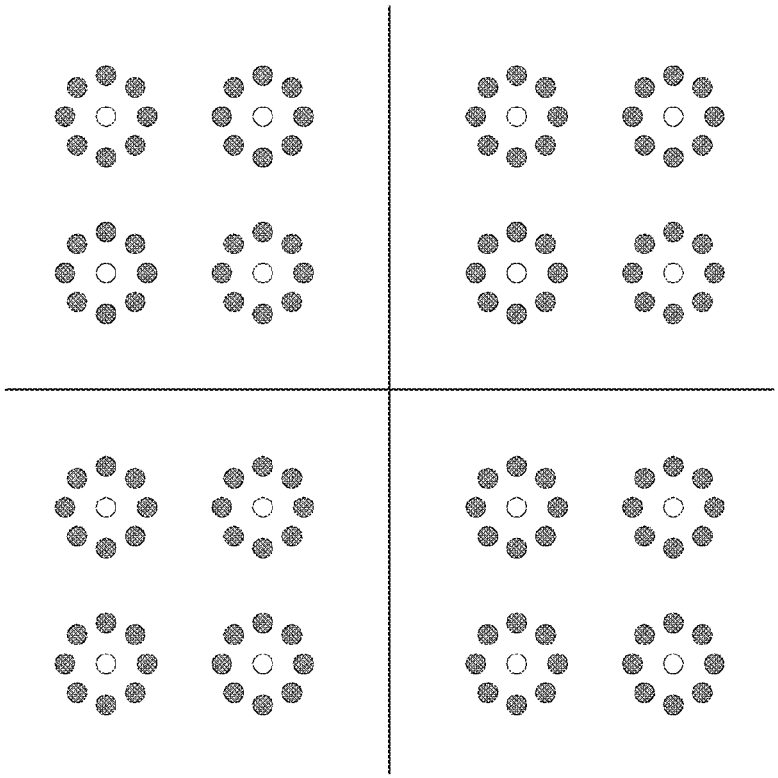


FIG. 39

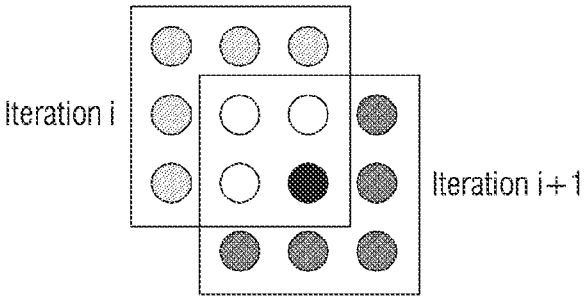


FIG. 40

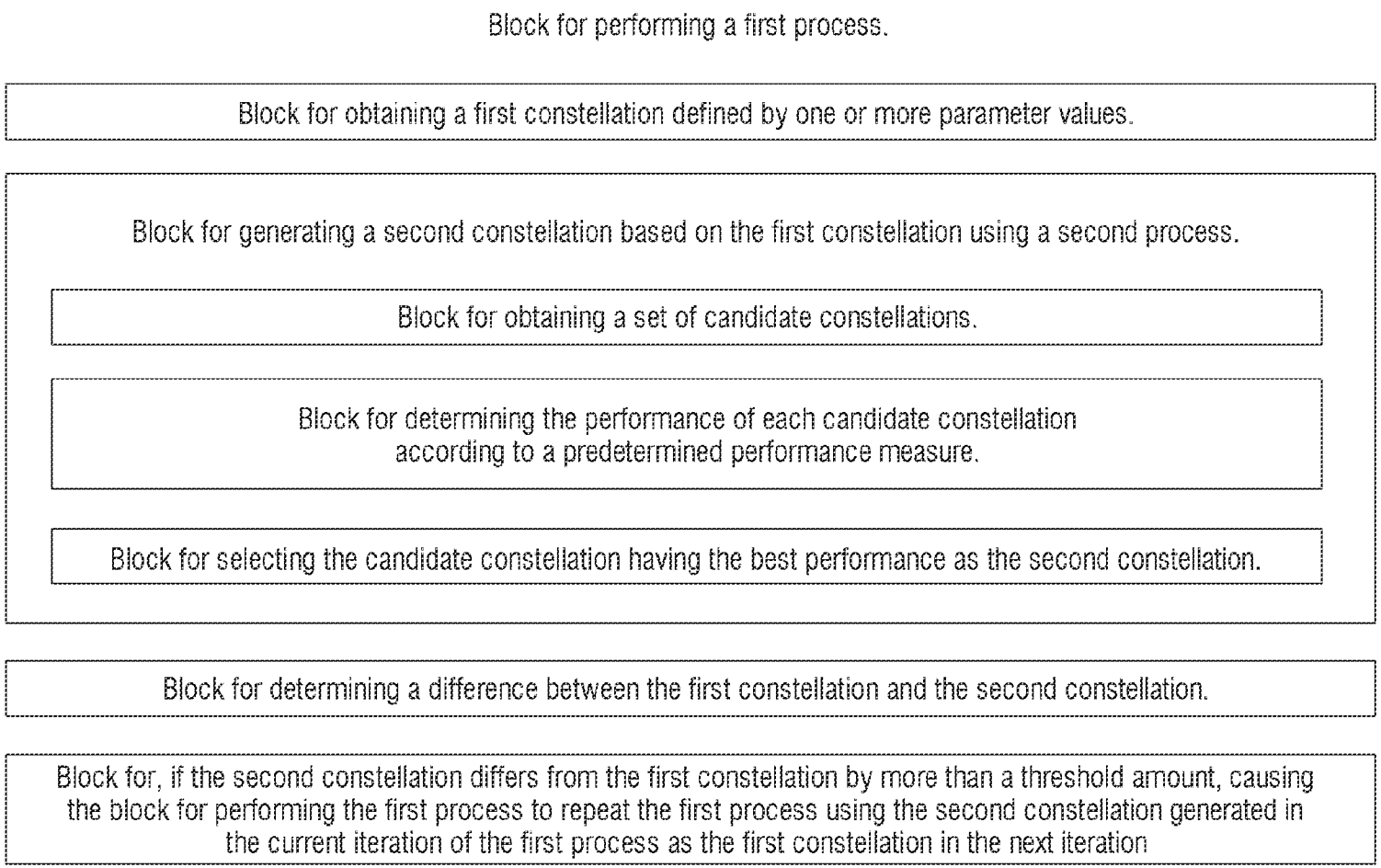

Block for performing a first process.
Block for obtaining a first constellation defined by one or more parameter values.
Block for generating a second constellation based on the first constellation using a second process.
Block for obtaining a set of candidate constellations.
Block for determining the performance of each candidate constellation according to a predetermined performance measure.
Block for selecting the candidate constellation having the best performance as the second constellation.
Block for determining a difference between the first constellation and the second constellation.
Block for, if the second constellation differs from the first constellation by more than a threshold amount, causing the block for performing the first process to repeat the first process using the second constellation generated in the current iteration of the first process as the first constellation in the next iteration

FIG. 41

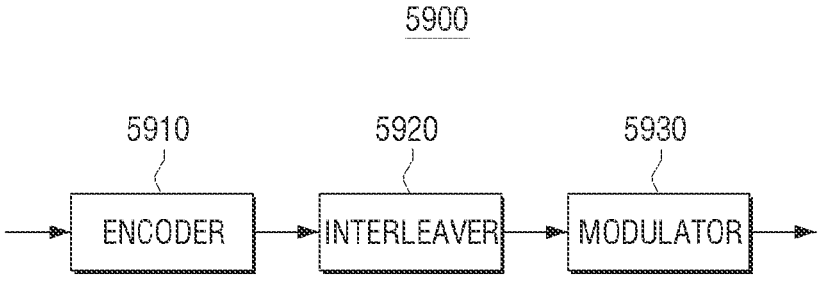


FIG. 42

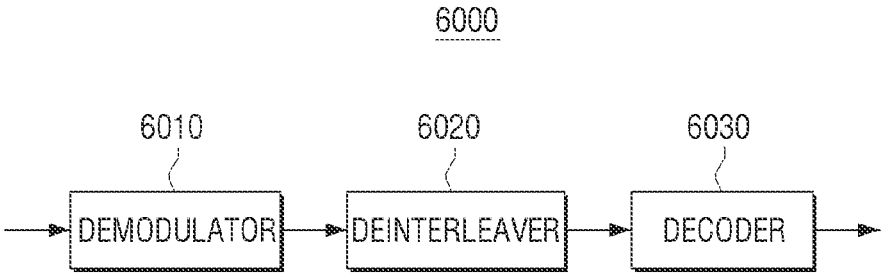


FIG. 43

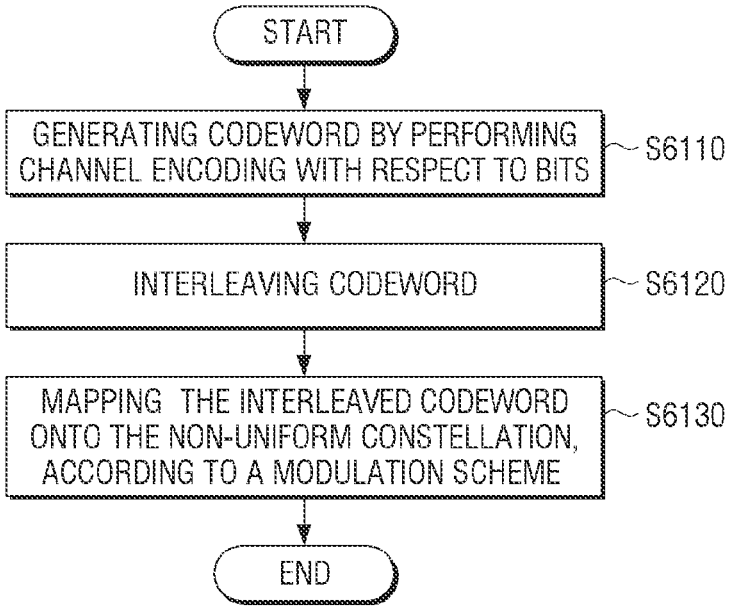

START
GENERATING CODEWORD BY PERFORMING CHANNEL ENCODING WITH RESPECT TO BITS
S6110
INTERLEAVING CODEWORD
S6120
MAPPING THE INTERLEAVED CODEWORD ONTO THE NON-UNIFORM CONSTELLATION, ACCORDING TO A MODULATION SCHEME
S6130
END

FIG. 44

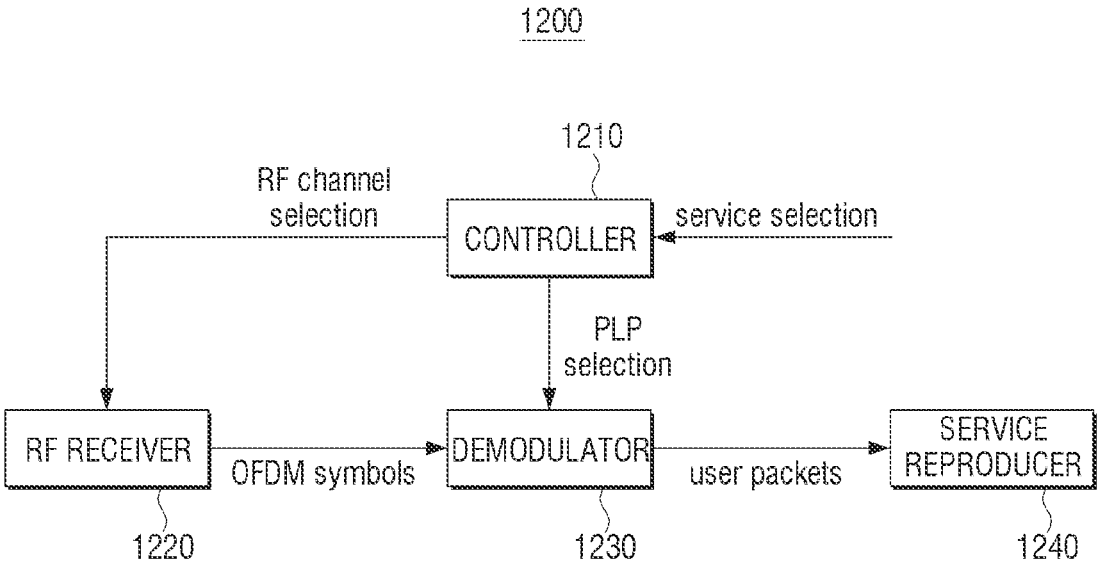

1200
1210
RF channel selection
CONTROLLER
service selection
PLP selection
RF RECEIVER
OFDM symbols
DEMODULATOR
user packets
SERVICE REPRODUCER
1220
1230
1240

FIG. 45

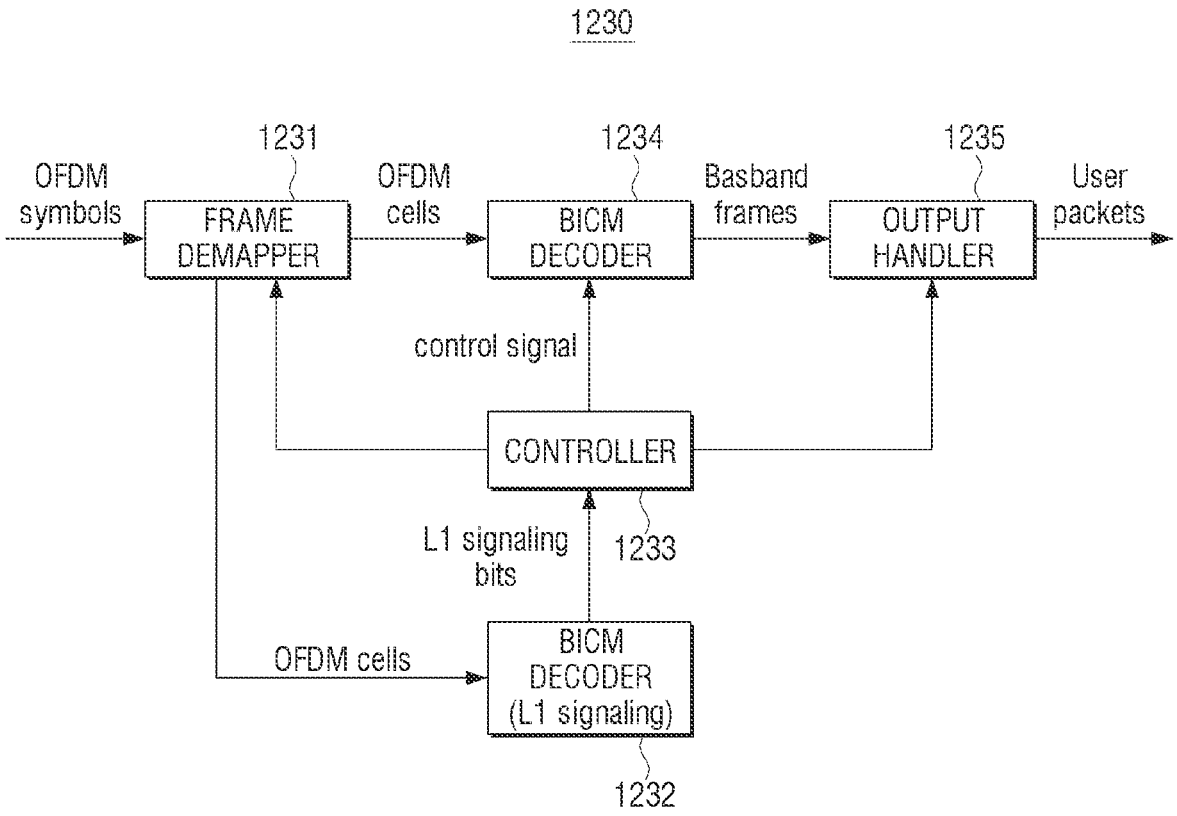
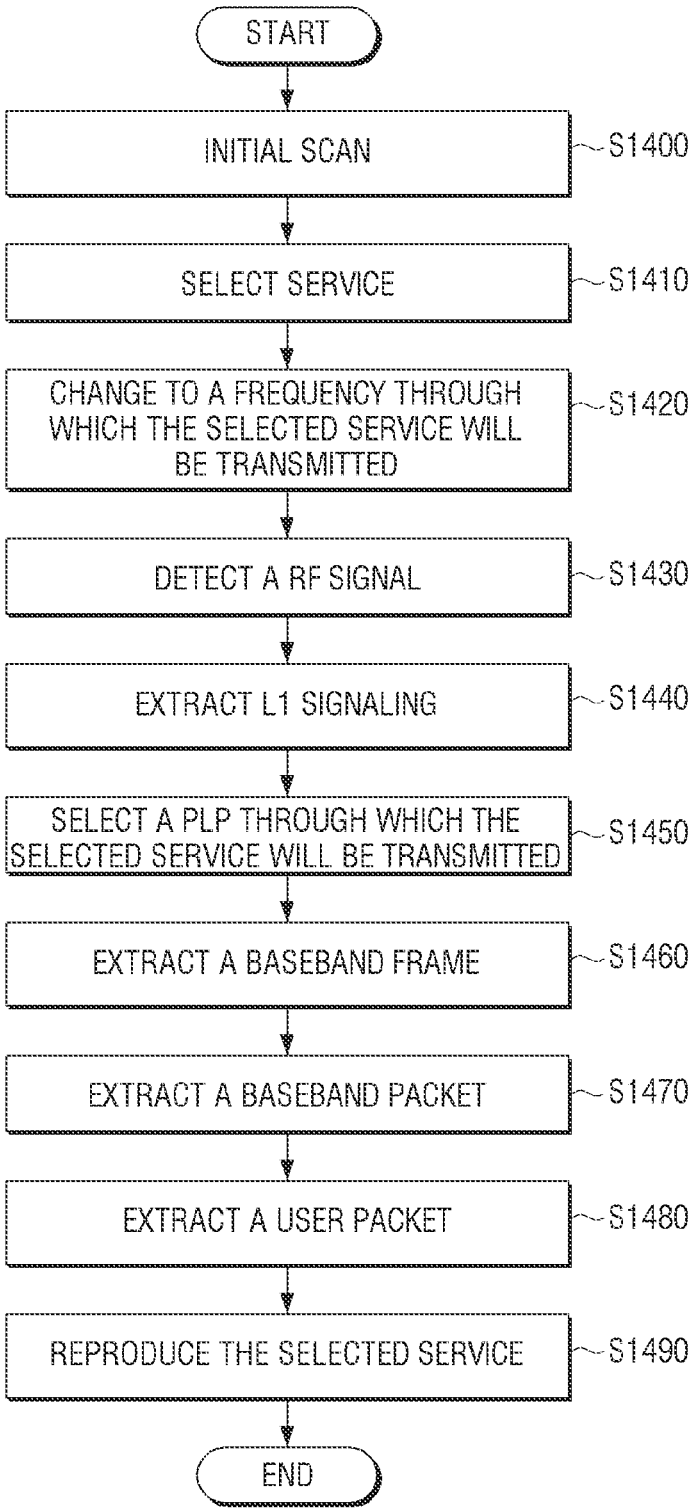

1230
1231
1234
1235
OFDM symbols
FRAME DEMAPPER
OFDM cells
BICM DECODER
Basband frames
OUTPUT HANDLER
User packets
control signal
CONTROLLER
L1 signaling bits
1233
OFDM cells
BICM DECODER (L1 signaling)
1232

FIG. 46


START
INITIAL SCAN
S1400
SELECT SERVICE
S1410
CHANGE TO A FREQUENCY THROUGH WHICH THE SELECTED SERVICE WILL BE TRANSMITTED
S1420
DETECT A RF SIGNAL
S1430
EXTRACT L1 SIGNALING
S1440
SELECT A PLP THROUGH WHICH THE SELECTED SERVICE WILL BE TRANSMITTED
S1450
EXTRACT A BASEBAND FRAME
S1460
EXTRACT A BASEBAND PACKET
S1470
EXTRACT A USER PACKET
S1480
REPRODUCE THE SELECTED SERVICE
S1490
END

TRANSMITTING APPARATUS AND MAPPING METHOD THEREOF

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This is a continuation of U.S. application Ser. No. 16/905,980 filed Jun. 19, 2020, which is a continuation of U.S. application Ser. No. 16/538,132 filed Aug. 12, 2019, which is a continuation of U.S. application Ser. No. 14/715,747 filed May 19, 2015, The disclosure of the aforementioned prior application is hereby incorporated by reference in its entirety.

BACKGROUND

1. Field

[0002] Apparatuses and methods consistent with exemplary embodiments of the inventive concept relate to transmitting and receiving date using broadcasting, more particularly, to the design of non-uniform constellations used in a Bit Interleaved Coded Modulation (BICM) mapping bits at an output of an encoder and interleaver to complex constellations.

2. Description of the Related Art

[0003] The current broadcasting systems consistent with the Digital Video Broadcasting Second Generation Terrestrial (DVB-T2) use a Bit Interleaved and Coded Modulation (BICM) chain in order to encode bits to be transmitted. The BICM chain includes a channel encoder like a Low Density Parity Check (LDPC) encoder followed by a Bit Interleaver and a Quadrature Amplitude Modulation (QAM) mapper. The role of the QAM mapper is to map different bits output from the channel encoder and interleaved using the Bit Interleaver to QAM cells. Each cell represents a complex number having real and imaginary part. The QAM mapper groups M bits into one cell. Each cell is translated into a complex number. M, which is the number of bits per cell, is equal to 2 for QPSK, 4 for 16QAM, 6 for 64QAM, and 8 for 256. It is possible to use a higher QAM size in order to increase a throughput. For example: 1K QAM is a constellation containing 1024 possible points and used to map M=10 bits. The DVB-T2 and previous standards use a uniform QAM. The uniform QAM has two important properties: possible points of constellation are rectangular, and spacing between each two successive points is uniform. The uniform QAM is very easy to map and demap.

[0004] The QAM is also easy to use since it does not need to be optimised as a function of the signal to noise ratio (SNR) or the coding rate of the channel code like the LDPC code. However, the capacity of the uniform QAM leaves a big gap from the theoretical limit, known as the Shannon limit. The performance in terms of bit error rate (BER) or frame error rate (FER) may be far from optimal.

SUMMARY

[0005] In order to reduce the gap from Shannon limit and provide a better BER/FER performance, a non-uniform constellation (NUC) is generated by relaxing the two properties of the uniform QAM, namely: the square shape and the uniform distance between constellations points.

[0006] It is an aim of certain exemplary embodiments of the present invention to address, solve and/or mitigate, at least partly, at least one of the problems and/or disadvantages associated with the related art, for example at least one of the problems and/or disadvantages described above. It is an aim of certain exemplary embodiments of the present invention to provide at least one advantage over the related art, for example at least one of the advantages described below.

[0007] The present invention is defined in the independent claims. Advantageous features are defined in the dependent claims.

[0008] Other aspects, advantages, and salient features of the invention will become apparent to those skilled in the art from the following detailed description, which, taken in conjunction with the annexed drawings, disclose exemplary embodiments of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The above and/or other aspects will be more apparent by describing certain exemplary embodiments with reference to the accompanying drawings, in which:

[0010] FIGS. **1**A to **12** are views to illustrate a transmitting apparatus according to exemplary embodiments;

[0011] FIGS. **13** to **18** are views to illustrate a receiving apparatus according to exemplary embodiments;

[0012] FIGS. **19** to **22** are views to illustrate an interleaving method of a block interleaver, according to exemplary embodiments;

[0013] FIG. **23** is a schematic diagram of a first algorithm according to an exemplary embodiment;

[0014] FIG. **24** is a flowchart illustrating the operations of the first algorithm, according to an exemplary embodiment;

[0015] FIG. **25** illustrates the convergence of C_last with respect to one of the parameters as the first algorithm of FIGS. **23** and **24** is performed, according to an exemplary embodiment;

[0016] FIG. **26** illustrates a second algorithm according to an exemplary embodiment for determining an optimal constellation at a given SNR value S in an AWGN channel;

[0017] FIG. **27** illustrates the convergence of the constellation C_best as the second algorithm of FIG. **4** is performed, according to an exemplary embodiment;

[0018] FIG. **28** illustrates a third algorithm according to an exemplary embodiment for determining the optimal constellation at a given SNR value S in a Rician fading channel for a desired Rician factor K_rice;

[0019] FIG. **29** illustrates a fourth algorithm according to an exemplary embodiment for determining the optimal constellation at a given SNR value S in a Rayleigh fading channel;

[0020] FIG. **30** illustrates a fifth algorithm according to an exemplary embodiment for determining an optimal constellation;

[0021] FIG. **31** illustrates a process for obtaining an optimal constellation for a specific system, according to an exemplary embodiment;

[0022] FIG. **32** illustrates an exemplary BER versus SNR plot for 64-QAM using a Low-Density Parity-Check, LDPC, coding rate (CR) of 2/3 from DVB-T2 in an AWGN channel, according to an exemplary embodiment;

[0023] FIG. **33** illustrates a sixth algorithm according to an exemplary embodiment for determining an optimal constellation;

[0024] FIG. **34** further illustrates the sixth algorithm illustrated in FIG. **33**, according to an exemplary embodiment;

[0025] FIG. **35** illustrates a process for obtaining the waterfall SNR for a certain channel type according to an exemplary embodiment;

[0026] FIG. **36** schematically illustrates a process for obtaining a weighted performance measure function for an input constellation based on different transmission scenarios according to an exemplary embodiment;

[0027] FIG. **37** illustrates a process for obtaining an optimum constellation according to an exemplary embodiment;

[0028] FIGS. **38**A and **38**B illustrate alternative schemes for generating a candidate constellation from a previous constellation according to exemplary embodiments;

[0029] FIG. **39** illustrates a technique for reducing complexity according to an exemplary embodiment;

[0030] FIG. **40** illustrates an apparatus for implementing an algorithm according to an exemplary embodiment;

[0031] FIG. **41** is a block diagram to describe a configuration of a transmitting apparatus according to an exemplary embodiment;

[0032] FIG. **42** is a block diagram to describe a configuration of a receiving apparatus according to an exemplary embodiment;

[0033] FIG. **43** is a flowchart to describe a modulation method according to an exemplary embodiment;

[0034] FIG. **44** is a block diagram illustrating a configuration of a receiving apparatus according to an exemplary embodiment;

[0035] FIG. **45** is a block diagram illustrating a demodulator according to an exemplary embodiment; and

[0036] FIG. **46** is a flowchart provided to illustrate an operation of a receiving apparatus from a moment when a user selects a service until the selected service is reproduced, according to an exemplary embodiment.

DETAILED DESCRIPTION OF THE EXEMPLARY EMBODIMENTS

[0037] Various exemplary embodiments will now be described in greater detail with reference to the accompanying drawings.

[0038] In the following description, same drawing reference numerals are used for the same elements even in different drawings. The matters defined in the description, such as detailed construction and elements, are provided to assist in a comprehensive understanding of the invention. Thus, it is apparent that the exemplary embodiments can be carried out without those specifically defined matters. Also, well-known functions or constructions are not described in detail since they would obscure the exemplary embodiments with unnecessary detail.

[0039] The following description of the exemplary embodiments with reference to the accompanying drawings is provided to assist in a comprehensive understanding of the inventive concept, as defined by the claims. The description includes various specific details to assist in that understanding but these are to be regarded as merely exemplary. Accordingly, those of ordinary skill in the art will recognize that various changes and modifications of the embodiments described herein can be made without departing from the scope of the inventive concept.

[0040] The same or similar components may be designated by the same or similar reference numerals, although they may be illustrated in different drawings.

[0041] Detailed descriptions of techniques, structures, constructions, functions or processes known in the art may be omitted for clarity and conciseness, and to avoid obscuring the subject matter of the exemplary embodiments.

[0042] The terms and words used herein are not limited to the bibliographical or standard meanings, but, are merely used by the inventors to enable a clear and consistent understanding of the exemplary embodiments.

[0043] Throughout the description and claims of this specification, the words "comprise", "contain" and "include", and variations thereof, for example "comprising", "containing" and "including", means "including but not limited to", and is not intended to (and does not) exclude other features, elements, components, integers, steps, operations, processes, functions, characteristics, and the like.

[0044] Throughout the description and claims of this specification, the singular form, for example "a", "an" and "the", encompasses the plural unless the context otherwise requires. For example, reference to "an object" includes reference to one or more of such objects.

[0045] Throughout the description and claims of this specification, language in the general form of "X for Y" (where Y is some action, process, function, activity or step and X is some means for carrying out that action, process, function, activity or step) encompasses means X adapted, configured or arranged specifically, but not necessarily exclusively, to do Y.

[0046] Features, elements, components, integers, steps, operations, processes, functions, characteristics, and the like, described in conjunction with a particular aspect, embodiment, example or claim of the inventive concept are to be understood to be applicable to any other aspect, embodiment, example or claim described herein unless incompatible therewith.

[0047] The exemplary embodiments may be implemented in the form of any suitable method, system and/or apparatus for use in digital broadcasting, for example in the form of a mobile/portable terminal (e.g. mobile telephone), hand-held device, personal computer, digital television and/or digital radio broadcast transmitter and/or receiver apparatus, set-top-box, etc. Any such system and/or apparatus may be compatible with any suitable existing or future digital broadcast system and/or standard, for example one or more of the digital broadcasting systems and/or standards referred to herein.

[0048] FIG. **1**A is provided to explain transmitting apparatus according to an exemplary embodiment.

[0049] According to FIG. **1**A, a transmitting apparatus **10000** according to an exemplary embodiment may include an Input Formatting Block (or part) **11000**, **11000-1**, a BIT Interleaved and Coded Modulation (BICM) block **12000**, **12000-1**, a Framing/Interleaving block **13000**, **13000-1** and a Waveform Generation block **14000**, **14000-1**.

[0050] The transmitting apparatus **10000** according to an exemplary embodiment illustrated in FIG. **1**A includes normative blocks shown by solid lines and informative blocks shown by dotted lines. Here, the blocks shown by solid lines are normal blocks, and the blocks shown by dotted lines are blocks which may be used when implementing an informative MIMO.

[0051] The Input Formatting block **11000**, **11000-1** generates a baseband frame (BBFRAME) from an input stream of data to be serviced. Herein, the input stream may be a transport stream (TS), Internet protocol (IP) stream, a generic stream (GS), a generic stream encapsulation (GSE), etc.

[0052] The BICM block **12000**, **12000-1** determines a forward error correction (FEC) coding rate and a constellation order depending on a region where the data to be serviced will be transmitted (e.g., a fixed PHY frame or mobile PHY frame), and then, performs encoding. Signaling information on the data to be serviced may be encoded through a separate BICM encoder (not illustrated) or encoded by sharing the BICM encoder **12000**, **12000-1** with the data to be serviced, depending on a system implementation.

[0053] The Framing/Interleaving block **13000**, **13000-1** combines time interleaved data with signaling information to generate a transmission frame.

[0054] The Waveform Generation block **14000**, **14000-1** generates an OFDM signal in the time domain on the generated transmission frame, modulates the generated OFDM signal to a radio frequency (RF) signal and transmits the modulated RF signal to a receiver.

[0055] FIGS. **1**B and **1**C are provided to explain methods of multiplexing according to an exemplary embodiment.

[0056] FIG. **1**B illustrates a block diagram to implement a Time Division Multiplexing according to an exemplary embodiment.

[0057] In the TDM system architecture, there are four main blocks (or parts): the Input Formatting block **11000**, the BICM block **12000**, the Framing/Interleaving block **13000**, and the Waveform Generation block **14000**.

[0058] Data is input and formatted in the Input Formatting block, and forward error correction applied and mapped to constellations in the BICM block **12000**. Interleaving, both time and frequency, and frame creation done in the Framing/Interleaving block **13000**. Subsequently, the output waveform is created in the Waveform Generation block **14000**.

[0059] FIG. **2**B illustrates a block diagram to implement a Layered Division Multiplexing (LDM) according to another exemplary embodiment.

[0060] In the LDM system architecture, there are several different blocks compared with the TDM system architecture. Specifically, there are two separate Input Formatting blocks **11000**, **11000-1** and BICM blocks **12000**, **12000-1**, one for each of the layers in LDM. These are combined before the Framing/Interleaving block **13000** in the LDM Injection block. The Waveform Generation block **14000** is similar to TDM.

[0061] FIG. **2** is a block diagram which illustrates detailed configuration of the Input Formatting block illustrated in FIG. **1**A.

[0062] As illustrated in FIG. **2**, the Input Formatting block **11000** consists of three blocks which control packets distributed into PLPs. Specifically, the Input Formatting block **11000** includes a packet encapsulation and compression block **11100**, a baseband framing block **11200** and a scheduler block **11300**.

[0063] Input data packets input to the Input Formatting block **11000** can consist of various types, but at the encapsulation operation these different types of packets become generic packets which configure baseband frames. Here, the format of generic packets is variable. It is possible to easily extract the length of the generic packet from the packet itself without additional information. The maximum length of the generic packet is 64 kB. The maximum length of the generic packet, including header, is four bytes. Generic packets must be of integer byte length.

[0064] The scheduler **11200** receives an input stream of encapsulated generic packets and forms them into physical layer pipes (PLPs), in the form of baseband frames. In the above-mentioned TDM system there may be only one PLP, called single PLP or S-PLP, or there may be multiple PLPs, called M-PLP. One service cannot use more than four PLPs. In the case of an LDM system consisting of two layers, two PLPs are used, one for each layer.

[0065] The scheduler **11200** receives encapsulated input packet streams and directs how these packets are allocated to physical layer resources. Specifically, the scheduler **11200** directs how the baseband framing block will output baseband frames.

[0066] The functional assets of the Scheduler **11200** are defined by data size(s) and time(s). The physical layer can deliver portions of data at these discrete times. The scheduler **11200** uses the inputs and information including encapsulated data packets, quality of service metadata for the encapsulated data packets, a system buffer model, constraints and configuration from system management, and creates a conforming solution in terms of configuration of the physical layer parameters. The corresponding solution is subject to the configuration and control parameters and the aggregate spectrum available.

[0067] Meanwhile, the operation of the Scheduler **11200** is constrained by combination of dynamic, quasi-static, and static configurations. The definition of these constraints is left to implementation.

[0068] In addition, for each service a maximum of four PLPs shall be used. Multiple services consisting of multiple time interleaving blocks may be constructed, up to a total maximum of 64 PLPs for bandwidths of 6, 7 or 8 MHz. The baseband framing block **11300**, as illustrated in FIG. **3**A, consists of three blocks, baseband frame construction **3100**, **3100-1**, . . . **3100-*n***, baseband frame header construction block **3200**, **3200-1**, . . . **3200-*n***, and the baseband frame scrambling block **3300**, **3300-1**, . . . **3300-*n***. In a M-PLP operation, the baseband framing block creates multiple PLPs as necessary.

[0069] A baseband frame **3500**, as illustrated in FIG. **3**B, consists of a baseband frame header **3500-1** and payload **3500-2** consisting of generic packets. Baseband frames have fixed length $K_{payload}$. Generic packets **3610-3650** shall be mapped to baseband frames **3500** in order. If generic packets **3610-3650** do not completely fit within a baseband frame, packets are split between the current baseband frame and the next baseband frame. Packet splits shall be in byte units only.

[0070] The baseband frame header construction block **3200**, **3200-1**, . . . **3200-*n*** configures the baseband frame header. The baseband frame header **3500-1**, as illustrated in FIG. **3**B, is composed of three parts, including the base header **3710**, the optional header (or option field **3720**) and the extension field **3730**. Here, the base header **3710** appears in every baseband frame, and the optional header **3720** and the extension field **3730** may not be present in every time.

[0071] The main feature of the base header **3710** is to provide a pointer including an offset value in bytes as an initiation of the next generic packet within the baseband frame. When the generic packet initiates the baseband frame, the pointer value becomes zero. If there is no generic packet which is initiated within the baseband frame, the pointer value is 8191, and a 2-byte base header may be used.

[0072] The extension field (or extension header) **3730** may be used later, for example, for the baseband frame packet counter, baseband frame time stamping, and additional signaling, etc.

[0073] The baseband frame scrambling block **3300**, **3300-1**, . . . **3300-*n*** scrambles the baseband frame.

[0074] In order to ensure that the payload data when mapped to constellations does not always map to the same point, such as when the payload mapped to constellations consists of a repetitive sequence, the payload data shall always be scrambled before forward error correction encoding.

[0075] The scrambling sequences shall be generated by a 16-bit shift register that has 9 feedback taps. Eight of the shift register outputs are selected as a fixed randomizing byte, where each bit from t his byte is used to individually XOR the corresponding input data. The data bits are XORed MSB to MSB and so on until LSB to LSB. The generator polynomial is $G(x)=1+X+X^3+X^6+X^7+X^{11}+X^{12}+X^{13}+X^{16}$.

[0076] FIG. **4** illustrates a shift register of a PRBS encoder for scrambling a baseband according to an exemplary embodiment, wherein loading of the sequence into the PRBS register, as illustrated in FIG. **4** and shall be initiated at the start of every baseband frame.

[0077] FIG. **5** is a block diagram provided to explain detailed configuration of the BICM block illustrated in FIG. **1**A.

[0078] As illustrated in FIG. **5**, the BICM block includes the FEC block **14100**, **14100-1**, . . . , **14100-*n***, Bit Interleaver block **14200**, **14200-1**, . . . , **14200-*n*** and Mapper blocks **14300**, **14300-1**, . . . , **14300-*n***.

[0079] The input to the FEC block **1400**, **14100-1**, . . . , **14100-*n*** is a Baseband frame, of length $K_{payload}$, and the output from the FEC block is a FEC frame. The FEC block **14100**, **14100-1**, . . . , **14100-*n*** is implemented by concatenation of an outer code and an inner code with the information part. The FEC frame has length N_{inner}. There are two different lengths of LDPC code defined: N_{inner}=64800 bits and N_{inner}=16200 bits

[0080] The outer code is realized as one of either Bose, Ray-Chaudhuri and Hocquenghem (BCH) outer code, a Cyclic Redundancy Check (CRC) or other code. The inner code is realized as a Low Density Parity Check (LDPC) code. Both BCH and LDPC FEC codes are systematic codes where the information part I contained within the codeword. The resulting codeword is thus a concatenation of information or payload part, BCH or CRC parities and LDPC parities, as shown in FIG. **6**A.

[0081] The use of LDPC code is mandatory and is used to provide the redundancy needed for the code detection. There are two different LDPC structures that are defined, these are called Type A and Type B. Type A has a code structure that shows better performance at low code rates while Type B code structure shows better performance at high code rates. In general N_{inner}=64800 bit codes are expected to be employed. However, for applications where latency is critical, or a simpler encoder/decoder structure is preferred, N_{inner}=16200 bit codes may also be used.

[0082] The outer code and CRC consist of adding M_{outer} bits to the input baseband frame. The outer BCH code is used to lower the inherent LDPC error floor by correcting a predefined number of bit errors. When using BCH codes the length of M_{outer} is 192 bits (N_{inner}=64800 bit codes) and 168 bits (for N_{inner}=16200 bit codes). When using CRC the length of M_{outer} is 32 bits. When neither BCH nor CRC are used the length of M_{outer} is zero. The outer code may be omitted if it is determined that the error correcting capability of the inner code is sufficient for the application. When there is no outer code the structure of the FEC frame is as shown in FIG. **6**B.

[0083] FIG. **7** is a block diagram provided to explain detailed configuration of the Bit Interleaver block illustrated in FIG. **5**.

[0084] The LDPC codeword of the LDPC encoder, i.e., a FEC Frame, shall be bit interleaved by a Bit Interleaver block **14200**. The Bit Interleaver block **14200** includes a parity interleaver **14210**, a group-wise interleaver **14220** and a block interleaver **14230**. Here, the parity interleaver is not used for Type A and is only used for Type B codes.

[0085] The parity interleaver **14210** converts the staircase structure of the parity-part of the LDPC parity-check matrix into a quasi-cyclic structure similar to the information-part of the matrix.

[0086] Meanwhile, the parity interleaved LDPC coded bits are split into $N_{group}=N_{inner}/360$ bit groups, and the group-wise interleaver **14220** rearranges the bit groups.

[0087] The block interleaver **14230** block interleaves the group-wise interleaved LDPC codeword.

[0088] Specifically, the block interleaver **14230** divides a plurality of columns into part 1 and part 2 based on the number of columns of the block interleaver **14230** and the number of bits of the bit groups. In addition, the block interleaver **14230** writes the bits into each column configuring part 1 column wise, and subsequently writes the bits into each column configuring part 2 column wise, and then reads out row wise the bits written in each column.

[0089] In this case, the bits constituting the bit groups in the part 1 may be written into the same column, and the bits constituting the bit groups in the part 2 may be written into at least two columns.

[0090] Back to FIG. **5**, the Mapper block **14300**, **14300-1**, . . . , **14300-*n*** maps FEC encoded and bit interleaved bits to complex valued quadrature amplitude modulation (QAM) constellation points. For the highest robustness level, quaternary phase shift keying (QPSK) is used. For higher order constellations (16-QAM up to 4096-QAM), non-uniform constellations are defined and the constellations are customized for each code rate.

[0091] Each FEC frame shall be mapped to a FEC block by first de-multiplexing the input bits into parallel data cell words and then mapping these cell words into constellation values.

[0092] FIG. **8** is a block diagram provided to explain detailed configuration of a Framing/Interleaving block illustrated in FIG. **1**A.

[0093] As illustrated in FIG. **8**, the Framing/Interleaving block **14300** includes a time interleaving block **14310**, a framing block **14320** and a frequency interleaving block **14330**.

[0094] The input to the time interleaving block **14310** and the framing block **14320** may consist of M-PLPs however the output of the framing block **14320** is OFDM symbols, which are arranged in frames. The frequency interleaver included in the frequency interleaving block **14330** operates an OFDM symbols.

[0095] The time interleaver (TI) configuration included in the time interleaving block **14310** depends on the number of PLPs used. When there is only a single PLP or when LDM

is used, a sheer convolutional interleaver is used, while for multiple PLP a hybrid interleaver consisting of a cell interleaver, a block interleaver and a convolutional interleaver is used. The input to the time interleaving block **14310** is a stream of cells output from the mapper block (FIG. **5**, **14300**, **14300-1**, . . . , **14300-***n*), and the output of the time interleaving block **14310** is also a stream of time-interleaved cells.

[0096] FIG. **9**A illustrates the time interleaving block for a single PLP (S-PLP), and it consists of a convolutional interleaver only.

[0097] FIG. **9**B illustrates the time interleaving block for a plurality of PLPs (M-PLP), and it can be divided in several sub-blocks as illustrated.

[0098] The framing block **14320** maps the interleaved frames onto at least one transmitter frame. The framing block **14320**, specifically, receives inputs (e.g. data cell) from at least one physical layer pipes and outputs symbols.

[0099] In addition, the framing block **14320** creates at least one special symbol known as preamble symbols. These symbols undergo the same processing in the waveform block mentioned later.

[0100] FIG. **10** is a view illustrating an example of a transmission frame according to an exemplary embodiment.

[0101] As illustrated in FIG. **10**, the transmission frame consists of three parts, the bootstrap, preamble and data payload. Each of the three parts consists of at least one symbol.

[0102] Meanwhile, the purpose of the frequency interleaving block **14330** is to ensure that sustained interference in one part of the spectrum will not degrade the performance of a particular PLP disproportionately compared to other PLPs. The frequency interleaver **14330**, operating on the all the data cells of one OFDM symbol, maps the data cells from the framing block **14320** onto the N data carriers.

[0103] FIG. **11** is a block diagram provided to explain detailed configuration of a Waveform Generation block illustrated in FIG. **1**A.

[0104] As illustrated in FIG. **11**, the Waveform Generation block **14000** includes a pilot inserting block **14100**, a MISO block **14200**, an IFFT block **14300**, a PAPR block **14400**, a GI inserting block **14500** and a bootstrap block **14600**.

[0105] The pilot inserting block **14100** inserts a pilot to various cells within the OFDM frame.

[0106] Various cells within the OFDM frame are modulated with reference information whose transmitted value is known to the receiver.

[0107] Cells containing the reference information are transmitted at a boosted power level. The cells are called scattered, continual, edge, preamble or frame-closing pilot cells. The value of the pilot information is derived from a reference sequence, which is a series of values, one for each transmitted carrier on any given symbol.

[0108] The pilots can be used for frame synchronization, frequency synchronization, time synchronization, channel estimation, transmission mode identification and can also be used to follow the phase noise.

[0109] The pilots are modulated according to reference information, and the reference sequence is applied to all the pilots (e.g. scattered, continual edge, preamble and frame closing pilots) in every symbol including preamble and the frame-closing symbol of the frame.

[0110] The reference information, taken from the reference sequence, is transmitted in scattered pilot cells in every symbol except the preamble and the frame-closing symbol of the frame.

[0111] In addition to the scattered pilots described above, a number of continual pilots are inserted in every symbol of the frame except for Preamble and the frame-closing symbol. The number and location of continual pilots depends on both the FFT size and scattered pilot pattern in use.

[0112] The MISO block **14200** applies a MISO processing.

[0113] The Transmit Diversity Code Filter Set is a MISO pre-distortion technique that artificially decorrelates signals from multiple transmitters in a Single Frequency Network in order to minimize potential destructive interference. Linear frequency domain filters are used so that the compensation in the receiver can be implemented as part of the equalizer process. The filter design is based on creating all-pass filters with minimized cross-correlation over all filter pairs under the constraints of the number of transmitters $M \in \{2,3,4\}$ and the time domain span of the filters $N \in \{64,256\}$. The longer time domain span filters will increase the decorrelation level, but the effective guard interval length will be decreased by the filter time domain span and this should be taken into consideration when choosing a filter set for a particular network topology.

[0114] The IFFT block **14300** specifies the OFDM structure to use for each transmission mode. The transmitted signal is organized in frames. Each frame has a duration of T_F, and consists of L_F OFDM symbols. N frames constitute one super-frame. Each symbol is constituted by a set of K_{total} carriers transmitted with a duration T_S. Each symbol is composed of a useful part with duration T_U and a guard interval with a duration Δ. The guard interval consists of a cyclic continuation of the useful part, T_U, and is inserted before it.

[0115] The PAPR block **14400** applies the Peak to Average Power Reduction technique.

[0116] The GI inserting block **14500** inserts the guard interval into each frame.

[0117] The bootstrap block **14600** prefixes the bootstrap signal to the front of each frame.

[0118] FIG. **12** is a block diagram provided to explain a configuration of signaling information according to an exemplary embodiment.

[0119] The input processing block **11000** includes a scheduler **11200**. The BICM block **15000** includes an L1 signaling generator **15100**, an FEC encoder **15200-1** and **15200-2**, a bit interleaver **15300-2**, a demux **15400-2**, constellation mappers **15500-1** and **15500-2**. The L1 signaling generator **15100** may be included in the input processing block **11000**, according to an exemplary embodiment.

[0120] An n number of service data are mapped to a PLP0 to a PLPn respectively. The scheduler **11200** determines a position, modulation and coding rate for each PLP in order to map a plurality of PLPs to a physical layer of T2. In other words, the scheduler **11200** generates L1 signaling information. The scheduler **11200** may output dynamic field information among L1 post signaling information of a current frame, using the framing/Interleaving block **13000** (FIG. **1**) which may be referred to as a frame builder. Further, the scheduler **11200** may transmit the L1 signaling information

to the BICM block **15000**. The L1 signaling information includes L1 pre signaling information and L1 post signaling information.

[0121] The L1 signaling generator **15100** may differentiate the L1 pre signaling information from the L1 post signaling information to output them. The FEC encoders **15200-1** and **15200-2** perform respective encoding operations which include shortening and puncturing for the L1 pre signaling information and the L1 post signaling information. The bit interleaver **15300-2** performs interleaving by bit for the encoded L1 post signaling information. The demux **15400-2** controls robustness of bits by modifying an order of bits constituting cells and outputs the cells which include bits. Two constellation mappers **15500-1** and **15500-2** map the L1 pre signaling information and the L1 post signaling information to constellations, respectively. The L1 pre signaling information and the L1 post signaling information processed through the above described processes are output to be included in each frame by the Framing/Interleaving block **13000** (FIG. **1**).

[0122] FIG. **13** illustrates a structure of an receiving apparatus according to an embodiment of the present invention.

[0123] The apparatus **20000** for receiving broadcast signals according to an embodiment of the present invention can correspond to the apparatus **10000** for transmitting broadcast signals, described with reference to FIG. **1**. The apparatus **20000** for receiving broadcast signals according to an embodiment of the present invention can include a synchronization & demodulation module **21000**, a frame parsing module **22000**, a demapping & decoding module **23000**, an output processor **24000** and a signaling decoding module **25000**. A description will be given of operation of each module of the apparatus **20000** for receiving broadcast signals.

[0124] The synchronization & demodulation module **21000** can receive input signals through m Rx antennas, perform signal detection and synchronization with respect to a system corresponding to the apparatus **20000** for receiving broadcast signals and carry out demodulation corresponding to a reverse procedure of the procedure performed by the apparatus **10000** for transmitting broadcast signals.

[0125] The frame parsing module **22000** can parse input signal frames and extract data through which a service selected by a user is transmitted. If the apparatus **10000** for transmitting broadcast signals performs interleaving, the frame parsing module **22000** can carry out deinterleaving corresponding to a reverse procedure of interleaving. In this case, the positions of a signal and data that need to be extracted can be obtained by decoding data output from the signaling decoding module **25200** to restore scheduling information generated by the apparatus **10000** for transmitting broadcast signals.

[0126] The demapping & decoding module **23000** can convert the input signals into bit domain data and then deinterleave the same as necessary. The demapping & decoding module **23000** can perform demapping for mapping applied for transmission efficiency and correct an error generated on a transmission channel through decoding. In this case, the demapping & decoding module **23000** can obtain transmission parameters necessary for demapping and decoding by decoding the data output from the signaling decoding module **25000**.

[0127] The output processor **24000** can perform reverse procedures of various compression/signal processing procedures which are applied by the apparatus **10000** for transmitting broadcast signals to improve transmission efficiency. In this case, the output processor **24000** can acquire necessary control information from data output from the signaling decoding module **25000**. The output of the output processor **24000** corresponds to a signal input to the apparatus **10000** for transmitting broadcast signals and may be MPEG-TSs, IP streams (v4 or v6) and generic streams.

[0128] The signaling decoding module **25000** can obtain PLS information from the signal demodulated by the synchronization & demodulation module **21000**. As described above, the frame parsing module **22000**, demapping & decoding module **23000** and output processor **24000** can execute functions thereof using the data output from the signaling decoding module **25000**.

[0129] FIG. **14** illustrates a synchronization & demodulation module according to an embodiment of the present invention.

[0130] As shown in FIG. **14**, the synchronization & demodulation module **21000** according to an embodiment of the present invention corresponds to a synchronization & demodulation module of an apparatus **20000** for receiving broadcast signals using m Rx antennas and can include m processing blocks for demodulating signals respectively input through m paths. The m processing blocks can perform the same processing procedure. A description will be given of operation of the first processing block **21000** from among the m processing blocks.

[0131] The first processing block **21000** can include a tuner **21100**, an ADC block **21200**, a preamble detector **21300**, a guard sequence detector **21400**, a waveform transform block **21500**, a time/frequency synchronization block **21600**, a reference signal detector **21700**, a channel equalizer **21800** and an inverse waveform transform block **21900**.

[0132] The tuner **21100** can select a desired frequency band, compensate for the magnitude of a received signal and output the compensated signal to the ADC block **21200**.

[0133] The ADC block **21200** can convert the signal output from the tuner **21100** into a digital signal.

[0134] The preamble detector **21300** can detect a preamble (or preamble signal or preamble symbol) in order to check whether or not the digital signal is a signal of the system corresponding to the apparatus **20000** for receiving broadcast signals. In this case, the preamble detector **21300** can decode basic transmission parameters received through the preamble.

[0135] The guard sequence detector **21400** can detect a guard sequence in the digital signal. The time/frequency synchronization block **21600** can perform time/frequency synchronization using the detected guard sequence and the channel equalizer **21800** can estimate a channel through a received/restored sequence using the detected guard sequence.

[0136] The waveform transform block **21500** can perform a reverse operation of inverse waveform transform when the apparatus **10000** for transmitting broadcast signals has performed inverse waveform transform. When the broadcast transmission/reception system according to one embodiment of the present invention is a multi-carrier system, the waveform transform block **21500** can perform FFT. Furthermore, when the broadcast transmission/reception system according to an embodiment of the present invention is a single carrier

system, the waveform transform block **21500** may not be used if a received time domain signal is processed in the frequency domain or processed in the time domain.

[0137] The time/frequency synchronization block **21600** can receive output data of the preamble detector **21300**, guard sequence detector **21400** and reference signal detector **21700** and perform time synchronization and carrier frequency synchronization including guard sequence detection and block window positioning on a detected signal. Here, the time/frequency synchronization block **21600** can feed back the output signal of the waveform transform block **21500** for frequency synchronization.

[0138] The reference signal detector **21700** can detect a received reference signal. Accordingly, the apparatus **20000** for receiving broadcast signals according to an embodiment of the present invention can perform synchronization or channel estimation.

[0139] The channel equalizer **21800** can estimate a transmission channel from each Tx antenna to each Rx antenna from the guard sequence or reference signal and perform channel equalization for received data using the estimated channel.

[0140] The inverse waveform transform block **21900** may restore the original received data domain when the waveform transform block **21500** performs waveform transform for efficient synchronization and channel estimation/equalization. If the broadcast transmission/reception system according to an embodiment of the present invention is a single carrier system, the waveform transform block **21500** can perform FFT in order to carry out synchronization/channel estimation/equalization in the frequency domain and the inverse waveform transform block **21900** can perform IFFT on the channel-equalized signal to restore transmitted data symbols. If the broadcast transmission/reception system according to an embodiment of the present invention is a multi-carrier system, the inverse waveform transform block **21900** may not be used.

[0141] The above-described blocks may be omitted or replaced by blocks having similar or identical functions according to design.

[0142] FIG. **15** illustrates a frame parsing module according to an embodiment of the present invention.

[0143] As shown in FIG. **15**, the frame parsing module **22000** according to an embodiment of the present invention can include at least one block interleaver **22100** and at least one cell demapper **22200**.

[0144] The block interleaver **22100** can deinterleave data input through data paths of the m Rx antennas and processed by the synchronization & demodulation module **21000** on a signal block basis. In this case, if the apparatus **10000** for transmitting broadcast signals performs pair-wise interleaving, the block interleaver **22100** can process two consecutive pieces of data as a pair for each input path. Accordingly, the block interleaver **22100** can output two consecutive pieces of data even when deinterleaving has been performed. Furthermore, the block interleaver **22100** can perform a reverse operation of the interleaving operation performed by the apparatus **10000** for transmitting broadcast signals to output data in the original order.

[0145] The cell demapper **22200** can extract cells corresponding to common data, cells corresponding to data pipes and cells corresponding to PLS data from received signal frames. The cell demapper **22200** can merge data distributed and transmitted and output the same as a stream as necessary. When two consecutive pieces of cell input data are processed as a pair and mapped in the apparatus **10000** for transmitting broadcast signals, the cell demapper **22200** can perform pair-wise cell demapping for processing two consecutive input cells as one unit as a reverse procedure of the mapping operation of the apparatus **10000** for transmitting broadcast signals.

[0146] In addition, the cell demapper **22200** can extract PLS signaling data received through the current frame as PLS-pre & PLS-post data and output the PLS-pre & PLS-post data.

[0147] The above-described blocks may be omitted or replaced by blocks having similar or identical functions according to design.

[0148] FIG. **16** illustrates a demapping & decoding module according to an embodiment of the present invention.

[0149] The demapping & decoding module **23000** shown in FIG. **16** can perform a reverse operation of the operation of the bit interleaved and coded & modulation module illustrated in FIG. **1**.

[0150] The bit interleaved and coded & modulation module of the apparatus **10000** for transmitting broadcast signals according to an embodiment of the present invention can process input data pipes by independently applying SISO, MISO and MIMO thereto for respective paths, as described above. Accordingly, the demapping & decoding module **23000** illustrated in FIG. **16** can include blocks for processing data output from the frame parsing module according to SISO, MISO and MIMO in response to the apparatus **10000** for transmitting broadcast signals.

[0151] As shown in FIG. **16**, the demapping & decoding module **23000** according to an embodiment of the present invention can include a first block **23100** for SISO, a second block **23200** for MISO, a third block **23300** for MIMO and a fourth block **23400** for processing the PLS-pre/PLS-post information. The demapping & decoding module **23000** shown in FIG. **16** is exemplary and may include only the first block **23100** and the fourth block **23400**, only the second block **23200** and the fourth block **23400** or only the third block **23300** and the fourth block **23400** according to design. That is, the demapping & decoding module **23000** can include blocks for processing data pipes equally or differently according to design.

[0152] A description will be given of each block of the demapping & decoding module **23000**.

[0153] The first block **23100** processes an input data pipe according to SISO and can include a time deinterleaver block **23110**, a cell deinterleaver block **23120**, a constellation demapper block **23130**, a cell-to-bit mux block **23140**, a bit deinterleaver block **23150** and an FEC decoder block **23160**.

[0154] The time deinterleaver block **23110** can perform a reverse process of the process performed by the time interleaving block **14310** illustrated in FIG. **8**. That is, the time deinterleaver block **23110** can deinterleave input symbols interleaved in the time domain into original positions thereof.

[0155] The cell deinterleaver block **23120** can perform a reverse process of the process performed by the cell interleaver block illustrated in FIG. **9***a*. That is, the cell deinterleaver block **23120** can deinterleave positions of cells spread in one FEC block into original positions thereof. The cell deinterleaver block **23120** may be omitted.

[0156] The constellation demapper block **23130** can perform a reverse process of the process performed by the mapper **12300** illustrated in FIG. **5**. That is, the constellation demapper block **23130** can demap a symbol domain input signal to bit domain data. In addition, the constellation demapper block **23130** may perform hard decision and output decided bit data. Furthermore, the constellation demapper block **23130** may output a log-likelihood ratio (LLR) of each bit, which corresponds to a soft decision value or probability value. If the apparatus **10000** for transmitting broadcast signals applies a rotated constellation in order to obtain additional diversity gain, the constellation demapper block **23130** can perform 2-dimensional LLR demapping corresponding to the rotated constellation. Here, the constellation demapper block **23130** can calculate the LLR such that a delay applied by the apparatus **10000** for transmitting broadcast signals to the I or Q component can be compensated.

[0157] The cell-to-bit mux block **23140** can perform a reverse process of the process performed by the mapper **12300** illustrated in FIG. **5**. That is, the cell-to-bit mux block **23140** can restore bit data mapped to the original bit streams.

[0158] The bit deinterleaver block **23150** can perform a reverse process of the process performed by the bit interleaver **12200** illustrated in FIG. **5**. That is, the bit deinterleaver block **23150** can deinterleave the bit streams output from the cell-to-bit mux block **23140** in the original order.

[0159] The FEC decoder block **23460** can perform a reverse process of the process performed by the FEC encoder **12100** illustrated in FIG. **5**. That is, the FEC decoder block **23460** can correct an error generated on a transmission channel by performing LDPC decoding and BCH decoding.

[0160] The second block **23200** processes an input data pipe according to MISO and can include the time deinterleaver block, cell deinterleaver block, constellation demapper block, cell-to-bit mux block, bit deinterleaver block and FEC decoder block in the same manner as the first block **23100**, as shown in FIG. **16**. However, the second block **23200** is distinguished from the first block **23100** in that the second block **23200** further includes a MISO decoding block **23210**. The second block **23200** performs the same procedure including time deinterleaving operation to outputting operation as the first block **23100** and thus description of the corresponding blocks is omitted.

[0161] The MISO decoding block **11110** can perform a reverse operation of the operation of the MISO processing in the apparatus **10000** for transmitting broadcast signals. If the broadcast transmission/reception system according to an embodiment of the present invention uses STBC, the MISO decoding block **11110** can perform Alamouti decoding.

[0162] The third block **23300** processes an input data pipe according to MIMO and can include the time deinterleaver block, cell deinterleaver block, constellation demapper block, cell-to-bit mux block, bit deinterleaver block and FEC decoder block in the same manner as the second block **23200**, as shown in FIG. **16**. However, the third block **23300** is distinguished from the second block **23200** in that the third block **23300** further includes a MIMO decoding block **23310**. The basic roles of the time deinterleaver block, cell deinterleaver block, constellation demapper block, cell-to-bit mux block and bit deinterleaver block included in the third block **23300** are identical to those of the corresponding blocks included in the first and second blocks **23100** and **23200** although functions thereof may be different from the first and second blocks **23100** and **23200**.

[0163] The MIMO decoding block **23310** can receive output data of the cell deinterleaver for input signals of the m Rx antennas and perform MIMO decoding as a reverse operation of the operation of the MIMO processing in the apparatus **10000** for transmitting broadcast signals. The MIMO decoding block **23310** can perform maximum likelihood decoding to obtain optimal decoding performance or carry out sphere decoding with reduced complexity. Otherwise, the MIMO decoding block **23310** can achieve improved decoding performance by performing MMSE detection or carrying out iterative decoding with MMSE detection.

[0164] The fourth block **23400** processes the PLS-pre/PLS-post information and can perform SISO or MISO decoding.

[0165] The basic roles of the time deinterleaver block, cell deinterleaver block, constellation demapper block, cell-to-bit mux block and bit deinterleaver block included in the fourth block **23400** are identical to those of the corresponding blocks of the first, second and third blocks **23100**, **23200** and **23300** although functions thereof may be different from the first, second and third blocks **23100**, **23200** and **23300**.

[0166] The shortened/punctured FEC decoder **23410** can perform de-shortening and de-puncturing on data shortened/punctured according to PLS data length and then carry out FEC decoding thereon. In this case, the FEC decoder used for data pipes can also be used for PLS. Accordingly, additional FEC decoder hardware for the PLS only is not needed and thus system design is simplified and efficient coding is achieved.

[0167] The above-described blocks may be omitted or replaced by blocks having similar or identical functions according to design.

[0168] The demapping & decoding module according to an embodiment of the present invention can output data pipes and PLS information processed for the respective paths to the output processor, as illustrated in FIG. **16**.

[0169] FIGS. **17** and **18** illustrate output processors according to embodiments of the present invention.

[0170] FIG. **17** illustrates an output processor **24000** according to an embodiment of the present invention. The output processor **24000** illustrated in FIG. **17** receives a single data pipe output from the demapping & decoding module and outputs a single output stream.

[0171] The output processor **24000** shown in FIG. **17** can include a BB scrambler block **24100**, a padding removal block **24200**, a CRC-8 decoder block **24300** and a BB frame processor block **24400**.

[0172] The BB scrambler block **24100** can descramble an input bit stream by generating the same PRBS as that used in the apparatus for transmitting broadcast signals for the input bit stream and carrying out an XOR operation on the PRBS and the bit stream.

[0173] The padding removal block **24200** can remove padding bits inserted by the apparatus for transmitting broadcast signals as necessary.

[0174] The CRC-8 decoder block **24300** can check a block error by performing CRC decoding on the bit stream received from the padding removal block **24200**.

[0175] The BB frame processor block **24400** can decode information transmitted through a BB frame header and

restore MPEG-TSs, IP streams (v4 or v6) or generic streams using the decoded information.

[0176] The above-described blocks may be omitted or replaced by blocks having similar or identical functions according to design.

[0177] FIG. **18** illustrates an output processor according to another embodiment of the present invention. The output processor **24000** shown in FIG. **18** receives multiple data pipes output from the demapping & decoding module. Decoding multiple data pipes can include a process of merging common data commonly applicable to a plurality of data pipes and data pipes related thereto and decoding the same or a process of simultaneously decoding a plurality of services or service components (including a scalable video service) by the apparatus for receiving broadcast signals.

[0178] The output processor **24000** shown in FIG. **18** can include a BB descrambler block, a padding removal block, a CRC-8 decoder block and a BB frame processor block as the output processor illustrated in FIG. **17**. The basic roles of these blocks correspond to those of the blocks described with reference to FIG. **17** although operations thereof may differ from those of the blocks illustrated in FIG. **17**.

[0179] A de-jitter buffer block **24500** included in the output processor shown in FIG. **18** can compensate for a delay, inserted by the apparatus for transmitting broadcast signals for synchronization of multiple data pipes, according to a restored TTO (time to output) parameter.

[0180] A null packet insertion block **24600** can restore a null packet removed from a stream with reference to a restored DNP (deleted null packet) and output common data.

[0181] A TS clock regeneration block **24700** can restore time synchronization of output packets based on ISCR (input stream time reference) information.

[0182] A TS recombining block **24800** can recombine the common data and data pipes related thereto, output from the null packet insertion block **24600**, to restore the original MPEG-TSs, IP streams (v4 or v6) or generic streams. The TTO, DNT and ISCR information can be obtained through the BB frame header.

[0183] An in-band signaling decoding block **24900** can decode and output in-band physical layer signaling information transmitted through a padding bit field in each FEC frame of a data pipe.

[0184] The output processor shown in FIG. **18** can BB-descramble the PLS-pre information and PLS-post information respectively input through a PLS-pre path and a PLS-post path and decode the descrambled data to restore the original PLS data. The restored PLS data is delivered to a system controller included in the apparatus for receiving broadcast signals. The system controller can provide parameters necessary for the synchronization & demodulation module, frame parsing module, demapping & decoding module and output processor module of the apparatus for receiving broadcast signals.

[0185] The above-described blocks may be omitted or replaced by blocks having similar r identical functions according to design.

[0186] Hereinafter, a method of interleaving of the block interleaver **14230** of FIG. **7** will be described

[0187] The block interleaver **14230** interleaves the plurality of bit groups the order of which has been rearranged. Specifically, the block interleaver **14230** may interleave the plurality of bit groups the order of which has been rearranged by the group-wise interleaver **14220** in bit group wise (or bits group unit). The block interleaver **14230** is formed of a plurality of columns each including a plurality of rows and may interleave by dividing the plurality of rearranged bit groups based on a modulation order determined according to a modulation method.

[0188] In this case, the block interleaver **14230** may interleave the plurality of bit groups the order of which has been rearranged by the group-wise interleaver **14220** in bit group wise. Specifically, the block interleaver **14230** may interleave by dividing the plurality of rearranged bit groups according to a modulation order by using a first part and a second part.

[0189] Specifically, the block interleaver **14230** interleaves by dividing each of the plurality of columns into a first part and a second part, writing the plurality of bit groups in the plurality of columns of the first part serially in bit group wise, dividing the bits of the other bit groups into groups (or sub bit groups) each including a predetermined number of bits based on the number of columns, and writing the sub bit groups in the plurality of columns of the second part serially.

[0190] Herein, the number of bit groups which are interleaved in bit group wise may be determined by at least one of the number of rows and columns constituting the block interleaver **14230**, the number of bit groups and the number of bits included in each bit group. In other words, the block interleaver **14230** may determine the bit groups which are to be interleaved in bit group wise considering at least one of the number of rows and columns constituting the block interleaver **14230**, the number of bit groups and the number of bits included in each bit group, interleave the corresponding bit groups in bit group wise, and divide bits of the other bit groups into sub bit groups and interleave the sub bit groups. For example, the block interleaver **14230** may interleave at least part of the plurality of bit groups in bit group wise using the first part, and divide bits of the other bit groups into sub bit groups and interleave the sub bit groups using the second part.

[0191] Meanwhile, interleaving bit groups in bit group wise means that the bits included in the same bit group are written in the same column. In other words, the block interleaver **14230**, in case of bit groups which are interleaved in bit group wise, may not divide the bits included in the same bit groups and write the bits in the same column, and in case of bit groups which are not interleaved in bit group wise, may divide the bits in the bit groups and write the bits in different columns.

[0192] Accordingly, the number of rows constituting the first part is a multiple of the number of bits included in one bit group (for example, 360), and the number of rows constituting the second part may be less than the number of bits included in one bit group.

[0193] In addition, in all bit groups interleaved by the first part, the bits included in the same bit group are written and interleaved in the same column of the first part, and in at least one group interleaved by the second part, the bits are divided and written in at least two columns of the second part.

[0194] As described above, the block interleaver **14230** may interleave the plurality of bit groups by using the plurality of columns each including the plurality of rows.

[0195] In this case, the block interleaver **14230** may interleave the LDPC codeword by dividing the plurality of columns into at least two parts. For example, the block

interleaver **14230** may divide each of the plurality of columns into the first part and the second part and interleave the plurality of bit groups constituting the LDPC codeword.

[0196] In this case, the block interleaver **14230** may divide each of the plurality of columns into N number of parts (N is an integer greater than or equal to 2) according to whether the number of bit groups constituting the LDPC codeword is an integer multiple of the number of columns constituting the block interleaver **14230**, and may perform interleaving.

[0197] When the number of bit groups constituting the LDPC codeword is an integer multiple of the number of columns constituting the block interleaver **14230**, the block interleaver **14230** may interleave the plurality of bit groups constituting the LDPC codeword in bit group wise without dividing each of the plurality of columns into parts.

[0198] Specifically, the block interleaver **14230** may interleave by writing the plurality of bit groups of the LDPC codeword on each of the columns in bit group wise in a column direction, and reading each row of the plurality of columns in which the plurality of bit groups are written in bit group wise in a row direction.

[0199] In this case, the block interleaver **14230** may interleave by writing bits included in a predetermined number of bit groups, which corresponds to a quotient obtained by dividing the number of bit groups of the LDPC codeword by the number of columns of the block interleaver **14230**, on each of the plurality of columns serially in a column direction, and reading each row of the plurality of columns in which the bits are written in a row direction.

[0200] Hereinafter, the group located in the j^{th} position after being interleaved by the group interleaver **14220** will be referred to as group Y_j.

[0201] For example, it is assumed that the block interleaver **14230** is formed of C number of columns each including R_1 number of rows. In addition, it is assumed that the LDPC codeword is formed of N_{group} number of bit groups and the number of bit groups N_{group} is a multiple of C.

[0202] In this case, when the quotient obtained by dividing N_{group} number of bit groups constituting the LDPC codeword by C number of columns constituting the block interleaver **14230** is A ($=N_{group}/C$) (A is an integer greater than 0), the block interleaver **14230** may interleave by writing A ($=N_{group}/C$) number of bit groups on each column serially in a column direction and reading bits written on each column in a row direction.

[0203] For example, as shown in FIG. **19**, the block interleaver **14230** writes bits included in bit group Y_0, bit group T_1, . . . , bit group Y_{A-1} in the 1^{st} column from the 1^{st} row to the $R_1{}^{th}$ row, writes bits included in bit group Y_A, bit group Y_{A+1}, . . . , bit group Y_{2A-1} in the 2nd column from the 1^{st} row to the $R_1{}^{th}$ row, . . . , and writes bits included in bit group Y_{CA-A}, bit group Y_{CA-A+1}, . . . , bit group Y_{CA-1} in the column C from the 1^{st} row to the $R_1{}^{th}$ row. The block interleaver **14230** may read the bits written in each row of the plurality of columns in a row direction.

[0204] Accordingly, the block interleaver **14230** interleaves all bit groups constituting the LDPC codeword in bit group wise.

[0205] However, when the number of bit groups of the LDPC codeword is not an integer multiple of the number of columns of the block interleaver **14230**, the block interleaver **14230** may divide each column into 2 parts and interleave a part of the plurality of bit groups of the LDPC codeword in bit group wise, and divide bits of the other bit groups into sub bit groups and interleave the sub bit groups. In this case, the bits included in the other bit groups, that is, the bits included in the number of groups which correspond to the remainder when the number of bit groups constituting the LDPC codeword is divided by the number of columns are not interleaved in bit group wise, but interleaved by being divided according to the number of columns.

[0206] Specifically, the block interleaver **14230** may interleave the LDPC codeword by dividing each of the plurality of columns into two parts.

[0207] In this case, the block interleaver **14230** may divide the plurality of columns into the first part and the second part based on at least one of the number of columns of the block interleaver **14230**, the number of bit groups of the LDPC codeword, and the number of bits of bit groups.

[0208] Here, each of the plurality of bit groups may be formed of 360 bits. In addition, the number of bit groups of the LDPC codeword is determined based on the length of the LDPC codeword and the number of bits included in the bit group.

[0209] For example, when an LDPC codeword in the length of 16200 is divided such that each bit group has 360 bits, the LDPC codeword is divided into 45 bit groups. Alternatively, when an LDPC codeword in the length of 64800 is divided such that each bit group has 360 bits, the LDPC codeword may be divided into 180 bit groups. Further, the number of columns constituting the block interleaver **14230** may be determined according to a modulation method. This will be explained in detail below.

[0210] Accordingly, the number of rows constituting each of the first part and the second part may be determined based on the number of columns constituting the block interleaver **14230**, the number of bit groups constituting the LDPC codeword, and the number of bits constituting each of the plurality of bit groups.

[0211] Specifically, in each of the plurality of columns, the first part may be formed of as many rows as the number of bits included in at least one bit group which can be written in each column in bit group wise from among the plurality of bit groups of the LDPC codeword, according to the number of columns constituting the block interleaver **14230**, the number of bit groups constituting the LDPC codeword, and the number of bits constituting each bit group.

[0212] In each of the plurality of columns, the second part may be formed of rows excluding as many rows as the number of bits included in at least some bit groups which can be written in each of the plurality of columns in bit group wise. Specifically, the number rows of the second part may be the same value as a quotient when the number of bits included in all bit groups excluding bit groups corresponding to the first part is divided by the number of columns constituting the block interleaver **14230**. In other words, the number of rows of the second part may be the same value as a quotient when the number of bits included in the remaining bit groups which are not written in the first part from among bit groups constituting the LDPC codeword is divided by the number of columns.

[0213] That is, the block interleaver **14230** may divide each of the plurality of columns into the first part including as many rows as the number of bits included in bit groups which can be written in each column in bit group wise, and the second part including the other rows.

[0214] Accordingly, the first part may be formed of as many rows as the number of bits included in bit groups, that is, as many rows as an integer multiple of M. However, since the number of codeword bits constituting each bit group may be an aliquot part of M as described above, the first part may be formed of as many rows as an integer multiple of the number of bits constituting each bit group.

[0215] In this case, the block interleaver **14230** may interleave by writing and reading the LDPC codeword in the first part and the second part in the same method.

[0216] Specifically, the block interleaver **14230** may interleave by writing the LDPC codeword in the plurality of columns constituting each of the first part and the second part in a column direction, and reading the plurality of columns constituting the first part and the second part in which the LDPC codeword is written in a row direction.

[0217] That is, the block interleaver **14230** may interleave by writing the bits included in at least some bit groups which can be written in each of the plurality of columns in bit group wise in each of the plurality of columns of the first part serially, dividing the bits included in the other bit groups except the at least some bit groups and writing in each of the plurality of columns of the second part in a column direction, and reading the bits written in each of the plurality of columns constituting each of the first part and the second part in a row direction.

[0218] In this case, the block interleaver **14230** may interleave by dividing the other bit groups except the at least some bit groups from among the plurality of bit groups based on the number of columns constituting the block interleaver **14230**.

[0219] Specifically, the block interleaver **14230** may interleave by dividing the bits included in the other bit groups by the number of a plurality of columns, writing each of the divided bits in each of a plurality of columns constituting the second part in a column direction, and reading the plurality of columns constituting the second part, where the divided bits are written, in a row direction.

[0220] That is, the block interleaver **14230** may divide the bits included in the other bit groups except the bit groups written in the first part from among the plurality of bit groups of the LDPC codeword, that is, the bits in the number of bit groups which correspond to the remainder when the number of bit groups constituting the LDPC codeword is divided by the number of columns, by the number of columns, and may write the divided bits in each column of the second part serially in a column direction.

[0221] For example, it is assumed that the block interleaver **14230** is formed of C number of columns each including R_1 number of rows. In addition, it is assumed that the LDPC codeword is formed of N_{group} number of bit groups, the number of bit groups N_{group} is not a multiple of C, and $A\times C+1=N_{group}$ (A is an integer greater than 0). In other words, it is assumed that when the number of bit groups constituting the LDPC codeword is divided by the number of columns, the quotient is A and the remainder is 1.

[0222] In this case, as shown in FIGS. **20** and **21**, the block interleaver **14230** may divide each column into a first part including R_1 number of rows and a second part including R_2 number of rows. In this case, R_1 may correspond to the number of bits included in bit groups which can be written in each column in bit group wise, and R_2 may be R_1 subtracted from the number of rows of each column.

[0223] That is, in the above-described example, the number of bit groups which can be written in each column in bit group wise is A, and the first part of each column may be formed of as many rows as the number of bits included in A number of bit groups, that is, may be formed of as many rows as A×M number.

[0224] In this case, the block interleaver **14230** writes the bits included in the bit groups which can be written in each column in bit group wise, that is, A number of bit groups, in the first part of each column in the column direction.

[0225] That is, as shown in FIGS. **20** and **21**, the block interleaver **14230** writes the bits included in each of bit group Y_0, bit group Y_1, . . . , group Y_{A-1} in the 1^{st} to $R_1{}^{th}$ rows of the first part of the 1^{st} column, writes bits included in each of bit group Y_A, bit group Y_{A+1}, . . . , bit group Y_{2A-1} in the 1^{st} to $R_1{}^{th}$ rows of the first part of the 2^{nd} column, . . . , writes bits included in each of bit group Y_{CA-A}, bit group Y_{CA-A+1}, . . . , bit group Y_{CA-1} in the 1^{st} to $R_1{}^{th}$ rows of the first part of the column C.

[0226] As described above, the block interleaver **14230** writes the bits included in the bit groups which can be written in each column in bit group wise in the first part of each column.

[0227] In other words, in the above exemplary embodiment, the bits included in each of bit group (Y_0), bit group (Y_1), . . . , bit group (Y_{A-1}) may not be divided and all of the bits may be written in the first column, the bits included in each of bit group (Y_A), bit group (Y_{A+1}), . . . , bit group (Y_{2A-1}) may not be divided and all of the bits may be written in the second column, and the bits included in each of bit group (Y_{CA-A}), bit group (Y_{CA-A+1}), . . . , group (Y_{CA-1}) may not be divided and all of the bits may be written in the C column. As such, all bit groups interleaved by the first part are written in the same column of the first part.

[0228] Thereafter, the block interleaver **14230** divides bits included in the other bit groups except the bit groups written in the first part of each column from among the plurality of bit groups, and writes the bits in the second part of each column in the column direction. In this case, the block interleaver **14230** divides the bits included in the other bit groups except the bit groups written in the first part of each column by the number of columns, so that the same number of bits are written in the second part of each column, and writes the divided bits in the second part of each column in the column direction.

[0229] In the above-described example, since $A\times C+1=N_{group}$, when the bit groups constituting the LDPC codeword are written in the first part serially, the last bit group $Y_{Ngroup-1}$ of the LDPC codeword is not written in the first part and remains. Accordingly, the block interleaver **14230** divides the bits included in the bit group $Y_{Ngroup-1}$ into C number of sub bit groups as shown in FIG. **20**, and writes the divided bits (that is, the bits corresponding to the quotient when the bits included in the last group ($Y_{Ngroup-1}$) are divided by C) in the second part of each column serially.

[0230] The bits divided based on the number of columns may be referred to as sub bit groups. In this case, each of the sub bit groups may be written in each column of the second part. That is, the bits included in the bit groups may be divided and may form the sub bit groups.

[0231] That is, the block interleaver **14230** writes the bits in the 1^{st} to $R_2{}^{th}$ rows of the second part of the 1^{st} column, writes the bits in the 1^{st} to $R_2{}^{th}$ rows of the second part of the 2^{nd} column, . . . , and writes the bits in the 1^{st} to $R_2{}^{th}$ rows

of the second part of the column C. In this case, the block interleaver **14230** may write the bits in the second part of each column in the column direction as shown in FIG. **20**.

[0232] That is, in the second part, the bits constituting the bit group may not be written in the same column and may be written in the plurality of columns. In other words, in the above example, the last bit group ($Y_{Ngroup-1}$) is formed of M number of bits and thus, the bits included in the last bit group ($Y_{Ngroup-1}$) may be divided by M/C and written in each column. That is, the bits included in the last bit group ($Y_{Ngroup-1}$) are divided by M/C, forming M/C number of sub bit groups, and each of the sub bit groups may be written in each column of the second part.

[0233] Accordingly, in at least one bit group which is interleaved by the second part, the bits included in the at least one bit group are divided and written in at least two columns constituting the second part.

[0234] In the above-described example, the block interleaver **14230** writes the bits in the second part in the column direction. However, this is merely an example. That is, the block interleaver **14230** may write the bits in the plurality of columns of the second part in the row direction. In this case, the block interleaver **14230** may write the bits in the first part in the same method as described above.

[0235] Specifically, referring to FIG. **21**, the block interleaver **14230** writes the bits from the 1^{st} row of the second part in the 1^{st} column to the 1^{st} row of the second part in the column C, writes the bits from the 2^{nd} row of the second part in the 1^{st} column to the 2^{nd} row of the second part in the column C, . . . , etc., and writes the bits from the R_2^{th} row of the second part in the 1^{st} column to the R_2^{th} row of the second part in the column C.

[0236] On the other hand, the block interleaver **14230** reads the bits written in each row of each part serially in the row direction. That is, as shown in FIGS. **20** and **21**, the block interleaver **14230** reads the bits written in each row of the first part of the plurality of columns serially in the row direction, and reads the bits written in each row of the second part of the plurality of columns serially in the row direction.

[0237] Accordingly, the block interleaver **14230** may interleave a part of the plurality of bit groups constituting the LDPC codeword in bit group wise, and divide and interleave some of the remaining bit groups. That is, the block interleaver **14230** may interleave by writing the LDPC codeword constituting a predetermined number of bit groups from among the plurality of bit groups in the plurality of columns of the first part in bit group wise, dividing the bits of the other bit groups and writing the bits in each of the columns of the second part, and reading the plurality of columns of the first and second parts in the row direction.

[0238] As described above, the block interleaver **14230** may interleave the plurality of bit groups in the methods described above with reference to FIGS. **19** to **21**.

[0239] In particular, in the case of FIG. **20**, the bits included in the bit group which does not belong to the first part are written in the second part in the column direction and read in the row direction. In view of this, the order of the bits included in the bit group which does not belong to the first part is rearranged. Since the bits included in the bit group which does not belong to the first part are interleaved as described above, bit error rate (BER)/frame error rate (FER) performance can be improved in comparison with a case in which such bits are not interleaved.

[0240] However, the bit group which does not belong to the first part may not be interleaved as shown in FIG. **20**. That is, since the block interleaver **14230** writes and reads the bits included in the group which does not belong to the first part in and from the second part in the row direction, the order of the bits included in the group which does not belong to the first part is not changed and the bits are output serially. In this case, the bits included in the group which does not belong to the first part may be output serially and mapped onto a modulation symbol.

[0241] In FIGS. **20** and **21**, the last single bit group of the plurality of bit groups is written in the second part. However, this is merely an example. The number of bit groups written in the second part may vary according to the total number of bit groups of the LDPC codeword, the number of columns and rows, the number of transmission antennas, etc.

[0242] The block interleaver **14230** may have a configuration as shown in tables 1 and 2 presented below:

TABLE 1

	N_{ldpc} = 64800					
	QPSK	16 QAM	64 QAM	256 QAM	1024 QAM	4096 QAM
C	2	4	6	8	10	12
R_1	32400	16200	10800	7920	6480	5400
R_2	0	0	0	180	0	0

TABLE 2

	N_{ldpc} = 16200					
	QPSK	16 QAM	64 QAM	256 QAM	1024 QAM	4096 QAM
C	2	4	6	8	10	12
R_1	7920	3960	2520	1800	1440	1080
R_2	180	90	180	225	180	270

[0243] Herein, C (or N_C) is the number of columns of the block interleaver **14230**, R_1 is the number of rows constituting the first part in each column, and R_2 is the number of rows constituting the second part in each column.

[0244] Referring to Tables 1 and 2, the number of columns has the same value as a modulation order according to a modulation method, and each of a plurality of columns is formed of rows corresponding to the number of bits constituting the LDPC codeword divided by the number of a plurality of columns.

[0245] For example, when the length N_{ldpc} of the LDPC codeword is 64800 and the modulation method is 16-QAM, the block interleaver **14230** is formed of 4 columns as the modulation order is 4 in the case of 16-QAM, and each column is formed of rows as many as R_1+R_2=16200 (=64800/4) In another example, when the length N_{ldpc} of the LDPC codeword is 64800 and the modulation method is 64-QAM, the block interleaver **14230** is formed of 6 columns as the modulation order is 6 in the case of 64-QAM, and each column is formed of rows as many as R_1+R_2=10800(=64800/6).

[0246] Meanwhile, referring to Tables 1 and 2, when the number of bit groups constituting an LDPC codeword is an integer multiple of the number of columns, the block interleaver **14230** interleaves without dividing each column. Therefore, R_1 corresponds to the number of rows constituting each column, and R_2 is 0. In addition, when the number

of bit groups constituting an LDPC codeword is not an integer multiple of the number of columns, the block interleaver **14230** interleaves the groups by dividing each column into the first part formed of R_1 number of rows, and the second part formed of R_2 number of rows.

[0247] When the number of columns of the block interleaver **14230** is equal to the number of bits constituting a modulation symbol, bits included in a same bit group are mapped onto a single bit of each modulation symbol as shown in Tables 1 and 2.

[0248] For example, when N_{ldpc}=64800 and the modulation method is 16-QAM, the block interleaver **14230** may be formed of four (4) columns each including 16200 rows. In this case, the bits included in each of the plurality of bit groups are written in the four (4) columns and the bits written in the same row in each column are output serially. In this case, since four (4) bits constitute a single modulation symbol in the modulation method of 16-QAM, bits included in the same bit group, that is, bits output from a single column, may be mapped onto a single bit of each modulation symbol. For example, bits included in a bit group written in the 1^{st} column may be mapped onto the first bit of each modulation symbol.

[0249] In another example, when N_{ldpc}=64800 and the modulation method is 64-QAM, the block interleaver **14230** may be formed of six (6) columns each including 10800 rows. In this case, the bits included in each of the plurality of bit groups are written in the six (6) columns and the bits written in the same row in each column are output serially. In this case, since six (6) bits constitute a single modulation symbol in the modulation method of 64-QAM, bits included in the same bit group, that is, bits output from a single column, may be mapped onto a single bit of each modulation symbol. For example, bits included in a bit group written in the 1^{st} column may be mapped onto the first bit of each modulation symbol.

[0250] Referring to Tables 1 and 2, the total number of rows of the block interleaver **14230**, that is, R_1+R_2, is N_{ldpc}/C.

[0251] In addition, the number of rows of the first part, R_1, is an integer multiple of the number of bits included in each group, M (e.g., M=360), and maybe expressed as $\lfloor N_{group}/C \rfloor \times M$ and the number of rows of the second part, R_2, may be $N_{ldpc}/C-R_1$. Herein, $\lfloor N_{group}/C \rfloor$ is the largest integer below N_{group}/C. Since R_1 is an integer multiple of the number of bits included in each group, M, bits may be written in R_1 in bit groups wise.

[0252] In addition, when the number of bit groups of the LDPC codeword is not a multiple of the number of columns, it can be seen from Tables 1 and 2 that the block interleaver **14230** interleaves by dividing each column into two parts.

[0253] Specifically, the length of the LDPC codeword divided by the number of columns is the total number of rows included in the each column. In this case, when the number of bit groups of the LDPC codeword is a multiple of the number of columns, each column is not divided into two parts. However, when the number of bit groups of the LDPC codeword is not a multiple of the number of columns, each column is divided into two parts.

[0254] For example, it is assumed that the number of columns of the block interleaver **14230** is identical to the number of bits constituting a modulation symbol, and an LDPC codeword is formed of 64800 bits as shown in Table 1. In this case, each bit group of the LDPC codeword is formed of 360 bits, and the LDPC codeword is formed of 64800/360(=180) bit groups.

[0255] When the modulation method is 16-QAM, the block interleaver **14230** may be formed of four (4) columns and each column may have 64800/4(=16200) rows.

[0256] In this case, since the number of bit groups of the LDPC codeword divided by the number of columns is 180/4(=45), bits can be written in each column in bit group wise without dividing each column into two parts. That is, bits included in 45 bit groups which is the quotient when the number of bit groups constituting the LDPC codeword is divided by the number of columns, that is, 45×360(=16200) bits can be written in each column.

[0257] However, when the modulation method is 256-QAM, the block interleaver **14230** may be formed of eight (8) columns and each column may have 64800/8(=8100) rows.

[0258] In this case, since the number of bit groups of the LDPC codeword divided by the number of columns is 180/8=22.5, the number of bit groups constituting the LDPC codeword is not an integer multiple of the number of columns. Accordingly, the block interleaver **14230** divides each of the eight (8) columns into two parts to perform interleaving in bit group wise.

[0259] In this case, since the bits should be written in the first part of each column in bit group wise, the number of bit groups which can be written in the first part of each column in bit group wise is 22, which is the quotient when the number of bit groups constituting the LDPC codeword is divided by the number of columns, and accordingly, the first part of each column has 22×360(=7920) rows. Accordingly, 7920 bits included in 22 bit groups may be written in the first part of each column.

[0260] The second part of each column has rows which are the rows of the first part subtracted from the total rows of each column. Accordingly, the second part of each column includes 8100−7920(=180) rows.

[0261] In this case, the bits included in the other bit groups which have not been written in the first part are divided and written in the second part of each column.

[0262] Specifically, since 22×8(=176) bit groups are written in the first part, the number of bit groups to be written in the second part is 180-176 (=4) (for example, bit group Y_{176}, bit group Y_{177}, bit group Y_{178}, and bit group Y_{179} from among bit group Y_0, bit group Y_1, bit group Y_2, . . . , bit group Y_{178}, and bit group Y_{179} constituting the LDPC codeword).

[0263] Accordingly, the block interleaver **14230** may write the four (4) bit groups which have not been written in the first part and remains from among the groups constituting the LDPC codeword in the second part of each column serially.

[0264] That is, the block interleaver **14230** may write 180 bits of the 360 bits included in the bit group Y_{176} in the 1^{st} row to the 180^{th} row of the second part of the 1^{st} column in the column direction, and may write the other 180 bits in the 1^{st} row to the 180^{th} row of the second part of the 2^{nd} column in the column direction. In addition, the block interleaver **14230** may write 180 bits of the 360 bits included in the bit group Y_{177} in the 1^{st} row to the 180^{th} row of the second part of the 3^{rd} column in the column direction, and may write the other 180 bits in the 1^{st} row to the 180^{th} row of the second part of the 4^{th} column in the column direction. In addition, the block interleaver **14230** may write 180 bits of the 360 bits included in the bit group Y_{178} in the 1^{st} row to the 180^{th} row of the second part of the 5^{th} column in the column direction, and may write the other 180 bits in the 1^{st} row to the 180^{th} row of the second part of the 6^{th} column in the column direction. In addition, the block interleaver **14230**

may write 180 bits of the 360 bits included in the bit group Y_{179} in the 1^{st} row to the 180^{th} row of the second part of the 7^{th} column in the column direction, and may write the other 180 bits in the 1^{st} row to the 180^{th} row of the second part of the 8^{th} column in the column direction.

[0265] Accordingly, the bits included in the bit group which has not been written in the first part and remains are not written in the same column in the second part and may be divided and written in the plurality of columns.

[0266] Hereinafter, the block interleaver **14230** according to an exemplary embodiment will be explained in detail with reference to FIG. **22**.

[0267] In a group-interleaved LDPC codeword (v_0, v_1, . . . , $v_{N_{ldpc}-1}$), Y_j is continuously arranged like $V=\{Y_0, Y_1, \ldots, Y_{N_{group}-1}\}$.

[0268] The LDPC codeword after group interleaving may be interleaved by the block interleaver **14230** as shown in FIG. **22**. In this case, the block interleaver **14230** divide a plurality of columns into the first part (Part 1) and the second part (Part 2) based on the number of columns of the block interleaver **14230** and the number of bits of bit groups. In this case, in the first part, the bits constituting the bit groups may be written in the same column, and in the second part, the bits constituting the bit groups may be written in a plurality of columns (i.e. the bits constituting the bit groups may be written in at least two columns).

[0269] Specifically, input bits vi are written serially from the first part to the second part column wise, and then read out serially from the first part to the second part row wise. That is, the data bits vi are written serially into the block interleaver column-wise starting in the first part and continuing column-wise finishing in the second part, and then read out serially row-wise from the first part and then row-wise from the second part. Accordingly, the bit included in the same bit group in the first part may be mapped onto a single bit of each modulation symbol.

[0270] In this case, the number of columns and the number of rows of the first part and the second part of the block interleaver **14230** vary according to a modulation format and a length of the LDPC codeword as in Table 3 presented below. That is, the first part and the second part block interleaving configurations for each modulation format and code length are specified in Table 3 presented below Herein, the number of columns of the block interleaver **14230** may be equal to the number of bits constituting a modulation symbol. In addition, a sum of the number of rows of the first part, N_{r1} and the number of rows of the second part, N_{r2}, is equal to N_{ldpc}/N_C (herein, Nc is the number of columns). In addition, since $N_{r1}(=\lfloor N_{group}/N_c \rfloor \times 360)$ is a multiple of 360, a multiple of bit groups may be written in the first part.

TABLE 3

	Rows in Part 1 N_{r1}		Rows in Part 2 N_{r2}		
Modulation	N_{ldpc} = 64800	N_{ldpc} = 16200	N_{ldpc} = 64800	N_{ldpc} = 16200	Columns N_c
QPSK	32400	7920	0	180	2
16-QAM	16200	3960	0	90	4
64-QAM	10800	2520	0	180	6
256-QAM	7920	1800	180	225	8
1024-QAM	6480	1440	0	180	10
4096-QAM	5400	1080	0	270	12

[0271] Hereinafter, an operation of the block interleaves **14230** will be explained in detail.

[0272] Specifically, as shown in FIG. **2**, the input bit v_i ($0 \le i < N_C \times N_{r1}$) is written in r_i row of c_i column of the first part of the block interleaver **14230**. Herein, c_i and r_i are

$$c_i = \left\lfloor \frac{i}{N_{r1}} \right\rfloor \text{ and } r_i = (i \bmod N_{r1}),$$

respectively.

[0273] In addition, the input bit v_i ($N_C \times N_{r1} \le i < N_{ldpc}$) is written in r_i row of c_i column of the second part of the block interleaver **14230**. Herein, c_i and r_i satisfy

$$c_i = \left\lfloor \frac{(i - N_C \times N_{r1})}{N_{r2}} \right\rfloor \text{ and } r_i = N_{r1} + \{(i - N_C \times N_{r1}) \bmod N_{r2}\},$$

respectively.

[0274] An output bit $q_j (0 \le j < N_{ldpc})$ is read from c_j column of r_j row. Herein, r_j and c_j satisfy

$$r_j = \left\lfloor \frac{j}{N_c} \right\rfloor \text{ and } c_j = (j \bmod N_C),$$

respectively.

[0275] For example, when the length N_{ldpc} of an LDPC codeword is 64800 and the modulation method is 256-QAM, the order of bits output from the block interleaver **14230** may be $(q_0, q_1, q_2, \ldots, q_{63357}, q_{63358}, q_{63359}, q_{63360}, q_{63361}, \ldots, q_{64799}) = (V_0, v_{7920}, v_{15840}, \ldots, v_{47519}, v_{55439}, v_{63359}, v_{63360}, v_{63540}, \ldots, v_{64799})$. Herein, the indexes of the right side of the foregoing equation may be specifically expressed for the eight (8) columns as 0, 7920, 15840, 23760, 31680, 39600, 47520, 55440, 1, 7921, 15841, 23761, 31681, 39601, 47521, 55441, . . . , 7919, 15839, 23759, 31679, 39599, 47519, 55439, 63359, 63360, 63540, 63720, 63900, 64080, 64260 64440, 64620, . . . , 63539, 63719, 63899, 64079, 64259, 64439, 64619, 64799.

[0276] Hereinafter, a method of mapping bits onto constellation points will be described in greater detail, according to various exemplary embodiments

[0277] A non-uniform constellation (NUC) according to an exemplary embodiment may be generated or obtained using any suitable method or algorithm including steps (or operations) for generating or obtaining such a non-uniform constellation. The non-uniform constellation according to the embodiment may be generated or obtained by any suitably arranged apparatus or system including means for generating or obtaining such a non-uniform constellation. The methods or algorithms described herein may be implemented in any suitably arranged apparatus or system including means for carrying out the method or algorithm steps.

[0278] Certain exemplary embodiments provide an algorithm for obtaining a non-uniform constellation. The non-uniform constellation obtained in the certain exemplary embodiments may provide a higher capacity than an equivalent uniform constellation (e.g. a uniform constellation of the same order). Certain exemplary embodiments may obtain an optimised non-uniform constellation using an algorithm with relatively low complexity and relatively high computational efficiency. For example, an algorithm in certain exemplary embodiments may obtain an optimised

non-uniform constellation much faster than an algorithm using a brute force method that searches all (or a high proportion of) possible candidate constellations. Certain exemplary embodiments provide an algorithm for obtaining optimised non-uniform constellations suitable for very high-order constellation (e.g. having more than 1024 constellation points).

[0279] Various embodiments are described below in which non-uniform (NU) Quadrature Amplitude Modulation (QAM) constellations are obtained. However, the skilled person will appreciate that the inventive concept is not limited to QAM constellations, but may be applied to other types of constellation.

[0280] As mentioned above, a constellation may be characterised by a number of parameters, for example specifying spacings between constellation points, or specifying the position of each positive real level (complete constellations may be obtained from these parameters because the constellations are the same for real and imaginary axes and the same for positive and negative values). In order to obtain an optimum constellation, a brute force approach may be taken in which combinations of values for each of the parameters are searched with a certain step size up to a certain maximum value. Each combination of values for each parameter corresponds to a distinct constellation. The constellation having the best performance is selected.

[0281] However, in certain exemplary embodiments, the number of parameters may be reduced by imposing one or more certain geometric and/or symmetry constraints on the constellations. For example, one constraint may be that the constellations are symmetric among the four quadrants of the constellations. In addition, the constellations may be constrained in that the constellation points are arranged in a QAM type lattice in which, within each quadrant, (i) constellation points are arranged in horizontal and vertical lines, (ii) the number of horizontal lines is the same as the number of vertical lines, (iii) the same number of constellation points are arranged in each horizontal line, and (iv) the same number of constellation points are arranged in each vertical line. In another example, a constellation may be constrained to be a circular constellation (e.g. a constellation having circular symmetry). Furthermore, constellations having the same relative arrangement, differing only in size, may be regarded as equivalent. In this case, one of the parameters may be set to a fixed value. The skilled person will appreciate that the inventive concept is not limited to the above examples, and that one or more additional or alternative constraints may be used.

[0282] In certain exemplary embodiments, a non-uniform QAM (NU-QAM) constellation may have a constellation conforming to one or more geometric and/or symmetry constraints, for example one or more, or all, of the above constraints, or a rotation and/or scaling thereof. A non-uniform N-QAM constellation may be a non-uniform QAM constellation including N constellation points.

[0283] By applying the constraints described above, the number of parameters may be reduced, for example to 1, 3, 7, 15, 31 and 63 parameter(s) for constellations including 16, 64, 256, 1024, 4096 and 16384 constellation points, respectively. The number of parameters in a reduced set of parameters may be denoted by b. For example b=1 for 16-QAM (in which there are 16 positions that are symmetric on the real/imaginary and positive/negative axes). Thus there are only 2 points to define. Since the total power of the constellation is typically normalized to one, fixing one parameter will fix the other. Thus b=1 for a square 16QAM.

[0284] In certain exemplary embodiments, combinations of values for each of the b parameters are searched with a step size d up to a maximum value A. Thus, the number of search iterations is equal to $(A/d)^b$.

[0285] A first algorithm according to certain exemplary embodiments for obtaining an optimum non-uniform constellation for a given SNR will now be described. The algorithm uses an iterative scheme to gradually modify an initial constellation until the constellation converges. For example, the initial constellation may be a uniform constellation, the constellation may be modified by changing the values of the parameters between iterations, and convergence occurs when the values of all the parameters change by less than a threshold amount between iterations. An optimum constellation may be defined as the constellation having the best performance according to any suitable measure. For example, the measure may include a coded modulation (CM) capacity or BICM capacity. In the following example a non-uniform 64-QAM constellation is obtained, in which the (reduced) number of variable parameters, b, is equal to 3.

[0286] FIG. **23** is a schematic diagram of a first algorithm, according to an exemplary embodiment, and FIG. **24** is a flowchart illustrating operations of the first algorithm, according to an exemplary embodiment. In the algorithm, the following variables are used. A parameter C_last denotes a particular constellation, corresponding to a particular set of values of the b parameters. A parameter C_last is initialised with a certain initial constellation, for example a uniform constellation. A parameter SNR denotes a signal-to-noise ratio. The SNR parameter is set to a desired value equal to an SNR for which an optimum constellation is desired. The parameter C_best denotes a constellation that maximises performance, for example maximises the CM capacity or BICM capacity, for a given SNR. The parameter d denotes a first step size used in the algorithm. The parameter d (or step) is initialised to a suitable value that may be determined theoretically and/or experimentally. A parameter Min_Step denotes a minimum allowed value for d, and is set to a fixed value.

[0287] In operation **201**, C_last is initialised to an input constellation. In a next operation **203**, step d is initialised to a value Ini_step. In operation **205**, a set of candidate constellations is obtained. The set of candidate constellations includes the constellation C_last and one or more modified constellations, where each modified constellation is obtained by modifying one or more of the parameter values defining C_last using any suitable scheme. In the illustrated example, the set of candidate constellations are created based on C_last and step size d, denoted by function CreateSet(C_Last, d). For example, for each constellation point, three derived constellations are generated [C_last, C_last+d, C_last−d]. Specifically, a set of constellations is derived such that the values of the b parameters in C_last are each set to one of n new values varying around the current parameter value. For example, three new values (n=3) may be used, which include (i) the current parameter value, (ii) a value d greater than the current parameter value, and (iii) a value d less than the current parameter value. For example, if there are two constellation levels to be defined then the number of combinations to be tested are 3×3 (corresponding to three positions for each level). All combinations of the new parameter values are used to generate the set of constellation. Thus, the set of constellations includes a total of n^b constellations. Although three new values for each parameter are used in the embodiment described above, any suitable number of new values may be used in another

embodiment. The set of new values may include the old (or current) value, or may not include the old value.

[0288] In certain exemplary embodiments, three values of each level are chosen so that the total number of possibilities to be tested is 3^b, where b is the number of levels (parameters) to be optimised. In the case of very high-order constellations, for example above 1K, 3^b may be very high. In this case, all the levels may be fixed except one, for which three possibilities are tested, C_last, C_last+d and C_last−d until convergence is achieved. The same operation may then be repeated for the other levels. The cost of this operation is multiplicative and not exponential (for example, if it is supposed that each level converges in one iteration then the cost will be 3×b instead 3^b.)

[0289] In operation **207**, the performance of each constellation in the set of derived (candidate) constellations is calculated or determined using any suitable performance measure (e.g. capacity). In operation **209**, the candidate constellation having the best performance (e.g. the candidate constellation that maximises the capacity) is assigned to C_best. In operation **211**, it is determined whether C_best differs from C_last by more than a threshold amount. For example, in the illustrated example, the threshold amount is equal to zero, so that it is determined whether C_best=C_last. That is, it is determined whether there is any difference between constellation C_best and constellation C_last (e.g. within a certain resolution). The difference may be any suitable measure of difference, for example including a difference based on geometry (e.g. differences in the locations of the constellation points of the constellations) and/or a performance measure (e.g. a difference in a certain performance measure between the constellations). If it is determined in operation **211** that C_best≠C_last, then in operation **213**, C_last takes the value C_best (i.e. so that the value of C_last in the next iteration is equal to the value of C_best in the current iteration) and the method returns to operation **205** in which a set of candidate constellations are created based on C_last and step size d, CreateSet(C_Last, d). On the other hand, if it is determined in operation **211** that C_best=C_last, then, in operation **215**, C_last takes the value C_best and the method moves to operation **217**.

[0290] In operation **217**, it is determined whether d<Min_Step. If it is determined in operation **217** that d≥Min_Step then the method moves to operation **219** in which the step size d is reduced. For example, d is divided by a certain factor (e.g. 2). Following operation **219**, the method returns to operation **205** in which a set of candidate constellations are created based on C_last and step size d (i.e., reduced d), CreateSet(C_Last, d). On the other hand, If it is determined in operation **217** that d<Min_Step then the value of C_best is saved and the algorithm ends.

[0291] FIG. **25** illustrates the convergence of C_last with respect to one of the parameters as the first algorithm of FIGS. **23** and **24** is performed. Initially, the value of the parameter converges to a certain value. When the value of the parameter has converged within a certain resolution, the step size d is reduced and the value of the parameter converges further, until the step size d has reached the minimum step size.

[0292] In the example shown in FIG. **25**, for each iteration, three new parameter values are tried, as represented by the vertical columns of circles. The best new parameter for each iteration is indicated in FIG. **25** as a filled circle. The best parameter value in one iteration is used as the new parameter value for the next iteration. Thus, in the example illustrated in FIG. **25**, in which three new parameter values are tried (including the current parameter and parameters an amount d above and below the current parameter), the filled circle of one iteration corresponds to the middle of the three circles arranged in a column for the next iteration.

[0293] In certain exemplary embodiments, operations **217** and **219** of the algorithm illustrated in FIG. **24** may be omitted so that operations **205**, **207**, **209**, **211**, **213** and **215** are performed using the initial step size. In this case, when it is determined in operation **215** that C_best=C_last, the step size is not reduced, but rather the value of C_best is saved and the algorithm ends. By omitting operations **217** and **219**, the algorithm may potentially complete more quickly. However, in this case the output constellation C_best may differ from the true optimum constellation more than the output constellation C_best obtained in the algorithm illustrated in FIG. **24** where the step size d is decreased. This may be seen in FIG. **25**, where it can be seen that the best parameter value in the final iteration lies closer to the optimal value (indicated by the horizontal line) than the best parameter value at the stage of convergence with the initial step size.

[0294] The first algorithm described above determines an optimum constellation based on a certain performance measure (e.g. capacity). In the following, various algorithms for determining an optimum constellation for a transmission system defined by a set of one or more system parameter values, where the constellation is optimised for a certain desired value of a system parameter (e.g. a certain SNR value or certain Ricean factor). In these embodiments, a system parameter value is set to an initial value (e.g. a relatively high value) and an optimum constellation is generated using an algorithm described above (e.g. the algorithm illustrated in FIG. **24**), wherein the performance measure is based on a defined transmission system having the set system parameter value. The system parameter value is then reset to a modified value (e.g. by reducing the value by a certain step size) and the algorithm is re-run. The other system parameter values may remain fixed. This process is repeated until the system parameter value reaches a certain desired value.

[0295] For example, FIG. **26** illustrates a second algorithm for determining an optimal constellation at a given SNR value S in an additive white Gaussian noise (AWGN) channel. In operation **401**, the algorithm is initialised by setting an SNR parameter to a high value N, where N is large. For example, the initial SNR value may be set to an SNR value above which a non-uniform constellation provides no better performance than an equivalent uniform constellation. This value may be determined, for example, theoretically and/or experimentally. In operation **401**, the parameter C_last is also initialised to a certain constellation, for example a uniform constellation.

[0296] In operation **403**, the first algorithm described above is run using the initialised constellation C_last as the input constellation and using the initialised SNR ratio. By applying the first algorithm, the constellation C_last will converge to an optimal constellation C_best for a specific input value of SNR. An output of operation **403** is C_best obtained using the first algorithm. In operation **405** the SNR value is reduced by a certain amount, for example one unit or step size. In operation **405**, C_last takes the value of C_best (i.e. so that the value of C_last in the next iteration is equal to the value of C_best in the current iteration). In operation **407** it is determined whether SNR<S. If it is determined in operation **407** that SNR≥S, then the method returns to operation **403**, in which the first algorithm is run with new values of C_last and SNR. On the other hand, if it is determined in operation **407** that SNR<S, then the value

of C_best is saved and the algorithm ends. By applying the second algorithm, the resulting constellation C_best is the optimal constellation for the desired SNR value S.

[0297] FIG. **27** illustrates convergence of the constellation C_best according to the second algorithm of FIG. **26** is performed. Each of the three curves represents variation in the value of a respective one of the three variable parameters. The solid constant line represents the fixed value of a fixed parameter. As shown in FIG. **27**, at the start of the second algorithm, starting from the right-hand side of FIG. **27**, the SNR value is high and the constellation is a uniform constellation, as defined by the values of the parameters on the right-hand side of FIG. **27**, labelled "Initial condition". At each iteration, an optimal constellation is obtained for the specific SNR value (indicated in FIG. **27** by the markers) The SNR is then reduced and the optimal constellation is obtained for the new SNR (this process being indicated for one of the parameters by the stepped line in FIG. **27**). As shown in FIG. **27**, the values of the parameters corresponding to the optimal constellation vary smoothly with varying SNR values. The iterations are repeated until the SNR value reaches the desired SNR value S.

[0298] By running the second algorithm illustrated in FIG. **26**, an optimal constellation is derived from each of a set of SNR values. These constellations may be stored in association with the corresponding SNR values, for example in a look-up table.

[0299] FIG. **28** illustrates a third algorithm for determining an optimal constellation at a given SNR value S in a Rician fading channel for a desired Rician factor K_rice. The Rician channel is given by

$$\sqrt{\frac{K}{K+1}}+\sqrt{\frac{1}{K+1}}\,h \qquad \text{[Equation 1]}$$

[0300] In Equation 1, K is the Rician factor and h is Rayleigh distributed (centred and normalised). Initially, the third algorithm applies the second algorithm described above to obtain the optimal constellation C_best at an SNR value S for an AWGN channel, C_best(AWGN) In operation **601**, parameter C_last is initialised to C_best(AWGN) In operation **601**, the Rician factor K is initialised to a high value, which may be determined theoretically and/or experimentally. For example, K may be initialised to a value K_rice+N, where N is large.

[0301] In operation **603**, the first algorithm described above is run using the initialised constellation C_last as the input constellation and using the initialised Rician factor K to obtain an optimal constellation C_best. In operation **605**, the Rician factor K is reduced by a certain amount, for example by one unit. In operation **605**, C_last takes the value of C_best (i.e. so that the value of C_last in the next iteration is equal to the value of C_best in the current iteration). In operation **607** it is determined whether K<K_rice. If it is determined in operation **607** that K≥K_rice, then the method returns to operation **603**, in which the first algorithm is run with new values of C_last and K. On the other hand, if it is determined in operation **607** that K<K_rice, then the value of C_best is saved and the algorithm ends. By applying the third algorithm, the resulting constellation C_best is the optimal constellation for the desired Rician factor K_rice.

[0302] FIG. **29** illustrates a fourth algorithm for determining an optimal constellation at a given SNR value S in a Rayleigh fading channel. A Rayleigh fading channel is a special case of Rician fading with the Rician factor K=0. Accordingly, the fourth algorithm is the same as the third algorithm described above, except that K_rice is set to zero.

[0303] Table 4 below compares the number of capacity calculation function calls for obtaining optimal constellations for various constellation sizes (16-QAM, 64-QAM and 256-QAM) using an exhaustive search, a restricted exhaustive search and an algorithm according to the present embodiment. The values in Table 4 are based on a step size d of 0.0125 and maximum value for the parameters of 10. Table 4 also indicates a factor difference between using a restricted exhaustive search and a search using an algorithm according to the present embodiment. As can be seen, the algorithm according to the present embodiment is significantly more efficient, for example by a factor of 1.15×10^{10} for 256-QAM.

TABLE 4

	Exhaustive search	Restricted exhaustive search	Algorithm according to the	Gain versus restricted
16QAM	800	800	21	38
64QAM	5.1e9	1.9e8	1701	117577
256QAM	2.1e21	2.5e15	216513	1.15e10

[0304] In Table 4, the difference between the exhaustive search and the restrictive exhaustive search is as following. It is assumed in the following that there are 4 levels (parameters) between 0 and 10. In the exhaustive search, each of the 4 parameters is searched over the whole range [0-10] with a certain granularity. In the case of the restricted exhaustive search, the range in which each level will fall is fixed. For example level 1 (first parameter) will be in the range [0-2.5], level 2 in the range [2.5-5], level 3 in the range [5-7.5], level 4 in the range [7.5-10]. In this manner, the number of possibilities is reduced.

[0305] FIG. **30** illustrates a fifth algorithm for determining an optimal constellation. This algorithm corresponds closely to the algorithm illustrated in FIG. **24**, but is modified to increase overall efficiency. This algorithm includes an inner loop for operations (operations **803-819**) corresponding to operations **203-219** of FIG. **24**. However, operation **805** for creating a set of candidate constellations is modified from the corresponding operation **205** of FIG. **24**. Specifically, in the algorithm of FIG. **30**, rather than modifying each of the b parameters and trying all combinations of the new parameters as in the algorithm of FIG. **24**, only one parameter is modified at a time. For example, within one iteration of the inner loop **803-819**, only one parameter (parameter i) is modified to produce a set of candidate constellation. The capacities of these constellations are calculated and the best constellation is selected, as in FIG. **24**.

[0306] In the algorithm of FIG. **30**, the value of i is varied from 1 to b using an outer loop (operations **821-825**). The algorithm of FIG. **30** is initialised in operation **801**, corresponding to operation **201** of FIG. **24**. It can be seen that, by using the algorithm of FIG. **30**, rather than the algorithm of FIG. **24**, the total number of candidate constellation tried (i.e. the total number of capacity calculations) is significantly reduced. However, in simulations, the optimal constellation obtained using the algorithm of FIG. **30** is very close to the optimal constellation obtained using the algorithm of FIG. **24**, which in turn is very close to the true optimal constellation obtained using an exhaustive search. The improvement in computational efficiency using the

algorithms according to the above embodiments, when compared to an exhaustive search, increases as the constellation order increases.

[0307] As with the algorithm illustrated in FIG. **24**, in certain exemplary embodiments, operations **817** and **819** of the algorithm illustrated in FIG. **30** may be omitted.

[0308] According to the above embodiments, optimal constellations may be obtained for particular parameters, for example SNR, Rician factor etc. These optimum constellations are obtained independently of any particular system implementation, for example independent of a particular coding scheme. In the following, various embodiments are described for obtaining an optimal constellation for a specific transmission system.

[0309] A transmission system may include a number of processes which may affect the optimal constellation, for example FEC encoding, bit interleaving, demultiplexing bits to cells, mapping cells to constellations, cell interleaving, constellation rotation, in-phase and quadrature phase (I/Q) component interleaving, inter-frame convolution and inter-frame block interleaving, and multiple-inputs-single-output (MISO) precoding. A QAM mapper is used in the Bit Interleaved Coded Modulation (BICM) chain to map bits to symbols. The QAM mapper may use a uniform constellation to map bits to cells (for example as done in DVB-T2). However, an increase in capacity may be achieved by using a fixed non-uniform constellation. A non-fixed non-uniform constellation (e.g. QAM) may be used to further increase capacity. The BICM capacity depends on the bit to cell mapping used. Optimisations are desirable in the LDPC design, the QAM mapping and the mapping of bits to cells.

[0310] In certain exemplary embodiments, different constellations are generated using a certain step size. A bit error rate (BER), a block error rate and/or a packet error rate corresponding to the constellations are obtained, and the best constellation is selected based on one or more of the aforementioned error rates.

[0311] In certain exemplary embodiments, the process illustrated in FIG. **31** may be carried out to obtain an optimal constellation for a specific system. In operation **901**, a uniform constellation (e.g. uniform QAM) is selected. In operation **903**, BER values for the selected uniform constellation are obtained over a range of SNR values (e.g. using simulation or by obtaining the BER values theoretically or experimentally). These values may be obtained based on a specific system, for example using a particular coding scheme (e.g. LDPC coding with a certain parity check matrix) with a certain coding rate and a certain bit interleaver and cell interleaver. FIG. **32** illustrates an exemplary plot for 64-QAM using an LDPC coding rate of 2/3 from DVB-T2 in an AWGN channel.

[0312] In operation **905**, an SNR at which the BER falls below a threshold value (e.g. 0.001) is determined. The threshold value may be selected such that the resulting SNR falls within a "waterfall zone" of the BER curve (i.e. the zone at which the BER falls relatively rapidly with increasing SNR). The determined SNR value may be denoted S and referred to as a "waterfall" SNR.

[0313] Next, an optimal constellation may be obtained for the SNR value S determined in operation **905**.

[0314] For example, in some exemplary embodiments, in operation **907***a*, an optimal constellation may be selected from optimal constellations obtained when performing the algorithms described above in relation to FIGS. **23-30** (and stored in a look-up table). Specifically, an optimal constellation previously determined for the SNR value S may be retrieved from the look-up table.

[0315] Alternatively, an iterative process may be performed to obtain an optimal (non-uniform) constellation. Specifically, after operation **905**, the process moves to operation **907***b* in which the algorithms described above in relation to FIGS. **23-30** are used to obtain an optimal constellation for the SNR value S (or for a value close to S). After operation **907***b*, the process returns to operation **903**, in which BER values are obtained over a range of SNR values. In this iteration, the BER values are obtained for the optimal constellation obtained in operation **907***b* (rather than for the initial uniform constellation as in the first iteration). In a similar manner as previously described, the SNR value at which the BER falls below a threshold value (using the new set of BER values for the optimal constellation) is determined in operation **905**, and a new optimal constellation for the newly determined SNR value is obtain in operation **907***b*. The previously described operations **903**, **905**, **907** may be repeated a certain number of time (for example a predetermined number of times). Alternatively, the algorithm may terminate when the waterfall SNR stops decreasing between iterations, and instead starts increasing.

[0316] FIGS. **33** and **34** illustrate a sixth algorithm for determining an optimal constellation. This algorithm corresponds closely to the algorithm illustrated in FIG. **30**, but is modified to improve performance. In particular, this algorithm introduces a concept of a direction of convergence of a parameter value. For example, within the inner loop of the algorithm, the direction is initialised to 0. When creating a set of candidate constellations, the candidate set depends on the direction parameter. When the best constellation is selected in operation **1109**, the direction of convergence of the value of parameter i is obtained. For example, if the parameter value is converging upwards, then the direction parameter may be set to +1, if the parameter is converging downwards, then the direction parameter may be set to −1, and if the parameter does not change, then the direction parameter may be set to 0. As illustrated in FIG. **34**, the number of candidate constellations may be reduced when the parameter value is converging upwards or downwards.

[0317] As described above, an optimum constellation may be obtained for a particular system implementation, and/or for certain system parameter values. For example, an optimum constellation (e.g. a constellation that optimises the BICM capacity) may be obtained for a certain propagation channel type (e.g. AWGN, Rayleigh or Typical Urban, TU6, channel) and for a certain SNR. However, in some cases, data may be transmitted in different scenarios. For example, data may be transmitted through different types of channels and may be received with different SNRs. Furthermore, it may be desirable or required that a data transmission system uses the same constellation, regardless of the scenario (e.g. channel type or SNR), for example in order to reduce system complexity. In some cases, a transmission system may use a certain constellation for many different scenarios (e.g. channel types and SNRs).

[0318] FIGS. **35-38**B illustrate an algorithm for obtaining a constellation that is optimised (e.g. achieves the best capacity) with respect to two or more different scenarios (e.g. different channel types and/or SNR values). The algorithm includes a number of different parts. First, the waterfall SNR for each channel type (e.g. propagation channel type) is obtained using an algorithm similar to the algorithm illustrated in FIG. **31**. A weighted performance measure function (e.g. weighted capacity) for an input constellation is defined, based on different scenarios (e.g. different channel types and SNR values). Then, an algorithm similar to the algorithms illustrated in FIG. **24**, **30** or **33** is applied to

determine an optimum constellation, where the performance measure is used based on the weighted performance measure.

[0319] FIG. **35** illustrates a process for obtaining the waterfall SNR for each channel type. Each channel type is treated separately in order to obtain its waterfall SNR. In particular, the process illustrated in FIG. **35** is repeated for each channel type to obtain a respective waterfall SNR for that channel type. The process illustrated in FIG. **35** operates in substantially the same manner as the algorithm illustrated in FIG. **31**, and therefore a detailed description will be omitted for conciseness. However, rather than outputting an optimal constellation, as in the algorithm illustrated in FIG. **31**, the process illustrated in FIG. **35** instead outputs the waterfall SNR determined in the final iteration of the process. The process illustrated in FIG. **35** (including BER simulation and capacity optimisation operations) is performed based on a certain channel type, and the output waterfall SNR is determined as the waterfall SNR associated with that channel type.

[0320] FIG. **36** schematically illustrates a process for obtaining a weighted performance measure function for an input constellation based on different transmission scenarios. In this example, the weighted performance measure is a weighted capacity, and the different scenarios include different channel types and associated waterfall SNR values. As illustrated in FIG. **36**, a candidate constellation is provided as an input. For each channel type and associated waterfall SNR, the BICM capacity for the input constellation based on the channel type and waterfall SNR is obtained. Each obtained BICM capacity is then multiplied by a respective weight and the weighted BICM capacities are added together to obtain an output weighted average BICM capacity. The weights may be selected according to any suitable criteria. For example, a relatively common or important channel type may be associated with a relatively large weight.

[0321] FIG. **37** illustrates a process for obtaining an optimum constellation. The process illustrated in FIG. **37** operates in substantially the same manner as the algorithm illustrated in FIG. **24**, **30** or **33**, and therefore a detailed description will be omitted for conciseness. However, when determining the performance of a candidate constellation in the process illustrated in FIG. **37**, the performance is determined based on the weighted performance measure described above in relation to FIG. **36**.

[0322] In the process illustrated in FIG. **37**, in some situation, a certain constellation may achieve the best performance with respect to the weighted performance measure, even though the performance of that constellation with respect to the BICM capacity based on an individual channel and SNR may be relatively low. In certain exemplary embodiments, to ensure that a constellation obtained using the algorithm is able to achieve at least a certain level of performance for one or more, or all, transmission scenarios, an additional criterion may be applied when testing each candidate constellation to obtain the constellation C_best. Specifically, any candidate constellation that does not achieve at least a threshold performance with respect to one or more certain individual scenarios, or all scenarios, is ignored and cannot be selected as C_best, even if that constellation achieves the best performance with respect to the weighted performance measure.

[0323] In the process illustrated in FIG. **37**, a set of candidate constellations may be derived using any suitable method, for example the method described above in relation to FIG. **31** based on a step size d. FIGS. **38**A and **38**B illustrate alternative schemes for generating a candidate constellation from a previous constellation, C_last, that may be used in certain exemplary embodiments. In FIGS. **38**A and **38**B, the open circles represent the constellation points of a previous constellation, C_last. For each constellation point of the previous constellation, a respective set of N modified constellation points are defined, as indicated in FIGS. **38**A and **38**B as filled circles. Each set of modified constellation points forms a pattern of constellation points located relatively close to the respective constellation point of the previous constellation.

[0324] For example, as illustrated in FIG. **38**A, each set of modified constellation points may form a square or rectangular lattice of N=8 constellation points surrounding a respective constellation point of the previous constellation. The lattice spacing is equal to d. Alternatively, as illustrated in FIG. **38**B, each set of modified constellation points may form a ring of N=8 constellation points surrounding a respective constellation point of the previous constellation. The radius of the ring is equal to d.

[0325] A candidate constellation may be obtained by selecting, for each constellation point in the previous constellation, either the constellation point of the previous constellation itself or one of the constellation points of a respective set of modified constellation points.

[0326] In the examples described above, a weighted performance measure is defined based on different transmission scenarios. For example, in the case illustrated in FIG. **36**, each transmission scenario includes a different channel type and an associated waterfall SNR value. Accordingly, a constellation optimised for a range of channel types and associated SNR values may be obtained. In an alternative embodiment, an optimal constellation may be obtained for different transmission scenarios, in the case where each transmission scenario includes the same channel type, but involves different SNR values (e.g. a set of SNR values S1, S1+d, S1+2d, S1+3d, . . . , S2, where d is a step size). That is, an optimal constellation may be obtained for a fixed channel type that is intended to be used over a range of SNR values. In this case, the algorithm described above in relation to FIGS. **35**-**38**B may be used, except that when determining the weighted performance measure as illustrated in FIG. **36**, instead of determining individual BICM capacities based on respective channel types and associated waterfall SNR values, the individual BICM capacities are determined based on the fixed channel type and respective SNR values S1, S1+d, S1+2d, S1+3d, . . . , S2.

[0327] In the algorithms described above, a technique may be applied to reduce the overall complexity. In particular, when a set of candidate constellations is generated and the performance of the candidate constellations are tested, those candidate constellations that have been previously tested (i.e. in one or more previous iteration) are not re-tested. That is, in a current iteration, only those candidate constellations that have not been tested in previous iterations are tested.

[0328] For example, as described above, a first set of candidate constellations, A, is generated in an iteration, and the best performing candidate constellation, a ($a \in A$), is selected from this set. In a next iteration, a second set of candidate constellations, B, is generated based on the previously selected constellation a ($a \in B$). In this next iteration, the best performing candidate constellation b ($b \in B$) from set B needs to be determined.

[0329] Typically, there will be at least some overlap between the two sets of candidate constellations A and B, such that one or more candidate constellations belong to both sets A and B (i.e. $A \cap B @ \emptyset$), including constellation a.

Since it is known that constellation a has the best performance of all the constellations in set A, then it is also known that constellation a has the best performance of all the constellations belonging to the overlap between sets A and B (i.e. A∩B).

[0330] Accordingly, when testing the constellations in set B to determine the best performing constellation, b, it is not necessary to re-test those constellations belonging to the overlap between sets A and B (i.e. it is not necessary to re-test those constellations in the set A∩B). Instead, rather than testing all constellations in set B, only those constellations belonging to the smaller set of constellations B*, including constellations belonging to set B but excluding any constellations that also belong to set A (i.e. B*=B−A) are tested. Then, the best performing constellation from the set formed from the union of B* and the previous best performing constellation, a (i.e. the best performing constellation from the set B*∪{a}) is selected as the best performing constellation, b, of set B.

[0331] An example of the above principle in relation to the example shown in FIG. **38**A is illustrated in FIG. **39**. In the example of FIG. **39**, at iteration i, it was found that the constellation point indicated as a black circle is the best performing constellation. At iteration i+1, there is no need to test the common subset (including the white circles and the black circle), because it was already tested before and gave an inferior performance. That is, at iteration i+1, only the dark grey circles need to be tested. Accordingly, in the illustrated example, a reduction in complexity of 44% (=4/9) is achieved.

[0332] FIG. **40** illustrates an apparatus for implementing an algorithm according to one or more of the embodiments described above. The apparatus is configured for generating a non-uniform constellation. The apparatus includes a block for performing a first process. The block for performing the first process includes: a block for obtaining a first constellation defined by one or more parameter values; and a block for generating a second constellation based on the first constellation using a second process. The block for generating the second constellation based on the first constellation using the second process includes: a block for obtaining a set of candidate constellations, wherein the set of candidate constellations includes the first constellation and one or more modified constellations, wherein each modified constellation is obtained by modifying the parameter values defining the first constellation; a block for determining performance of each candidate constellation according to a predetermined performance measure; and a block for selecting the candidate constellation having the best performance as the second constellation. The block for performing the first process further includes a block for determining a difference between the first constellation and the second constellation; and a block for, if the second constellation differs from the first constellation by more than a threshold amount, causing the block for performing the first process to repeat the first process using the second constellation generated in a current iteration of the first process as the first constellation in a next iteration.

[0333] The skilled person will appreciate that the functions of any two or more blocks illustrated in FIG. **40** may be performed by a single block, and that the functions of any block illustrated in FIG. **40** may be performed by two or more blocks. A block may be implemented in any suitable form, for example hardware, software, firmware, or any suitable combination of hardware, software and firmware.

[0334] A constellation obtained by a method according to the above exemplary embodiments may be used in a digital broadcasting system to transmit data from a transmitter side to a receiver side. In certain exemplary embodiments, the system includes a transmitter arranged to obtain data (e.g. a data stream), perform any required encoding and/or other processing of the data, modulate a signal using the data according to a modulation scheme corresponding to the constellation, and transmit the modulated signal. The system further includes a receiver configured to receive a modulated signal, demodulate the signal according to a demodulation scheme corresponding to the constellation (or a similar or corresponding constellation), and perform any necessary decoding and/or other processing to recover the original data. Certain exemplary embodiments may include a transmitter side apparatus only, a receiver side apparatus only, or a system including both a transmitter side apparatus and a receiver side apparatus.

[0335] In case of non-uniform constellations, it is possible to design constellations by relaxing only one constraint, i.e. by keeping a constellations square but changing a distance between constellation points. This form of non-uniform constellations (NUCs) can be referred to as one-dimensional (1D) NUCs. 1D NUCs can be described by the levels at which the constellations occur in the real positive part. The other points can be deduced by using the symmetry of the four quadrants as well as the real and imaginary symmetry. 1D-NUCs are simple to decode because of independence of the real and imaginary part. Two (2) pulse amplitude modulator (PAM) demappers can be used to decode 1D-NUCs.

[0336] It is possible to design a different type of NUC by relaxing both constraints: the square shape and the uniform distance between constellation points. Optimal constellations will have a tendency to look like a circular constellation. This type of NUC can be referred to as 2D-NUC. The 2D-NUC has a higher capacity than the 1D-NUC and a better BER/FER performance. However, performance of 2D-NUC comes at the expense of a more complex receiver demapper. Since the real and imaginary axes are not symmetrical, a 2D demapper is needed in order to decode a 2D-NUC constellation. In the case of 2D-NUC, a complete set of points need to be specified. It is possible to specify only the points belonging to the first quadrant and deduce the other points by supposing that the constellation is symmetrical.

[0337] Optimizing 1D and 2D NUCs depend on an SNR at which a capacity needs to be optimized. In the case of a BICM chain, an SNR can be selected to be a waterfall SNR of a BER/FER curve. The BER/FER waterfall can be defined as an SNR at which a BER curve falls below a certain level, for example 10e-6. The waterfall SNR depends on a coding rate of an LDPC encoder/decoder. As the code rate increases, the waterfall SNR increases. For this reason, a different NUC is associated with each LDPC encoder coding rate. The waterfall SNR also increases with a QAM constellation size (M). This is because the a receiver needs a higher SNR to decode a higher QAM constellation. Thus, a constellations size and a coding rate define the waterfall SNR. The waterfall SNR is used to optimize the constellations. Here, the coding rates include: 2/15, 3/15 and 4/15. The NUC sizes are: 16QAM, 64QAM, 256QAM and 1K QAM. For the first three QAM sizes only 2D constellations are proposed. For 1K QAM both 1D and 2D constellations are proposed.

[0338] Hereinafter, an example of constellation points of a constellation obtained by applying the algorithms described above by coding rates will be described.

[0339] In the following exemplary embodiment, a restriction is added to the process of determining capacity according to SNR with respect to the existing NUC designing method.

[0340] When SNR is given, it is general to calculate the maximum transmission capacity which can be transmitted with error-free. In other words, when calculating capacity by setting SNR with respect to BER/FER waterfall, the SNR indicates an area where a bit error or a frame error occurs, but actual capacity indicates transmission capacity under error-free circumstances, and thus, there may be a contradiction.

[0341] Therefore, in the present disclosure, when calculating capacity with respect to the SNR, a correction factor is added.

[0342] For example, when SNR1 is decided with respect to CH1 in FIG. **23**, if the capacity value under error-free state is C1, corrected C1 value, that is C1' is defined as shown below.

$$C1'=C1\times(1-H(P_b)) \quad \text{[Equation 2]}$$

[0343] Here, P_b indicates a BER value which determines a waterfall area, and function H(x) indicates binary entropy function, $H(x)=-x\times\log_2(x)-(1-x)\times\log_2(1-x)$.

[0344] In this case, the reason why the value (1−H(Pb)) which is equal to or smaller than 1 is multiplied to the existing capacity value as indicated in Equation 2 is as shown below.

[0345] With respect to a case where a bit error occurs as much as the probability Pb in a transmission channel, if it is assumed that a bit error as much as Pb has been occurred before transmitting source information, and error does not occur while source information is being transmitted, there is no difference in terms of a bit error in the final transmitting/receiving end. As described above, when the probability of error occurrence regarding a channel is considered a loss of source information, there is an effect as if lossy compression is applied as much as $H(P_b)$ compared to given data, by the rate distortion theory of Shannon. That is, in conclusion, it can be considered that data amount which can be transmitted through firstly given channel, that is capacity, will be reduced compared to the channel $(1-H(P_b))$ without an error or loss.

[0346] When the same value is applied to all the channels which consider Pb of FIG. **36**, the same factor $(1-H(P_b))$ is multiplied, and thus, the same factor $(1-H(P_b))$ is multiplied to the value of weighted capacity. Accordingly, there is no difference in the constellation points of the optimized NUC. However, target BER may be different for each channel, and therefore, the order of size of weighted capacity can be different. Accordingly, the constellation points of optimized NUC can be different. For example, an AWGN channel is applied to a fixed device in general, and thus, very low BER is required. Therefore, when calculating capacity, BER=1e-8 may be considered. As Rayleigh channel is largely considered for a mobile channel which experiences fading, BER higher than that of AWGN channel is required. Thus, BER=1e-6 can be considered. As described above, if different BER requirements are set for respective channels, values of factor for final capacity become different, and therefore, this value may be different from the NUC constellation points which are obtained in consideration of capacity only without reflecting BER.

[0347] To be specific, Table 5 indicates values of constellation points of a normalized 2D NU 16-QAM constellation (2D 16NUC) which is obtained by applying the algorithms described above using respective coding rates 11/15 and 13/15 for a single SNR value.

TABLE 5

	Coding Rate	
w/Shape	11/15	13/15
w0	0.934157+0.984668i	0.951702+0.951102i
w1	0.986649+0.29029i	0.952402+0.306101i
w2	0.271571+0.932473i	0.306701+0.952402i
w3	0.290092+0.269491i	0.306101+0.306701i
w4	−0.934157+0.984668i	−0.951702+0.951102i
w5	−0.986649+0.29029i	−0.952402+0.306101i
w6	−0.271571+0.932473i	−0.306701+0.952402i
w7	−0.290092+0.269491i	−0.306101+0.306701i
w8	0.934157−0.984668i	0.951702−0.951102i
w9	0.986649−0.29029i	0.952402−0.306101i
w10	0.271571−0.932473i	0.306701−0.952402i
w11	0.290092−0.269491i	0.306101−0.306701i
w12	−0.934157−0.984668i	−0.951702−0.951102i
w13	−0.986649−0.29029i	−0.952402−0.306101i
w14	−0.271571−0.932473i	−0.306701−0.952402i
w15	−0.290092−0.269491i	−0.306101−0.306701i

[0348] Table 6 indicates values of constellation points of a normalized 2D NU 64-QAM constellation (2D 64NUC) which is obtained by applying the algorithms described above using respective coding rates 5/15 and 11/15 for a single SNR value.

TABLE 6

	Coding Rate	
w/Shape	5/15	11/15
w0	0.573871+0.976254i	1.44428+0.26833i
w1	0.782918+1.24775i	0.747144+1.22429i
w2	0.297932+1.09229i	1.17488+0.773395i
w3	0.330876+1.43264i	0.713766+0.820077i
w4	0.976152+0.571463i	0.163802+1.07689i
w5	1.2484+0.780259i	0.292681+1.42171i
w6	1.09086+0.297075i	0.146222+0.745719i
w7	1.43269+0.330462i	0.413364+0.740848i
w8	0.28977+0.524646i	1.02034+0.151686i
w9	0.228711+0.395542i	0.665303+0.13565i
w10	0.247472+0.532718i	0.963923+0.446505i
w11	0.210804+0.391137i	0.674568+0.433914i
w12	0.523642+0.289419i	0.127098+0.142777i
w13	0.394508+0.228938i	0.378204+0.140639i
w14	0.531678+0.24753i	0.131136+0.428806i
w15	0.390066+0.211214i	0.391865+0.427618i
w16	−0.573871+0.976254i	−1.44428+0.26833i
w17	−0.782918+1.24775i	−0.747144+1.22429i
w18	−0.297932+1.09229i	−1.17488+0.773395i
w19	−0.330876+1.43264i	−0.713766+0.820077i
w20	−0.976152+0.571463i	−0.163802+1.07689i
w21	−1.2484+0.780259i	−0.292681+1.42171i
w22	−1.09086+0.297075i	−0.146222+0.745719i
w23	−1.43269+0.330462i	−0.413364+0.740848i
w24	−0.28977+0.524646i	−1.02034+0.151686i
w25	−0.228711+0.395542i	−0.665303+0.13565i
w26	−0.247472+0.532718i	−0.963923+0.446505i
w27	−0.210804+0.391137i	−0.674568+0.433914i
w28	−0.523642+0.289419i	−0.127098+0.142777i
w29	−0.394508+0.228938i	−0.378204+0.140639i
w30	−0.531678+0.24753i	−0.131136+0.428806i
w31	−0.390066+0.211214i	−0.391865+0.427618i
w32	0.573871−0.976254i	1.44428−0.26833i
w33	0.782918−1.24775i	0.747144−1.22429i
w34	0.297932−1.09229i	1.17488−0.773395i
w35	0.330876−1.43264i	0.713766−0.820077i
w36	0.976152−0.571463i	0.163802−1.07689i
w37	1.2484−0.780259i	0.292681−1.42171i
w38	1.09086−0.297075i	0.146222−0.745719i
w39	1.43269−0.330462i	0.413364−0.740848i
w40	0.28977−0.524646i	1.02034−0.151686i

TABLE 6-continued

w/Shape	Coding Rate 5/15	Coding Rate 11/15
w41	0.228711−0.395542i	0.665303−0.13565i
w42	0.247472−0.532718i	0.963923−0.446505i
w43	0.210804−0.391137i	0.674568−0.433914i
w44	0.523642−0.289419i	0.127098−0.142777i
w45	0.394508−0.228938i	0.378204−0.140639i
w46	0.531678−0.24753i	0.131136−0.428806i
w47	0.390066−0.211214i	0.391865−0.427618i
w48	−0.573871−0.976254i	−1.44428−0.26833i
w49	−0.782918−1.24775i	−0.747144−1.22429i
w50	−0.297932−1.09229i	−1.17488−0.773395i
w51	−0.330876−1.43264i	−0.713766−0.820077i
w52	−0.976152−0.571463i	−0.163802−1.07689i
w53	−1.2484−0.780259i	−0.292681−1.42171i
w54	−1.09086−0.297075i	−0.146222−0.745719i
w55	−1.43269−0.330462i	−0.413364−0.740848i
w56	−0.28977−0.524646i	−1.02034−0.151686i
w57	−0.228711−0.395542i	−0.665303−0.13565i
w58	−0.247472−0.532718i	−0.963923−0.446505i
w59	−0.210804−0.391137i	−0.674568−0.433914i
w60	−0.523642−0.289419i	−0.127098−0.142777i
w61	−0.394508−0.228938i	−0.378204−0.140639i
w62	−0.531678−0.24753i	−0.131136−0.428806i
w63	−0.390066−0.211214i	−0.391865−0.427618i

[0349] Table 7 indicates values of constellation points of a normalized 2D NU 256-QAM constellation (2D 256NUC) which is obtained by applying the algorithms described above using respective coding rates 9/15 for a single SNR value.

TABLE 7

w/Shape	Coding Rate 9/15
w0	0.0899+0.1337i
w1	0.0910+0.1377i
w2	0.0873+0.3862i
w3	0.0883+0.3873i
w4	0.1115+0.1442i
w5	0.1135+0.1472i
w6	0.2067+0.3591i
w7	0.1975+0.3621i
w8	0.1048+0.7533i
w9	0.1770+0.7412i
w10	0.1022+0.5904i
w11	0.1191+0.5890i
w12	0.4264+0.6230i
w13	0.3650+0.6689i
w14	0.3254+0.5153i
w15	0.2959+0.5302i
w16	0.3256+0.0768i
w17	0.3266+0.0870i
w18	0.4721+0.0994i
w19	0.4721+0.1206i
w20	0.2927+0.1267i
w21	0.2947+0.1296i
w22	0.3823+0.2592i
w23	0.3944+0.2521i
w24	0.7755+0.1118i
w25	0.7513+0.2154i
w26	0.6591+0.1033i
w27	0.6446+0.1737i
w28	0.5906+0.4930i
w29	0.6538+0.4155i
w30	0.4981+0.3921i
w31	0.5373+0.3586i
w32	0.1630+1.6621i
w33	0.4720+1.5898i
w34	0.1268+1.3488i

TABLE 7-continued

w/Shape	Coding Rate 9/15
w35	0.3752+1.2961i
w36	1.0398+1.2991i
w37	0.7733+1.4772i
w38	0.8380+1.0552i
w39	0.6242+1.2081i
w40	0.1103+0.9397i
w41	0.2415+0.9155i
w42	0.1118+1.1163i
w43	0.3079+1.0866i
w44	0.5647+0.7638i
w45	0.4385+0.8433i
w46	0.6846+0.8841i
w47	0.5165+1.0034i
w48	1.6489+0.1630i
w49	1.5848+0.4983i
w50	1.3437+0.1389i
w51	1.2850+0.4025i
w52	1.2728+1.0661i
w53	1.4509+0.7925i
w54	1.0249+0.8794i
w55	1.1758+0.6545i
w56	0.9629+0.1113i
w57	0.9226+0.2849i
w58	1.1062+0.1118i
w59	1.0674+0.3393i
w60	0.7234+0.6223i
w61	0.8211+0.4860i
w62	0.8457+0.7260i
w63	0.9640+0.5518i
w64	−0.0899+0.1337i
w65	−0.0910+0.1377i
w66	−0.0873+0.3862i
w67	−0.0883+0.3873i
w68	−0.1115+0.1442i
w69	−0.1135+0.1472i
w70	−0.2067+0.3591i
w71	−0.1975+0.3621i
w72	−0.1048+0.7533i
w73	−0.1770+0.7412i
w74	−0.1022+0.5904i
w75	−0.1191+0.5890i
w76	−0.4264+0.6230i
w77	−0.3650+0.6689i
w78	−0.3254+0.5153i
w79	−0.2959+0.5302i
w80	−0.3256+0.0768i
w81	−0.3266+0.0870i
w82	−0.4721+0.0994i
w83	−0.4721+0.1206i
w84	−0.2927+0.1267i
w85	−0.2947+0.1296i
w86	−0.3823+0.2592i
w87	−0.3944+0.2521i
w88	−0.7755+0.1118i
w89	−0.7513+0.2154i
w90	−0.6591+0.1033i
w91	−0.6446+0.1737i
w92	−0.5906+0.4930i
w93	−0.6538+0.4155i
w94	−0.4981+0.3921i
w95	−0.5373+0.3586i
w96	−0.1630+1.6621i
w97	−0.4720+1.5898i
w98	−0.1268+1.3488i
w99	−0.3752+1.2961i
w100	−1.0398+1.2991i
w101	−0.7733+1.4772i
w102	−0.8380+1.0552i
w103	−0.6242+1.2081i
w104	−0.1103+0.9397i
w105	−0.2415+0.9155i
w106	−0.1118+1.1163i
w107	−0.3079+1.0866i
w108	−0.5647+0.7638i

TABLE 7-continued

w/Shape	Coding Rate 9/15
w109	−0.4385+0.8433i
w110	−0.6846+0.8841i
w111	−0.5165+1.0034i
w112	−1.6489+0.1630i
w113	−1.5848+0.4983i
w114	−1.3437+0.1389i
w115	−1.2850+0.4025i
w116	−1.2728+1.0661i
w117	−1.4509+0.7925i
w118	−1.0249+0.8794i
w119	−1.1758+0.6545i
w120	−0.9629+0.1113i
w121	−0.9226+0.2849i
w122	−1.1062+0.1118i
w123	−1.0674+0.3393i
w124	−0.7234+0.6223i
w125	−0.8211+0.4860i
w126	−0.8457+0.7260i
w127	−0.9640+0.5518i
w128	0.0899−0.1337i
w129	0.0910−0.1377i
w130	0.0873−0.3862i
w131	0.0883−0.3873i
w132	0.1115−0.1442i
w133	0.1135−0.1472i
w134	0.2067−0.3591i
w135	0.1975−0.3621i
w136	0.1048−0.7533i
w137	0.1770−0.7412i
w138	0.1022−0.5904i
w139	0.1191−0.5890i
w140	0.4264−0.6230i
w141	0.3650−0.6689i
w142	0.3254−0.5153i
w143	0.2959−0.5302i
w144	0.3256−0.0768i
w145	0.3266−0.0870i
w146	0.4721−0.0994i
w147	0.4721−0.1206i
w148	0.2927−0.1267i
w149	0.2947−0.1296i
w150	0.3823−0.2592i
w151	0.3944−0.2521i
w152	0.7755−0.1118i
w153	0.7513−0.2154i
w154	0.6591−0.1033i
w155	0.6446−0.1737i
w156	0.5906−0.4930i
w157	0.6538−0.4155i
w158	0.4981−0.3921i
w159	0.5373−0.3586i
w160	0.1630−1.6621i
w161	0.4720−1.5898i
w162	0.1268−1.3488i
w163	0.3752−1.2961i
w164	1.0398−1.2991i
w165	0.7733−1.4772i
w166	0.8380−1.0552i
w167	0.6242−1.2081i
w168	0.1103−0.9397i
w169	0.2415−0.9155i
w170	0.1118−1.1163i
w171	0.3079−1.0866i
w172	0.5647−0.7638i
w173	0.4385−0.8433i
w174	0.6846−0.8841i
w175	0.5165−1.0034i
w176	1.6489−0.1630i
w177	1.5848−0.4983i
w178	1.3437−0.1389i
w179	1.2850−0.4025i
w180	1.2728−1.0661i
w181	1.4509−0.7925i
w182	1.0249−0.8794i

TABLE 7-continued

w/Shape	Coding Rate 9/15
w183	1.1758−0.6545i
w184	0.9629−0.1113i
w185	0.9226−0.2849i
w186	1.1062−0.1118i
w187	1.0674−0.3393i
w188	0.7234−0.6223i
w189	0.8211−0.4860i
w190	0.8457−0.7260i
w191	0.9640−0.5518i
w192	−0.0899−0.1337i
w193	−0.0910−0.1377i
w194	−0.0873−0.3862i
w195	−0.0883−0.3873i
w196	−0.1115−0.1442i
w197	−0.1135−0.1472i
w198	−0.2067−0.3591i
w199	−0.1975−0.3621i
w200	−0.1048−0.7533i
w201	−0.1770−0.7412i
w202	−0.1022−0.5904i
w203	−0.1191−0.5890i
w204	−0.4264−0.6230i
w205	−0.3650−0.6689i
w206	−0.3254−0.5153i
w207	−0.2959−0.5302i
w208	−0.3256−0.0768i
w209	−0.3266−0.0870i
w210	−0.4721−0.0994i
w211	−0.4721−0.1206i
w212	−0.2927−0.1267i
w213	−0.2947−0.1296i
w214	−0.3823−0.2592i
w215	−0.3944−0.2521i
w216	−0.7755−0.1118i
w217	−0.7513−0.2154i
w218	−0.6591−0.1033i
w219	−0.6446−0.1737i
w220	−0.5906−0.4930i
w221	−0.6538−0.4155i
w222	−0.4981−0.3921i
w223	−0.5373−0.3586i
w224	−0.1630−1.6621i
w225	−0.4720−1.5898i
w226	−0.1268−1.3488i
w227	−0.3752−1.2961i
w228	−1.0398−1.2991i
w229	−0.7733−1.4772i
w230	−0.8380−1.0552i
w231	−0.6242−1.2081i
w232	−0.1103−0.9397i
w233	−0.2415−0.9155i
w234	−0.1118−1.1163i
w235	−0.3079−1.0866i
w236	−0.5647−0.7638i
w237	−0.4385−0.8433i
w238	−0.6846−0.8841i
w239	−0.5165−1.0034i
w240	−1.6489−0.1630i
w241	−1.5848−0.4983i
w242	−1.3437−0.1389i
w243	−1.2850−0.4025i
w244	−1.2728−1.0661i
w245	−1.4509−0.7925i
w246	−1.0249−0.8794i
w247	−1.1758−0.6545i
w248	−0.9629−0.1113i
w249	−0.9226−0.2849i
w250	−1.1062−0.1118i
w251	−1.0674−0.3393i
w252	−0.7234−0.6223i
w253	−0.8211−0.4860i
w254	−0.8457−0.7260i
w255	−0.9640−0.5518i

[0350] Meanwhile, values of constellation points of one quadrant among values of constellation points described in Table 7 may be presented as below Table 7-1.

TABLE 7-1

w/Shape	Coding Rate 9/15
w0	0.0899+0.1337i
w1	0.0910+0.1377i
w2	0.0873+0.3862i
w3	0.0883+0.3873i
w4	0.1115+0.1442i
w5	0.1135+0.1472i
w6	0.2067+0.3591i
w7	0.1975+0.3621i
w8	0.1048+0.7533i
w9	0.1770+0.7412i
w10	0.1022+0.5904i
w11	0.1191+0.5890i
w12	0.4264+0.6230i
w13	0.3650+0.6689i
w14	0.3254+0.5153i
w15	0.2959+0.5302i
w16	0.3256+0.0768i
w17	0.3266+0.0870i
w18	0.4721+0.0994i
w19	0.4721+01206i
w20	0.2927+0.1267i
w21	0.2947+0.1296i
w22	0.3823+0.2592i
w23	0.3944+0.2521i
w24	0.7755+0.1118i
w25	0.7513+0.2154i
w26	0.6591+0.1033i
w27	0.6446+0.1737i
w28	0.5906+0.4930i
w29	0.6538+0.4155i
w30	0.4981+0.3921i
w31	0.5373+0.3586i
w32	0.1630+1.6621i
w33	0.4720+1.5898i
w34	0.1268+1.3488i
w35	0.3752+1.2961i
w36	1.0398+1.2991i
w37	0.7733+1.4772i
w38	0.8380+1.0552i
w39	0.6242+1.2081i
w40	0.1103+0.9397i
w41	0.2415+0.9155i
w42	0.1118+1.1163i
w43	0.3079+1.0866i
w44	0.5647+0.7638i
w45	0.4385+0.8433i
w46	0.6846+0.8841i
w47	0.5165+1.0034i
w48	1.6489+0.1630i
w49	1.5848+0.4983i
w50	1.3437+0.1389i
w51	1.2850+0.4025i
w52	1.2728+1.0661i
w53	1.4509+0.7925i
w54	1.0249+0.8794i
w55	1.1758+0.6545i
w56	0.9629+0.1113i
w57	0.9226+0.2849i
w58	1.1062+0.1118i
w59	1.0674+0.3393i
w60	0.7234+0.6223i
w61	0.8211+0.4860i
w62	0.8457+0.7260i
w63	0.9640+0.5518i

[0351] Meanwhile, according to Tables 5-7 and 7-1, when values of constellation points are determined in one quadrant, values of constellation points in other quadrants may be deduced by symmetry. For example, for each constellation point A in the top-right quadrant, corresponding constellation points may be present in three different quadrants (bottom-right, bottom-left and top-left) respectively, and they can be indicated as A*, −A*, and −A. Here, * indicates complex conjugation.

[0352] Table 8 indicates values of constellation points of a normalized 1D NU 1024-QAM constellation (1D 1024NUC) which is obtained by applying the algorithms described above using respective coding rates 13/15 for a single SNR value.

TABLE 8

u/CR	Coding Rate 13/15
u0	0.0325
u1	0.0967
u2	0.1623
u3	0.2280
u4	0.2957
u5	0.3645
u6	0.4361
u7	0.5100
u8	0.5878
u9	0.6696
u10	0.7566
u11	0.8497
u12	0.9498
u13	1.0588
u14	1.1795
u15	1.3184

[0353] In the case of the 1D NU 1K QAM constellation, rather than giving the values of the constellation points explicitly, a set of levels of the constellation points are given instead, from which actual values of the constellation points may be deduced. To be specific, given a set of m levels $A=[u_1, u_2, \ldots, u_m]$, a set of m^2 constellation point values C+Di may be deduced. Herein, C and D each may include a value selected from a level set u. A complete set of constellation points in the top-right quadrant may be obtained by considering all possible pairs of values C and D. The values of constellation points in the remaining three quadrants may be similarly deduced by symmetry. As an example, according to Table 8, when the coding rate is 2/15, A={0.0325, 0.0967, . . . , 1.3184}, and a group of C+Di corresponding to a constellation point set of the first quadrant has 256 elements such as {0.0325+0.0325i, 0.0325+0.0967i, 0.0967+0.0325i, . . . , 1.3184+1.3184i}, the complete set of 1D NU 1024-QAM constellation points may be obtained by indicating an arbitrary element a in the group, as a*, −a* and −a. Here, * indicates complex conjugation.

[0354] As described above, in the 1D-NUC of Table 8, the constellation can be described by the levels at which the constellations occur in the real positive part. The constellation points can be deduced by using the real/imaginary symmetry but also the symmetry of the four quadrants.

[0355] Meanwhile, the inventive concept is not limited to the constellations defined in Tables 5-8.

[0356] For example, when rounding-off is applied for the values of constellation points defined in Tables 5-6, the values can be indicated as in Tables 9-10. In this case, the constellation defined in Tables 5-6 can be an exemplary embodiment.

[0357] Tables 9-10 illustrate the set of constellation points for one quadrant only, but it is obvious to obtain a complete set of constellation points by indicating the constellation

point a in the one quadrant, as a*, −a*, and −a. Here, * indicates complex conjugation.

[0358] To be specific, Table 9 indicates the values of constellation points of 2D NU 16-QAM constellation (2D 16NUC) which is obtained by applying rounding-off of the values of constellation points defined in Table 5.

TABLE 9

w/Shape	NUC_16_11/15	NUC_16_13/15
w0	0.9342+0.9847i	0.9517+0.9511i
w1	0.9866+0.2903i	0.9524+0.3061i
w2	0.2716+0.9325i	0.3067+0.9524i
w3	0.2901+0.2695i	0.3061+0.3067i

[0359] In this case, the values of constellation points of other quadrants can be determined by symmetry.

[0360] Table 10 indicates values of constellation points of a 2D NU 64-QAM constellation (2D 64NUC) which is obtained by applying rounding-off of the values of constellation points defined in Table 6.

TABLE 10

w/Shape	NUC_64_5/15	NUC_64_11/15
w0	1.4327+0.3305i	1.4443+0.2683i
w1	1.0909+0.2971i	0.7471+1.2243i
w2	1.2484+0.7803i	1.1749+0.7734i
w3	0.9762+0.5715i	0.7138+0.8201i
w4	0.3309+1.4326i	0.1638+1.0769i
w5	0.2979+1.0923i	0.2927+1.4217i
w6	0.7829+1.2477i	0.1462+0.7457i
w7	0.5739+0.9763i	0.4134+0.7408i
w8	0.3901+0.2112i	1.0203+0.1517i
w9	0.5317+0.2475i	0.6653+0.1357i
w10	0.3945+0.2289i	0.9639+0.4465i
w11	0.5236+0.2894i	0.6746+0.4339i
w12	0.2108+0.3911i	0.1271+0.1428i
w13	0.2475+0.5327i	0.3782+0.1406i
w14	0.2287+0.3955i	0.1311+0.4288i
w15	0.2898+0.5246i	0.3919+0.4276i

[0361] In this case, the values of constellation points of other quadrants can be determined by symmetry.

[0362] Meanwhile, those skilled in the art may recognize that rotation, scaling (here, the scaling factor applied to a real axis and an imaginary axis can be the same or different) or other transformation can be applied with respect to the constellation described above. The constellation indicates a comparative position of constellation points, and other constellation can be deduced through rotation, scaling, or other transformation.

[0363] In addition, those skilled in the art can recognize that the inventive concept is not limited to constellation defined in Tables 5-10 described above.

[0364] For example, in certain exemplary embodiments, a constellation having different order and/or a constellation including a different arrangement or a comparative position of constellation points can be used. As another example, a constellation which is similar to one of constellations defined in Tables 5-10 can be used.

[0365] For example, a constellation which has values of constellation points with differences which do not exceed a predetermined threshold (or error) from the values indicated in Tables 5-10 can be used. Here, the threshold value can be expressed as comparative numbers (for example, 0.1%, 1%, 5%, etc.), absolute numbers (for example, 0.001, 0.01, 0.1, etc.) or appropriate methods (rounding-off, flooring, ceiling, or the like). As an example of rounding-off, constellation point "0.707316+0.707473i" can be approximated to "0.7073+0.7075i" by rounding-off at the five decimal places.

[0366] In addition, a transmitter and a receiver may use different constellations. For example, a transmitter and a receiver may use respective constellations which have at least one constellation point that has a difference which does not exceed a predetermined threshold value. For example, a receiver may use a constellation having at least one round off/round down constellation point (for example, A2) to demap constellation points, whereas a transmitter may use a constellation having non-round off/round-down constellation points (for example, A1).

[0367] In addition, even if an order of the values in Tables 5-10 is changed, the set of constellation points itself is not changed, and thus, it is possible to arrange the values by changing the order of values as in Tables 5-10.

[0368] As described above, rotation, scaling, and other transformation can be applied to the constellations defined by the various exemplary embodiments

[0369] To be specific, the constellation point of which the absolute values of the constellation points defined in tables 5-7 and 9-10 are not changed and indicated as constellation points which are converted to 1:1 can be considered the equivalent constellation as the constellation defined in tables 5-7 and 9-10. Here, conversion to 1:1 in dictates that inverse conversion is available.

[0370] For example, in a constellation, the below is applied to the constellation points defined in tables 5-7 and 9-10.

[0371] i) rotation->conjugation->rotation or

[0372] ii) rotation->conjugation->multiplying a constant with absolute value of 1

[0373] The above constellation where the above is applied is considered to be the same as the constellation defined in tables 5-7 and 9-10.

[0374] As an example the constellation points (B+Ai)=(0.9342+0.9847i, 0.9866+0.2903i, 0.2716+0.9325i, 0.2901+0.2695i) which are obtained by rotating the constellation points defined in table 9 (A+Bi)=(0.9847+0.9342i, 0.2903+0.9866i, 0.9325+0.2716i,0.2695+0.2901i) in a counter-clockwise direction by 90°->conjugation->counterclowise direction by 180° (or, multiplying−1) can be considered the same constellation as the constellation defined in table 9. Meanwhile, in the aforementioned example, table 9 is explained as an example, but the constellation points A+Bi defined in tables 5-7 and 9-10 and the constellation points B+Ai which are converted from A+Bi are considered the same as the constellation defined in tables 5-7 and 9-10.

[0375] Hereinbelow, an example of a normalization method and an exemplary embodiment of constituting 2D constellation from a 1D level set will be described.

[0376] For example, in Table 11, it is assumed that values of constellation points of a 1D NU 1K QAM constellation for a 13/15 coding rate are as shown below.

TABLE 11

Level	Coding Rate 13/15
1	1
2	2.975413
3	4.997551
4	7.018692
5	9.102872
6	11.22209
7	13.42392
8	15.69921
9	18.09371
10	20.61366
11	23.2898
12	26.15568
13	29.23992
14	32.59361
15	36.30895
16	40.58404

[0377] Here, when a level vector A is indicated as $A=(a_i)$, (I=0, 1, 2, . . . , L−1), first of all, the vector A is normalized using Equation 3 shown below, and normalized vector $\overline{A}$ can be obtained.

$$\overline{A} = \frac{A}{\sqrt{\frac{2}{L}\Sigma_i a_i^2}} \quad \text{[Equation 3]}$$

[0378] In above Equation 3, L indicates the number of level (that is, dimensionality of A) For example, in the case of 16-QAM, 64-QAM, 256-QAM, 1024-QAM, and 4096-QAM, dimensionality of level can be 4, 6, 8, 10 and 12 respectively.

[0379] In the example described above, the normalized vector $\overline{A}$ can be indicated as Table 12 shown below.

TABLE 12

Level	Coding Rate 13/15
1	0.0325
2	0.0967
3	0.1623
4	0.228
5	0.2957
6	0.3645
7	0.4361
8	0.51
9	0.5878
10	0.6696
11	0.7566
12	0.8497
13	0.9498
14	1.0588
15	1.1795
16	1.3184

[0380] Next, a final constellation is generated such that all the possible combinations of the real part and the imaginary part, which are the same as one of the entries (that is, components). In this case, for an example, gray mapping can be used.

[0381] In the example described above, constellation points in the final first quadrant can be indicated as in Table 13 shown below.

TABLE 13

Label (int.)	Constellation Point
1	1.3184 + 1.3184i
2	1.3184 + 1.1795i
3	1.1795 + 1.3184i
4	1.1795 + 1.1795i
5	1.3184 + 0.9498i
6	1.3184 + 1.0588i
7	1.1795 + 0.9498i
8	1.1795 + 1.0588i
9	0.9498 + 1.3184i
10	0.9498 + 1.1795i
11	1.0588 + 1.3184i
12	1.0588 + 1.1795i
13	0.9498 + 0.9498i
14	0.9498 + 1.0588i
15	1.0588 + 0.9498i
16	1.0588 + 1.0588i
17	1.3184 + 0.5878i
18	1.3184 + 0.6696i
19	1.1795 + 0.5878i
20	1.1795 + 0.6696i
21	1.3184 + 0.8497i
22	1.3184 + 0.7566i
23	1.1795 + 0.8497i
24	1.1795 + 0.7566i
25	0.9498 + 0.5878i
26	0.9498 + 0.6696i
27	1.0588 + 0.5878i
28	1.0588 + 0.6696i
29	0.9498 + 0.8497i
30	0.9498 + 0.7566i
31	1.0588 + 0.8497i
32	1.0588 + 0.7566i
33	0.5878 + 1.1795i
34	0.6696 + 1.3184i
35	0.6696 + 1.1795i
36	0.5878 + 0.9498i
37	0.5878 + 1.0588i
38	0.6696 + 0.9498i
39	0.6696 + 1.0588i
40	0.8497 + 1.3184i
41	0.8497 + 1.1795i
42	0.7566 + 1.3184i
43	0.7566 + 1.1795i
44	0.8497 + 0.9498i
45	0.8497 + 1.0588i
46	0.7566 + 0.9498i
47	0.7566 + 1.0588i
48	0.5878 + 0.5878i
49	0.5878 + 0.6696i
50	0.6696 + 0.5878i
51	0.6696 + 0.6696i
52	0.5878 + 0.8497i
53	0.5878 + 0.7566i
54	0.6696 + 0.8497i
55	0.6696 + 0.7566i
56	0.8497 + 0.5878i
57	0.8497 + 0.6696i
58	0.7566 + 0.5878i
59	0.7566 + 0.6696i
60	0.8497 + 0.8497i
61	0.8497 + 0.7566i
62	0.7566 + 0.8497i
63	0.7566 + 0.7566i
64	1.3184 + 0.0325i
65	1.3184 + 0.0967i
66	1.1795 + 0.0325i
67	1.1795 + 0.0967i
68	1.3184 + 0.2280i
69	1.3184 + 0.1623i
70	1.1795 + 0.2280i
71	1.1795 + 0.1623i
72	0.9498 + 0.0325i
73	0.9498 + 0.0967i
74	1.0588 + 0.0325i

TABLE 13-continued

Label (int.)	Constellation Point
75	1.0588 + 0.0967i
76	0.9498 + 0.2280i
77	0.9498 + 0.1623i
78	1.0588 + 0.2280i
79	1.0588 + 0.1623i
80	1.3184 + 0.5100i
81	1.3184 + 0.4361i
82	1.1795 + 0.5100i
83	1.1795 + 0.4361i
84	1.3184 + 0.2957i
85	1.3184 + 0.3645i
86	1.1795 + 0.2957i
87	1.1795 + 0.3645i
88	0.9498 + 0.5100i
89	0.9498 + 0.4361i
90	1.0588 + 0.5100i
91	1.0588 + 0.4361i
92	0.9498 + 0.2957i
93	0.9498 + 0.3645i
94	1.0588 + 0.2957i
95	1.0588 + 0.3645i
96	0.5878 + 0.0325i
97	0.5878 + 0.0967i
98	0.6696 + 0.0325i
99	0.6696 + 0.0967i
100	0.5878 + 0.2280i
101	0.5878 + 0.1623i
102	0.6696 + 0.2280i
103	0.6696 + 0.1623i
104	0.8497 + 0.0325i
105	0.8497 + 0.0967i
106	0.7566 + 0.0325i
107	0.7566 + 0.0967i
108	0.8497 + 0.2280i
109	0.8497 + 0.1623i
110	0.7566 + 0.2280i
111	0.7566 + 0.1623i
112	0.5878 + 0.5100i
113	0.5878 + 0.4361i
114	0.6696 + 0.5100i
115	0.6696 + 0.4361i
116	0.5878 + 0.2957i
117	0.5878 + 0.3645i
118	0.6696 + 0.2957i
119	0.6696 + 0.3645i
120	0.8497 + 0.5100i
121	0.8497 + 0.4361i
122	0.7566 + 0.5100i
123	0.7566 + 0.4361i
124	0.8497 + 0.2957i
125	0.8497 + 0.3645i
126	0.7566 + 0.2957i
127	0.7566 + 0.3645i
128	0.0325 + 1.3184i
129	0.0325 + 1.1795i
130	0.0967 + 1.3184i
131	0.0967 + 1.1795i
132	0.0325 + 0.9498i
133	0.0325 + 1.0588i
134	0.0967 + 0.9498i
135	0.0967 + 1.0588i
136	0.2280 + 1.3184i
137	0.2280 + 1.1795i
138	0.1623 + 1.3184i
139	0.1623 + 1.1795i
140	0.2280 + 0.9498i
141	0.2280 + 1.0588i
142	0.1623 + 0.9498i
143	0.1623 + 1.0588i
144	0.0325 + 0.5878i
145	0.0325 + 0.6696i
146	0.0967 + 0.5878i
147	0.0967 + 0.6696i
148	0.0325 + 0.8497i

TABLE 13-continued

Label (int.)	Constellation Point
149	0.0325 + 0.7566i
150	0.0967 + 0.8497i
151	0.0967 + 0.7566i
152	0.2280 + 0.5878i
153	0.2280 + 0.6696i
154	0.1623 + 0.5878i
155	0.1623 + 0.6696i
156	0.2280 + 0.8497i
157	0.2280 + 0.7566i
158	0.1623 + 0.8497i
159	0.1623 + 0.7566i
160	0.5100 + 1.3184i
161	0.5100 + 1.1795i
162	0.4361 + 1.3184i
163	0.4361 + 1.1795i
164	0.5100 + 0.9498i
165	0.5100 + 1.0588i
166	0.4361 + 0.9498i
167	0.4361 + 1.0588i
168	0.2957 + 1.3184i
169	0.2957 + 1.1795i
170	0.3645 + 1.3184i
171	0.3645 + 1.1795i
172	0.2957 + 0.9498i
173	0.2957 + 1.0588i
174	0.3645 + 0.9498i
175	0.3645 + 1.0588i
176	0.5100 + 0.5878i
177	0.5100 + 0.6696i
178	0.4361 + 0.5878i
179	0.4361 + 0.6696i
180	0.5100 + 0.8497i
181	0.5100 + 0.7566i
182	0.4361 + 0.8497i
183	0.4361 + 0.7566i
184	0.2957 + 0.5878i
185	0.2957 + 0.6696i
186	0.3645 + 0.5878i
187	0.3645 + 0.6696i
188	0.2957 + 0.8497i
189	0.2957 + 0.7566i
190	0.3645 + 0.8497i
191	0.3645 + 0.7566i
192	0.0325 + 0.0325i
193	0.0325 + 0.0967i
194	0.0967 + 0.0325i
195	0.0967 + 0.0967i
196	0.0325 + 0.2280i
197	0.0325 + 0.1623i
198	0.0967 + 0.2280i
199	0.0967 + 0.1623i
200	0.2280 + 0.0325i
201	0.2280 + 0.0967i
202	0.1623 + 0.0325i
203	0.1623 + 0.0967i
204	0.2280 + 0.2280i
205	0.2280 + 0.1623i
206	0.1623 + 0.2280i
207	0.1623 + 0.1623i
208	0.0325 + 0.5100i
209	0.0325 + 0.4361i
210	0.0967 + 0.5100i
211	0.0967 + 0.4361i
212	0.0325 + 0.2957i
213	0.0325 + 0.3645i
214	0.0967 + 0.2957i
215	0.0967 + 0.3645i
216	0.2280 + 0.5100i
217	0.2280 + 0.4361i
218	0.1623 + 0.5100i
219	0.1623 + 0.4361i
220	0.2280 + 0.2957i
221	0.2280 + 0.3645i
222	0.1623 + 0.2957i

TABLE 13-continued

Label (int.)	Constellation Point
223	0.1623 + 0.3645i
224	0.5100 + 0.0325i
225	0.5100 + 0.0967i
226	0.4361 + 0.0325i
227	0.4361 + 0.0967i
228	0.5100 + 0.2280i
229	0.5100 + 0.1623i
230	0.4361 + 0.2280i
231	0.4361 + 0.1623i
232	0.2957 + 0.0325i
233	0.2957 + 0.0967i
234	0.3645 + 0.0325i
235	0.3645 + 0.0967i
236	0.2957 + 0.2280i
237	0.2957 + 0.1623i
238	0.3645 + 0.2280i
239	0.3645 + 0.1623i
240	0.5100 + 0.5100i
241	0.5100 + 0.4361i
242	0.4361 + 0.5100i
243	0.4361 + 0.4361i
244	0.5100 + 0.2957i
245	0.5100 + 0.3645i
246	0.4361 + 0.2957i
247	0.4361 + 0.3645i
248	0.2957 + 0.5100i
249	0.2957 + 0.4361i
250	0.3645 + 0.5100i
251	0.3645 + 0.4361i
252	0.2957 + 0.2957i
253	0.2957 + 0.3645i
254	0.3645 + 0.2957i
255	0.3645 + 0.3645i
256	1.3184 − 1.3184i

[0382] FIG. **41** is a block diagram to describe a configuration of a transmitting apparatus according to an exemplary embodiment. Referring to FIG. **41**, the transmitting apparatus **5900** includes an encoder **5910**, an interleaver **5920**, and a modulator **5930** (or, 'mapper', 'constellation mapper').

[0383] The encoder **5910** performs channel encoding with respect to input bits and generates a codeword.

[0384] For example, the encoder **5910** may perform LDPC encoding with respect to the bits and generate an LDPC codeword using an LDPC encoder (not shown).

[0385] Specifically, the encoder **5910** may perform LDPC encoding with the input bits as the information word bits, and generate the LDPC codeword constituting the information word bits and parity bits (that is, the LDPC parity bits). In this case, the LDPC code is a systematic code, the information word may be included in the LDPC codeword as it is.

[0386] Herein, the LDPC codeword is constituted by the information word bits and parity bits. For example, the LDPC codeword has N_{ldpc} bits, and may include the information word bits formed of K_{ldpc} bits and parity bits formed of $N_{parity}=N_{ldpc}-K_{ldpc}$ parity bits.

[0387] In this case, the encoder **5910** may perform LDPC encoding based on a parity check matrix and generate the LDPC codeword. That is, a process of performing LDPC encoding is a process of generating the LDPC codeword satisfying $H \cdot C^T=0$, and thus, the encoder **5910** may use the parity check matrix when performing LDPC encoding. Herein, H is a parity check matrix and C is an LDPC codeword.

[0388] To do this, the transmitting apparatus **5900** may include a separate memory and prestore various types of a parity check matrix.

[0389] However, this is merely exemplary, and channel encoding may be performed in various schemes.

[0390] The encoder **5910** may perform channel encoding using various coding rates such as 2/15, 3/15, 4/15, 5/15, 6/15, 7/15, 8/15, 9/15, 10/15, 11/15, 12/15 and 13/15. In addition, the encoder **5910** may generate a codeword having various lengths such as 16200 and 64800 based on a length of the bits and coding rate.

[0391] An interleaver **5920** interleaves the codeword. That is, the interleaver **5920**, based on various interleaving rules, may perform bit-interleaving of the codeword generated by the encoder **5910**. In this case, the interleaver **5920** may include the parity interleaver **14210**, group-wise interleaver **14229**, and block interleaver **14230** of FIG. **7**

[0392] A modulator **5930** maps the codeword which is interleaved according to a modulation scheme onto a non-uniform constellation.

[0393] Specifically, the modulator **5930** may perform serial-to-parallel conversion with respect to the interleaved codeword, and demultiplex the interleaved codeword into a cell (or a cell word) formed of a certain number of bits.

[0394] For example, the modulator **5930** may receive the codeword bits $Q=(q_0, q_1, q_2, \ldots)$ output from the interleaver **5920**, and generates cells.

[0395] In this case, the number of bits constituting each cell may be the same as the number of bits constituting a modulation symbol (that is, a modulation order). For example, when the modulator **5930** performs modulation using QPSK, 16-QAM, 64-QAM, 256-QAM, 1024-QAM, 4096-QAM, the number of bits η_{MOD} constituting the modulation symbol may be 2, 4, 6, 8, 10 and 12.

[0396] For example, when the modulation scheme is 64-QAM, η_{MOD} is 6 ($\eta_{MOD}=6$), and thus, each cell may be composed as $(q_0, q_1, q_2, q_3, q_4, q_5)$, $(q_6, q_7, q_8, q_9, q_{10}, q_{11})$, $(q_{12}, q_{13}, q_{14}, q_{15}, q_{16}, q_{17})$, ...

[0397] In addition, the modulator **5930** may perform modulation by mapping the cells onto the non-uniform constellation.

[0398] Specifically, each cell includes bits as many as the number constituting the modulation symbol, and thus, the modulator **5930** may generate the modulation symbol by sequentially mapping each cell onto a constellation point of the non-uniform constellation. Herein, the modulation symbol corresponds to a constellation point of a constellation.

[0399] In this case, constellation may include constellation points which are defined based on Tables 5-10 according to a modulation scheme.

[0400] To be specific, the constellation may include the constellation points which are defined by a constellation position vector as in Tables 5-7 and 9-10 according to a modulation scheme. Or, the constellation may include the constellation points which are defined by the constellation position vector which is generated based on the level set as in Table 8 according to a modulation scheme.

[0401] That is, the modulator **5930**, in consideration of the coding rate used for encoding by the encoder **5910**, may perform modulation by mapping cells onto the set of constellation points which corresponds to the coding rate from among the sets of constellation points which are defined based on Tables 5-10 according to the coding rates.

[0402] For example, constellation may include constellation points which are defined based on Table 11, when a modulation scheme is 16-QAM.

[0403] To be specific, the modulator **5930**, when encoding is performed with the coding rate of 11/15 by the encoder **5910**, may map the interleaved codeword onto the non-uniform constellation which includes constellation points defined by NUC_16_11/15 of Table 9.

[0404] That is, when the coding rate is 11/15 and modulation is performed to 2D 16NUC, the constellation points in the first quadrant of constellation can be expressed as the constellation position vector $\{w_0,w_1,w_2,w_3\}$={0.9342+0.9847i, 0.9866+0.2903i, 0.2716+0.9325i, 0.2901+0.2695i} which is defined as NUC_16_11/15 of Table 9.

[0405] In addition, the modulator **5930**, when encoding is performed with the coding rate of 13/15 by the encoder **5910**, may map the interleaved codeword onto the non-uniform constellation which includes the constellation points defined by NUC_16_13/15 of Table 9.

[0406] That is, when the coding rate is 13/15 and modulation is performed to 2D 16NUC, the constellation points in the first quadrant of constellation can be expressed as the constellation position vector $\{w_0,w_1,w_2,w_3\}$={0.9517+0.9511i, 0.9524+0.3061i, 0.3067+0.9524i, 0.3061+0.3067i} which is defined as NUC_16_13/15 of Table 9.

[0407] Table 9 indicates the constellation points in one quadrant of constellation, and the constellation points in remaining quadrants of constellation may be obtained by indicating each constellation point a, which is defined in Table 9, as a*, −a*, and −a respectively. (Here, * indicates complex conjugation).

[0408] As another example, constellation may include constellation points which are defined based on Table 10, when a modulation scheme is 64-QAM.

[0409] To be specific, the modulator **5930**, when encoding is performed with the coding rate of 5/15 by the encoder **5910**, may map the interleaved codeword onto the non-uniform constellation which includes constellation points defined by NUC_64_5/15 of Table 10.

[0410] That is, when the coding rate is 5/15 and modulation is performed to 2D 64NUC, the constellation points in the first quadrant of constellation can be expressed as the constellation position vector $\{w_0, w_1, w_2, \ldots, w_{14}, w_{15}\}$={1.4327+0.3305i, 1.0909+0.2971i, 1.2484+0.7803i, . . . , 0.2287+0.3955i, 0.2898+0.5246i} which is defined as NUC_64_5/15 of Table 10.

[0411] In addition, the modulator **5930**, when encoding is performed with the coding rate of 11/15 by the encoder **5910**, may map the interleaved codeword onto the non-uniform constellation which includes the constellation points defined by NUC_64_11/15 of Table 10.

[0412] That is, when the coding rate is 11/14 and modulation is performed to 2D 64NUC, the constellation points in the first quadrant of constellation can be expressed as the constellation position vector $\{w_0, w_1, w_2, \ldots, w_{14}, w_{15}\}$={1.4443+0.2683i, 0.7471+1.2243i, 1.1749+0.7734i, . . . , 0.1311+0.4288i, 0.3919+0.4276i} which is defined as NUC_64_11/15 of Table 10.

[0413] Table 10 indicates the constellation points in one quadrant of constellation, and the constellation points in remaining quadrants of constellation may be obtained by indicating each constellation point a, which is defined in Table 10, as a*, −a*, and −a respectively. (Here, * indicates complex conjugation).

[0414] As another example, when the modulation scheme is 256-QAM, the constellation points which are defined based on Table 12 may be included.

[0415] Specifically, the modulator **5930**, when encoding is performed with the coding rate of 9/15 by the encoder **5910**, may map the interleaved codeword onto the non-uniform constellation which includes the constellation points defined by NUC_256_9/15 of Table 7.

[0416] That is, when coding rate is 9/15 and modulation is performed to 2D 256NUC, the constellation points of the first quadrant of constellation may be expressed as the constellation position vector $\{w_0, w_1, w_2, \ldots, w_{62}, w_{63}\}$ {0.0899+0.1337i, 0.0910+0.1377i, 0.0873+0.3862i, . . . , 0.8457+0.7260i, 0.9640+0.5518i} which is defined as NUC_256_9/15 of Table 7.

[0417] As another example, constellation, when the modulation scheme is 1024-QAM, may include the constellation points which are defined based on Table 8.

[0418] Specifically, the modulator **5930**, when encoding is performed with the coding rate of 13/15 by the encoder **5910**, may map the interleaved codeword onto the non-uniform constellation which includes the constellation points defined by NUC_1k_13/15 of Table 130.

[0419] That is, when coding rate 13 s 2/15 and modulation is performed to 1D 1024NUC, the level set may be A={0.0325, 0.0967, 0.1623, . . . , 1.1795, 1.3184} as Table 8, and the constellation position vector indicating the constellation points in the first quadrant may be expressed as {0.0325+0.0325i, 0.0325+0.0967i, 0.0967+0.0325i, . . . , 1.3184+1.3184i}.

[0420] Table 8 is used to define the constellation points in one quadrant of constellation, and the constellation points in remaining quadrants may be obtained by indicating each constellation point, which is defined based on Table 8, as a*, −a*, and −a (Here, * indicates complex conjugation).

[0421] In the above-described examples, it is described that the cells are mapped onto the set of constellation points which correspond to coding rate used for encoding, but this is merely exemplary, and in some cases, the modulator **5930** may map the cells onto the set of constellation points which do not correspond to coding rate which is used for encoding.

[0422] As an example, when 64-QAM is used, even if encoding is performed with the coding rate of 5/15, the modulator **5930** may map the cells onto the set of constellation points which are defined as NUC_64_11/15 of Table 6, instead of the set of constellation points which are defined as NUC_64_5/15 of Table 6.

[0423] The transmitting apparatus **5900** may modulate a signal which is mapped onto the constellation and transmit the signal to a receiving apparatus (for example, 6000 of FIG. **42**). For example the transmitting apparatus **5900** may map the signal which is mapped to the constellation onto an orthogonal frequency division multiplexing (OFDM) frame by using an OFDM scheme, and may transmit the signal to the receiving apparatus **6000** via an allocated channel.

[0424] In other word, in mapping method for 16-QAM, 64-QAM and 256-QAM, each input data cell word $(y_{0,s}, \ldots, y_{\eta MOD-1,s})$ shall be modulated using a 2D non-uniform constellation to give a constellation point z_s. Index s denotes the discrete time index, $\eta_{MOD}=\log_2(M)$, M being the number of constellation points, e.g., M=64 for 64-QAM. The vector of complex constellation points $x=(x_0, \ldots, x_{M-1})$ includes all M constellation points of the QAM alphabet. The k-th element of this vector, x_k, corresponds to the QAM constel-

lation point for the input cell word ($y_{0,s}$, . . . , $y_{\eta MOD-1,s}$), if these bits take on the decimal number k ($y_{0,s}$ being the most significant bit (MSB), and $y_{\eta MOD-1,s}$ being the least significant bit (LSB)). Due to the quadrant symmetry, the complete vector x can be derived by defining just the first quarter of the complex constellation points, i.e., (x_0, . . . , $x_{M/4-1}$), which corresponds to the first quadrant. The generation rule for the remaining points is described below. Defining b=M/4, the first quarter of complex constellation points is denoted as the NUC position vector w=(w_0, . . . , w_{b-1}). The position vectors are defined the above tables. As an example, the NUC position vector for a 16-QAM comprises the complex constellation points with the labels corresponding to the decimal values 0, i.e., ($y_{0,s}$, . . . , $y_{\eta MOD-1,s}$)=0000, to b−1, i.e., ($y_{0,s}$, . . . , $y_{\eta MOD-1,s}$)=0011. The remaining constellation points are derived as follows:

(x_0, . . . , x_{b-1})=w (first quarter)

(x_b, . . . , x_{2b-1})=−conj(w) (second quarter)

(x_{2b}, . . . , x_{3b-1})=conj(w) (third quarter)

(x_{3b}, . . . , x_{4b-1})=−w (fourth quarter), with conj being the complex conjugate.

[0425] As an example, the NUC position vector for 16-QAM and code rate 2/15 is constructed as follows. From Table 10, w=(0.7073+0.7075i, 0.7073+0.7074i, 0.7060+0.7077i, 0.7065+0.7071i). Here and in the following, i=√(−1) is the imaginary unit. Assuming the input data cell word is ($y_{0,s}$, . . . , $y_{\eta MOD-1,s}$)=(1100), the corresponding QAM constellation point at time index s is $z_s=x_{12}=-w_0=-0.7073-0.7075i$.

[0426] Also, in mapping method for 1024-QAM and 4096-QAM, Each input data cell word ($y_{0,s}$, . . . , $y_{\eta MOD-1,s}$) at discrete time index s shall be modulated using a 1-dimensional non-uniform QAM constellation to give a constellation point z_s prior to normalization. 1-dimensional refers to the fact that a 2-dimensional QAM constellation can be separated into two 1-dimensional PAM constellations, one for each I and Q component. The exact values of the real and imaginary components Re(z_s) and Im(z_s) for each combination of the relevant input cell word ($y_{0,s}$, . . . , $y_{\eta MOD-1,s}$) are given by a 1D-NUC position vector u=(u_0, . . . , u_v), which defines the constellation point positions of the non-uniform constellation in one dimension. The number of elements of the 1D-NUC position vector u is defined by

$$v = \frac{\sqrt{M}}{2}.$$

[0427] As an example the 1024-NUC for code rate 2/15 is defined by the NUC position vector NUC_1k_2/15. From Table 13, u=(u_0, . . . , u_{15})=(0.3317, 0.3321, 0.3322, 0.3321, 0.3327, 0.3328, 0.3322, 0.3322, 0.9369, 0.9418, 0.9514, 0.9471, 0.9448, 0.9492, 0.9394, 0.9349). Assuming the input data cell ($y_{0,s}$, . . . , $y_{\eta MOD-1,s}$)=(0010011100) the corresponding QAM constellation point z_s has Re(z_s)=u_3=0.3321 (defined by even index bit labels, i.e., 01010) and Im(z_s)=u_{11}=0.9471 (defined by odd index bit label, i.e., 00110).

[0428] FIG. **42** is a block diagram to describe a configuration of the receiving apparatus according to an exemplary embodiment. Referring to FIG. **42**, the receiving apparatus **6000** includes a demodulator **6010**, a deinterleaver **6020**, and a decoder **6030**.

[0429] The demodulator **6010** receives and demodulates a signal transmitted from the transmitting apparatus **5900**. Specifically, the demodulator **6010** may generate a value corresponding to the codeword by demodulating the received signal.

[0430] In this case, the demodulator **6010** may perform demodulation to correspond to the modulation scheme which is used by the transmitting apparatus **5900**. To do this, the transmitting apparatus **5900** may transmit information on the modulation scheme to the receiving apparatus **6000**, or the transmitting apparatus **5900** may perform modulation using the modulation scheme which is predefined between the transmitting apparatus **5900** and the receiving apparatus **6000**.

[0431] Meanwhile, a value which corresponds to the codeword may be expressed as a channel value with respect to the received signal. There may be various methods for determining the channel value, for example, a method for determining a log likelihood ratio (LLR) value is an example of the method for determining the channel value.

[0432] The LLR value may indicate a log value for a ratio of the probability that the bit transmitted from the transmitting apparatus **3500** is 0 and the probability that the bit is 1. In addition, the LLR value may be a bit value which is determined by a hard decision, or may be a representative value which is determined according to a section to which the probability that the bit transmitted from the transmitting apparatus **3500** is 0 or 1 belongs.

[0433] The demodulator **6010** may perform cell-to-bit conversion with respect to a value corresponding to the codeword and output an LLR value in the unit of bits.

[0434] The deinterleaver **6020** deinterleaves an output value of the demodulator **6010**, and outputs the value to the decoder **6030**.

[0435] To be specific, the deinterleaver **6020** is an element corresponding to the interleaver **5920** of the transmitting apparatus **5900** and performs an operation corresponding to the interleaver **5920**. That is, the deinterleaver **6020** performs the interleaving operation of the interleaver **5920** inversely and deinterleaves an LLR value.

[0436] The decoder **6030** may perform channel decoding based on the output value of the deinterleaver **6020**.

[0437] Specifically, the decoder **6030** is an element corresponding to the encoder **5910** of the transmitting apparatus **5900**, which may correct an error by performing decoding by using the LLR value output from the deinterleaver **6020**.

[0438] For example, the decoder **6030** may include an LDPC decoder (not shown) to perform LDPC decoding.

[0439] In this case, the decoder **6030** may perform LDPC decoding using an iterative decoding scheme based on a sum-product algorithm. Herein, the sum-product algorithm refers to an algorithm by which messages (e.g., LLR value) are exchanged through an edge on a bipartite graph of a message passing algorithm, and an output message is calculated from messages input to variable nodes or check nodes, and is updated.

[0440] Meanwhile, the decoder **6030** may use a parity check matrix for LDPC decoding.

[0441] In this case, the parity check matrix which is used for decoding may have the same structure as the parity check matrix which is used for encoding.

[0442] Meanwhile, information on the parity check matrix or information on the code rate used for LDPC decoding

may be prestored in the receiving apparatus **6000** or provided by the transmitting apparatus **5900**.

[0443] The foregoing is merely exemplary, and channel decoding may be performed by various schemes which correspond to the channel coding which is performed by the transmitting apparatus **3500**.

[0444] FIG. **43** is a flowchart to describe a method for transmitting of a transmitting apparatus according to an exemplary embodiment.

[0445] First of all, a codeword is generated (S**6110**) by performing channel encoding with respect to the bits, and the codeword is interleaved (S**6120**).

[0446] Thereafter, the interleaved codeword is mapped onto the non-uniform constellation according to a modulation scheme (S**6130**).

[0447] In this case, constellation may include the constellation points which are defined based on Tables 5-10 according to a modulation scheme.

[0448] As an example, when, the modulation scheme is 16-QAM, constellation may include the constellation points which are defined based on Table 9.

[0449] Specifically, when encoding is performed with the coding rate of 11/15, at S**6130**, an interleaved codeword may be mapped onto the non-uniform constellation which includes the constellation points defined by NUC_16_11/15 of Table 9.

[0450] When encoding is performed with the coding rate of 13/15, at S**6130**, the interleaved codeword may be mapped onto the non-uniform constellation which includes the constellation points which are defined by NUC_16_13/15 of Table 9.

[0451] As another example, when the modulation scheme is 64-QAM, constellation may include the constellation points which are defined based on Table 10.

[0452] Specifically, when encoding is performed with the coding rate of 5/15, at S**6130**, an interleaved codeword may be mapped onto the non-uniform constellation which includes the constellation points defined by NUC_64_5/15 of Table 10.

[0453] When encoding is performed with the coding rate of 11/15, at S**6130**, the interleaved codeword may be mapped onto the non-uniform constellation which includes the constellation points which are defined by NUC_64_11/15 of Table 10.

[0454] Meanwhile, Tables 9 and 10 indicate the constellation points in one quadrant of the constellation, and the constellation points in the remaining quadrants of constellation may be obtained by indicating each constellation point a, which is defined in Tables 11 and 12, as a*, −a*, and −a respectively (Here, * indicates complex conjugation).

[0455] As another example, when the modulation scheme is 256-QAM, constellation may include the constellation points which are defined based on Table 7,

[0456] Specifically, when encoding is performed with the coding rate of 9/15, at S**6130**, the interleaved codeword may be mapped onto the non-uniform constellation which includes the constellation points defined by NUC_256_9/15 of Table 7.

[0457] As another example, constellation may include, when the modulation scheme is 1024-QAM, the constellation points which are defined based on Table 8.

[0458] Specifically, when encoding is performed with the coding rate of 13/15, at S**6130**, the interleaved codeword may be mapped onto the non-uniform constellation which includes the constellation points defined based on of Table 8.

[0459] Meanwhile, Table 8 are used to define the constellation points in one quadrant, and the constellations in the remaining quadrants of constellation may be obtained by indicating each constellation point a, which is defined based on Table 8, as a*, −a*, and −a respectively (Here, * indicates complex conjugation).

[0460] In the present disclosure, in order to generate the optimized constellation, capacity needs to be determined. To do this, SNR is an important parameter. However, optimizing with respect to the SNR does not necessarily mean that an environment which satisfies the SNR is necessary. Though it is highly likely that the optimized performance may be obtained in the environment which satisfies the SNR, but in general, receiving SNR may change frequently according to system environment, and it is obvious that different SNR or different channel coding rate may be used according to not only complexity of realizing the system but also various purposes to support several channel environments using the modulation scheme to which one NUC constellation point is applied.

[0461] FIG. **44** is a block diagram illustrating a configuration of a receiving apparatus according to an exemplary embodiment.

[0462] Referring to FIG. **44**, a receiving apparatus **3800** may comprise a controller **3810**, an RF receiver **3820**, a demodulator **3830** and a service regenerator **3840**.

[0463] The controller **3810** determines an RF channel and a PLP through which a selected service is transmitted. The RF channel may be identified by a center frequency and a bandwidth, and the PLP may be identified by its PLP ID. A specific service may be transmitted through at least one PLP included in at least one RF channel, for each component constituting the specific service. Hereinafter, for the sake of convenience of explanation, it is assumed that all of data needed to play back one service is transmitted as one PLP which is transmitted through one RF channel. In other words, a service has only one data obtaining path to reproduce the service, and the data obtaining path is identified by an RF channel and a PLP.

[0464] The RF receiver **3820** detects an RF signal from an RF channel selected by a controller **3810** and delivers OFDM symbols, which are extracted by performing signal processing on the RF signal, to the demodulator **3830**. Herein, the signal processing may include synchronization, channel estimation, equalization, etc. Information required for the signal processing may be a value predetermined by the receiving apparatus **3810** and a transmitter according to use and implementation thereof and included in a predetermined OFDM symbol among the OFDM symbols and then transmitted to the receiving apparatus.

[0465] The demodulator **3830** performs signal processing on the OFDM symbols, extracts user packet and delivers the user packet to a service reproducer **3740**, and the service reproducer **3840** uses the user packet to reproduce and then output a service selected by a user. Here, a format of the user packet may differ depending on a service implementation method and may be, for example, a TS packet or a IPvv4 packet.

[0466] FIG. **45** is a block diagram illustrating a demodulator according to an exemplary embodiment.

[0467] Referring to FIG. **45**, a demodulator **3830** may include a frame demapper **3831**, a BICM decoder **3832** for L1 signaling, a controller **3833**, a BICM decoder **3834** and an output handler **3835**.

[0468] The frame demapper **3831** selects a plurality of OFDM cells constituting an FEC block which belongs to a selected PLP in a frame including OFDM symbols, based on control information from the controller **3833**, and provides the selected OFDM cells to the BICM decoder **3834**. The frame demapper **3831** also selects a plurality of OFDM cells corresponding to at least one FEC block which includes L1 signaling, and delivers the selected OFDM cells to the BICM decoder **3832** for L1 signaling.

[0469] The BICM decoder for L1 signaling **3832** performs signal processing on an OFDM cell corresponding to an FEC block which includes L1 signaling, extracts L1 signaling bits and delivers the L1 signaling bits to the controller **3833**. In this case, the signal processing may include an operation of extracting an LLR value for decoding an LDPC codeword and a process of using the extracted LLR value to decode the LDPC codeword.

[0470] The controller **3833** extracts an L1 signaling table from the L1 signaling bits and uses the L1 signaling table value to control operations of the frame demapper **3831**, the BICM decoder **3834** and the output handler **3835**. FIG. **45** illustrates that the BICM decoder **3832** for L1 signaling does not use control information of the controller **3833**. However, when the L1 signaling has a layer structure similar to the layer structure of the above described L1 pre signaling and L1 post signaling, it is obvious that the BICM decoder **3832** for L1 signaling may be constituted by at least one BICM decoding block, and operation of this BICM decoding block and the frame demapper **3831** may be controlled by L1 signaling information of an upper layer.

[0471] The BICM decoder **3834** performs signal processing on the OFDM cells constituting FEC blocks which belong to a selected PLP to extract BBF (Baseband frame)s and delivers the BBFs to the output handler **3835**. In this case, the signal processing may include an operation of extracting an LLR value for decoding an LDPC codeword and an operation of using the extracted LLR value to decode the LDPC codeword, which may be performed based on control information output from the controller **3833**.

[0472] The output handler **3835** performs signal processing on a BBF, extracts a user packet and delivers the extracted user packet to a service reproducer **3840**. In this case, the signal processing may be performed based on control information output from the controller **3833**.

[0473] According to an exemplary embodiment, the output handler **3835** comprises a BBF handler (not shown) which extracts BBP (Baseband packet) from the BBF.

[0474] FIG. **46** is a flowchart provided to illustrate an operation of a receiving apparatus from a moment when a user selects a service until the selected service is reproduced, according to an exemplary embodiment.

[0475] It is assumed that service information on all services selectable by a user are acquired at an initial scan (S**4010**) prior to the user's service selection (S**4020**). Service information may include information on a RF channel and a PLP which transmits data required to reproduce a specific service in a current receiving apparatus. As an example of the service information, program specific information/service information (PSI/SI) in an MPEG2-TS is available, and normally can be achieved through L2 signaling and an upper-layer signaling.

[0476] In the initial scan (S**4010**), comprehensive information on a payload type of PLPs which are transmitted to a specific frequency band. As an example, there may be information on whether every PLP transmitted to the frequency band includes a specific type of data.

[0477] When the user selects a service (S**4020**), the receiving apparatus transforms the selected service to a transmitting frequency and performs RF signaling detection (S**4030**). In the frequency transforming operation (S**4020**), the service information may be used.

[0478] When an RF signal is detected, the receiving apparatus performs an L1 signaling extracting operation from the detected RF signal (S**4050**). Then, the receiving apparatus selects a PLP transmitting the selected service, based on the extracted L1 signaling, (S**4060**) and extracts a BBF from the selected PLP (S**4070**). In S**4060**, the service information may be used.

[0479] The operation to extract a BBF (S**4070**) may include an operation of demapping the transmitted frame and selecting OFDM cells included in a PLP, an operation of extracting an LLR value for LDPC coding/decoding from an OFDM cell, and an operation of decoding the LDPC codeword using the extracted LLR value.

[0480] The receiving apparatus, using header information of an extracted BBF, extracts a BBP from the BBF (S**4080**). The receiving apparatus also uses header information of an extracted BBP to extract a user packet from the extracted BBP (S**4090**). The extracted user packet is used to reproduce the selected service (S**4100**). In the BBP extraction operation (S**4080**) and user packet extraction operation (S**4090**), L1 signaling information extracted in the L1 signaling extraction operation may be used.

[0481] According to an exemplary embodiment, the L1 signaling information includes information on types of a user packet transmitted through a corresponding PLP, and information on an operation used to encapsulate the user packet in a BBF. The foregoing information may be used in the user packet extraction operation (S**1480**). Specifically, this information may be used in an operation of extracting the user packet which is a reverse operation of encapsulation of the user packet in the BBF. In this case, process for extracting user packet from the BBP (restoring null TS packet and inserting TS sync byte) is same as above description.

[0482] A non-transitory computer readable medium in which a program which sequentially performs non-uniform constellation generation method is stored therein may be provided.

[0483] The non-transitory computer-recordable medium is not a medium configured to temporarily store data such as a register, a cache, or a memory but an apparatus-readable medium configured to semi-permanently store data. Specifically, the above-described various applications or programs may be stored in the non-transitory apparatus-readable medium such as a compact disc (CD), a digital versatile disc (DVD), a hard disc, a Blu-ray disc, a universal serial bus (USB), a memory card, or a read only memory (ROM), and provided.

[0484] In the block diagram which illustrates the transmitting apparatus and receiving apparatus, bus is not illustrated, but communication among elements of each apparatus can be done through the bus. In addition, each apparatus

may further include CPU performing the steps described above and processors such as a micro processor.

[0485] The components, elements, modules or units may be embodied as various numbers of hardware, software and/or firmware structures that execute respective functions described above, according to an exemplary embodiment. For example, these components, elements, modules or units may use a direct circuit structure, such as a memory, processing, logic, a look-up table, etc. that may execute the respective functions through controls of one or more microprocessors or other control apparatuses. Also, these components, elements, modules or units may be specifically embodied by a program or a part of code, which contains one or more executable instructions for performing specified logic functions. Also, at least one of these components, elements, modules or units may further include a processor such as a central processing unit (CPU) that performs the respective functions, a microprocessor, or the like.

[0486] The foregoing exemplary embodiments and advantages are merely exemplary and are not to be construed as limiting the inventive concept. The exemplary embodiments can be readily applied to other types of device or apparatus. Also, the description of the exemplary embodiments is intended to be illustrative, and not to limit the scope of the inventive concept, and many alternatives, modifications, and variations will be apparent to those skilled in the art.

What is claimed is:

1. A broadcast signal receiving method comprising:

receiving a signal from a broadcast signal transmitting apparatus;

demodulating the signal to generate values;

deinterleaving the values; and

decoding the deinterleaved values based on a low density parity check (LDPC) code, a code rate of the LDPC code being 13/15,

wherein the signal is demodulated based on constellation points for 1024-quadrature amplitude modulation (QAM), and

wherein each of the constellation points is represented as one of A+Bi, −A+Bi, A−Bi and −A−Bi,

where A is a real component, B is an imaginary component, and each of A and B comprises one of values listed below:

0.0325
0.0967
0.1623
0.2280
0.2957
0.3645
0.4361
0.5100
0.5878
0.6696
0.7566
0.8497
0.9498
1.0588
1.1795
1.3184

2. The broadcast signal receiving method of claim **1**, wherein a structure of the constellation points is a non-uniform constellation structure.

3. A transmitting method comprising:

interleaving a codeword comprising input bits and parity bits, wherein the parity bits are generated by encoding the input bits based on a low density parity check (LDPC) code, a code rate of the LDPC code being 13/15;

demultiplexing bits of the interleaved codeword to generate cells;

mapping the cells to constellation points for 1024-quadrature amplitude modulation (QAM); and

transmitting a signal which is generated based on the constellation points,

wherein each of the constellation points is represented in one of A+Bi, −A+Bi, A—Bi and −A−Bi, where A is a real component, B is an imaginary component, and each of A and B is one of values listed below:

0.0325
0.0967
0.1623
0.2280
0.2957
0.3645
0.4361
0.5100
0.5878
0.6696
0.7566
0.8497
0.9498
1.0588
1.1795
1.3184

4. The transmitting method of claim **3**, wherein a structure of the constellation points is a non-uniform constellation structure.

* * * * *