(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
05 July 2018 (05.07.2018)

W!P O | PCT

(10) International Publication Number
WO 2018/122855 Al

(51) International Patent Classification:
H04B 3/32 (2006.0 1) *H04L 1/00* (2006.01)
H04J 11/00 (2006.01)

(21) International Application Number:
PCT/IL20 17/05 1402

(22) International Filing Date:
28 December 2017 (28.12.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

62/439,501	28 December 2016 (28.12.2016)	US
62/610,75 1	27 December 2017 (27.12.2017)	US

(71) Applicant: SCKIPIO TECHNOLOGIES S.I LTD. [IL/IL]; 10 Ahaliav St., 5252263 Ramat Gan (IL).

(72) Inventors: CHISKIS, Alexander; 8 Hankin St., 6250620 Tel Aviv (IL). REINA, Guy; 1 Hamorad St., 5322001 Givatyaim (IL). VERBIN, Rami; 7 Beit-Tzuri St., 6912207 Tel Aviv (IL).

(74) Agent: KORAKH, Eliav et al; Korakh & Co. Advocates & Patent Attorneys, P.O.B. 58100, 6158002 Tel Aviv (IL).

(81) Designated States *(unless otherwise indicated, for every kind cf national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

(54) Title: SYSTEM AND METHOD UNIFYING LINEAR AND NONLINEAR PRECODING FOR TRANSCEIVING DATA

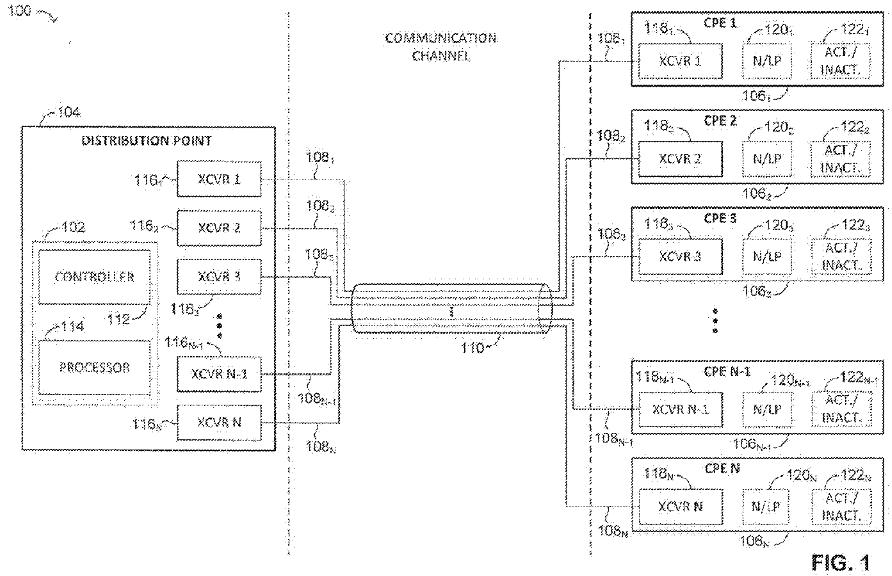

100
COMMUNICATION CHANNEL
104
DISTRIBUTION POINT
102
CONTROLLER
112
114
PROCESSOR
116_1 XCVR 1
116_2 XCVR 2
XCVR 3 116_3
116_{N-1} XCVR N-1
116_N XCVR N
108_1
108_2
108_3
108_{N-1}
108_N
110
CPE 1 118_1 XCVR 1 120_1 N/LP 122_1 ACT./INACT. 106_1
CPE 2 118_2 XCVR 2 120_2 N/LP 122_2 ACT./INACT. 106_2
CPE 3 118_3 XCVR 3 120_3 N/LP 122_3 ACT./INACT. 106_3
CPE N-1 118_{N-1} XCVR N-1 120_{N-1} N/LP 122_{N-1} ACT./INACT. 106_{N-1}
CPE N 118_N XCVR N 120_N N/LP 122_N ACT./INACT. 106_N

FIG. 1

(57) Abstract: A method for precoding an information symbol conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels over a subcarrier frequency, the number of transmitters (N) is different than the number of active receivers (K) for that subcarrier frequency, the method comprising: receiving by said transmitters information pertaining to supportabilities of said receivers to decode non-linearly precoded data; determining a precoding scheme defining for which of said receivers said data to be transmitted by said transmitters shall be precoded using at least one of linear precoding and non-linear precoding, according to said supportabilities; constructing a signal by applying a reversible mapping to said information symbol, said reversible mapping includes elements each respectively associated with a particular one of said receivers, such that those said receivers supporting the decoding of said non-linearly precoded data are capable of reversing said reversible mapping to said information symbol,

[Continued on nextpage]

SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind cf regional protection available):* ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *before the expiration cf the time limit for amending the claims and to be republished in the event cf receipt cf amendments (Rule 48.2(h))*

while for those said receivers not supporting the decoding of said non-linearly precoded data said information symbol is unaffected by said reversible mapping; constructing a precoder characterized by N≠K such that said precoder is configured to perform regularized generalized inversion of a communication channel matrix.

SYSTEM AND METHOD UNIFYING LINEAR AND NONLINEAR PRECODING FOR TRANSCEIVING DATA

FIELD OF THE DISCLOSED TECHNIQUE

The disclosed technique relates to communication systems and methods in general, and to a system and method for employing linear and nonlinear precoding, in particular.

BACKGROUND OF THE DISCLOSED TECHNIQUE

In multi-user communications where a centralized transmitter transmits data to a plurality of independent (e.g., non-cooperative) receivers (users), the transmitted data may be subject to inter-user noise, known as crosstalk, which interferes with the communication between different communication entities. Attaining an effective contrivance to eliminate or at least partially reduce crosstalk is therefore of high importance. Crosstalk may generally occur in both wireless and wire-line communications systems, that utilize linear precoding (LP) and nonlinear precoding (NLP) techniques, and particularly, in the Gigabit Internet "G.fast" wire-line standard.

Crosstalk cancellation techniques that employ precoding of data prior to its transmission are known in the art as "vectoring". Crosstalk cancellation typically requires taking into account of power restrictions, which often involve hardware-related considerations. Additionally, the number of bits is limited to predefined constellation sizes. The linear precoder may eliminate crosstalk in part or fully by using an inverse of a channel matrix. Linear precoding, however, may typically require equalization of gains introduced by the inversion operation (i.e., the gains must be suppressed to satisfy power restrictions, which in turn cause a diminished bitrate). The NLP schemes avoid this problem by use of Tomlinson-Harashima Precoding (THP) scheme working through the modulus operation or by seeking a perturbation vector associated with transmission symbols, thereby reducing power consumption. NLP schemes work seamlessly at the receiver after application of the modulus operation.

Systems and method that combine linear precoding and nonlinear precoding, in general, are known in the art. A World Intellectual Property Organization (WIPO) Patent Cooperation Treaty (PCT) international Publication Number WO 2014/054043 A 1 to Verbin et al. to the same present Applicant, entitled "Hybrid Precoder" is directed to a hybrid precoder system and method employing linear precoding and nonlinear precoding to provide far-end crosstalk (FEXT) cancellation that enhances performance and lowers complexity during transmission and reception of data between transmitters and receivers of the communication system. The hybrid precoder system and method employs linear precoding and non-linear precoding for transmitting data between at least two transmitters and a plurality of receivers via a plurality of communication channels over a plurality of subcarrier frequencies. The at least two transmitters are communicatively coupled, respectively, with the plurality of receivers. The hybrid precoder system includes a linear precoder, a non-linear precoder, a controller, and an input selector. The linear precoder is for linearly precoding the data. The non-linear precoder is for non-linearly precoding the data. The controller is coupled with the linear precoder, and with the non-linear precoder. The input selector as well, is coupled with the linear precoder and with the non-linear precoder. The controller at least partly evaluates channel characteristics of at least part of the communication channels. The controller further determines a precoding scheme selection that defines for at least part of the communication channels, over which of the subcarrier frequencies the data to be transmitted shall be preceded using either one of linear precoding and non-linear precoding, according to determined channel characteristics. The input selector selects which of the linear precoded data and the non-linear precoded data is outputted by the hybrid precoder system, according to the precoding scheme selection.

U.S. Patent Application Publication No.: US 201 7/0279490 A 1 to Maes, entitled "Non-linear Precoding with a Mix of NLP Capable and NLP Non-capable Lines" is directed at a method for achieving crosstalk mitigation in the presence of nonlinear precoding (NLP) non-capable and NLP capable multiple customer premises equipment (CPE). Maes provides a particular solution to the general interoperability problem of using different precoding-capable CPE units where the number active CPE units, is equal to the total number of CPE units.

SUMMARY OF THE DISCLOSED TECHNIQUE

It is an object of the disclosed technique to provide a method for preceding an information symbol conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels over a subcarrier frequency, where the number of transmitters (N) is different than the number of active receivers (K) for that subcarrier frequency. The method includes the following steps. The method initiates with a step of receiving by the transmitters, information pertaining to supportabilities of the receivers to decode non-linearly preceded data. The method continues with a step of determining a precoding scheme defining for which of the receivers the data to be transmitted by the transmitters shall be precoded using at least one of linear precoding and non-linear precoding, according to the supportabilities. The method continues with a step of constructing a signal by applying a reversible mapping to the information symbol, where the reversible mapping includes elements each respectively associated with a particular one of the receivers, such that those receivers supporting the decoding of non-linearly precoded data are capable of reversing the reversible mapping to the information symbol, while for those receivers not supporting the decoding of non-linearly precoded data the information symbol is unaffected by the reversible mapping. The method continues with a step of constructing a precoder characterized by N ≠ K such that the precoder is configured to perform regularized generalized inversion of a communication channel matrix.

It is a further object of the disclosed technique to provide a hybrid precoder system for precoding an information symbol conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels over a subcarrier frequency, where the number of transmitters (N) is different than the number of active receivers (K) for that subcarrier frequency. The hybrid precoder system includes a controller and a processor (coupled therebetween). The controller is configured for receiving information pertaining to supportabilities of the receivers to decode non-!inearly precoded data, and for determining a precoding scheme defining for which of the receivers the data to be transmitted by the transmitters shall be precoded using at least one of linear precoding and non-linear precoding, according to the supportabilities. The processor is configured for constructing a signal for transmission, according b the determined precoding scheme, by applying a reversible mapping to the information symbol, where the reversible mapping includes elements each respectively associated with a particular one of the receivers, such that those receivers supporting the decoding of non-linearly precoded data are capable of reversing the reversible mapping to the information symbol, while for those receivers not supporting the decoding of non-linearly precoded data the information symbol is unaffected by the reversible mapping. The processor is further configured for constructing a precoder characterized by N ≠ K such that the precoder is configured to perform regularized generalized inversion of a communication channel matrix.

It is a further object of the disclosed technique to provide a method for nonlinear precoding of an information symbol at a given precoder input. The information symbol is in a symbol space having a given symbol space size. The nonlinear precoding involves modulo arithmetic and having a plurality of inputs. The method includes the following steps. The method includes an initial step of determining a reference symbol space size which is common to all of the inputs. The method continues with the steps of determining a modulus value according to the reference symbol space size, adapting the given symbol space size according to the reference symbol space size, and **nonlinearly** precoding the information symbol according to the modulus value, common to all of the inputs.

It is a further object of the disclosed technique to provide a system for nonlinear precoding of an information symbol at a given precoder input, where the information symbol is in a symbol space having a given symbol space size. The nonlinear precoding involves modulo arithmetic and having a plurality of inputs. The system includes a controller and a processor (coupled therebetween). The controller is configured for determining a reference symbol space size that is common to all of the inputs, and for determining a modulus value according to the reference symbol space size. The processor is configured for adapting the given symbol space size according to the reference symbol space size, and for nonlinearly precoding the information symbol according to the modulus value, common to all of the inputs.

In is another object of the disclosed technique to provide a method for nonlinear precoding an information symbol conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels defining a channel matrix $\boldsymbol{H}$ over a particular subcarrier frequency. The method includes the steps of determining a weighting matrix $\boldsymbol{G}$, whose number of rows is equal to the number of transmitters; then determining a modified channel matrix equal to EG; and constructing a nonlinear precoder for performing nonlinear precoding of the modified channel matrix.

It is a further object of the disclosed technique to provide a system for nonlinear precoding an information symbol conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels defining a channel matrix H over a particular subcarrier frequency. The system includes a processor configured for determining a weighting matrix $\boldsymbol{G}$, whose number of rows is equal to the number of transmitters; for determining a modified channel matrix equal to EG; and for constructing a nonlinear precoder for performing nonlinear precoding of the modified channel matrix.

BRIEF DESCRIPTION OF THE DRAWINGS

The disclosed technique will be understood and appreciated more fully from the following detailed description taken in conjunction with the drawings in which:

Figure 1 is a schematic diagram illustrating an overview of a communication system, showing a system of the disclosed technique, constructed and operative according to an embodiment of the disclosed technique;

Figure 2A is schematic diagram illustrating a prior art zero-forcing (ZF) linear precoding scheme;

Figure 2B is a schematic diagram illustrating a prior art ZF nonlinear vector precoding scheme;

Figure 2C is a schematic diagram illustrating a prior art QR nonlinear precoding scheme, generally referenced 50;

Figure 3A is a schematic diagram illustrating an overview of a general hybrid-interoperability precoding scheme supporting both linear and nonlinear precoding, constructed and operative according to the embodiment of the disclosed technique;

Figure 3B is a schematic diagram illustrating an overview of another general hybrid-interoperability precoding scheme including permutations supporting both linear and nonlinear precoding, constructed and operative according to the embodiment of the disclosed technique;

Figure 3C is a schematic diagram illustrating an overview of a further general hybrid-interoperability precoding scheme including permutations supporting both linear and nonlinear vector precoding, constructed and operative according to the embodiment of the disclosed technique;

Figure 4 is a schematic diagram illustrating an example of a specific implementation of the general hybrid-interoperability precoding scheme, utilizing a QR nonlinear precoder and permutations, configured and operative in accordance with the disclosed technique;

Figure 5 is a schematic diagram illustrating a partition of vector variables into two groups, one group associated with linear precoding (LP) and the other group associated with nonlinear precoding (NLP);

Figure 6 is a schematic diagram illustrating an example permutation configuration in the specific implementation of the general hybrid-interoperability precoding scheme of Figure 4;

Figure 7 is a schematic diagram illustrating an example configuration of an internal structure of the permutation block in Figure 6;

Figure 8 is a schematic illustration detailing a partition of a lower-diagonal matrix L having the dimensions $(K_1 + K_2)$ x $(K_1 + K_2)$ into three matrices;

Figure 9 is a schematic illustration detailing an example configuration of an internal structure of the preprocessing block in Figure 6;

Figure 10 is a schematic illustration detailing an example configuration of an internal structure of preprocessing block and permutation block in Figure 6, shown in a particular 5-user configuration;

Figure 11 is a schematic illustration showing a particular implementation of NLP/LP control mechanisms in an internal structure of preprocessing block;

Figure 12 is a schematic illustration showing another particular implementation of the preprocessing block of Figure 6;

Figure 13 is a schematic illustration showing further particular implementation of the preprocessing block of Figure 6;

Figure 14A is a table showing a database of supportabilities and activity levels of each CPE unit at a particular point in time, constructed and operative in accordance with the disclosed technique;

Figure 14B is a schematic diagram showing a graph of a particular example of activity levels of CPE units ordered according to (relative) communication link quality as a function of subcarrier frequency at a particular point in time, in accordance with the disclosed technique;

Figure 15 is a schematic block diagram illustrating a specific implementation of the general hybrid-interoperability precoding scheme, specifically showing delineation into two paths, constructed and operative in accordance with the disclosed technique;

Figure 16 is a schematic block diagram illustrating another specific implementation of the general hybrid-interoperability precoding scheme, specifically showing delineation into two paths, constructed an operative in accordance with the disclosed technique;

Figure 17A is a schematic diagram illustrating an example of a specific implementation of the general hybrid-interoperability precoding scheme, utilizing different scalar factors, configured and operative in accordance with the disclosed technique;

Figure 17B is a schematic block diagram of a method for a specific implementation of nonlinear precoding utilizing different scalar factors, configured and operative in accordance with the disclosed technique; Figure 18 is a schematic block diagram of a method for a hybrid-interoperability precoding scheme supporting both linear and nonlinear precoding, constructed and operative according to the embodiment of the disclosed technique;

Figure 19 is a schematic block diagram of a system for nonlinear precoding exhibiting a modulus size that is constellation-independent, constructed and operative in accordance with another embodiment of the disclosed technique;

Figure 20 is a schematic illustration detailing an example configuration of an internal structure of a vectoring processor in the system of Figure 19, constructed and operative in accordance with an embodiment of Figure 19 of the disclosed technique;

Figure 21 is a schematic diagram of is a schematic illustration of Tomlinson-Harashima precoding used per subcarrier frequency being applied to chosen users only, constructed and operative in accordance with an embodiment of Figure 19 of the disclosed technique;

Figure 22A is a schematic diagram showing an example of a non-scaled 4-QAM constellation, constructed and operative in accordance with the disclosed technique;

Figure 22B is a schematic diagram showing an example of non-scaled 16-QAM constellation;

Figure 22C is a schematic illustration showing a boundary of a modulo operation representing a square of size τ;

Figure 22D is a schematic diagram showing an example of $\tau = 8 = 2^3$ chosen as a constant modulo for all constellations; and

Figure 23 is a schematic block diagram of a method for nonlinear precoding where the modulus size is constellation-independent, constructed and operative in accordance with the embodiment of Figure 19 of the disclosed technique.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The disclosed technique overcomes the disadvantages of the prior art by proposing a general solution to the interoperability problem between a data providing entity communicatively coupled with multiple data subscriber entities in a communication network, where part of the data subscriber entities do not support nonlinear precoding (NLP) while another part does. The disclosed technique generally relates to multi-user multiple input multiple output (MIMO) communications systems in which there is a data provider side, typically embodied in the form of a data providing entity, such as a distribution point (DP) that is interconnected via a plurality of communication channels to a plurality of data subscriber entities (i.e., a data subscriber side), typically embodied in the form of multiple corresponding customer premises equipment (CPE) units. The terms "data provider side", "data provider", "transmitter side", "distribution point", and "distribution point unit" (DPU) used herein are interchangeable. The terms "data subscriber side", "data subscriber", "receiver side", "CPE", "CPE unit", and "CPE receiver unit" used herein are interchangeable. The disclosed technique proposes a system and a method configured and operative to unite or unify linear and nonlinear schemes, for a particular subcarrier frequency, at the data provider side, in such a way that enables interoperability in the use of nonlinear precoding for CPE units supporting nonlinear precoding and linear precoding for CPE units not supporting nonlinear precoding.

The system and method of the disclosed technique is configured and operative for precoding an information symbol conveying data for transmission between a plurality of transceivers at the data provider side (i.e., operating in the downstream (DS) direction as transmitters) and a plurality of transceivers at the data subscriber side (i.e., operating in the DS direction as receivers) via a plurality of communication channels over a subcarrier frequency (denoted herein interchangeably 'tone'). A symbol generally refers to a waveform, a signal, or a state of a communication medium (e.g., link, channel) that transpires over a particular time period (e.g., a time slot). A symbol typically encodes bits (i.e., "information symbol"). According to one (typical) implementation the communication channels are wire-lines (e.g., physical wire conductors such as twisted pairs). According to another implementation the communication channels are realized by the transmission and reception of signals (e.g., via antennas in wireless communication techniques) propagating through a wireless medium (e.g., air). The communication channels whether wired or wireless are susceptible to interference known as crosstalk between the communication channels, more specifically known as far-end crosstalk (FEXT). The communication channels may as be susceptible to near-end crosstalk (NEXT). Preceding is used for mitigating the effects of FEXT, while adaptive filtering may be used for mitigating the effects of NEXT.

The prior art teaches a specific solution, limited to a very special case where the number of transmitters at the transmitter side is equal to the number of active receivers at the receiver side at any particular subcarrier frequency. The disclosed technique offers a solution to the general case, where the number of transmitters is not necessarily equal to the number of active receivers at the receiver side at a particular subcarrier frequency. An 'active receiver' (e.g., an active CPE unit) is a receiver that (i) is switched on and is ready to receive data or in a process of receiving data; (ii) is switched on and is ready to receive data or in a process of receiving data at a particular subcarrier frequency (or plurality thereof) and not for other subcarrier frequencies (i.e., the receiver is 'active' at particular subcarrier frequencies and 'inactive' at other subcarrier frequencies) or **(iii)** is either one of (i) and (ii) stipulated by a decision rule determined by at least one criterion related to communication performance (e.g., max-mean rate, max-min rate, minimal bit loading value, etc). Examples for (ii) include situations where a particular active CPE unit is unable to receive information at high subcarrier frequencies due to a low signal-to-noise ratio (SNR), or due to the peculiarity of the communication channel or binder structure, etc. Thus, a CPE unit may be active at some subcarrier frequencies, and inactive at others. The system and method of the disclosed technique provide a general solution to the more difficult interoperability problem for the general case, where the number of transmitters *(N)* is not necessarily equal to the number of active receivers *(K).* The general solution also solves the special case where *N* is equal to *K*.

The following is a succinct summary of the system and method of the disclosed technique; the summary is followed by a comprehensive description. The system includes a controller and a processor implemented at the transmitter side (e.g., in the DPU). The controller is typically embodied in the form of, and interchangeably denoted herein, a 'vectoring control entity' (VCE). The processor is typically embodied in the form of, and interchangeably denoted herein, a Vectoring processing entity' (VCE). The controller is configured and operative for receiving information pertaining to supportabilities of the receivers (i.e., CPE units) to decode nonlinearly precoded data. In general, a 'supportability' of a receiver defines whether that receiver supports the decoding of nonlinearly precoded data (and linear precoded data). It is assumed herein that if a particular receiver does not support the decoding of nonlinearly precoded data, then its default supportability is the capability to decode linearly precoded data. The controller is further configured and operative for determining a precoding scheme defining for which of the receivers (at the receiver side) the data to be transmitter by the transmitters (at the transmitter side) shall be precoded using at least one of linear precoding (LP) and nonlinear precoding (NLP) according to the supportabilities of the receivers.

The processor is configured and operative for constructing a signal for transmission, according to the determined precoding scheme, by applying a reversible mapping (i.e., a reversible transformation) to the information symbol. A reversible mapping is a function or algorithm that can be reversed (i.e., reversibility yields the operand that is, the object of the mapping operation). As will be described in greater detail heresnbelow, a reversible mapping can be realized by various entities and techniques. Several such techniques include use of modulo arithmetic, a use of a perturbation vector, use of transformations in a lattice reduction technique, and the like. The reversible mapping includes elements (e.g., represented by matrix elements) each respectively associated with a particular one of the receivers, whereby those receivers supporting the decoding of nonlinearly precoded data are capable of reversing the reversible mapping to the information symbol, while for those receivers not supporting the decoding of nonlinearly precoded data the information symbol is unaffected by the reversible mapping. The processor is then configured and operative to construct a precoder characterized by $N \neq K$ such that the precoder is configured to perform regularized generalized inversion of a communication channel matrix. The communication channel matrix (or simply "channel matrix") represents the channel information conveyed between transmitter side and receiver side. Regularized generalized inversion, which will be discussed in greater detail hereinbelow and used in the context of the disclosed technique, relates to a generalization of generalized inversion. Basically, generalized inversion of the channel matrix essentially involves finding a matrix that serves as an inverse of the channel matrix that is not necessarily invertible. An example of generalized inverse includes the Moore-Penrose pseudoinverse. Regularization of the generalized inverse, known herein as "regularized generalized inversion" involves use of the principles of regularization by introducing a regularization term to the mathematical expression representing the generalized inverse.

According to the disclosed technique there is thus provided a method for preceding an information symbol conveying data for transmission between a plurality of transmitters (i.e., defining a transmitter side) and a plurality of receivers (i.e., defining a receiver side) via a plurality of communication channels over a subcarrier frequency. The number of transmitters *(N)* is different than the number of active receivers *(K)* for that subcarrier frequency. The method includes the following steps including an initial step of receiving by the transmitter side (e.g., transmitters), information pertaining to supportabilities of the receivers to decode nonlinearly precoded data. The method proceeds with the step of determining a **precoding** scheme defining for which of the receivers the data to be transmitted by the transmitters shall be precoded using at least one of linear precoding and nonlinear precoding, according the supportabilities. The method proceeds with the step of constructing a signal by applying a reversible mapping to the information symbol. The reversible mapping includes elements each respectively associated with a particular one of the receivers, such that those receivers supporting the decoding of nonlinearly precoded data are capable of reversing the reversible mapping to the information symbol, whereas the information symbol is unaffected by the reversible mapping for those receivers not supporting the decoding of the nonlinearly precoded data. The method proceeds with the step of constructing a precoder characterized by $N \neq K$ such that the precoder is configured to perform regularized generalized inversion of a communication channel matrix. The method proceeds with the step of transmitting the signal by the transmitters.

At the receiver side, the CPE units are configured and operative to receive the signal from the transmitters. Both types of CPE units, namely, those supporting the decoding of linearly precoded data as well as those supporting the decoding of nonlinearly precoded data are configured and operative to perform equalization by multiplication of the received signal by a scalar (i.e., not necessarily the same scalar for each CPE unit). The CPE units supporting the decoding of nonlinearly precoded data are further configured and operative to reverse the reversible mapping (e.g., by applying modulo operation in accordance with the selected reversible mapping).

In other words, users who have hardware (e.g., DSL modems), software, firmware, and the like supporting nonlinear precoding (e.g., modulo arithmetic capable) may choose to use at least one of nonlinear precoding and linear precoding, whereas users who don't have hardware (software, firmware, etc.) supporting nonlinear precoding may still use linear precoding. A particular CPE unit whose supportability includes nonlinear precoding is not necessarily limited only to nonlinear precoding as that CPE unit may opt to employ linear precoding for the benefit of system performance, or alternatively, for the reduction of nonlinear precoder dimensionality (consequently reducing computational complexity).

Particularly, in the case of the system employing orthogonal frequency-division multiplexing (OFDM) for encoding data on multiple subcarriers, all users (corresponding to CPE units) may be divided into three groups at every tone: (1) a group of CPE unit(s) (user(s)) employing nonlinear precoding; (2) a group of CPE unit(s) employing linear precoding; and (3) a group of CPE unit(s) that are inactive (i.e., are not precoded at that particular tone and are thus excluded from the transmitted signal). There are also derivative logical groups, e.g., a group that is an intersection of groups (1) and (2), and the like. The communication channels (e.g., lines) of the inactive CPE unit(s), at a specific tone, are exploited to transmit information for the benefit of other CPE unit(s). The division of CPE unit(s) (user(s)) between these three groups may vary from tone to tone since the channel matrix and the signal-to-noise ratio (SNR) are usually frequency dependent. For the above mentioned group (2) of users which employ linear precoding, modulo arithmetic (e.g., the modulus operation) is not utilized (applied) at the receiver side (CPE units).

The ability to allocate CPE units that support nonlinear precoding among linear and nonlinear precoding schemes also improves system performance by diverting nonlinear precoding enabled CPE units exhibiting large power and coding gain losses (typically belonging to small bit constellations) to utilize linear precoding. In addition, this allocation enables part of the CPE units to process data via linear precoding, while enables the remaining (active) CPE units to utilize techniques of nonlinear precoding, such as vector precoding, which reduces the dimensionality of the search space for a perturbation vector (i.e., given that the complexity of such a search is known to lie between polynomial and exponential in dimension size). Consequently, the allocation of the CPE units into three groups for every tone (i) effectively facilitates attainment a solution of the interoperability problem between different CPE units, which either have or don't have nonlinear precoding supporting hardware, and (ii) serves as an instrument to achieve system performance optimization.

Where reversible mapping is implemented via use of modulo arithmetic, the disclosed technique also proposes an option of utilizing a constant modulus size for NLP, and particularly for Tomiinson-Harashima precoding (THP) schemes. This brings about a constant power requirement to be satisfied automatically, facilitating an increase in hardware efficiency involving execution of the modulus operation, as well as averts the need for different modulus values for different constellation sizes. The disclosed technique is implementable to any number of CPE units that may be split arbitrarily between NLP-supporting CPE units and CPE units not supporting NLP.

Notation: The notation used herein for the operation *diag*(*A*) applied to a matrix *A* yields a vector equal to its diagonal, and the operation *diagia)* applied to a vector $\boldsymbol{a}$ yields a matrix with a diagonal $\boldsymbol{a}$ and all other non-diagonal elements are zeroes. The notation for the component-wise multiplication (also known as the Hadamard product) is $\boldsymbol{a} \odot b$, which signifies that every component of the product, $\boldsymbol{c}$, is constructed by multiplication of components $\boldsymbol{a}$ and *b*: $c_k = a_k b_k$. The Hermitian conjugation of a matrix *R* (i.e., matrix transpose and complex conjugation of every element) is denoted by R^H. The notation R^{-H} is used herein to denote (J?$^{-1})^{H}$. In the Figures and Detailed Description, the prime symbol, ' , denotes the Hermitian conjugation: e.g., *H'* signifies a Hermitian conjugation of matrix ***H*.** The inverse of a diagonal matrix *D* is also a diagonal matrix given by simple inversion of the diagonal components: for *G* = *Lnv(D)*, $G_{ii} = \frac{1}{D_{ii}}$ and all non-diagonal components being zero. Vectors and matrices are represented in bold-italics.

Without loss of generality, the disclosed technique will be described in the context of a wire-line communication system, though the principles of the disclosed technique likewise apply for wireless communication systems. Reference is now made to Figure 1, which is a schematic diagram illustrating an overview of a communication system, generally referenced 100, showing a system of the disclosed technique, generally referenced 102, constructed and operative according to an embodiment of the disclosed technique. Figure 1 shows a distribution point unit (DPU) 104 (also denoted herein interchangeably as "network side entity", and "distribution point" (DP)), communicatively coupled to at least one (typically a plurality of) customer premises equipment (CPE) unit(s) 106-j, 106_2, 106_3,..., 106_{N-1}, 106_N (also denoted herein interchangeably as a "node" or "nodes" in plural, where $N \geq 1$ is an integer) via a plurality of *N* communication lines (also denoted herein interchangeably as "communication channels") 108i, 108_2, 108_3,..., 108N-I , 108$_N$ (typically and at least partially passing through a binder 110, in the wire-line case). DPU 104 includes a controller 112, a processor 114 embodying the system of the disclosed technique, generally referenced 102. DPU 104 further includes a plurality of *N* network-side fast transceiver unit (FTU-O) transceivers (XCVRs) 116$_1$, 116$_2$, 116$_3$,.. .,1 16$_{N-i}$, 116$_N$.

At the receiver side, each one of CPE units 106$_1$, 106$_2$, 106$_3$,.. . ,106$_{N-1,}$ 106$_N$ respectively includes a corresponding transceiver (XCVR), also denoted herein interchangeably as "remote-end fast transceiver unit" (FTU-R) 118,, 118$_2$, 118$_3$,...,118$_{N-1,}$ 118$_N$. Particularly, each transceiver (XCVR) 116$_1$, 116$_2$, 116$_3$,.. ., 116$_{N-1}$, 116$_N$ at the network side is communicatively coupled with its respective CPE unit 106$_1$, 106$_2$, 106$_3$,.. ., 106N-I , 106$_N$ via its respective communication line IO8 $_1$, 108$_2$, 108$_3$,.. ., 108$_{N-1}$, 108$_N$ (i.e., index-wise). Each remote-end fast transceiver unit is configured and operative to receive and to transmit data to-and-fro its respective FTU-0 at the DPU 104. Specifically, each transceiver at the network side, i.e., FTU-O$_i$ (where / is an integer running index) is configured to be in communication with a corresponding transceiver at the receiver side, i.e., FTU-Rj.

The terms "communication channel", "communication line", "communication link" or simply "link" are interchangeable and are herein defined as a communication medium (e.g., physical conductors, air) (whether wired or wireless) configured and operative to communicatively couple between DPU 104 and CPE units 106·,, 106$_2$, 106$_3$,.. ., 106$_{N-1}$,106$_N$. The communications channels are configured and operative to be propagation media for signals (information symbols) for wireless as well as wire-line communication methods (e.g., xDSL, G.fast services). DPU 104 is typically embodied as a multiple-link enabled device (e.g., a multi-port device) having a capability of communicating with a plurality of nodes (e.g., CPE units). Alternatively, DPU 104 is a single-link device (not shown) having a capability of communicating with one node (e.g., a CPE unit). A transmission from DPU 104 to at least one of CPE units 106_1, 106_2, 106_3,.. .,106N-I ,106N is defined herein as a downstream (DS) direction. Conversely, a transmission from at least one of CPE units 106_1, 106_2, 106_3,.. ., 106N-I ,106N to DPU 104 is defined herein as an upstream (US) direction.

Each one of CPE units 106_1, 106_2, 106_3,..., 106_{N-1},106_N is partly characterized by its respective inherent supportability 120i, 120_2, 120_3,.. ., 120_{N-i},120_N to decode at least one type of precoded data, namely, linear preceded (LP) data, and nonlinear precoded (NLP) data (or both linear and nonlinear precoded data). It is assumed that if a particular CPE unit does not support the decoding of NLP data it supports by default the decoding of LP data. In a typical case, it is assumed that all CPE units possess supportability to decode LP data. The point is which of the CPE units further possess supportability to decode NLP data. The system and method of the disclosed technique are configured and operative to receive information pertaining to supportabilities 120i, 120_2, 120_3,.. ., 120N-I ,120N to decode LP and especially NLP data (represented in Figure 1 as "N/LP"). Information pertaining to the supportabilities of CPE units is typically acquired in an initialization process, in which initial communication parameters are determined between the DPU and the CPE units. Initialization typically involves a plurality of steps or phases, such as a handshake and discovery phase, a training phase, and a channel evaluation and analysis phase. The initialization phase, and specifically the handshaking and discovery phase, involves the CPE units communicating to DPU 104 information pertaining to their supported capabilities to decode nonlinear precoded data. For example, DPU 104 is configured and operative to send a message (request) to each of the CPE units to report their respective supportability. In response, the CPE units are configured and operative to reply in a return message, specifying information pertaining to their respective supportabi!ity (e.g., 'NLP supported'/'NLP not supported'). Once initialization has been completed communication system 100 enters a data exchange phase and in particular, when bearer data (e.g., payload data) is being transmitted, this is what is typically known as "showtime". Alternatively, information pertaining to the supportabiiities of the CPE units is determined independently from the CPE units (e.g., in an initial setup of system 100, by controller 112 receiving information such as a lookup table of supportabiiities, and the like).

DPU 104 (controller 112 thereof) is further configured and operative to continually determine activity levels 122_1, 122_2, 122_3,..., **122** $_{N-1}$,122_N (represented "ACT./INACT." in Figure 1) of each of CPE units 106_1, 106_2, 106_3,..., 106_{N-1},$106_{N.}$ The activity level generally defines a current degree of operation or function of a CPE unit. An 'active receiver' (interchangeably denoted herein as an "active CPE unit") is a receiver that is switched on and ready to receive data or in a process of transceiving (i.e., receiving and/or transmitting) data per tone. An Inactive receiver' (interchangeably denoted herein as an "inactive CPE unit") is a receiver that is not ready to receive data or otherwise unable to communicate with the network side (e.g., switched off, not functioning, malfunctioning, not initialized, not connected, etc.). The activity level will be discussed in greater detail hereinbelow in conjunction with Figures 14A and 14B. The maximum number of active receivers (K) is as the total number of CPE units *(N)* per tone. Typically, in the routine operative state of system 100, at any particular time, the number of active receivers may usually be less than the total number of CPE units $(K < N)$ per tone.

Following the determination of the supportabiiities (and activity levels) of the CPE units, the system and method of the disclosed technique are configured and operative to determine a precoding scheme defining for which of the CPE units the, data transmitted by the DPU shall be precoded using at least one of linear preceding and nonlinear precoding, according to the determined supportabilities (and activity levels). In other words, signals transmitted from transceivers 116_1, 1162, 1163,..., 116_{N-1},116_N of DPU 104 to respective transceivers 118 1, 118_2, 1183,..., **118N-I**,1 18_n of CPE units 106_1, 106_2, 106_3,..., 106_N.i,1 06_N (respectively) have to take into account respective supportabilities120i, 120_2, $12O_3$,.. ., 120_{N-1},120_N and respective activity levels 122_1, 122_2, 122_3,.. ., 122_{N-1},122_N. As will be described in greater detail hereinbelow, the preceding scheme defines how a transmitted signal by the DPU is precoded given the supportabilities and activity levels of the CPE units.

At this stage, for the purpose of highlighting the differences between the solution of the disclosed technique and known solutions of the prior art, reference is now further made to Figures 2A, 2B, and 2C. Figure 2A is schematic diagram illustrating a prior art zero-forcing (ZF) linear precoding scheme, generally referenced 10. Figure 2B is a schematic diagram illustrating a prior art ZF nonlinear vector precoding scheme, generally referenced 30. Figure 2C is a schematic diagram illustrating a prior art QR nonlinear precoding scheme, generally referenced 50. Figures 2A, 2B, and 2C show a transmitter side, a communication channel (represented by block *H),* and a receiver side. The precoding scheme shown in Figures 2A, 2B, and 2C are described in the context of vector precoding that utilizes a perturbation vector, c.

With reference to linear precoding scheme 10 of Figure 2A, at the transmitter side, information symbols denoted by a vector s, constitute an input 12 intended for precoding. All components of a perturbation vector 13, c, which are all zero for the linear precoding (LP) scheme, are added 14 with vector s (i.e., which remains the same) and inputted 15 into a linear precoder block 16. Hence, for LP a perturbation is not added, and c is equal to the zero vector. The linear precoder (linear precoder block 16) is denoted by F, which is given by: $P - pinv(H) * D$. The operator

inv() denotes an inverse of a matrix (inscribed between parentheses), ***pinvQ*** denotes a pseudoinverse of a matrix (inscribed between parentheses). LP block 16 applies a pseudoinverse operation to a channel matrix H and calculates a product with a diagonal matrix D. Diagonal matrix D is used to scale the precoding matrix. LP block 16 yields a signal or group of signals as preceded information symbols denoted by an output 17, o, which is then communicated via a communication channel 18 to a receiver side. Output 17 from the transmitter side, o, to the communication lines, may further be scaled by a scalar gain factor, a (not shown). Signals received at the receiver side denoted by a received vector 22, $\boldsymbol{r}$, is a sum of a vector 19, y, and an additive noise vector 20, $\mathbf{n}$, denoting that the precoded information symbols propagated through the communication channel includes additive 21 noise 20. Scaling at the receiver side is performed independently for every component by an $inv(D)$ block 24, the result of which is an estimated output symbol 26, $\hat{s}$. No modulus operation is performed at the receiver side.

With reference to nonlinear precoding scheme 30 of Figure 2B, a perturbation vector 33, c, is determined and added 34 to input information symbols 32 denoted by a vector s, constituting an input 35 intended for precoding. Perturbation vector c is a nonzero vector, which forms an essential part of a nonlinear precoder (NLP). Perturbation **vector c** is constructed as an (signed) integer (i.e., or for complex constellations it is a pair of two signed complex integers $i_1 + j * i_2$ where j is the imaginary component) multiplied by a modulus component, having different values per component (as is common in the case when NLP is the Tomlinson-Harashima precoder (THP). A nonlinear precoder (NLP block 36), denoted by P is given by $P = \boldsymbol{inv(H)} * D$. NLP block 36 applies an inverse operation to a channel matrix $\boldsymbol{H}$ and calculates a product with a diagonal matrix $\boldsymbol{D}$. Diagonal matrix D is used to scale the precoding matrix and is predefined for THP. NLP block 36 yields a signal or group of signals as precoded information symbols denoted by an output 37, ***o***, which then is communicated via a communication channel 38 to a receiver side. An output 37 to communication channel (e.g., the communication lines) is denoted by *o* representing the precoded symbols. Output vector ***o*** has *N* components: $\boldsymbol{o} \in \mathbf{C}^{N\times 1}$ (where **C** denotes the complex field). The received vector 42, r , is the sum of a vector 39, *y*, and an additive noise vector 40, **n**, denoting that the precoded information symbols propagated through the communication channel includes additive 41 noise 40. Scaling at the receiver side is performed independently for every component by an *inv(D)* block 44, an output 45 of which is provided to a modulus operation block 46. The notation *mod()* denotes the modulo operation (also interchangeably denoted "modulus" operation) applied to an operand (inscribed between parentheses) per component (i.e., real and imaginary parts). The modulus operation, with its corresponding modulus value, is determined by the constellation (i.e., via simultaneous transmission of constellations of different sizes for different users for different moduli values). The output of modulus operation block 46 results in a estimated symbol 48, $\hat{s}$.

With reference to the nonlinear precoding scheme 50 of Figure 2C, at the transmitter side, information symbols denoted by a vector s, constitute an input 52 intended for precoding. Figure 2C illustrates a Tomlinson-Harashima precoder. All components of the perturbation vector 53, ***c***, which are not all zero for the nonlinear precoding (NLP) scheme, are added 54 with vector *s* (i.e., which remains the same) and inputted 55 into a nonlinear precoder block 56. The nonlinear precoder (NLP block 56), denoted is given by *inv(R') *D.* NLP block 56 applies an inverse operation to a matrix *R* and calculates a product with a diagonal matrix ***D*** (i.e., *D* is the diagonal of a matrix R^H (*i.e.,* ***D*** = *diag(diag(*R^{H}*))).* Diagonal matrix *D* is used to scale the precoding matrix and is predefined for THP. An output 57 of NLP block 56, ***m*,** is inputted to a block 58, *Q{\.e.,* of the QR decomposition process) which yields a signal or group of signals as precoded information symbols denoted by an output 59, $\mathbf{o} = Q * inv(R^H) * D$, which in turn then is communicated via a communication channel 60 to a receiver side. An output 59 to communication channel (e.g., the communication lines) is denoted by $\boldsymbol{o}$ representing the precoded symbols.

The QR preceding scheme in Figure 2C employs QR decomposition of a channel matrix H which symbolizes a communication channel (denoted by block **60),** mathematically represented by $\boldsymbol{H} = (QR)'$ =- ***R' * Q'.*** Matrix ***Q*** is an orthonormal matrix and J? is an upper triangular matrix. A received vector 65, $\boldsymbol{r}$, at the receiver side is the sum of a vector 62, y, and an additive noise vector 63, $\mathbf{n}$, denoting that the precoded information symbols propagated through the communication channel includes additive 64 noise 63. Scaling is performed independently for every component by an $inv(D)$ block 66, an output 68 of which is provided to a modulus operation block 70. The output of modulus operation block 70 results in a estimated output symbol 72, $\hat{s}$.

in conjunction with Figure 1, reference is now further made to Figures 3A, 3B, and 3C. Figure 3A is a schematic diagram illustrating an overview of a general hybrid-interoperability **precoding** scheme supporting both linear and nonlinear precoding, generally referenced 150A, constructed and operative according to the embodiment of the disclosed technique. Figure 3B is a schematic diagram illustrating an overview of another general hybrid-interoperability precoding scheme including permutations supporting both linear and nonlinear precoding, generally referenced 150B, constructed and operative according to the embodiment of the disclosed technique. Figure 3C is a schematic diagram illustrating an overview of a further general hybrid-interoperability precoding scheme including permutations supporting both linear and nonlinear vector precoding, generally referenced 150C, constructed and operative according to the embodiment of the disclosed technique. Figures 3A, 3B, and 3C illustrate different implementations of hybrid-interoperability precoding constructed and operative in accordance with the principles disclosed technique. Similarly numbered reference numbers in Figures 3A, 3B, and 3C differentiated respectively by suffixes A, B and C represent similar (but not necessarily identical) entities. Since the different implementations shown in Figures 3A, 3B, and 3C are similar in some respects and dissimilar in others, for the purpose of simplifying the explanation of the principles and particulars of the disclosed technique reference is specifically made to general hybrid-interoperability precoding scheme 150A of Figure 3A with further reference being made to Figures 3B and 3C pertaining to differences in implementation.

In hybrid-interoperability precoding scheme 150A information symbols (represented by vector s) are inputted to a reversible mapping block 154A. Reversible mapping block 154A is configured and operative to apply a reversible mapping to the information symbols. A reversible mapping (i.e., a reversible transformation) is defined as a function or algorithm that can be reversed so as to yield the operand that is, the object of the reversible mapping. For example, a reversible mapping is an association between two sets S_1 and S_2 in which to every element of $\mathbf{S}_1$ there is an associated element in S_2, and to every element in S_2 there is the same associated element in *S^.* Reversible mapping block 154A may be implemented by various techniques, such as dirty paper coding techniques of applying modulo operation to the information symbols, vector precoding by the use of a perturbation vector, lattice precoding techniques, and the like.

The reversible mapping includes elements, each of which is respectively associated with a particular one of receivers (CPE units) 106_1, 106_2, 106_3,...,106_{N-1}, 106_N. For example, a reversible mapping is represented by a vector of rank N: $W = \{w_1, w_2, \mathrm{w}_3 \ldots ¼_{\mathrm{N}-1}¾'_{\mathrm{N}}\}$, where each i-th vector element, w_i, is associated with the i-th CPE unit (i.e., index-wise). An example of a simple reversible mapping is an offset function, where information symbol vector s is vector added (i.e., element-wise) with an offset function (vector), i.e.: s + W. For a NLP supporting CPE unit, its associated reversible mapping element (of the reversible mapping (vector)) is offset by a nonzero set value (e.g., an integer value). For example, for CPE unit 108_4, its associated reversible mapping element $w_4 \neq 0$ equals the reversible offset integer. For a CPE unit not supporting NLP, its associated reversible mapping element is zero. For example, if CPE unit 106_5 does not support NLP, its associated reversible mapping element $w_5 = 0$. In general, for those CPE units supporting NLP, their respectively associated (e.g., index-wise) reversible mapping elements (e.g., of a reversible mapping vector) are nonzero, while for those CPE units not supporting NLP, their respectively associated reversible mapping elements are zero.

According to one implementation, hybrid-interoperability precoding scheme 150A employs dirty paper coding techniques in which case the reversible mapping is an offset function typically represented by a vector whose integer elements are nonzero for those NLP-supporting CPE units, and zero for those CPE units not supporting NLP (as exemplified above). According to another implementation, hybrid-interoperability precoding scheme 150A employs vector precoding in which case the reversible mapping is a perturbation vector whose elements are nonzero (i.e., at least one vector element or component is nonzero) for those NLP-supporting CPE units, and zero for those CPE units not supporting NLP. According to yet another implementation, hybrid-interoperability precoding scheme 150A employs lattice techniques for precoding in which symbols are reversibly mapped to lattice points having known boundaries. According to this implementation, lattice precoding (or lattice dirty-paper coding) is employed. Generally, a lattice is a regular arrangement of a set of distanced-apart points (a discrete subgroup of E"(n-dimensional field of rational numbers). A symbol constellation S is a subset of size $\mathbf{2}^{D}$ of a D-dimensional lattice. information symbols are reversibly mapped to the symbol constellation. Lattice precoding involves performing modulo reduction in relation to a precoding lattice A_p into a fundamental (e.g., Voronoi) region $\mathcal{R}(A_p)$ of precoding lattice A_p (practical for a finite number of points). The symbol constellation is an intersection of the lattice symbol space (e.g., $2\mathbb{Z}^2$, where D=2) and $\mathcal{R}$ of precoding lattice A_p, namely $\mathcal{R}(\mathrm{A}_p)$. (A Voronoi region of a lattice is a region having a Voronoi site, where all points are distanced closer to the Voronoi site than to another Voronoi site in the lattice.) The symbol constellation is extended in a periodic manner via addition (or other reversible mapping or transformation) as: $V = S + \boldsymbol{A}_p$ = $\{s + d \backslash s\ E S \mathrm{n}\, \mathcal{R}(A_p),\ d\ E\, \mathrm{A}_p\}$, where each point $v \in V$ is equivalent modulo Λ_p. Thus, symbols propagating via the communication channel ("channel symbols") from transmitter side to receiver side consequently fall **within** $\mathcal{R}(\Lambda_\rho)$. At the receiver side NLP supporting CPE units reverse the reversible mapping by reducing the received signal to the region 7? via the modulo operation: $\nu\iota o\, \acute{\alpha}(\mathrm{A_P})$. Generally, reversible mapping block 154A is configured and operative to apply a reversible mapping to input information symbols 154A as described, and to output a result 156A to a precoding matrix block 158A, *P.*

Precoding matrix block 158A, ***P,*** is configured and operative to perform regularized generalized inversion (hereby denoted *"reg.-gen.-* ***inv.")*** of a channel matrix ***H.*** Regularized generalized inversion is a generalization of general inversion, which in turn is a generalization of inversion. Inversion of a matrix or "matrix inversion" of an invertible matrix ***A*** is a procedure of finding a matrix B such that $AB = \boldsymbol{I}$, where

I represents the n x n identity matrix. In the simplest (and less typical) case where the number of transmitters *(N)* is equal to the number of active receivers *(K)* (i.e., *N = K)* the preceding matrix 158A is reduced to a conventional inverse of the channel matrix. At any point in time, however, there is no assurance that the number of active receivers (users) would be equal to the total number of transmitters. This case can be encapsulated by $N \neq K$. In this case (A/ $\neq$ *K)* channel matrix *H* is **noninvertible** by conventional inversion, as non-square matrices (i.e., *m* x *n*, where *m* $\neq$ *n)* don't have a conventional inverse. (Conventional matrix inversion is limited to regular (non-generate square *n* x *n*, and non-singular) matrices.)

Generalized inversion of the channel matrix *H* essentially involves finding a matrix that serves as an inverse of the channel matrix that is not necessarily invertible. An example of a generalized inverse is generally given by:

$$\mathrm{P} = \mathrm{A}^{H}(\mathrm{H}\mathrm{A}^{H})^{-1} \qquad (1A),$$

where $H \ \varepsilon \ \mathbb{C}^{K\times N}$ and $A \ e\mathbb{C}^{K\times N}$. In the case for channel matrix *H* where: *A = H* an example of generalized inverse can be a pseudoinverse (notation: *pinv(H)},* such as the Moore-Penrose pseudoinverse given by:

$$P = H^{H}(HH^{H})^{-1} \qquad (1B),$$

employed when the total number of active receivers (users), *K*, is less than (also viable when equal to) the total number of transmitters *N*: $K \leq N$ where $H \ \varepsilon \ \mathbb{C}^{K\times \mathrm{N}}$. This case is reduced to the simple case when *K = N* thus we obtain *pinv(H) = inv(If),* i.e., the pseudoinverse is a conventional ("simple") inverse of channel matrix *H*. It is noted that for the *case* $K \leq N$, the construction of the pseudoinverse is not unique. For example, given $\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ there may be found matrices $P_A = \begin{pmatrix} 0.5 & -0.5 \\ 0.5 & -0.5 \\ 0 & 1 \end{pmatrix}$ and $\boldsymbol{P_B} = \begin{pmatrix} 1 & -2 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$ such that $\boldsymbol{HP_A} = \boldsymbol{HP_B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. It is emphasized that the disclosed technique is not restricted b the specific pseudoinverse or to the method of pseudoinverse construction. In general, one may observe that for the case $K < N$ there may be an infinite number of pseudoinverses satisfying the relation $HP = I$, where H ε $\mathbb{C}^{KxN}$, $P \in \mathbb{C}^{N\times K}$ and I is the identity matrix $(K$ x $K)$. To demonstrate this let: $P = A''(HA'')^{-1}$, where M ε $\mathbb{C}^{K\times N}$ and A e C^{KxN} . A is an arbitrary matrix such that the matrix (HA^{H}) whose dimension is K x K is invertible. Invertibility implies:

$$HP = HA''(HA'')\text{-}^{1} = I \qquad (2).$$

It may be seen that an arbitrary matrix A is a generality of a matched filter (for the Moore-Penrose pseudoinverse $A = H$) while the term HA^{H} is a generality of a correlator (for the Moore-Penrose pseudoinverse this term \s HH^{H}), and consequently the term $(HA'')^{-1}$ is a generality of a de-correlator. Since there are an infinite number of possible matrices A $e.$ C^{KxN} , there may be an infinite number of generalized inverses. For the case of $K = N$ the relation of generalized inverse reduces b the conventional inverse matrix: $A''(HA'')^{-1} = A''(A'')^{-1}(H)^{-1} = H^{-1}$. Returning to the above-presented purely illustrative example, for $H = \begin{pmatrix}1 & 1 & 1\\ 0 & 0 & 1\end{pmatrix}$, we obtain P_A by taking $A = \begin{pmatrix}1 & 1 & 1\\ 0 & 0 & 1\end{pmatrix}$, which leads to a particular case of the Moore-Penrose pseudoinverse, and we obtain P_B by taking $A =$ (J $\begin{matrix}1\\ 1\end{matrix}$ J).

Regulanzation of the generalized inverse, known herein as "regularized generalized inversion" involves use of the principles of regulanzation by introducing a regulanzation term to the mathematical expression representing the generalized inverse. A regularized generalized inverse is generally given by:

$$p = A''(HA'' + fn)^{-1} \qquad (3A),$$

where the scalar β≥0 is the regulanzation factor, and 1 is the regulanzation term (in this case, the identity matrix, also interchangeably denoted herein /). A typically employed regularized generalized pseudoinverse of the disclosed technique, *for* $A = H$ is given by:

$$P = H^H(HH^H + \beta\mathbb{1})^{-1} \qquad (3B),$$

The regularization factor controls the impact of the regularization term, and can be selected to optimize (e.g., maximize) the signal-to-interference-plus-noise ratio (SINR) at the receiver side. Note that the case where $\beta=Q$ reduces equation (3) into the Moore-Penrose pseudoinverse of equation (1B). In an analogous manner regarding the selection of the pseudoinverse, the disclosed technique is likewise not limited to a particular regularized generalized inverse. Precoding matrix block 158A is configured and operative to perform precoding in the sense of the disclosed technique, i.e., regularized generalized inversion of the channel matrix according to equation (3B) and to produce an output 160A, denoted by output vector o representing preceded symbols, thenceforward communicated via communication channel 162A to the receiver side.

At the receiver side, a received vector 180A, r , is the sum of a vector 164A, *y*, and an additive noise vector 166A, n, denoting that the precoded information symbols propagated through the communication channel include additive 168A noise 186A. For each one of CPE units 106_1, 106_2, 106_3,..., 106_{N-1}, 106_N (Figure 1) received vector 180A (Figure 3A) can follow one of two different paths: 184A ("a") or 188A ("b"), according to its respective supportability $12Q_1$, 120_2, 120_3,..., 120_N-i,120_N. Specifically, for a CPE unit supporting the decoding of NLP data, received vector 180A follows path 184A and enters an inverse mapping block 186A, otherwise (i.e., CPE unit not supporting the decoding of NLP data) received vector follows path 188; this is illustratively shown by a receiver supportability-dependent pseudo-block 182A (i.e., not a true operational block).

Inverse mapping block 186A is configured and operative to "reverse" the reversible mapping applied at the transmitter side by reversible mapping block 154A. For example, for a reversible mapping that is a function, the inverse mapping is a corresponding inverse function. If hybrid-interoperability precoding scheme 150A employs dirty paper coding and vector precoding techniques, inverse mapping block 186 is typically embodied as a modulo operation block (e.g., a modulus operation applied to an operand (e.g., an integer offset)). In the case hybrid-interoperability precoding scheme 150A employs lattice techniques it is as aforementioned.

For a CPE unit supporting the decoding of NLP data, the output of inverse mapping block 186A (following path 184A **("a"))** results in an estimated output symbol 190A, ***3.*** For a CPE unit not supporting the decoding of NLP data, received vector 180A follows path 188A **("b")** the output of which is an estimated output symbol 190A, $\hat{\boldsymbol{s}}$.

With reference now to hybrid-interoperability precoding scheme 150B of Figure 3B, information symbols 152B are inputted to a reversible mapping block 154B, which is the same as 154A of Figure 3A. Reversible mapping block 154B is configured and operative to apply a reversible mapping to input information symbols 154B and to output a result 156B to a permutation block 157B, which in turn is configured and operative to permute information symbol elements in vector $\boldsymbol{s}$ according to a permutation matrix $\boldsymbol{\Pi}'$. Vector $\boldsymbol{s}$ includes vector elements each of which is associated with a particular CPE unit. The permutation introduces additional degrees of freedom which facilitates the optimization of performance. A particular example permutation that will be hereinafter discussed in greater detail in conjunction with Figures 6 and 7 involves grouping information symbol elements associated with different CPE units according to their respective **supportability** such to form two successive aggregate groups. Alternatively, arbitrary permutations of information symbol elements are also viable.

Precoding matrix block 158B, ***P,*** is configured and operative to perform regularized generalized inversion of a permuted channel matrix ***W*** = $\Pi^T H$, where Π^T represents the transpose of permutation matrix Π and ***H*** represents the channel matrix (in accordance with the principles of equations (3A) and (3B)). The precoding matrix is denoted by ***P*** = reg. - gen. ***-inv(W)D,*** where ***D*** represents a scaling matrix that depends on the permutation used. Precoding matrix block 158B is configured and operative to perform precoding thereby producing an output 160B that is communicated via a communication channel 162B that is represented by a block 162B and given by ***H = TIW,*** to the receiver side.

At the receiver side, a received signal 180B, represented by a vector, *r* , is a sum of a vector 164B, y , and an additive 168B noise vector 166B, *n* (respectively similar to 180A, 164A, and 166A of Figure 3A). Received signal 180B is inputted to a block 181B configured and operative to perform scaling by inverting the permuted diagonal vector $\mathbb{D}$, (i.e., inv($\mathbb{D}$)) where $\mathbb{D} = \Pi D \Pi^T$ and to output a signal to a receiver supportability-dependent pseudo-block 182B (whose operation is identical to 182A). Two paths184B ("a") or 188B ("b"), including inverse mapping block 186B are respectively identical with paths 184A, 188A and inverse mapping block 186A (of Figure 3A). Receiver supportability-dependent pseudo-block 182B outputs an estimated output symbol 190B, $\hat{s}$.

In an alternative optional implementation to hybrid-interoperability precoding scheme 150B, reversible mapping block 154B and permutation block 157B are in reversed order (i.e., information symbols 152 are inputted first to permutation block 157B an output of which is provided to reversible mapping block 154B) (not shown).

According to a different implementation, with reference being made to hybrid-interoperability preceding scheme 150C of Figure 3C, information symbols 152C are inputted to a permutation block 157C configured and operative to permute information symbol elements in vector s according to a permutation matrix Π^T and to output permuted information symbols 153C. Vector s includes vector elements each of which is associated with a particular CPE unit. An adder 155C is configured and operative to combine a vector of permuted information symbols 153C with a perturbation vector 154C, denoted by c, an output 156C of which is inputted to a precoding block 158C. Perturbation vector 154C constitutes as a reversible mapping in the nonlinear vector precoding scheme of Figure 3C. The reversible mapping, in this case implemented by a perturbation vector c includes vector elements each respectively associated with a particular one of the receivers, such that those receivers supporting the decoding of noniineariy precoded data are configured to apply an inverse mapping to the reversible mapping, whereas for i-th vector elements of c associated with receivers not supporting the decoding of noniineariy precoded data information symbol are unaffected by the reversible mapping (i.e., $c_i = 0$).

The following operational blocks and procedures of general hybrid-interoperability precoding scheme 150C in Figure 3C are substantially similar to those respectively in general hybrid-interoperability precoding scheme 150B in Figure 3B. Precoding matrix block 158C, F, is configured and operative to perform regularized generalized inversion of a permuted channel matrix $W = \Pi^T H$, where Π^T represents the transpose of permutation matrix Π and H represents the channel matrix (in accordance with the principles of equations (3A) and (3B)). The precoding matrix is given by $P = \text{reg.-gen.-}inv(W)D$, where D represents a scaling matrix that depends on the permutation used. Precoding matrix block 158C is configured and operative to perform precoding thereby producing an output 160C that is communicated via a communication channel 162B that is represented by a block 162C and given by $\boldsymbol{H} = \boldsymbol{\Pi} \boldsymbol{w}$, to the receiver side.

At the receiver side, a received signal 180C, represented by a vector, r , is a sum of a vector 164C, y , and an additive 168C noise vector 166C, n (respectively similar to 180B, 164B, and 166B of Figure 3B). Received signal 180C is inputted to a block 181C configured and operative to perform scaling by inverting the permuted diagonal vector $\mathbb{D}$), (i.e., inv(O)) where $\mathbb{D} = \boldsymbol{\Pi} \mathbf{D} \boldsymbol{\Pi}^T$) and to output a signal to a receiver supportability-dependent pseudo-block 182C (whose operation is identical to 182B). Two paths184C ("a") or 188C ("b"), including inverse mapping block 186C are respectively identical with paths 184B, 188B and inverse mapping block 186B (of Figure 3B). Receiver supportability-dependent pseudo-block 182C outputs an estimated output symbol 190C, $\hat{s}$.

To explicate the particulars of the disclosed technique in greater detail, an example of a specific implementation will now be described. Reference is now made to Figures 4 and 5. Figure 4 is a schematic diagram illustrating an example of a specific implementation of the general hybrid-interoperability precoding scheme, utilizing a QR nonlinear precoder and permutations, generally referenced 200. Figure 5 is a schematic illustration of a partition of vector variables into two groups, linear precoding (LP) and nonlinear precoding (NLP), generally referenced 250. The example specific implementation of the disclosed technique involves dividing users (CPE units) into two groups at every frequency tone (given, without loss of generality the use of orthogonal frequency-division multiplexing (OFDM) transmission or flat fading environment): those which use LP (linear precoder) and those which use NLP (nonlinear precoder). We herein denote information symbols for these groups s_1 and s_2, respectively. The following description discusses the disclosed technique in the context of one particular subcarrier frequency (interchangeably herein "frequency tone" or simply "tone"). Generally, the partition of users into different groups (linear and nonlinear) may differ at different frequency tones (e.g., in consideration with performance optimization issues). It is emphasized that the general precoding scheme according to the disclosed technique involving NLP supporting CPE units and NLP non-supporting CPE units, does not necessitate any specific ordering of the CPE units (e.g., general hybrid-interoperability precoding scheme 150A, Figure 3A).

As an example to the dynamic nature of this division, users with relatively small constellation sizes are assigned to a first group (e.g., 1 or 2 bits) employing LP, while remaining users of relatively larger constellation sizes are assigned to a second group employing NLP. Alternatively, constellations are not assigned for those users having a low signal-to-noise ratio (SNR) thus not allowing the loading of constellations of larger size than the predetermined value (e.g., the bit loading is less than one bit). This state effectively corresponds to the case where $\kappa < N$. The precoder employed in this case is determined according to the regularized generalized inverse of the channel matrix (utilizing communication channels of inactive CPE units at a specific tone). Noteworthy also in this case is the analogous wireless implementation where the number of transmitting antennas is greater than the number of receiving antennas.

The dynamic division of users into groups involves allocating: group 1 for users employing LP and having no supportability for inverse mapping (e.g., modulo operation capable) functionality (e.g., hardware, software, firmware) or having such functionality but preferably using linear precoding due to optimization issues; and group 2 for users employing NLP having supportability for inverse mapping functionality. This dynamic division merges system optimization (i.e., according to different criteria, such as, max-mean rate, max-min rate, etc.) with resolution of the interoperability issues. This issue is elaborated further hereinbelow. In the description that follows, the operation per frequency tone is described, assuming principles of an OFDM system.

With reference to Figure 4, implementation of general hybrid-interoperability preceding scheme 200 initiates by information symbols 202, $\boldsymbol{s}$, being inputted into a permutation block 204 , which is configured and operative to permute information symbol elements in vector s, according to a permutation matrix Π'. Each information symbol element in vector $\boldsymbol{S}$ is associated with a particular CPE unit. The permutation takes into consideration the division of CPE units into two groups (those that employ LP and those that employ NLP). In its general form, the permutation arbitrarily intermixes between information symbol elements. Alternatively, permutation block 204 permutes the information symbol elements into two successive aggregate groups, namely, a first aggregate group of symbol elements $\boldsymbol{S}_1$ successively followed by a second aggregate group of symbol elements s_2. These information symbols are described in terms of a complex vector per user, represented as:

$$\boldsymbol{S} = [\boldsymbol{S}_1^T . \boldsymbol{S}_2^T]^T \text{, where } \boldsymbol{S}_1 \; e \; C^{K_1\times 1}, s_2 \; e \; C^{K_2\times 1}, K_1 + K_2 = K \qquad (4).$$

With further reference to Figure 5, which shows different vector variables (shown also in Figure 4), whose vector elements are partitioned into two distinct and separate aggregate groups, where each group is associated with either one of NLP supporting CPE units (a "NLP group" and NLP non-supporting CPE units (a "LP group"). Specifically, Figure 5 illustrates the partitioning of vectors $\boldsymbol{c}, \boldsymbol{s}, \boldsymbol{m}, y, r$ (which are all complex vectors of $\mathbb{C}^{(K_1+3/4)\times 1}$) into two groups of sizes K_1 and K_2. Partitioning of a vector into groups means that a vector's elements (also known as components) are allocated to these groups. Their first K_1 components are associated with the LP group (i.e., the group of CPE units employing LP and not supporting NLP), whilst the second remaining K_2 components are related to the NLP group (i.e., the group of CPE units supporting NLP).

To elucidate the role of permutations in the system and method of the disclosed technique, reference is now further made to Figure 6, which is a schematic diagram illustrating an example permutation configuration in the specific implementation of the general hybrid-interoperability **precoding** scheme of Figure 4, generally referenced 300. Figure 6 shows a particular example of a permutation configuration for five users. Permutation block 304 (Figure 6) is a particular example of permutation block 204 (Figure 4), in which there are a total of five CPE units (users): {1,2,3,4,5} (in "natural ordering") where the group of users {1,3,5} are the LP group and group of users {2,4} are the NLP group. Similarly to permutation block 204 (Figure 4), permutation block 304 (Figure 6) is configured and operative to permute CPE units (users) into two aggregate groups according to supportability, e.g.: {{1 ,3,5},{2,4}}.

Further detail of permutation block 304 is described in conjunction with Figure 7, which is a schematic diagram illustrating an example configuration of an internal structure of the permutation block of Figure 6. Permutation block 304 includes two permutation stages: a stage 1 referenced 330 and a stage 2 referenced 332. Input Information symbols 302,, 302_2, 302_3, 302_4, and 302_5 (collectively denoted "302^") are inputted into permutation block 304 (specifically, into stage 1). As shown in Figure 7, input information symbols s(1), s(2), s(3), s(4), s(5) are in the natural ordering. Stage 1 330 is configured and operative b divide the input information symbols into two aggregate groups: the LP group and the NLP group.

Generally, an arbitrary permutation of information symbol elements is achieved by permuting their indices via a permutation matrix n $_0$, i.e.:

$$[\boldsymbol{s}_1^{\mathrm{T}}, \boldsymbol{s}_2^{\mathrm{T}}]^{\mathrm{T}} = \boldsymbol{\Pi}_0^{\mathrm{T}} \boldsymbol{s} \quad (5).$$

The example given in Figures 6 and 7, shows 5 users of whom the group of users {2,4} (i.e., CPE units **106_2, 106_4)** support the decoding of NLP data, while the group of users {1,3,5} (i.e., CPE units **106i,** 106_3, 106_5) don't support the decoding of NLP data (i.e., the LP group of users). The permutation Π_0^T permutes the natural ordered list of users 1 through 5 {1,2,3,4,5} into two successive aggregate groups of different supportabi!ities, **thusly:** {1,3,5,2,4} (i.e., **{{LP},{NLP}}).** Stage 2 includes permutation blocks 334 and 336. In stage 2 each of these two aggregate groups of users may be further permuted within each group by employing permutation blocks 334 and 336 respectively implementing permutation matrices Π_1^T and Π_2^T, namely:

$$\mathbf{s}_1^{(\Pi_1)} = \Pi_1^T s_1 \text{ and } \mathbf{s}_2^{(\Pi_2)} = \Pi_2^T s_2 \qquad (6).$$

The following short notations are hereby introduced: $\mathbf{x}_1 = s_1^{(\Pi_1)}$ and $\chi_2 = \mathbf{s}_2^{(\Pi_2)}$. In general, the permutation matrix corresponding to the operation of permutation block 204 (Figure 4) and by permutation block 304 (in the particular example shown in Figure 6) is:

$$\Pi_{K\times K} = \Pi_0 \begin{pmatrix} \Pi_1 & 0_{K_1\times K_2} \\ 0_{K_2\times K_1} & \Pi_2 \end{pmatrix} \qquad (7).$$

Permutation block 204 (Figure 4) outputs a signal 206 (i.e., permuted information symbols represented in Figure 7 as a vector x = $[\chi(1), \chi(2), \chi(3), \chi(4), \chi(5)]^T$ in the 5-user example) in which the outputted permuted symbols can generally be represented by:

$$]\mathbf{x}_1, x_2]^T = \left[s_1^{(\Pi_1)}, s_2^{(\Pi_2)}\right]^T = \Pi^T x = \begin{pmatrix} \Pi_1^T & 0_{..\times K_2} \\ 0_{K_2\times K_1} & \Pi_2^T \end{pmatrix} \Pi_0^T x \qquad (8).$$

Output signal 206 (Figure 4) having permuted information symbols (Figure 7) are added 210 with a perturbation vector 208, c. Perturbation vector 208, c, for the first K_1 components, denoted as c_1 are zeroes: $c_1 = 0$, signifying that no perturbation is added 210 to the K_1 components of output vector 206. The remaining K_2 components, are denoted as $c_2 \neq 0$, signifying that a perturbation value is added 210 b the

K_2 components output vector 206. Vector c is defined by the following expressions: c = $[c_1^T, c_2^T]^T$, $c_1 = 0$ means $0^{K_1 \times 1}$, $c_2 \in C^{K_2 \times 1}$ where $K_1 + K_2 = K$. Vector x is defined by the following expressions x = [xl. xl $]^T$, $x_1 \in C^{K_1 \times 1}$, x_2 ε $C^{K_2 \times 1}$, where $\kappa_1 + K_2 = K$.

The perturbation vector, which can be any addition that may be eliminated by modulo arithmetic operations at the receiver, is one particular example of a reversible mapping. The employment of the perturbation vector approach effectively connotes that for every k-th index the point $(s_k + c_k)$ belongs to an expanded constellation set for the k-th user. Generally, various algorithms may be used to determine the perturbation vector. Particularly, perturbation vector c is attained by taking into account system constraints that serve several objectives. An example objective involves minimizing the output power of the precoder signal (thus aiding to avoid intensive pre-scaling of the transmitted signal, consequently increasing the SNR at the receiver). Another example objective concerns the minimization of the bit error rate (BER) of the received signal.

Alternatively, the permutation matrix is applied directly to the vector sum of the symbol vector and the perturbation vector, $s + c$, since $c + x = c + \Pi^T s = \Pi^T(s + \Pi c) = \Pi^T(s + c)$, where $\tilde{c}$ represents an auxiliary perturbation vector. The perturbation vector and auxiliary perturbation vector are related thusly: $\tilde{c} = c^{(\Pi)} = \Pi c$. Arranging c as $c = [c_1^T, c_2^I Y$ we obtain:

$$\tilde{c} = \Pi[c_1^T, c_2^T]^T \qquad (9).$$

Perturbation vector c_1, whose components are all zero, is added to s_1. $c_1^{(\Pi_1)}$ is also a zero vector. Obtaining an implicit expression to perturbation vector c in terms of the auxiliary perturbation vector $\tilde{c}$ (by multiplying the auxiliary vector by Π^T from the left and using the identity $\Pi^T\Pi$) yields: $c = \Pi^T\tilde{c}$. An output 212 (Figure 4) from adding operation 210, $x + c$, constitutes as an input to a block 214 configured and operative to perform the pseudo-inversion operation: $pinv(R')*D$, and to output a signal 218 (Figure 4) represented by a vector m referenced 256 in Figure 5. In the 5-user example of Figure 6, a pre-processing block 306 performs the same operation as block 214 (note that $L = R'$). Outputted signal 216, represented by vector $\boldsymbol{m}$ is partitioned (Figure 5) into two sub-vectors 256_1 and 256_2 of vector 256 corresponding to two aggregate groups of vector elements, namely, a first aggregate group, sub-vector $\pi\iota_1$, successively followed by a second aggregate group, sub-vector $\boldsymbol{m}_2$. Block 214 (Figure 4) in conjunction with a Q block 218, a permutation block 204, and a perturbation vector 208, $\boldsymbol{c}$, collectively constitute a precoder 220, denoted by $\boldsymbol{P}$, which in turn is configured and operative to perform pseudo-inversion of a communication channel matrix $\boldsymbol{H}$, in accordance with a ZF (zero-forcing) scheme involving permutations. Analogously, in the 5-user example of Figure 6, permutation block 304, a pre-processing block 306, an orthonormal matrix "Q" block 308, and a plurality of scalar power gain blocks $310_{1\text{-}5}$ collectively constitute a precoder 312, which in turn is configured and operative to perform inversion of a communication channel 314.

Block 214 (Figure 4) and pre-processing block 306 (in the 5-user example in Figure 6) utilize a diagonal matrix $D \varepsilon C^{K\times K}$ (i.e., $D = diagid))$ configured and operative to perform power scaling for precoder 220 (e.g., for meeting power constraints). Vector d is defined by the following expressions: $d \tilde{-}\text{-} [d_1^T, d_2^T]^T$, where $\boldsymbol{d}_1$ e $C^{K_1\times 1}$, and d_2 £ $C^{K_2\times 1}$, and where $K_1 + K_2 = K$. Q block 218 (Figure 4) and a Q block 308 (in the 5-user example in Figure 6) are configured to perform the following QR decomposition for a Hermitian conjugated and permuted channel according to:

$$\boldsymbol{H}'' = QR\Pi^T \quad \text{hence} \quad \boldsymbol{H} = \Gamma I \boldsymbol{R}^H \boldsymbol{Q}^H \qquad (10),$$

signifying that the QR decomposition is constructed from:

$$H^{H}\Pi = QR \qquad (11),$$

where H e $C^{K\times N}$ is the channel matrix, R e $C^{K\times K}$ is an upper triangular matrix, Q £ $C^{N\times K}$ is an orthogonal matrix such that $Q^{H}Q = I_{K\times K}$, and Π e C^{KxK} is a permutation matrix. The multiplication expression ΠX permutes the rows of a matrix X; the multiplication expression $X\Pi^{T}$ permutes the columns of matrix X. In particular, the determination of matrix R in the QR factorization process depends not only on the channel matrix but also on the permutation matrix Π. Permutation matrix Π consists of K elements, such that all elements but one is zero and just one element is equal to 1, for each row and each column of Π. The position (i.e., index value) of this element (the **1)** is different for every row and column so that these positions determine the permutation sequence. With reference to the 5-user example of Figures 6 and 7, let us consider the following a 5-dimensional vector (1,2,3,4,5) and its **5-element** permutation into {2,4,1 ,3,5}, represented by the following matrix:

$$\Pi = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \Pi \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} a_2 \\ a_4 \\ a_1 \\ a_3 \\ a_5 \end{bmatrix} \qquad (12).$$

In particular, this also demonstrates an example of:

$$\Pi = \begin{pmatrix} \Pi_1 & 0_2 \\ 0_{3\times 2} & \Pi r \end{pmatrix} \qquad (13),$$

where the 2 x 2 permutation matrix Π_1 permutes the first two indices and the 3 x 3 permutation matrix Π_2 permutes last three indices.

The construction of precoder a P represented by precoder matrix $P \in C^{N\times K}$ (referenced 220 in Figure 4 and 312 in Figure 6) is achieved according to the following equations:

$$P = aQR^{-H}D\Pi^{T}, \varnothing = P(s + c), \qquad (14).$$

$P \in \mathbb{C}^{N\times K}$ is a generalized vector precoder matrix, s is the information symbol vector and c is the perturbation vector. For the LP group of CPE units the respective components of c are zero, while for the NLP group of

CPE units the respective components of $\mathbf{c}$ are nonzero. Alternativeiy, there might be NLP supporting CPE units whose respective components of c are constrained (forced) to be zero for the purposes of optimization (e.g., dimensionality reduction in decreasing the number of nonzero components of c, thus reducing computational load), effectively treating NLP supporting CPE units as part of the LP group of CPE units. The perturbation vector is added for the NLP group of CPE units for example via search-based criteria, implicitly via the THP scheme, and the like.

Scalar power gain blocks **310**$_1$, **31 0$_2$, 31 0$_3$,** 3 10$_4$, and 3 10$_5$, (collectively denoted 310$_{1\_5}$) (Figure 6) apply a scalar α factor, which is common to all users (may be used optionally), matrices $\boldsymbol{Q}$ and R are determined according to equation (1 1), and matrix D is used for power scaling. Specifically, matrix R^H is a lower triangular matrix hereby denoted interchangeably ***by** $\boldsymbol{L}$,* (i.e., $\boldsymbol{L} = R^H$). The input to block 218 in Figure 4 and block 308 in Figure 6, matrix $\boldsymbol{Q}$, is given by:

$$_m = R^{\sim H}Dil\ ^{\mathrm{T}}(s + \mathbf{c}) = L^{-1}Dn^{\mathrm{T}}(s + \mathbf{c}) \qquad (16),$$

which means that:

$$\boldsymbol{Lm} = \boldsymbol{D}\begin{pmatrix} \boldsymbol{\Pi}_1^T & \boldsymbol{0}_{K_1\times K_2} \\ \boldsymbol{0}_{K_2\times K_1} & \boldsymbol{\Pi}_2^T \end{pmatrix}\boldsymbol{\Pi}_0^T(\boldsymbol{s}+\boldsymbol{c}) = \boldsymbol{D}\left[\left(\mathbf{s}_1^{(\Pi_1)}\right)^T, \left(\mathbf{s}_2^{(\Pi_2)}\right)^{\mathrm{T}} + \boldsymbol{c}_2^T\right]^T \qquad (17).$$

Since $\boldsymbol{L}$ is lower diagonal, for the LP group we obtain:

$$\boldsymbol{L}\mathbf{i}\boldsymbol{m}_1 = \boldsymbol{D}_1\mathrm{s}_1^{(\Pi_1)} \qquad (18),$$

or equivalently,

$$\boldsymbol{m_1} = \boldsymbol{L}_1^{-1}\boldsymbol{D_1}\boldsymbol{s}_1^{(\Pi l_1)} \qquad (18^*),$$

where $D_1 = \boldsymbol{diag(d_1)}$.

Reference is now further made b Figure 8, which is a schematic illustration, generally referenced 350, detailing a partition of a lower-diagonal matrix $\boldsymbol{L}$ having the dimensions (¾ + K_2) x (K_1 + K_2) into three matrices L_1, $\boldsymbol{L_2}$, and M$_{i\ 3}$, and a zero matrix $\boldsymbol{0}_{\boldsymbol{KlxK2}}$ having zero-valued elements. Matrices L_1^{-1}, $\boldsymbol{L}_1$, $\boldsymbol{D}_1$, U_1 are of size K_1 x K_1 and vectors m_1, $\mathbf{s}_1^{(\Pi_1>}$, $\boldsymbol{d}_1$ are of dimension K_1 x 1. Note that permutation matrix Π_1 has only K_1 non-zero elements and is defined by the vector of size K_1 specifying the permutations. Matrix D_1 has K_1 non-zero elements on its principal diagonal. Matrix L_1^{-1} is a lower diagonal matrix whose inverse is matrix L_1, the latter of which is also a lower diagonal matrix consisting of K_1 first rows of matrix L.

Diagonal scaling vector d_1 can represent the degrees of freedom for optimization under constraints. Optimization involves use of the diagonal scaling vector d_1 in conjunction with the degrees of freedom afforded by the permutation. The elements of the diagonal (power) scaling vector d ($D = diag(d)$) are typically selected to be real-valued and non-negative (although it is not a required restriction). It is noted that different precoder outputs may be scaled by gains of different values (not shown in Figure 6). The corresponding degrees of freedom rendered by the diagonal of the corresponding components of matrix D are chosen to satisfy and ensure that power constraints are met per communication channel (e.g., line, antenna), as well as to optimize the performance (e.g. affect communication rates: mean or max-min (maximum-minimum) or max-min with constraints etc). For example, the optimization, may seek maximization of the average rate, the max-min rate, or alternatively other multi-criterion optimizations. Typical constraints include the power density mask (PSD), and the minimal bit loading value.

Following determination of diagonal scaling vector d_1, vector m_1 is calculated directly via equation (18*), or sequentially via equation (18). The direct path allows calculation of the components of ??¾ independently from each other, thus allowing for parallel processing to be employed.

The NLP scheme for the NLP group of CPE units (users) will now be described in greater detail. Reference is now made to Figure 9, which is a schematic illustration detailing an example configuration, of an internal structure of the preprocessing block in Figure 6, generally referenced 370. Figure 9 shows an internal structure of preprocessing block 306 in Figure 8, for the 5-user example. Also shown is permutation block 304 of Figures 6 and 7, configured and operative to provide permutated information symbols (vector ***x)*** as input to preprocessing block 306. Preprocessing block 306 includes a plurality of gain blocks 372_1, 372_2, 372_3, 372_4, and 372_5, a plurality of NLP/LP control mechanisms 374_1, 374_2, 374_3, 374_4, and 374_5, a plurality of adders 376_2, 376_3, 376_4, and 376_5, and a plurality of modular arithmetic calculation units **378i,** 378_2, 378_3, 378_4, and 378_5. Permuted information symbols ***x*** = {x(l),x(2),x(3), x(4), x(5)} are respectively inputted into gain blocks 372-, , 372_2, 372_3, 372_4, and 372_5 (index-wise). The gain blocks are configured and operative to multiply the permuted information symbols by respective gain components /(l), /(2), /'(3), $f(4)$, and $f(5)$ of a gain vector / . NLP/LP control mechanisms are each 374_1, 374_2, 374_3, 374_4, and 374_5 are configured and operative to control application of reversible mapping 154A (Figure 3), for example, a modulo-Z adder (where Z ***e*** $\mathbb{Z}$***,*** i.e., a whole number), according to the respective supportabilities of the CPE units. For the LP group of users, NLP/LP control mechanisms 374·|, 374_2, 374_3, 374_4, and 374_5 are configured and operative to direct relevant input signals in(1), **in(2), in(3), in(4), in(5)** to the output of preprocessing block 306, thereby respectively bypassing modular arithmetic calculation units 378!, 378_2, **378_3,** 378_4, and **378_5.** For the NLP group of users, NLP/LP control mechanisms 374-j, 374_2, 374_3, 374_4, and 374_5 are configured and operative to direct relevant input signals **in(1), in(2), in(3), in(4),** in(5) respectively to modular arithmetic calculation units **378i,** 378_2, 378_3, 378_4, and 378_5.

The operation of adders 376_2, 376_3, 376_4, and 376_5, and that of modular arithmetic calculation units 378_1, 378_2, 378_3, 378_4, and 378_5, which are relevant to the group of NLP users, will now be described in detail. Analogously to equation (18) relating to the LP group of users, for all LP and NLP groups of users we have:

$$\boldsymbol{Lm = D\Pi(s + c)} \qquad (19).$$

Referencing Figure 8, which shows a partition of the lower-diagonal $\boldsymbol{L}$ matrix into (sub)matrices, where the lower-diagonal matrix is denoted $\boldsymbol{L}_1$, the rectangular matrix is denoted $\boldsymbol{M}_{12}$, and the lower-diagonal is denoted $\boldsymbol{L_2}$. All elements above the main diagonal are zeroes. The dimensions of $\boldsymbol{L}_1$, $\boldsymbol{M}_{12}$ and $\boldsymbol{L_2}$ are $\boldsymbol{K}_1$ x $\boldsymbol{K}_1$, $\boldsymbol{K}_2\ \chi\ \mathrm{K}_1$, and $\boldsymbol{K}_2\ x\ \boldsymbol{K}_2$ respectively. From equation (19) we obtain:

$$\boldsymbol{l}_2\boldsymbol{m}_2 + \boldsymbol{M}_{21\mathrm{0}\ 1} = \boldsymbol{D}_{\mathbf{2}}\mathbf{nJ}(\mathbf{s_2} + \mathrm{c}_2) \qquad (20).$$

The following notations are hereby defined: $\boldsymbol{D}_2 = \mathrm{diag}(\mathrm{d}_2)$, where $\boldsymbol{d}_2 = \mathrm{diag}(£_2)$. The objective now is to construct a unit diagonal matrix $\boldsymbol{L}_U$:

$$\boldsymbol{L}_U = \boldsymbol{Dl}^{1}\boldsymbol{L} \qquad (21),$$

where $\boldsymbol{D}_L = \boldsymbol{diag}(\boldsymbol{d}_L)$, and $\boldsymbol{d}_L = \boldsymbol{[diag(L_1)^T, diag(L_2)^T]^T}$. The modular arithmetic calculation units are configured and operative to facilitate construction of the diagonal matrix $\boldsymbol{L}_U$ in a recursively manner according to: $\boldsymbol{L}_U(\boldsymbol{k}, \boldsymbol{n}) = \frac{L(k,n)}{L(k,k)}$.

Particularly, multiplication of equation (17) the left side by $\boldsymbol{D}_L^{-1}$ results in:

$$\boldsymbol{L}_U\boldsymbol{m} = \left[\left(\boldsymbol{f}_{(1)} \odot \boldsymbol{s}_1^{(\Pi_1)}\right)^T, \boldsymbol{s}_2^{(\Pi_2)\mathrm{T}} + \boldsymbol{c}_2^T\right]^T \qquad (22),$$

where Q denotes the component-wise multiplication, and / is a gain vector calculated as according to:

$$\mathrm{di}a\mathrm{g}(\boldsymbol{f}_{(1)}) = \wedge\mathbf{diag(diag(L_1))}\Big)^{-1}\boldsymbol{D}_1 \qquad (23),$$

hence:

$$f_1(k) = {}^{-}_{l}\text{¾¾} = \frac{d_1(k)}{L(k,k)} \ \blacksquare\ \text{ɒ}^{r} * = 1, \dots, \text{¾} \qquad (24).$$

According to equation (20) the first $\boldsymbol{a}$ x $\boldsymbol{a}$ elements of the scaled matrix:

$$\boldsymbol{L}_1 = \mathbf{diag(diag(I}_1\mathbf{))L}_{u1} \qquad (25).$$

Thus:

$$L_1^{-1} = L_u^{-}\backslash \wedge\text{diag}(\text{diag}(\mathrm{L}_1))\Big)^{-1} \qquad \textbf{(26).}$$

By using equation **(23)** we may rewrite equation **(18*)** as:

$$m_1 = L_1^{-1}D_1\mathrm{s}_1^{\mathrm{i,1,)}} = L_{U1}^{-1}\,(\text{diag}(\text{diag}\,(L\mathrm{i}))\Big)^{-1} D_I S_1^{(\Pi_1)} \qquad \textbf{(27),}$$

then obtain an expression for m_1 by means **of** the scaled matrix, L_{u1}^{-1}, and gain vector f:

$$\mathbf{a}_1 = L_{U1}^{-1}\,\text{diag}(f_{(1)})\,\mathrm{s}_1^{(\mathrm{i}\Pi_1)} \qquad \textbf{(28),}$$

(which is an alternative expression for $m^\wedge$.

We observe that a sequential solution of the system of equations $L_U m - b + \tilde{c}$ can be given by:

$$m(\mathbf{l}) = h(l) + \mathrm{c(l)}$$

$$m(\mathbf{fc}) = b(k) + \tilde{c}(k) - \textstyle\sum_{n=1}^{k-1} L\mathbf{y}(\mathit{l}\mathrm{c}, \mathbf{n})\mathbf{m(n)} \quad \text{for } k = 2,\dots,K \qquad \textbf{(29),}$$

where we denote:

$$b = \text{diag}(f)\Pi\mathrm{s} = \text{diag}(f)\,\mathrm{s}^{(\Pi)} \qquad \textbf{(30),}$$

and discuss the following general scheme derived from equation **(19):**

$$l_u m = \text{diag}(f)\,\mathrm{s}^{(\Pi)} + \tilde{c} \qquad \textbf{(31).}$$

From equations **(29)** and **(30)** we arrive at:

$$\mathbf{m(l)} = f(\mathrm{l})\mathrm{s}^{(\Pi)}(\mathrm{l}) + c(V)$$

$$ni(k) = /\,(k)\mathbf{s}^{(\mathbf{n})}(k) + c(k) - \textstyle\sum_{n=1}^{k-1} L_U(k,n)m_2(n)\ ,\ \textbf{for } k = 2,\dots,K \qquad \textbf{(32).}$$

For any index k we may choose an auxiliary perturbation vector c(fc) to be proportional to the modulus size such that:

$$m(\mathrm{fc}) = mod\{f(k)s^{(\Pi}\langle k) - \textstyle\sum_{n=1}^{k-1} L_U(k,n)m_2(n)) \qquad \textbf{(33).}$$

Reference is now made to Figure **10**, which is a schematic illustration detailing an example configuration of an internal structure of preprocessing block and permutation block in Figure **6**, shown in a particular **5**-user configuration, generally referenced **400.** Figure **10** shows an example internal structure of permutation block **304** of Figure **6**, including permutation sub-blocks **334** and **336** (Figure **7).** As described in conjunction with Figure **7**, permutation sub-blocks **334** and **336** each respectively permute the two aggregate groups LP and NLP of users. Preprocessing block 306 employs equation (22) for the NLP group of users such that:

(1) The first $\boldsymbol{K}_1$ permuted information symbols $s_1^{(\Pi_1)}(k)$ (i.e., $\mathbf{x}(1)$, $\mathbf{x}(2)$, $\mathbf{x}(3)$) are inputted to preprocessing block 306 and gain blocks 372-, , 372_2, 372_3 apply (e.g., multiply by) gain vector components $f_{(1)}(k)$ i.e., as $f_{(1)}(k)s_1^{(\Pi_1)}(k)$ (i.e., $\mathbf{f}(1)$, $\mathbf{f}(2)$, $\mathbf{f}(3)$). The rest of $\boldsymbol{K_2}$ permuted information symbols, corresponding to the NLP users enter directly as $s_2^{(\Pi_2)}(k)$ (i.e., gain blocks 372_4, and 372_5 don't apply gains, i.e., $\mathbf{f}(4) = 1$, $\mathbf{f}(5) = 1$). For the LP group of users, outputs from gain blocks 372i, 372_2, 372_3 correspond to respective inputs in(1), **in**(2), **in**(**3**) to respective modular arithmetic calculation units 378i, 378_2, 378_3 (i.e., effectively bypassed), which in turn correspond to respective outputs **m**(**1**), **m**(2), **m**(3). For the NLP group of users, outputs from gain blocks 372_4, 372_5 correspond to respective inputs in(4), **in**(**5**) to respective modular arithmetic calculation units 378_4, 378_5, which **in** turn correspond to respective outputs **m**(4), and **m**(5).

(2) NLP/LP control mechanisms 374_1, 374_2, 374_3, control the application of modulus operations, which are not applied for the A_1 first equations corresponding to the LP users. NLP/LP control mechanisms 374_4 and 374_5 control the application of modulus operations applied to the subsequent $\boldsymbol{K_2}$ equations corresponding to the NLP users. The perturbation vectors are implicitly calculated via modulus operations according to:

$$m\eta(1) = \boldsymbol{l}_{(\mathbf{1})}(\mathbf{l})\ s_{\mathbf{l}}^{(\mathbf{n}_1)}(\mathbf{l}) \quad (34)$$

$$m(k) = f_{(1)}(k)s_1^{(\Pi_1)}(k) - \textstyle\sum_{n=1}^{k-1} L_U(k, n)m(\mathrm{n}), \text{for } \mathrm{e} = 2, \ldots, A_1 \quad (35)$$

$$\boldsymbol{m}(k) = \mathrm{mod}\left(S_2^{(\Pi_2)}(k)\right) - \textstyle\sum_{n=1}^{*-1} L_U(k, \mathrm{n})\mathrm{m}(\mathrm{n})) \quad \boldsymbol{, io}\mathrm{r}\ \boldsymbol{k} = \boldsymbol{K}_1 + 1, \boldsymbol{\ldots}, \boldsymbol{K_2}$$

(36).

The gains, represented by vector / , are calculated according to: f(k)=d_1(k)/L(k,k), where d_1 represents (pre-calculated) diagonal coefficients for the linear precoder.

The presented scheme is a convenient way to embed the LP and NLP groups of users together into the THP structure, concomitantly with NLP/LP control mechanisms configured and operative to respectively apply modulus arithmetic operations to groups of users classified according to supportability. The implementation of permutations by permutation block 304 is achieved via equation (8) such that the permutation block matrix can be represented in the form of $\begin{pmatrix}\mathbf{\Pi}_1^T & \mathbf{0}_{K_1\times K_2}\\ \mathbf{0}_{K_2\times K_1} & \mathbf{\Pi}_2^T\end{pmatrix}$ acting on $[S_1^T, s_2^T]^T$ as its input. Alternatively, the permutation block may also include a matrix Ħj (see equation (8)) acting on the natural order of information symbols, which can be represented by: $\begin{pmatrix}\mathbf{\Pi}_1^T & \mathbf{0}_{K_1\times K_2}\\ \mathbf{0}_{K_2\times K_1} & \mathbf{\Pi}_2^T\end{pmatrix}\mathbf{\Pi}_0^T$.

There are various implementations to preprocessing block 306 (Figure 6), several of which are hereby given by the following examples. Reference is now further made to Figure 11, which is a schematic illustration showing a particular implementation of NLP/LP control mechanisms in an internal structure of preprocessing block, generally referenced 410. Particular implementation 410 of preprocessing block 306 is the same as particular implementation 400 of processing block 306, apart from Figure 11 showing another particular implementation of NLP/LP control mechanisms. Specifically, NLP/LP control mechanisms 374_1, 374_2, 374_3, 374_4, and 374_5 are implemented as electronic switches configured and operative to route input signals in(1), sn(2), in(3), in(4), in(5) either to respective modular arithmetic calculation units 378i, 378_2, 378_3, 378_4, and 378_5, or directly to respective outputs m(1), m(2), m(3), m(4), m(5), thereby controlling the application of the modulo operation (i.e., the reversible mapping) according to the respective supportabilities of the CPE units.

In accordance with another example implementation of preprocessing block 306, reference is now further made to Figure 12, which is a schematic illustration showing another particular implementation of the preprocessing block of Figure 6, generally referenced 420. Figure 12 is similar to Figure 10, apart from implementation of a calculation block 422 for the LP group of users. For those LP group of users, permuted information symbols x(l), *x(2),* and x(3) are inputted into calculation block 422, which is configured and operative to calculate $\mathrm{inv}(L_{u1})$ * diag(f), (i.e., using the scaled to unit diagonal $\boldsymbol{L_u}$ matrix and the scaled gains /') the respective outputs of which are m(l), m(2), and $m(3)$. For the NLP group of users, modular arithmetic calculation units 378_4, and 378_5 are used. Figure 12 shows a combination of a united direct non-sequential action for the LP group of users, in conjunction with a sequential THP preprocessing action for the NLP group of users. Alternatively, according ಶ another implementation (not shown), calculation block 422, is configured and operative to calculate: $\mathrm{inv}(\mathbf{L}_1)$ * D_1. For both implementations, it is emphasized that only $m(k)$ for the NLP group of users is calculated sequentially. For the LP group of users, m(/c) is calculated either sequentially or directly according to equation (18*), namely: $\boldsymbol{m}_1 = L_1^{-1} D_1 \mathbf{s}_1^{(\Pi_1)}$ or alternatively, by opting to use the scaled matrix: $\boldsymbol{m}_1 = \boldsymbol{L}_{\tilde{u}1}^{-1}\,\mathrm{diag}(/\,_{(1)})\,\mathrm{s}_1^{(\Pi_1)}$.

In accordance with another example implementation of preprocessing block 306, reference is now further made to Figure 13, which is a schematic illustration showing further particular implementation of the preprocessing block of Figure 6, generally referenced 430. Figure 13 is generally similar to Figure 12, apart from a different configuration of modular arithmetic calculation units 378_4, and 378_5 and the inclusion of a calculation block 432 that includes a plurality of recursive calculation blocks and adders 434 and 436. The example implementation shown in Figure 13 is configured and operative to reduce the real-time precoding complexity, due to preprocessing block 306 requiring the execution of fewer computations involving the sequential part of the computation process (i.e., as compared with the configuration shown in Figure 12).

In particular, preprocessing block 306 performing the sequential calculations for determining the k-th output, $\boldsymbol{m}(k)$, given in equation (36), may be viewed as follows. Subsequent to determining $\boldsymbol{m}_1$ we may calculate:

$$\delta_1(k) = -\sum_{n=1}^{K_1} L_U(k, n) m_1(n) \qquad (37).$$

Then, the remaining K_2 sequential equations may be represented by:

$$m(k) = \mathrm{mod}\,(s_2^{(\Pi_2)}(/c)) + 5_1(\mathrm{fe}) - \sum_{n=K_1+1}^{k-1} \mathrm{I}_U(k,n)m(n)\Big) \qquad (38),$$

for $k = K_1 + 1, \ldots, K_2$, where δ is an offset vector, which may be represented in the vector form as:

$$\boldsymbol{\delta} = -\boldsymbol{M}_{U21}\boldsymbol{m}_1 \qquad (39),$$

where matrix M_{u12} is based on partition of the scaled unit diagonal matrix L_u (i.e., $M_{u12} = L_U(rows: K_i + 1\ to\ K, columns:\ 1\ to\ K_1))$. Offset vector δ may also be written in terms of D_2 as:

$$\delta = -D_2^{-1} M_{21}\boldsymbol{m}_1, \text{ where } D_2 = \mathrm{diag}(\mathrm{diag}(\mathrm{I}_2)) \qquad (40).$$

Since the offset vector δ is known, it may be separately calculated (independently of the sequential calculations). The outputs of adders 434 and 436 constitute outputs of calculation block 432 represented by offset vector s, whose two components are δ_1 (inputted to adder 376_4) and δ_2 (inputted to adder 376_5). This configuration typically reduces real-time precoding complexity as noted.

Alternatively, according to another implementation (not shown), calculation block 422 in Figure 13, is configured and operative to instead calculate: $\mathrm{inv}(\boldsymbol{L}_1) * D_1$.

Returning now to Figure 4 (and Figure 6 in the 5-user example), block 214 in Figure 4 (and preprocessing block 306 of Figure 6) output a signal represented by vector m to the input of Q block 218 (and Q block 308 of Figure 6). Q block 218 (Figure 4) and a Q block 308 (in the 5-user example in Figure 6) are configured to perform QR decomposition for a Hermitian conjugated and permuted channel according to equations (10) and (11), and to output an outputted signal 222 represented by vector o in Figure 4. Scalar power gain blocks 310_1, 310_2, 310_3, 310_4, and 310_5, (collectively denoted 310_{1-5}), which are configured and operative to apply a scalar factor a common to all users, are optional and shown only in Figure 6. Alternatively, the scalar factor may be different for different users as will be described in greater detail in conjunction with Figures 17A and 17B. In the case where the system of the disclosed technique utilizes the scalar power gain blocks (Figure 8), these in turn output the signal represented by vector $\boldsymbol{o}$ (see equations (14)).

Precoder 220 (Figure 4) and precoder 312 (Figure 6), represented by precoder matrix P e. $C^{N\times K}$ are configured and operative to perform pseudo-inversion of the communication channel matrix, and to generate outputted signal 222, represented by vector $\boldsymbol{o}$ (recapping that $\boldsymbol{o}$ = F(s + $\mathbf{c}$) and $P = \alpha QR^{-H} D \Pi^T$ (equations (14)) to the communication channel 224 (Figure 4) and 314 (Figure 8 in the 5-user example). P represents a generalized inverse of the communication channel matrix H, given by:

$$\boldsymbol{H} = (\boldsymbol{Q} * \boldsymbol{R} * \Pi^T)^H = \Pi * R'' * \boldsymbol{Q}'' \quad (41).$$

If a direct substitution of the factorized channel matrix (equation (41) above) is made into an expression for the pseudoinverse $pinv(R')*D$ of block 214 (Figure 4), the result yields basic elements of THP based on QR decomposition. In particular, if we denote the diagonal scaling of the communication channel pseudoinverse as $\tilde{\boldsymbol{D}}$ and the scaling of the THP block 220 as D then:

$$\text{pmv}\,(H)\tilde{\boldsymbol{D}} = H^{\mathrm{H}}(H\boldsymbol{H}^{\mathrm{H}})^{-1}\tilde{D} = \boldsymbol{QR\Pi}^{\mathrm{T}}(IJR^{\mathrm{H}}\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{Q}Ri\mathrm{1}^{\mathrm{T}})^{-1}\tilde{\Omega} =$$

$$\boldsymbol{QR}\Pi^{\mathrm{T}}\Pi^{-\mathrm{T}}(\boldsymbol{R}^{\mathrm{H}}\boldsymbol{R})^{-1}\Pi^{-1}\tilde{\mathrm{D}} = \boldsymbol{QRR}^{-1}(\boldsymbol{R}^{\mathrm{H}})^{-1}\Pi^{\mathrm{T}}\tilde{\boldsymbol{D}} = \boldsymbol{Q}(\boldsymbol{R}^{\mathrm{H}})^{-\mathbf{1}}\boldsymbol{D}\Pi^{\mathrm{T}} \qquad (42),$$

where $\boldsymbol{n}^{\mathrm{T}}\tilde{\boldsymbol{D}} = \boldsymbol{D}\mathrm{H}^{\mathrm{T}}$, $\mathrm{Q} \in \mathrm{C}^{N\times K}$, $R\ e\ \mathbb{C}^{K\times K}$ and I) is a diagonal matrix of size (dimension) K x K and Π^{τ} is the permutation matrix of size $K\ x\ K$ (where the columns of matrix ***Q*** are orthonormal) based on the so-called thin (or reduced) QR factorization (due to $K < N$). The relation between the diagonal matrix which scales the channel pseudoinverse and the scaling utilized by THP block 220 is D = $\mathbf{n}^{-1}\tilde{\boldsymbol{D}}\mathbf{n} = \text{ĭ}\mathrm{I}^{\mathrm{Y}}\tilde{\boldsymbol{D}}\text{ĭ}\mathrm{I}$. Alternatively, it is: $\tilde{D} = \mathrm{II}\boldsymbol{D}\mathrm{I}\bar{\mathrm{I}}^{\mathrm{T}}$. The scaling is not arbitrary (e.g., as in the prior art) but THP block 220 is configured to scale according **to** $\boldsymbol{D}$ = diag(diag(J? ^{H})). Note that the QR decomposition presented is based on THP that represents the precoder utilizing the **Moore-Penrose** pseudoinverse and its corresponding specifically chosen diagonal: $\tilde{D}$ = ndiag(diag ***(R***^{H}))n $^{\mathrm{T}}$.

The transmitter side outputs **outputted** signal 222 (vector ***o),*** which propagates through communication channel 224, the result of which is a signal 226 (represented by a vector y) received at the receiver side. A received signal 232 at the receiver side, represented by a vector r is the sum of signal 228 ***(y)*** with additive noise 228 (represented by a vector **n),** signifying that the preceded information symbols propagated through the communication channel includes additive 230 noise. Received signal 232, vector **r** may be represented by:

$$\boldsymbol{r} = \boldsymbol{Ho} + \boldsymbol{n} = \boldsymbol{HP}(\boldsymbol{s} + \mathbf{c}) + \mathbf{n} = \Pi R^{H}\boldsymbol{Q}^{H}\{xQR^{\sim H}DiI\ ^{T}(s + \mathbf{c}) + \mathbf{n} =$$

$$= \alpha\Pi\boldsymbol{D}\Pi^{T}(\boldsymbol{s} + c) + 7\!\!/i \qquad (43),$$

where **r** £ $\mathbb{C}^{\mathrm{K}\times 1}$ is the received signal, ***o*** *e* $\mathbb{C}^{\mathrm{K}\times 1}$ is the output to the communication lines (i.e., the transmitted signal) and n ε $\mathbb{C}^{\mathrm{K}\times 1}$ is the additive noise at the receiver side. The expression $\Pi D\Pi^{T}$ represents the permuted diagonal:

$$\boldsymbol{D}_{\Pi} = \Pi\boldsymbol{D}\Pi^{T} \qquad (44),$$

therefore:

$$r = \alpha D_{\Pi}(s + c) + 7\!\!/ = \alpha \acute{\alpha}_{\gamma_{I}}\, 0\ (s + c) + n \qquad (45).$$

The permutation matrix Π orders (i.e., permutes) the constructed diagonal:

$$\boldsymbol{d}_{\Pi} = \mathrm{diag}(\boldsymbol{D}_{\Pi}) = \mathrm{Fid} \qquad (46),$$

where **d =** $\mathrm{diag}(\boldsymbol{D})$. In THP, the input signal is first permuted as $\boldsymbol{\Pi}^{T}\boldsymbol{s}$, then the sequence $\tilde{c}$ is added via the THP scheme:

$$\Pi^{T}\mathbf{5} + \tilde{\boldsymbol{c}} = \mathbf{n}^{T}(s + \boldsymbol{Tic}) = \mathbf{n}^{T}(\mathbf{s} + c) \qquad (47),$$

where c = ***lie.***

Continuing at the receiver side, received signal 232 (Figure 4) is inputted to a block 234 configured and operative to perform scaling by inverting the permuted diagonal vector $\mathbf{D}_{\Pi}$**,** as defined in equation (44), the output of which is denoted by an output signal 236. With reference to the 5-user example of Figure 6, at the receiver side, complex-valued scalar gain blocks **316_1**, **316_2,** 316_3, 316_4, 316_5 (collectively denoted by **$316_{1\text{-}5}$)** are configured and operative to apply corresponding complex-valued scalar gain factors **F(1), F(2), F(3),** F(4), and **F(5).** (A special case of complex-valued scalar gain factors is where they are real values). For NLP supporting CPE units, output signal 236 (Figure 4) represents an input to a modulo operation block 238, which is configured and operative to apply modulo arithmetic operation to received signal 236, and to output an output signal 240 of estimated symbols, represented by a vector $\hat{\mathbf{s}}$**.** For those CPE units whose supportability does not include NLP (i.e., LP group of users) output signal 236 is effectively output signal 240 (i.e., no modulo operation is applied). With reference to the 5-user example of Figure 6, at the receiver side, modulo arithmetic blocks 318_2 and 318_4 are employed by respectively employed respectively by CPE units 108_2 and 108_4 (Figure 1, for example) having supportability to decode NLP encoded data (i.e., the NLP group of users). Modulo arithmetic blocks 318_2 and 318_4 are configured and operative to encode the NLP encoded data by applying modulo operation, the generated result of which are output as estimated symbols **320_2** (i.e., **s(2))** and 320_4 (i.e., **s(4)).** CPE units not supporting the decoding of NLP encoded data (LP group of users) don't have modulo arithmetic blocks, and consequently the outputs from complex-valued scalar gain blocks 3161, 316_3, 316_5 constitute as estimated symbols 320i (i.e., $\hat{s}(1)$) 320_3 (i.e., s(3)), and **320_5** (i.e., s(5)).

For non-cooperative receivers (i.e., receivers that do not share or use information they process between themselves) received signals, r_1 and $\mathbf{r}_2$, are processed: (i) independently for every user (as the users are non-cooperative); and (ii) by using scalar scaling of the received signals. These can be represented in the matrix-vector form as diagonal gain corrections, $diag(g_{(1)})$ and $diag(g_{(2)}))$ for both LP and NLP groups of users, and additionally by modulus operations applied only to the NLP group of users, as may be represented by the following equations for outputted estimated symbols:

$$\hat{s}_1 = G_{(1)}r_1 = diag(g_{(1)})\, r_1 = g_{(1)} \mathbf{Q}\ r_1 \qquad (48),$$

$$\hat{\mathbf{s}}_2 = mod(G_{(2)}r_2) = mod(dl.ag(g_{(2)})\, \mathbf{r}_2) = mod(\mathbf{g}_{(2)} \odot \mathbf{r}_2) \qquad (49),$$

where s_1 and $\mathbf{s}_2$ respectively signify information symbols for the LP and NLP groups of users and $\hat{s}_1$ and $\hat{s}_2$ denote their corresponding estimates (which may be further inputted into error correction blocks (not shown) as known in the art).

The application of complex-valued scalar gain factors F(1), F(2), F(3), F(4), **F**(5) shown in Figure 6 for the NLP supporting CPE receiver units, in the case where $F(k) \neq 1$ is generally achieved in two steps. The first step involves compensating for the communication channel attenuation gains, given by the diagonal of the L matrix (i.e., diag(L)). The second step involves compensating for gains affected by the modulus operation. In this case an estimated symbol for a k-th NLP supporting CPE receiver unit is given by:

$$\hat{s}_{NLP}(k) = \frac{1}{f_{\Pi}(k)} mod\left(\frac{1}{D_{\Pi}(k,k)}\, r(k))\right) \qquad (50),$$

where $D_{\Pi} = \Pi D_{\mathrm{L}} \Pi^{\mathrm{T}}$; D_{L} = **diag(diag(L));** and $/_{\Pi}$ = **diag(n diag(/)** Π^{τ}**),** as the received symbol is prior to frequency equalization at the receiver side (CPE units) is given by (prior to addition of noise):

$$y = \mathrm{r}\Pi D_{\mathrm{L}}(\mathrm{diag}((f))\Pi^{\mathrm{T}}\mathrm{s} + \tilde{c}) = \mathbf{I}\Pi D_{\mathrm{L}}\mathbf{n}^{\mathrm{T}}(\mathbf{ndiag((/))n} \quad {}^{\mathrm{T}}\mathbf{s} + \Pi/\tilde{c}) \quad \mathbf{(51).}$$

Recalling that the diagonal matrix ***D*** (after factorization) in Figure 4 is $\boldsymbol{D}$ = $\mathrm{D}_{L}\mathrm{diag}(/)$, vector r is scaled by components of the diagonal matrix $\Pi D_{L} \Pi^{T}$ thus obtaining:

$$(\Pi D_{\mathrm{L}}\Pi^{\mathrm{T}})^{--1}y = \mathbf{ndiag((/))n} \quad {}^{\mathrm{T}}\mathbf{s} + \Pi\tilde{c} \qquad (52).$$

Subsequently the modulo operation is performed on the permuted auxiliary **perturbation** $\tilde{c} = \Pi c$ after which the second scaling step is performed, in accordance with equation (50), thereby obtaining:

$$\mathbf{(ndiag((/))n} \quad {}^{\mathrm{T}})^{\sim 1}\mathbf{mod}(\mathbf{n}D_{\mathrm{L}}\mathbf{n}^{\mathrm{T}})\text{-}{}^{1}y = s \qquad (53).$$

In practice, this procedure is applied to received vector?-, thereby arriving at equation (50).

Note that for the standard THP case where *F(k)* ≡ 1, the *k-t*h estimated symbol reduces to:

$$\hat{s}_{NLP}(k) = mod\left(\frac{1}{D_{\Pi}(k,k)}\,\mathrm{r}\,(k))\right) \qquad (54).$$

The estimated symbol $\hat{s}_{NLP}(k)$ corresponds to the NLP group of users at the receiver side (CPE units). At the transmitter side (DPU) the processor applies the modulus operation to symbols whose indices correspond to the NLP group of users after the permutation block (i.e., while not applied to symbols whose indices correspond to the LP group of users. For the LP group of CPE receiver units, the estimated symbol for a **Z-th** CPE receiver unit is:

$$\hat{s}_{,,}\ (l) = \frac{1}{f_{\Pi}(l)}\frac{1}{\overline{D_{\Pi}}(l,\overline{l})}\ r(l)) \qquad \mathbf{(55),}$$

which is performed **in** one step (while two steps are also possible). Equation (31) may be rewritten to account for the THP scheme in the case where $\boldsymbol{f(k)} = 1$ for index $\boldsymbol{k}$ corresponding to the permuted NLP group of users as: $\boldsymbol{L_v m} = \mathrm{diag}(\boldsymbol{f})\mathbf{n}^{\mathrm{T}}(\mathbf{s} + \mathbf{c})$, where $\mathbf{c} = \mathbf{n}^{\mathrm{T}}\tilde{\mathbf{c}}$.

It is noted that the precoder given by: $\boldsymbol{P} = QR^{-H}$, where $Q \in \mathbb{C}^{\boldsymbol{N}x\boldsymbol{K}}$ and $R \in \mathbb{C}^{KxK}$ (where the columns of matrix $\boldsymbol{Q}$ are orthonormal) based on the so-called thin (or reduced) QR factorization (due to $\boldsymbol{K < N}$) performed for H^H as $H^H = QR$ and hence $\boldsymbol{H = R^H Q^H}$) leads to the Moore-Penrose pseudoinverse. We straightforwardly observe that:

$$P = H^H(HH^H)^{-1} = QR(R^H Q^H QR)^{-1} = QR(R^H R)^{-1} =$$

$$= \boldsymbol{QR(R^{-1}R^{-H}) = QR^{-H}} \qquad (56).$$

Therefore, the precoder based on the QR decomposition represents a particular pseudoinverse. One may readily observe that the same is true for QR decomposition with permutations, as described hereinabove.

It is assumed that the number of transmitters is $\boldsymbol{N}$, and that $\boldsymbol{K \leq N}$. In a wireless system, $\boldsymbol{N}$ represents the number of transmitting antennas and in a wire-line system $\boldsymbol{N}$ is the number of CPE units (e.g., modems) transmitting through binder 110 (Figure 1). The $\boldsymbol{K}$ CPE units which receive information (at a particular frequency) are regarded as active CPE units ("active users") at that particular frequency. The same CPE (user) may be active at some frequency tones and non-active at the other frequency tones. To further explain the meaning of an activity level of a particular CPE unit at a particular subcarrier frequency, reference is further made to Figures 14A and 14B. Figure 14A is a table showing a database of supportabilities and activity levels of each CPE unit at a particular point in time, generally referenced 450, constructed and operative in accordance with the disclosed technique. Figure 14B is a schematic diagram showing a graph of a particular example of activity levels of CPE units ordered according to (relative) communication link quality as a function of subcarrier frequency at a particular point in time, generally referenced 480, in accordance with the disclosed technique.

Database 450 shown in tabulated form includes a CPE number field 452, a supportability field 454, and an activity level field 458. CPE number field 452 includes a numbered list of CPE units 106i, 106_2, 106_3,..., 106_{N-1},106_N (Figure 1). Supportability field 454 includes an ordered **list of supportabilities** 120_1, 120_2, 120_3,..., 120_{N-1},120_N (Figure 1) each associated (index-wise) with a particular CPE unit. Activity level field 456 includes an ordered list of activity levels activity levels 122_1, 122_2, 122_3,..., 122_{N-1},122_N (Figure 1) (per tone), each associated (index-wise) with a particular CPE unit. Activity level field 454 includes a plurality of subfields: an ON/OFF subfield 458, a synchronization subfield 480, a frequency response subfield 462, and an activity level per tone **(subcarrier** frequency) subfield 464. Supportability field 454 includes a list of supportabilities 120_1, 120_2, 120_3,..., 120_{N-1},120_N defining for each CPE unit its ability to decode at least one of NLP data and LP data (i.e., with respect to hardware, software, firmware, etc.). ON/OFF subfield 458 lists for each CPE unit whether it is switched 'on' or otherwise not (e.g., 'off', non-existent, etc.). Synchronization subfield 460 lists for each CPE unit whether it is synchronized for transceiving data with corresponding transceivers 116_1, 116_2, 1163,..., **116_N-i**, 116_N of DPU 104 **(Figurel)** (e.g., in the "showtime" state). Frequency response subfield 462 lists for each CPE unit its corresponding frequency response (i.e., a measure of the magnitude (and phase) of a signal propagating via the communication channels as a function of subcarrier frequency). Activity level per tone subfield 464 lists for each CPE unit whether it is considered active or inactive according to a determination made by DPU 104. Specifically, at least one of controller 112 (Figure 1) and processor 114 of DPU 104 is configured and operative to employ a decision rule (e.g., an algorithm) for determining active CPE units **(*K*)** and inactive CPE units from a total number of CPE units *(N)* according to various criteria including data from ON/OFF subfield 458, synchronization subfield 460, frequency response subfield 462 (e.g., taking account SNR), and according to various constraints including optimization criteria such as the maximization of average rate, maximization of max-min rate, and the like. It is noted that data stored by database 450 is time-dependent (i.e., are subject to change over time). Consequently, activity level for each CPE is both subcarrier frequency-dependent as well as time-dependent.

Further reference is made to Figure 14B showing a graph 480 of a particular example of activity levels of CPE units ordered according to (relative) communication link quality (performance, e.g., channel capacity) as a function of subcarrier frequency at a particular point in time. Graph 480 includes a vertical axis 482, and a horizontal axis 484. The CPE units are ordered along vertical axis 482 according to relative communication link quality, i.e., CPE units exhibiting a relatively high communication link quality (e.g., SNR) are positioned at a relative higher position along vertical axis 482 in comparison to CPE units exhibiting relatively lower communication link quality. For demonstration purposes, the CPE unit exhibiting the highest communication link quality (in relative terms) is referenced 488 in Figure 14B, whereas the CPE unit exhibiting the lowest link quality is referenced 486. Horizontal axis 484 represents frequency (i.e., a range of subcarrier frequencies) the higher the frequency the farther along it is on this axis. A point in graph 480 is defined by a (horizontal, vertical) coordinate that corresponds to a particular CPE unit at a particular subcarrier frequency. A point located on a curve 490 or within a shaded area 492 represents that it is inactive, whereas a point located outside shaded area 492, i.e., within a non-shaded area 494 represents that it is active. Shaded area 492 is defined by curve 490 that is characteristic to the communication system 100. Curve 490 is a typical representation of a characteristic tendency of SNR decline with higher frequencies but is given only as an example for explicating the disclosed technique, for it can assume other forms (e.g., non-monotonic, continuous, discontinuous, etc.). Figure 14B illustrates that a particular CPE unit may be active at some subcarrier frequencies, and inactive at others. In particular, for the example given, CPE unit 488 exhibiting the relatively highest communication link quality is active at all subcarrier frequencies, whereas CPE unit 486 exhibiting the lowest link quality is inactive from a threshold subcarrier frequency f_{γ}. Each CPE unit may have its respective threshold subcarrier frequency. In the wireless case, a drop in the SNR at a specific frequency maybe due to a property of the communication channel (including the environment). Alternatively, inactive CPE units may be determined according to an optimization algorithm (e.g., that maximizes the average rate for all CPE units).

A memory (not shown) is configured and operative to store database 450. The memory is typically configured to be coupled with at least one of controller 112 and processor 114 of DPU 104 (Figure 1). In one implementation, memory is intrinsic to DPU 104. In another implementation, memory is distinct and external to DPU 104, accessible via known communication methods (e.g., wire-line, wireless, internet, intranet communication techniques, etc.).

The precoder constructed according to the disclosed technique to solve the interoperability problem of CPE units having different supportabilities relates to the general case where $K \leq N$, but more specifically relates to the more typical and tougher case where $K < N$ at a particular subcarrier frequency. As aforementioned, a solution to the special simple case where $K = N$ is already proposed by the prior art. The case where $K < N$ may also arise for example in situations where some CPE units might not have a large enough SNR (e.g., due to significant channel signal attenuation in a high frequency portion of the transmitted spectrum of frequencies (e.g., corresponding to moderate and lengthy communication lines)). This case can be expressed, for example, when there's an insufficient SNR for transmitting a certain number of bits in a constellation at a particular frequency (e.g., calculated when bit-loading is less than a specific value). The communication lines corresponding to these CPE units may be reused to transmit information (i.e., preceding them with information symbols for other users) intended for the benefit of the remaining CPE units (i.e., or subset thereof), instead of for themselves. For example, if the ZF precoder is utilized it can be based on the communication channel pseudoinverse. In an extreme case (i.e., beam forming) where all but one CPE unit is active (or operational), ail communication lines are utilized to transmit to that CPE unit. Note that a particular CPE unit may be active at some subcarrier frequencies and inactive at the other subcarrier frequencies. The solution of the disclosed technique relates and is attained on a subcarrier frequency basis.

The disclosed technique is configured and operative to employ an algorithm for determining active users from a total number of users according to various criteria, e.g., depending on optimization criteria such as the maximization of average rate, maximization of max-min rate, etc. In that regard, let's recall equation (19), which may be rewritten as:

$$\boldsymbol{L}_2\boldsymbol{m}_2 = D_2 n_2^{\mathrm{T}}(s_2 + c_2) - \boldsymbol{M}_{21}\boldsymbol{m}_1 \qquad (57).$$

Noting that $\boldsymbol{m}_1$ is already known, we introduce:

$$\tilde{\boldsymbol{\delta}}_1 = - \mathbf{D}_2^{-1}\boldsymbol{\Pi}_2\boldsymbol{M}_{21}\boldsymbol{m}_1 \qquad (58).$$

For a general vector precodsng scheme there is not assumption that $D_2 = \mathrm{diag}(\mathrm{diag}(X_2))$. The diagonal D_2 represents additional degrees of freedom. Then we have:

$$\boldsymbol{m}_2 = \boldsymbol{L}_2^{-1}\boldsymbol{D}_2\,\mathbf{n}_2^{\mathrm{T}}(\mathbf{s}_2 + \tilde{\boldsymbol{\delta}}_1 + c_2) \qquad (59),$$

where perturbation vector c_2 is determined by applying optimization criteria which for a given D_2 that minimizes a power criterion applied to vector $\boldsymbol{m}_2$. For example, optimization involves minimization of a **^-dimensional** (absolute-value) norm $\|\boldsymbol{m}_2\|_q$, where the following standard notation is used: $\|\boldsymbol{z}\|_q = (\sum_{l=1}^{L}|z_l|^q)^{\frac{1}{q}}$ for a vector z of a length L. Such minimization may be applied for every realization of $\mathbf{s_2}$ and s_1 (where s_1 governs m_1 and therefore $\tilde{\delta}$j). Alternatively, minimization is applied b the statistical power averages. Different types of norms of different dimension q may be used. For example, the norm $\|m_2\|_2$ is related to the total power of all $\boldsymbol{m}_2$ components, while minimization of $\|\mathbf{m}_2\|_\infty$ minimizes the maximal power per component. The components of $\mathbf{c_2}$ are $c_2(k) = \tau_k p_k$. The modulus ¾ depends on the constellation size of s_2 or according to $c_2(/i) = \tau p_k$, where the same modulus value is applied to all constellations (i.e., determined by the maximal allowed constellation size as described below). The complex number p_k is what is sought via the optimization process (typically in the form of signed integer components (i.e., its real and imaginary components are signed integers)). For a constant modulus, the disclosed technique may minimize the instantaneous power value related to the currently transmitted symbols or, alternatively, the average power value for a given time period.

To further elucidate the disclosed technique, reference is now made to Figure 15, which is a schematic block diagram illustrating a specific implementation of the general hybrid-interoperability precoding scheme, specifically showing delineation into two paths, generally referenced 500, constructed an operative in accordance with the disclosed technique. Figure 15 essentially shows a hybridization arrangement of two delineated data paths, namely, a LP path 502 corresponding to the LP group of CPE receiver units (users) implementing LP, and a NLP path 504 corresponding to the NLP group of CPE receiver units implementing NLP. The principles of general hybrid-interoperability precoding scheme 500 conforms to principles of the disclosed technique heretofore described.

Initially, it is noted that vectors s_1, $\boldsymbol{m}_1$, δ_1, $\boldsymbol{d}_1$, $\boldsymbol{y}_1$, n_1, $\boldsymbol{r}_1$, and $\hat{s}_1$ are of dimension K_1, and vectors s_2, $\boldsymbol{m}_2$, $\boldsymbol{d}_2$, y_2, n_2, $\boldsymbol{r}_2$, and $\hat{s}_2$ are of dimension K_2. Starting from LP path 502, information symbols, represented by a vector s_1 (intended for being linearly precoded) are inputted to a permutation block 506 configured and operative to perform permutation of vector elements of s_1 according to a permutation matrix Π_1' (where Π_1' 'is a $Ki \times K_1$ permutation matrix) (similarly to permutation matrix of block 334 in Figure 7). The permuted information symbols from permutation block 506 enter a block 508 configured and operative to perform power scaling according to a matrix D_1 $(D_, = diag(d_1))$ the output of which is directed to a block 510, which in turn is configured and operative to perform inversion of matrix L_1 (i.e., calculate invCL$_1$), where L_1 is a lower diagonal matrix of dimensions K_1 x K_1.) (Analogous to block 422 in Figure 13). Initially, the system and method of the disclosed technique is configured and operative to optimize the linear precoder by optimized selection of diagonal scaling values of D_1 and permutation matrix Π_1. Block 510 outputs linearly precoded symbols, denoted by a vector m_1, which constitutes as an input to a concatenation block 526, as well as to a block 512, which in turn is utilized in NLP path 504. Block 512 is configured and operative to multiply vector m_1 **by** a matrix M_{21} (Figure **8)** of dimensions of K_2 x K_1 (where L_1, L_2 and M_{21} are partitions of a lower diagonal L matrix of dimensions K x K, and $K = K_1 + K_2$) and to output an offset **vector** δ_1**,** which in turn is input to a block 514. Block 514 is configured and operative to multiply offset vector δ_1 by **(-1)** which yields a result in the form of equation (39).

Referring now to NLP path 504, information symbols represented by a vector s_2 (intended for being nonlinearly precoded) are inputted to a permutation block 516 configured and operative to perform permutation of vector elements of §$_2$ according to a permutation matrix Π_2' (where Π_2**'** is a K_2 x K_2 permutation matrix) (similarly to permutation matrix of block 336 in Figure **7).** Adder 518 combines the permuted information symbols with a permutation vector c_2 and outputs the result to a block 520 configured and operative to perform power scaling according to a matrix D_2 $(D_2 = diag(d_2))$. Similarly to LP, the system and method of the disclosed technique is configured and operative to optimize the nonlinear precoder by optimized selection of diagonal scaling values of D_2 and permutation matrix Π_2, as well as optimized selection of the perturbation vector c_2. Adder 522 is configured and operative to combine the output of block 520 and block 514 and produce a result which is directed to a block 524, which in turn is configured and operative to perform inversion of matrix $\boldsymbol{L_2}$ (i.e., calculate inv(L_2), where L_2 is a lower diagonal matrix of dimensions K_2 x K_2.). It is noted that permutation blocks 506 and 516 may be regarded as a second permutation stage (e.g., stage 2 referenced 332 in Figure 7) that follows from a first permutation (e.g., stage 1 referenced 330 in Figure 7). Block 524 outputs nonlinearly preceded symbols, denoted by a vector m_2, which constitutes as another input to concatenation block 526.

Concatenation block 526 is constructed and operative to perform concatenation of K_1 vector components of $\boldsymbol{m}_1$ and K_2 vector components of m_2 into a concatenated vector m. of K vector components. Essentially, vector 256, $\boldsymbol{m}$ (Figure 5), consists of two sub-vectors 256_1, $\boldsymbol{m}_1$, and 256_2, $\boldsymbol{m_2}$ corresponding to two aggregate groups of vector elements respectively associated with the LP group and NLP group of CPE units. The outputted concatenated vector m is fed into a Q block 528 configured and operative to perform the QR decomposition $QR = H^H\Pi$ (where the channel matrix is H and Π represents a block-diagonal permutation matrix constructed from Π_1 and Π_2). A matrix $\boldsymbol{L}$ is defined by $\boldsymbol{L} = R^H$. Q block 528 outputs an output signal represented by an output vector $\boldsymbol{o}$, which constitutes also as a transmitted output signal propagating from the transmitter side that via a communication channel 530 for reception at the receiver side.

At the receiver side, LP path 502 corresponds to LP CPE units configured and operative to receive a signal, represented by a received vector $\boldsymbol{r}_1$ which is the sum of a vector $\boldsymbol{y}_1$ and noise $\boldsymbol{n}_1$, denoting that the received information symbols propagated through communication channel 530 includes additive 532 noise. The received signal (vector $\boldsymbol{r}_1$) is inputted to a block 534, which in turn is configured and operative to perform scaling of the received signal per component according to $\mathrm{inv}(\Pi_1 \boldsymbol{D}_1 \Pi_1^T)$ the result of which yields outputted estimated symbols $\hat{s}_1$. NLP path 504 corresponds to NLP-supporting CPE units configured and operative to receive a signal, represented by and operative to receive a signal, represented by a received vector $\boldsymbol{r_2}$ which is the sum of a vector y_2 and noise n_2, denoting that the received information symbols propagated through communication channel 530 includes additive 536 noise. The received signal (vector r_2) is inputted to a block 538, which in turn is configured and operative to perform scaling of the received signal per component according to $\mathrm{inv}(\Pi_2 \boldsymbol{D_2} \Pi_2^T)$ the resulting signal is fed into modulo operation block 540, which in turn is configured and operative to apply a modulo operation to the signal outputted by block 538, the result of which yields outputted estimated symbols $\hat{\boldsymbol{s}}_2$.

The optimized selection of the scaling values of D_1, $\boldsymbol{D_2}$, the permutation matrices Π_1, Π_2, as well as the perturbation vector c_2 may be determined, for example according to various criteria such as the minimization of the power involving m_2 as discussed hereinabove in conjunction with equation (59). It is noted that the dimension of the perturbation vector is $\boldsymbol{K_2}$ corresponds to the dimension of the input information symbol vector $\boldsymbol{s_2}$. The functional split to linear and nonlinear precoders solves the interoperability problem, improves performance of the nonlinear precoder for small constellations, as well as well as reduces the dimensionality of the nonlinear precoder. Regarding NLP dimensionality reduction, it is known that determining the perturbation vector may involve large computational complexity (increasing significantly with vector dimension), therefore it may be beneficial to reduce it by reducing the number of the NLP users and the system and method of the disclosed technique allows to perform this by allocating part of the NLP users to the LP users (i.e., all users remain precoded).

Alternatively, optional scaling of the output from the transmitter side by use of a scaling gain factor α is also viable. To further elucidate this implementation, reference is now made to Figure 16, which is a schematic block diagram illustrating another specific implementation of the general hybrid-interoperability preceding scheme, specifically showing delineation into two paths, generally referenced 550, constructed an operative in accordance with the disclosed technique. Figure 16 essentially shows a similar hybridization arrangement of two delineated data paths of Figure 15 apart from several modifications described hereinbelow. Figure 16 shows two paths, namely, a LP path 552 corresponding to the LP group of CPE receiver units (users) implementing LP, and a NLP path 554 corresponding to the NLP group of CPE receiver units implementing NLP. The principles of general hybrid-interoperability preceding scheme 550 conforms to principles of the disclosed technique heretofore described. General hybrid-interoperability preceding scheme 550 hybridizes between linear preceding and Tomlinson-Harashima **vector** preceding.

Initially, and similarly to the description heretofore described in conjunction with Figure 14, it is noted that vectors s_1, $\boldsymbol{m}_1$, δ_1, $\boldsymbol{d}_1$, y_1, $\boldsymbol{n}_1$, r_1, and $\hat{s}_1$ are of dimension K_1, and vectors s_2, $\boldsymbol{m_2}$, d_2, y_2, $\boldsymbol{n_2}$, $\boldsymbol{r_2}$, and $\hat{s}_2$ are of dimension K_2. $D_1 = diagid$^), and $D_2 = \boldsymbol{diag(d_2)}$. a is a scalar used for (optional) power scaling. ꞵ¾ is a $\boldsymbol{K_1}$ x $\boldsymbol{K_1}$ permutation matrix, and Π_2 is a K_2 x K_2 permutation matrix. L_1 is a lower diagonal matrix of K_1 x K_1. L_2 is a lower diagonal matrix of $K_2 \times K_2$. M_{12} is of $\boldsymbol{K_2} \times K_1$. $\boldsymbol{L}_1$, $\boldsymbol{L_2}$ and M_{12} are partitions of lower diagonal $\boldsymbol{L}$ matrix of dimensions K x K. Π represents a block-diagonal permutation matrix constructed from Π_1 and Π_2.

Starting from LP path 552, information symbols, represented by a vector $\boldsymbol{s}_1$ are inputted to the following blocks 556, 558, 560, 562 which are all respectively configured and operative identically to blocks 506, 508, 510, 512 in Figure 15. Owing to the lower triangular structure of the $\boldsymbol{L}_1$ matrix, its optimization is not influenced and independent of the nonlinear precoder. Block 562 outputs an offset **vector** δ_1**,** which in turn is input to a block 564, which in turn is configured and operative to apply inversion to the diagonal of vector L_2 (i.e., $inv(diag(L_2)))$ and to output a signal to a block 566 configured and operative identically to block 514 in Figure 15. An output of LP path 552 is a vector m_1.

Turning now to NLP path 554, information symbols represented by a vector s_2 are inputted to a permutation block 516 configured and operative identically to permutation block 516 in Figure 15. Adder 570 combines the permuted information symbols generated from permutation block 516 with an output of block 556, the result of which is fed into a THP block 572, which in turn is configured and operative to perform THP with respect to $\boldsymbol{L_2}$ (i.e., inversion of matrix L_2 according to the principles of THP). THP is performed on permuted information symbols $\boldsymbol{s_2}$ (via permutation matrix Π_2) from which a component-wise scaled vector δ_1 (i.e., S_1 multiplied by diagonal matrix $inv\{D_2))$ is subtracted (corresponding to the subtraction of linear precoded users from nonlinearly precoded users). in this scheme, THP block 572 determines the vector $D_2 = \boldsymbol{diag(L_2)}$**,** thereby yielding an output signal represented by a vector m_2.

Vectors m_1, and m_2 are inputted to a concatenation block 574 and then to a Q block 576, both of which are respectively identical in construction and operation to blocks 526 and 528 in Figure 15. An output from Q block 576 is inputted into a scalar gain block 578, which in turn is configured and operative to apply scalar gain factors to an output signal, represented by a vector $\boldsymbol{o}$, which constitutes as an output from the transmitter side communicated via a communication channel 580 to the receiver side.

At the receiver side, following LP path 552 corresponds to LP CPE units configured and operative to receive a signal, represented by a received vector r_1 which is the sum of a vector **^and** additive 582 noise n_1. The received signal (vector r_1) is inputted to a block 584, which in turn is configured and operative to perform scaling of the received signal per component according to $\left(\frac{1}{\alpha}\right)$ * inv(ιτ$_1$ * D_1 * Π_1^T) the result of which yields outputted estimated symbols $\hat{s}_1$. NLP path 554 corresponds to NLP-supporting CPE units configured and operative to receive a signal, represented by and operative to receive a signal, represented by a received vector r_2 which is the sum of a vector y_2 additive 586 and noise n_2. The received signal (vector r_2) is inputted to a block 588, which in turn is configured and operative to perform scaling of the received signal per component according to (-) * inv(Π_2 * D_2 * Π_2^T) the resulting signal is fed into a modulo operation block 590, which in turn is configured and operative to apply a modulo operation to the signal outputted by block 588, the result of which yields outputted estimated symbols $\hat{s}_2$.

Alternatively, according to another implementation of system 102, the scalar factor may be distinctive (e.g., not common, different) for different CPE units (users). In accordance with this alternative implementation, reference is now made to Figure 17A, which is a schematic diagram illustrating an example of a specific implementation of the general hybrid-interoperability precoding scheme, utilizing different scalar factors, generally referenced 600, configured and operative in accordance with the disclosed technique. Implementation 600 generally provides a system and method for nonlinear precoding of an information symbol conveying data for transmission (over a particular **subcarrier** frequency) between a plurality of transmitters **116**$_1$,... .,11**6**$_N$ (Figure 1) and a plurality of receivers 11**8i,.. .,1** 18$_N$ (Figure 1) via a plurality of communication channels **108i,.. .,108$_N$** (Figure 1), where the communication channels define a channel matrix ***H*.** A processor (e.g., 114, Figure 1) is configured to determine a weighting matrix G, whose number or rows is equal to the number of the transmitters; and further for determining a modified channel matrix equal to ***HG***; so as to construct a nonlinear precoder for performing nonlinear precoding of the modified channel matrix. The weighting matrix G can be square (i.e., where the number of rows equals the number of columns).

The weighting matrix G is diagonal having diagonal elements representing individual gains each associated with a particular one of the transmitters. In this case we denote the individual scalar factors by means of a vector α, where each vector element represents an individual scalar factor, each of which is associated with a particular CPE unit (i.e., elements in vector α represent specific power scaling factors that are distinctive for each one of the CPE units). Permutation of inputted information symbols 602 may be performed optionally. In this case, information symbols 602, ***s*,** are inputted into a permutation block 604, which is configured and operative to permute information symbol elements in vector ***s*,** according to a permutation matrix **IT.** Each information symbol element **in** vector ***s*** is associated with a particular CPE unit. The permutation takes into consideration the partition of CPE units into two groups (those that employ NLP and those that don't). Adder 610 combines an output from permutation block 604 with a perturbation vector 608 denoted by **c**, the result of which is a signal 612 that is inputted to a block 614, which int turn is configured and operative to perform the operation: $inv(R') * D$, and to output a signal 616 represented by a vector m. Signal 616 is inputted into a Q block 618 configured to perform QR decomposition the result of which is inputted into a gain block 619 denoted by $\alpha * G$. Gain block 619 is configured and operative to construct a diagonal square weighing matrix G, according to: $\tilde{G} = diag(\alpha)$ whose number of rows is equal to the number of transmitters, and to output an output signal 622 that is communicated over a communication channel, denoted by a block 624. Block 614 in conjunction with Q block 618, permutation block 604, block 619, and perturbation vector 608 collectively constitute a precoder 620, denoted by $\boldsymbol{P}$, which in turn is configured and operative to perform pseudo-inversion of a communication channel matrix $\boldsymbol{H}$. In the general case, implementation 600 of the disclosed technique uses QR decomposition to determine a modified channel matrix equal *to HG.* In a particular case involving permutations (i.e., a permutation block 604 Π'), where the processor is further configured for permuting information symbol elements in the information symbol vector, per subcarrier frequency, where each information symbol element is associated with a particular one of the CPE units. The disclosed technique utilizes an inverse of matrix $\boldsymbol{G}$ for **precoding,** and in particular, for factorizing the communication channel matrix $\boldsymbol{H}$ according to (in the specific case utilizing permutations):

$$\boldsymbol{H} = \Pi \boldsymbol{R}^{H} \boldsymbol{Q}^{H} \tilde{G}^{-1} \qquad (60).$$

A signal 626 is a result from propagation through communication channel 624, received at the receiver side. A received signal 632 at the receiver side, represented by a vector r is the sum of signal 626 *(y)* with additive noise 628 (represented by a vectors), signifying that the preceded information symbols propagated through the communication channel includes additive 630 noise. Received signal 632 is inputted to a block 634 configured and operative to perform scaling by inverting the permuted diagonal vector $\mathbb{D}$ the output of which is denoted by an output signal 636. For NLP supporting CPE units, output signal 636 represents an input to a modulo operation block 638, which is configured and operative to apply modulo arithmetic operation to received signal 636, and to output an output signal 640 of estimated symbols, represented by a vector $\hat{s}$. For those CPE units whose supportability does not include NLP (i.e., LP group of users) output signal 636 is effectively output signal 640 (i.e., no modulo operation is applied).

Alternatively, $\tilde{G}$ consists of a combination of a variable gain factor with a common gain factor (to all CPE units). This may be expressed by: $\delta = \alpha * \eta$, where a scalar η represents a common gain factor and the vector $\boldsymbol{\alpha}$ represents the variable gain factor (individual to each CPE unit). Thus, we may choose:

$$\tilde{\boldsymbol{G}} = \mathrm{diag}(\boldsymbol{\delta}) \qquad (61).$$

The QR decomposition in such case is calculated via the modified channel as follows:

$$H\tilde{G} = \boldsymbol{\Pi R}^{\mathrm{H}} Q^{\mathrm{H}} \qquad (62),$$

i.e., $\boldsymbol{n}^{\mathrm{H}}\boldsymbol{H\tilde{G}} = R^{\mathrm{H}}\boldsymbol{Q}^{\mathrm{H}}$, hence $\left(H\tilde{\boldsymbol{G}}\right)^{\mathrm{H}}\boldsymbol{\Pi} = \boldsymbol{QR}$ and thus $\tilde{G}^{\mathrm{H}}H^{\mathrm{H}}\boldsymbol{\Gamma}I = \boldsymbol{QR}$. The introduced gains do not affect channel diagonalization. They represent additional degrees of freedom, which may be used for the performance optimization.

Reference is now further made to Figure 17B, which is a schematic block diagram of a method for a specific implementation of nonlinear precoding utilizing different scalar factors, generally referenced 650, configured and operative in accordance with the disclosed technique. Method 650 initiates with procedure 652. In procedure 652 a weighting matrix C, whose number of rows is equal to the number of transmitters is determined, where nonlinear precoding of an information symbol is employed for conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels defining a channel matrix $\boldsymbol{H}$, over a particular subcarrier frequency. With reference to Figures 1 and 17A, processor 114 (Figure 1) executes block 620 (Figure 17A) by determining a weighting matrix G via block 619 (Figure 17A).

In procedure 654, a modified channel matrix equal to HG is determined. With reference to Figures 1 and 17A, processor 114 (Figure

1) executes block 620, specifically uses QR decomposition (blocks 614, Q block 618, and block 619) to determine a modified channel matrix equal to *EG.*

In procedure 656, a nonlinear precoder is constructed for performing nonlinear precoding of the modified channel matrix. With reference to Figures 1 and 17A, processor 114 (Figure 1) executes block 620 by constructing a precoder via blocks 614, 618 and 619 (Figure 17A).

Reference is now made to Figure 18, which is a schematic block diagram of a method for a hybrid-interoperability precoding scheme supporting both linear and nonlinear precoding, generally referenced 680, constructed and operative according to the embodiment of the disclosed technique. Method 680 initiates with procedure 682. In procedure 682, information pertaining to supportabilities of a plurality of receivers to decode nonlinearly preceded data is received by a plurality of transmitters. The transmitters are communicatively enabled to transmit an information symbol conveying data to the receivers via a plurality of communication channels over a subcarrier frequency, where the number of transmitters *(N)* is different than the number of active receivers *(K')* for that subcarrier frequency. With reference to Figures 1 and 3A, DPU 104 (Figure 1) (includes transceivers 116$_1$, 116$_2$, 116$_3$,...,116$_N$-i, 116$_N$) acquires supportabilities 120i, 12Q$_2$, 120$_3$,..., 120$_{N-1}$,120$_N$ (to decode at least one type of preceded data: LP data, NLP data, or both) respectively from receivers 106$_1$, 106$_2$, 106$_3$,..., 106$_N$-i,106$_N$ during initialization. DPU 104 (Figure 1) including its transceivers 116$_1$, 116$_2$, 116$_3$,..., 116$_{N-}$i, 116$_N$ are communicatively enabled to transmit information symbol 160A (represented by vector s, Figure 3A) conveying data to receivers 106i, 106$_2$, 106$_3$,..., **106N-I**,106$_N$ (Figure 1) via respective communication channels 108$_1$, 108$_2$, 108$_3$,..., 108$_{N-1}$,108$_N$ over a subcarrier frequency, where the number of transmitters is different than the number of active receivers for that subcarrier frequency. DPU 104 (Figure 1) continually determines (e.g., via pings, via bi-directiona! messages, etc.) activity levels 122_1, 122_2, 122_3,..., 122_{N-1},122_N (per tone) of each of CPE units 106!, 106_2, **IO63,..., 106N-I** ,106_n.

In procedure 684, a precoding scheme defining for which of the receivers the data to be transmitted by the transmitters shall be precoded using at least one of linear precoding and non-linear precoding, according to the supportabilities is determined. With reference to Figures 1, 3A, 4, 5, and 15, processor 114 (Figure 1) determines a precoding scheme that associates each information symbol element of an inputted symbol vector 152A (Figure 3A), *s,* to either LP or NLP, according to supportabilities **12G**$_1$, 120_2, $12O_3$,..., $120_{N\text{-}I}$,120_N (Figure 1). In a particular implementation, permutation block 204 (Figure 4) permutes information symbol elements in vector *s* into two successive aggregate groups (represented respectively by sub-vectors 254$_!$ and 254_2 in Figure 5), namely, a first aggregate group of symbol elements s_1 (LP path 502 in Figure 15) corresponding to the LP group of CPE units implementing LP, successively followed by a second aggregate group of symbol elements s_2 (NLP path 504, Figure 15) corresponding to the NLP group of CPE units implementing NLP.

In procedure 686, a signal is constructed by applying a reversible mapping to the information symbol, where the reversible mapping includes elements each respectively associated with a particular one of the receivers, such that those receivers supporting the decoding of nonlinearly precoded data are capable of reversing the reversible mapping to the information symbol, while for those receivers not supporting the decoding of nonlinearly precoded data the information symbol is unaffected by the reversible mapping. With reference to Figures 1, 3A, 4, 5, 6, and 10, processor 114 (Figure 1) constructs a signal 156A (Figure 3A) by applying a reversible mapping 154A (Figure 3), e.g., perturbation vector 208, *c* (Figure 4) and 252 (Figure 5) whose vector elements are defined by sub-vector 252_1 c_1: $c_1(1)$, $c_1(2)$,..., $c_1(K_1)$, and further by vector elements defined by sub-vector 252_2, c_2: $c_2(1)$, $c_2(2)$,.. ., $c_1(K_2)$ are each respectively associated with a particular one of CPE units 106_1, 106_2, 106_3,.. ., 106_{N-1},106_N (Figure 1) such that those receivers supporting the decoding of NLP data are capable of reversing the effect of the perturbation vector, while for those receivers not supporting the decoding of NLP data, the information symbol is unaffected by the reversible mapping, e.g., the perturbation sub-vector for that case: $c_1 = 0$ does not affect output signal 212 by adder 210 (Figure 4). Preprocessing block 306 (Figures 6, 10) and particularly NLP/LP control mechanisms 374i, 374_2, 374_3, 374_4, and 374_5 (Figure 10) control application of reversible mapping 154A (Figure 3) (e.g., a modu!a-Z adder, a perturbation vector) according to respective supportabilities 120i, 120_2, 120_3,.. ., 120_{N-1},120_N (Figure 1) of CPE units 106i, 106_2, 106_3,.. ., 106_N-i,1 06_N.

In procedure 888, a precoder characterized by $N \neq K$ is constructed, such that the precoder is configured to perform regularized generalized inversion of a communication channel matrix. With reference to Figures 1, 3A, and 4, processor 114 constructs a precoder 158A (Figure 3), e.g., precoder 220 (Figure 4) such that the precoder is configured to perform regularized inversion (e.g., equation (3B)) of a communication channel matrix 162A, H, (Figure 3A) and 224 (Figure 4).

The system and method of the disclosed technique combines linear and nonlinear precoding (precoders) and have the following advantages:

1. Providing a general solution to the interoperability problem between a data providing entity (e.g., a DPU having multiple transceivers) communicatively coupled with multiple data subscriber entities (e.g., CPE units) in a communication network, where part of the data subscriber entities do not support nonlinear precoding (NLP) while another part does. The general case is where the number of

transmitters *(N)* (i.e., transceivers operating as transmitters in the DL direction) is not necessarily equal to the number of active receivers (K) (per tone) (i.e., transceivers operating as receivers in the DL direction). The general solution also includes the special case where N is equal to K. This enhances interoperability of the communication system in whole imparting it with the ability to operate with receivers of different types of supportabilities.

Using LP for raising bitrates of inferior communication channels (i.e., those enabling a relatively low number of bits per constellation (compared with superior communication channels enabling a relatively high number of bits per constellation)). It is known **in** the art that the standard THP scheme may exhibit large power gain loss as well as coding gain loss for small constellations (e.g., 1 or 2 bits, and moderate losses for 3 or 4 bits). For these constellations, the disclosed technique optionally employs optimized LP to avert the aforesaid losses. The optimization of LP is essential and controlled via diagonal gains D_1 as well as by the degrees of freedom conferred the permutations $\mathbf{\Pi}_l$, since non-optimized LP introduces loses due to the use of power scaling per output. Optimization may generally achieve greater performance compared with simple scalar scaled ZF.

2. Facilitating a reduction in size of the NLP system (i.e., which effectively translates to the dimension of input vector s_2 which is defined to equal to the number of users undergoing or employing NLP), and in so doing, reducing the computational complexity required for determining the perturbation vector c_2. While size reduction may not pose a problem for systems employing the THP scheme, however, it may be an issue for schemes employing vector precoding. This may be of importance and interest since the determination of the perturbation vector may be computationally

expensive. The exact complexity is difficult to quantify and has been estimated to be between polynomial to exponential on the order of the search space size. Hence, the reduction in size of the NLP system may be very desirable. To determine the perturbation vector, the disclosed technique may typically employ algorithms, which include, for example sphere decoding, the Blockwise Korkine Zolotarev (BKZ) algorithm, and the like.

According to another embodiment of the disclosed technique it is thus provided a method and a system for NLP where the modulus value is constellation-independent. In the prior art, NLP techniques utilize a modulus value that is constellation-dependent. The disclosed technique proposes a method and system for nonlinear precoding of an information symbol at a given precoder input, the information symbol is in a symbol space (i.e., a construct for representing different symbols, e.g., a constellation) having a given symbol space size (e.g., a constellation size). The nonlinear precoding involves modulo arithmetic and has a plurality of inputs. The method includes initially the step of determining a reference symbol space size, which is common to all inputs. Then, the method determines a modulus value according to the reference symbol space size. The method adapts the given symbol space size according to the reference symbol space size. The step of adapting involves scaling the given symbol space size to the boundaries of the reference symbol space size. Subsequently the method nonlinearly precodes the information symbol according to the determined modulus value, common to all of the inputs. The disclosed technique further provides a system for nonlinear precoding of an information symbol at a given precoder input, where the information symbol is in a symbol space size having a given symbol space size. The nonlinear precoding involves modulo arithmetic and has a plurality of inputs. The system includes a controller and a processor. The controller is configured for determining a reference symbol space size that is common to all of the inputs, and for determining a modulus value according to the reference symbol space size. The processor is configured for adapting the given symbol space size according to the reference symbol space size, and for nonlinearly precoding the information symbol according to the determined modulus value common to all of the inputs.

The disclosed technique is implementable for various NLP techniques such as THP, including inflated lattice precoding techniques, which work with non-Gaussian interference and typically outperform THP for all SNR levels, including low SNRs.

In the traditional Tomlinson-Harashima precoding design, the modulo operation is defined per symbol space size (e.g., constellation size), and it is different for different symbol spaces (e.g., constellations). The modulo operation for user number *k* is denoted as $\Gamma_{3/4}[\chi]$, where $\Gamma_{\tau}[\chi]$ is (see e.g., Ginis and J. M. Cioffi, "A multi-user precoding scheme achieving crosstalk cancellation with applications b DSL systems," in *Proc. 34th Asi!omar Conf. Signals, Systems, and Computers,* Pacific Grove, CA, Oct. 2000, pp. 1627-1631) denoted by:

$$\Gamma_{\tau}[x] = x - \tau \left\lfloor \frac{x}{\tau} + \frac{1}{2} \right\rfloor \quad (63),$$

where $\lfloor \ldots \rfloor$ is the floor function, and for a real-valued x, whereas for complex values the modulo operation is performed separately for the real, $Re(x)$, and imaginary, $Irn(x)$, parts [we denote the imaginary unit as j]:

$$\Gamma_{\tau}[x] = x - \tau \left\lfloor \frac{Re(x)}{\tau} + \frac{1}{2} \right\rfloor - j\tau \left\lfloor \frac{Im(x)}{\tau} + \frac{1}{2} \right\rfloor \quad (64).$$

Reference is now made to Figures 19, 20, and 21. Figure 19 is a schematic block diagram of a system for nonlinear precoding exhibiting a modulus size that is constellation-independent, generally referenced 700, constructed and operative in accordance with another embodiment of the disclosed technique. Figure 20 is a schematic illustration detailing an example configuration of an internal structure of a vectoring processor in the system of Figure 19, generally referenced 720, constructed and operative in accordance with an embodiment of Figure 19 of the disclosed technique. Figure 21 is a schematic diagram of Tomlinson-Harashima preceding used per subcarrier frequency being applied to chosen users only, generally referenced 730, constructed and operative in accordance with an embodiment of Figure 19 of the disclosed technique. System 700 (Figure 19) includes a vectoring processor 702 (herein denoted interchangeably "processor") and a controller 704. Vectoring processor 702 includes a scaler 706 and a nonlinear precoder (NLP) 708. Controller 704 is typically implemented by a physical layer (PHY) controller of a DPU (e.g., DPU 104 of Figure 1).

Figure 20 shows an internal structure of vectoring processor 702 of system 700 including a mapper 722 coupled with vectoring processor 702. Vectoring processor 702 includes a preprocessing subblock 710, which in turn includes scaler 706, NLP 708, a plurality of adders 712_2, 712_3, 712_4, 712_4, a Q block 714, and a G block 716. Without loss of generality and for the purposes of simplicity, the internal structure of vectoring processor 702 in Figure 20 shows an implementation in the 5-user example, although the principles analogously extended and apply in the general case for N users. Scaler 706 includes a plurality of gain blocks 706i, 706_2, 706_3, 706_4, and 706_5. NLP 708 includes a plurality of modular arithmetic calculation units 708i, 708_2, 708_3, 708_4, and 708_5.

Information bits $B = \{h(l), b(2), b(3), h(4), b(5)\}$ (interchangeably denoted "data bits") are inputted into mapper 722, which is configured and operative to map the information bits (bit streams) into respective information symbols $\boldsymbol{x}$ =- {x(l),x(2),.r(3),x(4), x(5)}, which in turn are inputted into vectoring processor 702. Specifically, the information symbols are correspondingly inputted into individual gain blocks 706_1, 706_2, 706_3, 706_4, and 706_5 of preprocessing subblock 710 of vectoring processor 702. The gain blocks are configured and operative to apply respective gain components /'(1), $f(2)$, /'(3), $f'(4)$, and /(5) of gain a vector/ to the corresponding information symbols (i.e., index-wise) the outputs of which are inputted to adders 712_2, 712_3, 712_4, 712_4 (except for an output of gain block 706_1 that is directly provided into modular arithmetic calculation unit 708i). Each adder 712_2, 712_3, 712_4, 712_4 can be respectively viewed as a part of a respective modular arithmetic calculation unit 708_2, 708_3, 708_4, and 708_5. Modular arithmetic calculation units 708_1, 708_2, 708_3, 708_4, and 708_5 are generally configured and operative to facilitate construction of the diagonal matrix L_U in a recursively manner according to: Ly(fc, n) = $\frac{L(k,n)}{L(k,k)}$. Figure 21 illustrates a schematic diagram of Tomlinson-Harashima preceding (THP), employing recursive calculations (of matrix $L_{i;}$) used per subcarrier frequency being applied to chosen users only, constructed and operative in accordance with an embodiment of Figure 19 of the disclosed technique.

The disclosed technique formulates a way to set the modulo value that is employed in THP to be independent on the symbol space size (constellation size). In the prior art, and in particular with the Tomlinson-Harashima precoder, the modulo value is dependent on a symbol space size (e.g., the constellation size). Figure 21 shows information symbols x(I), x(2),.. ., $x(K)$ (or permuted information symbols) are inputted to the precoder having a plurality of inputs. K is the number of the active users. Figure 21 shows the recursive NLP procedure for a more general case of K active users (in comparison with the more specific 5-user example shown in Figure 20). The modulo sizes, which are denoted **in** Figure 21 as *M2,.. .,MK* (i.e., $M2 \equiv \tau_2, \ldots, MK \equiv \tau_K$) depend in the prior art on the constellation size of the input symbols. L is a lower-diagonal matrix (equal to R^H, where R is the upper-diagonal matrix obtained by QR decomposition of the permuted channel, i.e., $H^H\Pi = QR$ (see description hereinabove)). The number of the NLP users is designated K_2. The modulus value may be selected to be symbol space (constellation) dependent **(as** known in the prior art), or alternatively, symbol space (constellation) independent, as proposed by the disclosed technique, the particulars of which follow. For a mixture of linear and nonlinear precoders acting on the same subcarrier frequency, inputs *x(k),* represent an "impure" information symbol (optionally permuted) having an offset that takes into account the influence of the other users processed in previous step(s) via the linear precoder. This offset, as disclosed, need not be processed sequentially by processor 702, is determined at a processing stage of NLP 708. This facilitates a reduction in the burden of real-time performance since fewer operations are needed to be executed for real-time implementations.

A customary modulus value is $\tau = M\mathrm{d}_{\mathrm{min}}$ for pulse amplitude modulation (PAM) and $\tau = \sqrt{M}\mathrm{d}_{\mathrm{min}}$ for a two-dimensional (2-D) square symbol space (constellations) of digital quadrature amplitude modulation (QAM), where d_{min} is the quantized symbol position (constellation point) spacing. See for example G. Ginis and J. M. Cioffi, "A multi-user precoding scheme achieving crosstalk cancellation with applications to DSL systems," in *Proc. 34th Asilomar Conf. Signals, Systems, and Computers,* Pacific Grove, CA, Oct. 2000, pp. 1627-1631 , and the discussion for square constellations. For 4-QAM the modulus value is $\tau = 2\mathrm{d}_{\mathrm{min}}$ (due to V4=2), and for 16-QAM the modulus value is $\tau = 4\mathrm{d}_{\mathrm{min}}$ (while the symbol positions (constellation points) in the symbol space are traditionally located at integer positions (n_1, n_2) taking independently the values {-3, -1,1,3})- Other, non-integer even values, are different for different symbol space sizes (constellation sizes), where $\mathrm{d}_{\mathrm{min}}$ (i.e., a symbol position spacing for a symbol space).

The average energy, $\bar{E}$ (or *(E))* in M-QAM, where *M* is the number of symbol positions (e.g., constellation points) in the symbol space (e.g., constellation) and is square (i.e., for square symbol spaces (constellations)) is given by: $\bar{E}$ = (M - l)$d_{min}{}^{2}/6$, thus the average energy for 4-QAM with constellation points $\pm 1 \pm j$, where M = 4 (points) and

$d_{min} = 2$ is: $(4 - 1) * \frac{2^2}{6} = 3 * \frac{4}{6} = 2$. and the constellation square boundary is at 2 (and hence the initial modulo value is 4). The average energy for a symbol space size (constellation) having M-symbol positions (constellation points) is given by:

$$\bar{E} = \frac{1}{M}\sum_{m=1}^{M}|z_m|^2 = \frac{1}{M}\sum_{m=1}^{M}(x_m^2 + y_m^2) \qquad (65),$$

where the m-th constellation point is expressed as $z_m = x_m + jy_m$ and x_m and y_m are respectively real and imaginary coordinate values in the complex plane. For 16-QAM, the average energy for we obtain:

$$\bar{E} = \frac{4}{16}\sum_{m=1}^{4}(x_m^2 + y_m^2) = \frac{1}{4}\{2 + 10 + 10 + 18\} = 10 \qquad (66).$$

Reference is now further made to Figure 22A, which is a schematic diagram showing an example of a non-scaled 4-QAM constellation diagram, generally referenced 740, constructed and operative in accordance with the disclosed technique. Particularly, Figure 22A (as well as with Figures 22C, and 22D) illustrates a scatter diagram of symbols (dots) distributed in quantized symbol positions (constellation points) in a symbol space (constellation) that is represented on a complex plane having a real axis I-axis ("in phase") and an imaginary axis Q-axis ("quadrature"). The square-shaped dotted lines represents a boundary (or frame) of the symbol space. A symbol space size is defined by its boundary. The symbol space size (constellation size) is scaled by the root of the average energy, which yields: $\tau_{4-QAM} = \frac{4}{\sqrt{2}} = 2\sqrt{2}$. For 64-QAM it is readily observed that the average energy is equal to 10 and the initial modulo size is 8. Scaling of the initial modulus value by the root of the average energy yields $\tau_{16QAM} = \frac{8}{\sqrt{10}}$. Thus, unit energy scaling produces closed but different modulus values for every symbol space (constellation) which are mutually exclusive (i.e., they may include irrational numbers, such as: $\frac{(2\sqrt{2})}{\left(\frac{8}{\sqrt{10}}\right)} = \frac{\sqrt{5}}{2}$). It is noted that if one approximates the symbol space (constellation) square with a large number of uniformly distributed integer symbol positions (constellation points), then the modulo value of the symbol space (constellation) with unit average energy would equal $\tau = \sqrt{6}$ (i.e., for M-QAM the un-scaled $\tau_{unscaled}$ = VMd$_{min}$ and after **rescaling** τ with $\sqrt{E}$ we obtain: $\tau = \frac{\tau_{unscaled}}{\sqrt{F}} = \frac{\sqrt{M}d_{min}}{\sqrt{\frac{(M-1){d_{min}}^2}{6}}} = \sqrt{\frac{M}{M-1}} \cdot \sqrt{6}$. For large M we obtain $\tau \rightarrow$ V6). For finite and especially small values for M the scaled modulus value is symbol space size (constellation size) M-dependent, (i.e., consequently depending on d$_{min}$). Such moduli are a costly operation and they may be made much less costly as discussed hereinbelow. Reference is now further made to Figures 22B and 22C. Figure 22B is a schematic diagram showing an example of non-scaled 16-QAM symbol space (constellation), generally referenced 742. Figure 22C is a schematic illustration showing a boundary or frame of a modulo operation representing a square of size τ, generally referenced 744. This frame corresponds to a reference symbol space size. Controller 704 (Figure 19) is configured and operative to determine this reference symbol space size that is common to ail inputs x(l), x(2),.. ., $x(K)$ of the NLP 720 (Figures 20 and 21). The initial modulus value equals 8. Controller 704 is configured and operative to further determine a modulus value according to the determined reference symbol space size.

According to a typical implementation, controller 704 determines the reference symbol space size that is a maximum symbol space size thereby enabling the transmission of a maximum number of bits per information symbol. Vectoring processor 702 and particularly scaler 706 (Figure 19), having individual gain blocks 706i, 706_2, 706_3, 706_4, and 706_5, is configured and operative to adapt (e.g., scale) the given symbol space size according to the determined modulus value. The process of adapting of involves scaling of symbol positions (constellation points) in the symbol space according to the reference symbol space size.

The reference symbol space size can be selected to be numerically equal to $2\sqrt{M}$, where M is a maximal number of symbol positions (constellation points) in the symbol space. The modulus value can be selected to be numerically equal to said reference symbol space size.

NLP 708 (Figure 19), having individual modular arithmetic calculation units 708_1, 708_2, 708_3, 708_4, and 708_5 (Figure 20) is configured and operative to nonlinearly precode inputted information symbol $x(1)$, $x(2)$,..., $x(K)$ (Figure 21) according to the determined modulus value that is common to all of the inputs of NLP 708.

Q block 714 is configured to perform QR decomposition the result of which is inputted into G block 716. G block 716 is a gain scaling block configured and operative to apply gain scaling and to output signals (not shown) that are communicated over a communication channel (e.g., 108_1,...,108_N, Figure 1). G block 716 is typically used for gain scaling (e.g., power normalization) so to apply the same gain to all outputs. Alternatively, G block 716, which can be represented by a vector G, has vector elements that can be different for each output, for applying different gain scaling to different outputs. G block 716 is dependent on the number of lines L.

The system and method of disclosed technique uses of same modulus value for any symbol space size (constellation size). This serves several purposes:

1. It scales all constellations automatically to the same power. This follows from known observations that THP transmission results in homogeneous distribution of points over the whole constellation area (see short discussion of this in, e.g., Ginis and J. M. Cioffi, "A multi-user precoding scheme achieving crosstalk cancellation with applications to DSL systems," in *Proc. 34th Asiiomar Conf. Signals, Systems, and Computers,* Pacific Grove, CA, Oct. 2000, pp. 1627-1631). It is stressed that the constraint of the average energy of the

constellation points equaling one is a mathematical statement but not directly related to the actual transmitted energy of the THP per output. This is a typical scaling employed in digital communications. The constellation square size (for QAM modulation) determines the average energy per THP communication channel (e.g., line, antenna) output. If all squares are designed to be of the same size, the average power consumption will be the same. The constellation points inside of this square are determined from the standard requirement that the minimal distance from a constellation point to the square border is equal to half of the minimal distance between constellation points. For example, if the modulo value is chosen to be $2^7 = 2 * 64 = 128$, then for the traditional points of 16-QAM which are $n_1 + jn_2$ with $n_1, n_2\ E$ {-3, -1,1,3} and the square boundary placed at coordinates ±4 so that the modulus is $2 * 4 = 8$ along the real and imaginary axes, therefore the are scaling factor is $(\frac{128}{8}) = 16$. Hence, the new locations of the 16-QAM constellation points are at $n_1 + j\,n_2$ with $n_1, n_2 \in \{-3 \cdot 16, -16 \cdot 1, 16 \cdot 1, 3 \cdot 16\}$ = {-48, -16, 16, 48}. Note, that multiplication by 16 may be achieved efficiently just as a shift by 4 binary positions. The above chosen modulo example namely $\tau = 128$ is convenient for constellations up to 4096-QAM (i.e., representing 12-bits, since $2^{12} = 4096$), which can be validated via the above-mentioned general relation $\tau = 2\sqrt{M} = 2\sqrt{2^{12}} = 2 * 2^6 = 128$); the points of this constellation, denoted as $n_1 + j\,n_2$, are integers $n_1, n_2\ E$ {-63, -61, ..., -1,1, ...,61,63}. If a 14-bit constellation is to be included a modulus value of: $2 \cdot \sqrt{2''} = 2^8 = 256$ is proposed. The 12 and 14-bit constellations mentioned above are given as examples. These are the current maximum constellation sizes in the current G.fast standard. In another example, for a modulo value of 8, symbol positions (constellation points) for 16-QAM (are not needed to be scaled) and 4-QAM (which needs to be scaled by factor of 2, is again effectively represented by a shift), with reference now being made to Figure 22D, which is a schematic diagram showing an example of $\tau = 8 = 2^3$ chosen as a constant modulo for all constellations, generally referenced 746. Such constellations are 16-QAM designated by black points in Figure 22D, which is not scaled, since it is in its natural size, and the 4-QAM designated by white points/rings. The 4-QAM is scaled by factor of 2 (where the scaling is applied via a shift and there's no need in multiplication). Specifically, for modulo values of 2, processor 702 performs the modulus operation for binary represented values in a very efficient form (e.g., shift operations that do not require use of multiplication operations or associated hardware). The modulus operation performed on complex numbers (representing constellation points), which are outside of the constellation square scales these constellation points into the constellation square. The scaling of constellations to the unit energy is performed by using a scalar gain factor a. For an illustrative example, if $\tau = 128$, which corresponds to the maximum number of bits in the constellation of 12-bits, then α = V6/128 performs the desired scaling to the unit energy. Similarly, when the maximum number of bits in the constellation is 14, we obtain $\tau = 2 * \sqrt{2^{14}} = 256$ then α = V6/256. As an option, the precoder matrix ***L*** may incorporate α as a part or the whole.

2. it simplifies the system since there's no need to take into account the symbol space size (constellation size) for performing modulo operation in decoding and encoding operations. Having the same modulus value for all symbol spaces makes these operations independent from applied permutations, which would otherwise have to be taken into account.

3. It simplifies hardware implementations. For example, if the modulo has a degree of 2 : $\tau = 2^p$ (where the degree p is a positive integer), the divisions in the above moduli relations may be re-substituted for hardware efficient shift operations. Further hardware specific simplifications, may involve mask techniques.
4. The proposed modulo operation is both applied at the centralized transmission unit (DPU 104, Figure 1) performing the broadcast (i.e. performing the downlink transmission) as well as at the receivers (CPE units 106, . , 106_2, 106_3,..., 106_{N-1}, 106_N, Figure 1). The constellations for the NLP group of users and optionally the LP group of users are scaled to the same size and then the same modulo (e.g., $\tau = 256$) is applied at the transmitter side (DPU unit) for the NLP group of users. At the receiver side, the NLP group of users employs the same modulo size for any constellation.
5. This implementation is more general than the transmission to the decentralized receivers and it may be used in any up-link or down-link (or both) for virtually any MIMO system that employs the THP (and vector precoding) performing modulo operations at both receiver and transmitter sides.

Note that while "conventional" modulo arithmetic operation may be performed according to $d_{min} * \sqrt{M}$, the system of method of the disclosed technique typically select to perform this operation according to $2\sqrt{M_{max}}$ where M_{max} represents the maximal number of symbol positions M for a modulation scheme (e.g., M-ary QAM) of a communication standard (e.g., 12 or 14 for G.fast).

Reference is now made to Figure 23, which is a schematic block diagram of a method for nonlinear precoding where the modulus size is constellation-independent, generally referenced 800, constructed and operative in accordance with the embodiment of Figure 19 of the disclosed technique. Method 800 initiates in procedure 802. In procedure

802 a reference symbol space size thai is common to all inputs of nonlinear precoding involving modulo arithmetic is determined. The nonlinear precoding is of an information symbol at a given precoder input, the information symbol is in a symbol space having a given symbol space size. With reference to Figures 19, 20, 21, and 22C, controller 704 (Figure 19) determines a reference symbol space size 744 that is common to all inputs x(l), *x(2),.. ., x(K)* (Figures 20, and 21) of NLP 708 (Figures 19 and 20) and NLP 720 (Figures 20 and 21) involving modulo arithmetic (shown by recursive calculation blocks in Figures 20 and 21). The information symbol is in a symbol space (e.g., Figures 22A, 22B, 22D) having a given symbol space size 744 (Figure 22C) (constellation size τ).

In procedure 804 a modulus value is determined according to the reference symbol space size). With reference to Figures 19 and 22B, controller 704 (Figure 19) determines a modulus value according to reference symbol space size τ 744 (Figure 22C).

In procedure 806, the given symbol space size is adapted according to the reference symbol space size. With reference to Figures 19, 20, 21C, and 22D, scaler 706 (Figures 19 and 20) adapts (e.g., scales) the given symbol space size according to the determined reference symbol space size in procedure 802. The determined modulus value for all symbol spaces (constellations) is depicted by 746 in Figure 22D.

In procedure 808, the information symbol is nonlinearly precoded according to the modulus value that is common to all of the inputs. With reference to Figures 19, 20, and 21, NLP 708 (Figures 19 and 20) and NLP 720 (Figures 20 and 21) nonlinearly precode the information symbol (i.e., for each of the inputted information symbols x(l), *x(2),.. .,* x(K)) according to the determined modulus value that is common to all of the inputs.

System 700 and method 800 are configured and operative for utilization in a multiple-user multiple-input-multiple-output {MiMO) matrix channel environment, employing NLP, for normalizing the given symbol space size according to the determined modulus symbol space size (i.e., fundamentally, based on the boundary and not on energy considerations).

It will be appreciated by persons skilled in the art that the disclosed technique is not limited to what has been particularly shown and described hereinabove. Rather the scope of the disclosed technique is defined only by the claims, which follow.

CLAIMS

1. A method for **precoding** an information symbol conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels over a subcarrier frequency, the number of transmitters (A/) is different than the number of active receivers *(K)* for that subcarrier frequency, the method comprising:

    receiving by said transmitters information pertaining to supportabilities of said receivers to decode non-linearly precoded data;

    determining a precoding scheme defining for which of said receivers said data to be transmitted by said transmitters shall be precoded using at least one of linear precoding and non-linear precoding, according to said supportabilities;

    constructing a signal by applying a reversible mapping to said information symbol, said reversible mapping includes elements each respectively associated with a particular one of said receivers, such that those said receivers supporting the decoding of said non-linearly precoded data are capable of reversing said reversible mapping to said information symbol, while for those said receivers not supporting the decoding of said non-linearly precoded data said information symbol is unaffected by said reversible mapping;

    constructing a precoder characterized by $N \neq K$ such that said precoder is configured to perform regularized generalized inversion of a communication channel matrix.

2. The method according to claim 1, further comprising determining which of said receivers constitutes an active receiver per said subcarrier frequency.

3. The method according to claim 2, wherein each said active receiver is characterized by an activity level defined by one of:

(a) is switched on and ready to receive said data at a particular said subcarrier frequency;

(b) is switched on and is **in** a process of receiving said data at a particular said subcarrier frequency;

(c) is either one of (a) and (b) but not for other said subcarrier frequency; and

(d) is either one of (a), **(b),** and (c) stipulated by a decision rule determined by at least one criterion related to communication performance.

4. The method according b claim 1, wherein said communication channels facilitate propagation of said signal, and are selected from a list consisting of: wire-lines; and antennas in wireless techniques.

5. The method according to claim 1, wherein said supportabilities further include capability for decoding linearly precoded data.

8. The method according to claim 1, wherein said receiving of said information pertaining to said supportabilities is acquired in an initialization process between said transmitters and said receivers.

7. The method according to claim 1, wherein said receiving of said supportabilities is determined independently from said receivers.

8. The method according to claim 1, wherein said reversible mapping is selected from a list consisting of:

at least one of a function and an algorithm that can be reversed so as to yield an operand of said reversible mapping;

an association between two sets S_1 and S_2 in which every member of S_1 there is an associated member in S_2, and for every member in S_2 there is the same associated member in S_1;

an entity that is reversible via application of modulo arithmetic; and

a perturbation vector.

9. The method according to claim 8, wherein said reversible mapping is said perturbation vector, where said elements in said perturbation vector associated with said receivers not supporting the decoding of said non!inearly precoded are zero, and said perturbation vector includes at least one nonzero element associated with respective at least one said receivers supporting the decoding of said nonlinearly precoded data.

10. The method according to claim 1, wherein said regularized generalized inversion is performed according to: $\boldsymbol{P = A^H(HA^H + fiV)^{-1}}$, where β is a regularization factor, and 1 is a regularization term.

11. The method according to claim 1, further comprising precoding according to constructed said precoder; and transmitting said signal by said transmitters via said communication channels.

12. The method according to claim 1, further comprising reversing said reversible mapping by said receivers supporting the decoding of said non!inear!y precoded data.

13. The method according to claim 12, wherein said reversing of said reversible mapping is performed by modulo arithmetic.

14. The method according to claim 1, further comprising permuting information symbol elements in said information symbol per said subcarrier frequency, each information symbol element is associated with a particular one of said receivers.

15. The method according to claim 14, wherein said permuting involves grouping said information symbol elements into distinct and successive aggregate groups according to said supportabilities of respective said receivers.

16. The method according to claim 15, wherein said distinct and successive aggregate groups include a linear **precoding** (LP) group not supporting the decoding of said nonlinear precoded data, and a nonlinear precoding (NLP) group supporting the decoding of said nonlinear precoded data.

17. The method according to claim 15, further comprising permuting said information symbol elements in each said distinct and successive aggregate groups.

18. The method according to claim 14, wherein said permuting is performed according to either one of prior to applying said reversible mapping; and after applying said reversible mapping.

19. The method according to claim 14, wherein said precoder performs said regularized generalized inversion of a permuted channel matrix.

20. The method according to claim 14, further comprising applying power scaling to said signal constructed, following said permuting.

21. The method according to claim 14, further comprising applying power scaling to a received signal by said receivers.

22. The method according to claim 9, wherein at least one of said elements in said perturbation vector associated with said receivers supporting the decoding of said nonlinearly preceded data is forced to be zero for performance considerations, thereby effectively rendering those said receivers for decoding said linearly precoded data.

23. The method according to claim 11, further comprising applying a power scaling factor common to all of said receivers after said preceding and before said transmitting.

24. The method according to claim 23, further comprising determining said power scaling factor for optimizing performance, and meeting power constraints per said communication channel.

25. The method according to claim 3, further comprising storing said supportabilities, and said activity levels corresponding to said receivers.

26. The method according to claim 11, further comprising applying a specific power scaling factor that is distinctive to each one of said receivers.

27. A hybrid precoder system for precoding an information symbol conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels over a subcarrier frequency, the number of transmitters *(N)* is different than the number of active receivers (*K*) for that subcarrier frequency, the hybrid precoder system comprising:

a controller configured for receiving information pertaining to supportabilitses of said receivers to decode non-linearly preceded data, and for determining a precoding scheme defining for which of said receivers said data to be transmitted by said transmitters shall be precoded using at least one of linear precoding and non-linear precoding, according to said supportabilities; and

a processor configured for constructing a signal for transmission, according to determined said precoding scheme, by applying a reversible mapping to said information symbol, said reversible mapping includes elements each respectively associated with a particular one of said receivers, such that those said receivers supporting the decoding of said non-linearly precoded data are capable of reversing said reversible mapping to said information symbol, while for those said receivers not supporting the decoding of said non-linearly precoded data said information symbol is unaffected by said reversible mapping, said processor constructing a precoder characterized by $N \neq K$ such that said precoder is configured to perform regularized generalized inversion of a communication channel matrix.

28. The system according to claim 27, wherein said controller is further configured for determining which of said receivers constitutes an active receiver per said subcarrier frequency.

29. The system according to claim 28, wherein each said active receiver is characterized by an activity level defined by one of:

(a) is switched on and ready to receive said data at a particular said subcarrier frequency;

(b) is switched on and is in a process of receiving said data at a particular said subcarrier frequency;

(c) is either one of (a) and **(b)** but not for other said subcarrier frequency; and

(d) is either one of (a), **(b),** and (c) stipulated by a decision rule determined by at least one criterion related to communication performance.

30. The system according to claim 27, wherein said communication channels are configured to facilitate propagation of said signal, and are selected from a list consisting of: wire-lines; and antennas in wireless techniques.

31. The system according to claim 27, wherein said supportabilities further include capability for decoding linearly preceded data.

32. The system according to claim 27, wherein said controller is further configured for receiving said information pertaining to said

supportabilities in an initialization process between said transmitters and said receivers.

33. The system according to claim 27, wherein said controller is further configured to receive said information pertaining to said supportabilities independently from said receivers.

34. The system according to claim 27, wherein said reversible mapping is selected from a list consisting of:

at least one of a function and an algorithm that can be reversed so as to yield an operand of said reversible mapping;

an association between two sets S.i and S_2 in which every member of S_1 there is an associated member in S_2, and for every member in S_2 there is the same associated member in Si;

an entity that is reversible via application of modulo arithmetic; and

a perturbation vector.

35. The system according to claim 34, wherein said reversible mapping is said perturbation vector, where said elements in said perturbation vector associated with said receivers not supporting the decoding of said nonlsnearly precoded are zero, and said perturbation vector includes at least one nonzero element associated with respective at least one said receivers supporting the decoding of said nonlinearly precoded data.

36. The system according to claim 27, wherein said processor performs said regularized generalized inversion according to: $P = A^H(HA^H + \beta \mathbb{1})^{-1}$. where β is a regularization factor, and 1 is a regularization term.

37. The system according to claim 27, wherein said system is configured to precode constructed said precoder; and to transmit said signal to said receivers via said communication channels.

38. The system according to claim 27, wherein said receivers supporting the decoding of said nonlinearly preceded data are configured for reversing said reversible mapping.

39. The system according to claim 38, wherein said receivers are configured to employ modulo arithmetic for said reversing said reversible mapping.

40. The system according to claim 27, wherein said processor is further configured for permuting information symbol elements in said information symbol per said subcarrier frequency, each information symbol element is associated with a particular one of said receivers.

41. The system according to claim 40, wherein said permuting involves grouping said information symbol elements into distinct and successive aggregate groups according to said supportabilities of respective said receivers.

42. The system according to claim 41, wherein said distinct and successive aggregate groups include a linear precoding (LP) group not supporting the decoding of said nonlinear precoded data, and a nonlinear precoding (NLP) group supporting the decoding of said nonlinear precoded data.

43. The system according to claim 41, wherein said processor is further configured to further permute said information symbol elements in each said distinct and successive aggregate groups.

44 The system according to claim 40, wherein said permuting is performed according to either one of: prior to applying said reversible mapping; and after applying said reversible mapping.

45. The system according to claim 40, wherein said precoder performs said regularized generalized inversion of a permuted channel matrix.

46. The system according to claim 40, wherein said system is configured to for applying power scaling to said signal constructed, following said permuting.

47. The system according to claim 40, wherein said receivers are configured for applying power scaling to received signals.

48. The system according to claim 35, wherein said processor constrains at least one of said elements in said perturbation vector associated with said receivers supporting the decoding of said nonlinearly precoded to be zero for performance considerations, thereby effectively rendering those said receivers for decoding said linearly precoded data.

49. The system according to claim 37, wherein said transmitters are configured to apply a power scaling factor common to all of said receivers after said precoding and before said transmitting.

50. The system according to claim 49, further comprising determining said power scaling factor for optimizing performance, and meeting power constraints per said communication channel.

51. The system according to claim 29, further comprising a memory configured for storing said supportabilities, and said activity levels corresponding to said receivers.

52. The system according to claim 37, further comprising applying a specific power scaling factor that is distinctive to each one of said receivers.

53. A method for nonlinear precoding of an information symbol at a given precoder input, the information symbol is in a symbol space having a given symbol space size, the nonlinear precoding involving modulo arithmetic and having a plurality of inputs, the method comprising:

determining a reference symbol space size which is common to all of said inputs;

determining a modulus value according b said reference symbol space size;

adapting said given symbol space size according to said reference symbol space size; and

nonlinearly precoding said information symbol according to said modulus value, common to all said inputs.

54. The method according to claim 53, wherein said adapting involves scaling of constellation points in said symbol space according to said reference symbol space size.

55. The method according to claim 53, wherein said modulus value is equal to said reference symbol space size.

56. The method according to claim 53, wherein said reference symbol space size is equal to $2\sqrt{M}$, where M is a maximal number of constellation points in said symbol space.

57. The method according to claim 53, where said reference symbol space size is a maximum symbol space size that enables transmission of a maximum number of bits per said information symbol.

58. A system for nonlinear precoding of an information symbol at a given precoder input, the information symbol is in a symbol space having a given symbol space size, the nonlinear precoding involving modulo arithmetic and having a plurality of inputs, the system comprising:

a controller configured for determining a reference symbol space size which is common to all of said inputs, and for determining a modulus value according to said reference symbol space size;

a processor configured for adapting said given symbol space size according to said reference symbol space size, and for nonlinearly precoding said information symbol according to said modulus value, common to all said inputs.

59. The system according to claim 58, wherein said adapting involves scaling of constellation points in said symbol space according to said reference symbol space size.

60. The system according to claim 58, wherein said modulus value is equal to said reference symbol space size.

61. The system according to claim 58, wherein said reference symbol space size is equal to $2\sqrt{M}$, where ***M*** is a maximal number of constellation points in said symbol space.

62. The system according to claim 58, where said reference symbol space size is a maximum symbol space size that enables transmission of a maximum number of bits per said information symbol.

63. A method for nonlinear precoding an information symbol conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels defining a channel matrix ***H*** over a particular subcarrier frequency, the method comprising:

determining a weighting matrix ***G***, whose number of rows is equal to the number of said transmitters;

determining a modified channel matrix equal to ***H G***; and

constructing a nonlinear precoder for performing nonlinear precoding of said modified channel matrix.

64. The method according to claim 63, wherein said weighting matrix ***G*** is diagonal having diagonal elements representing individual gains each associated with a particular one of said transmitters.

65. The method according to claim 63, further comprising permuting information symbol elements in said information symbol per said

subcarrier frequency, each information symbol element is associated with a particular one of said receivers.

66. A system for nonlinear precoding an information symbol conveying data for transmission between a plurality of transmitters and a plurality of receivers via a plurality of communication channels defining a channel matrix $\boldsymbol{H}$ over a particular subcarrier frequency, the system comprising:

a processor configured for:

determining a weighting matrix $\boldsymbol{G}$, whose number of rows is equal to the number of said transmitters;

determining a modified channel matrix equal to HG; and

constructing a nonlinear precoder for performing nonlinear precoding of said modified channel matrix.

87. The system according to claim 66, wherein said weighting matrix G is diagonal having diagonal elements representing individual gains each associated with a particular one of said transmitters.

68. The system according to claim 66, wherein said processor is further configured for permuting information symbol elements in said information symbol per said subcarrier frequency, each information symbol element is associated with a particular one of said receivers.

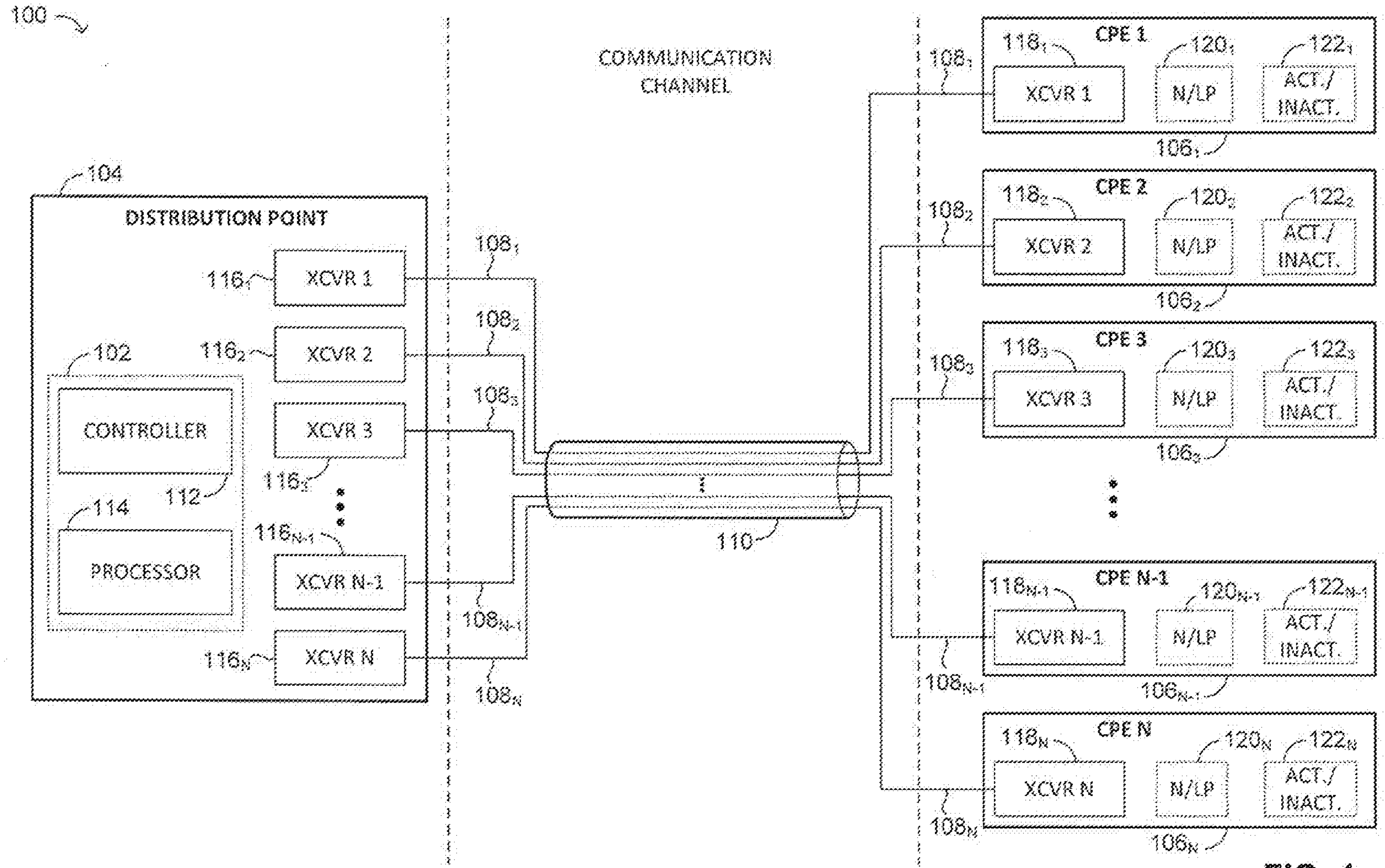

100
104
DISTRIBUTION POINT
102
CONTROLLER
112
114
PROCESSOR
116$_1$
XCVR 1
116$_2$
XCVR 2
XCVR 3
116$_3$
116$_{N-1}$
XCVR N-1
116$_N$
XCVR N
108$_1$
108$_2$
108$_3$
108$_{N-1}$
108$_N$
COMMUNICATION CHANNEL
110
CPE 1
118$_1$
XCVR 1
120$_1$
N/LP
122$_1$
ACT./ INACT.
106$_1$
CPE 2
118$_2$
XCVR 2
120$_2$
N/LP
122$_2$
ACT./ INACT.
106$_2$
CPE 3
118$_3$
XCVR 3
120$_3$
N/LP
122$_3$
ACT./ INACT.
106$_3$
CPE N-1
118$_{N-1}$
XCVR N-1
120$_{N-1}$
N/LP
122$_{N-1}$
ACT./ INACT.
106$_{N-1}$
CPE N
118$_N$
XCVR N
120$_N$
N/LP
122$_N$
ACT./ INACT.
106$_N$

FIG. 1

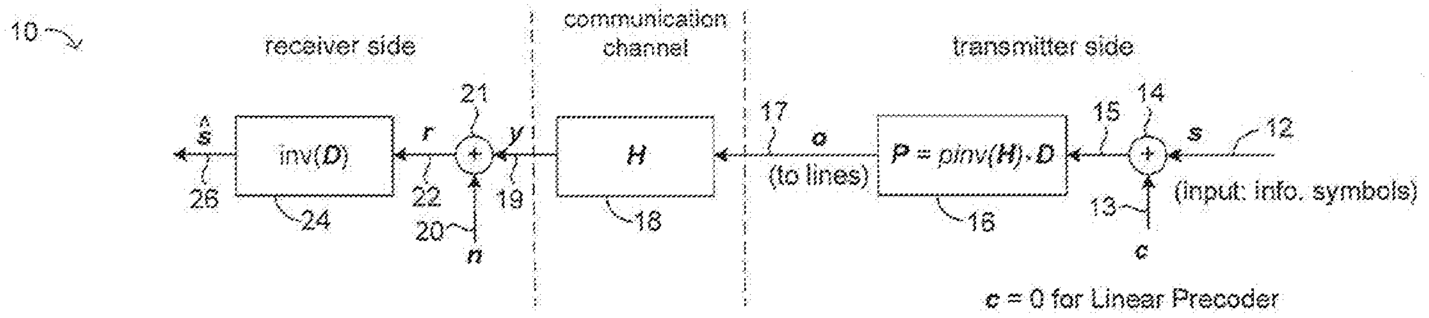


FIG. 2A
(PRIOR ART)

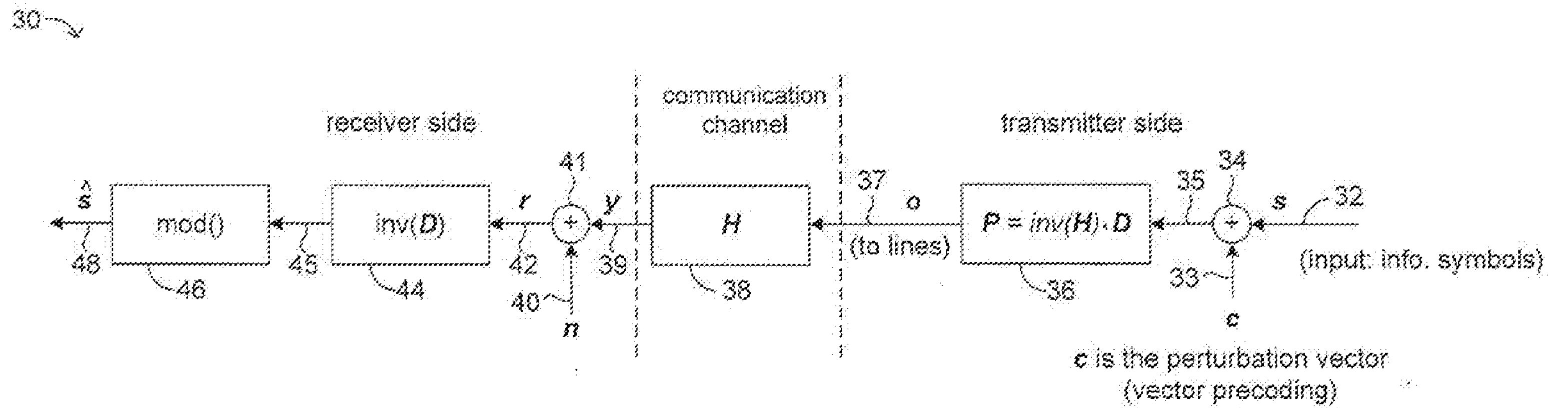


FIG. 2B
(PRIOR ART)

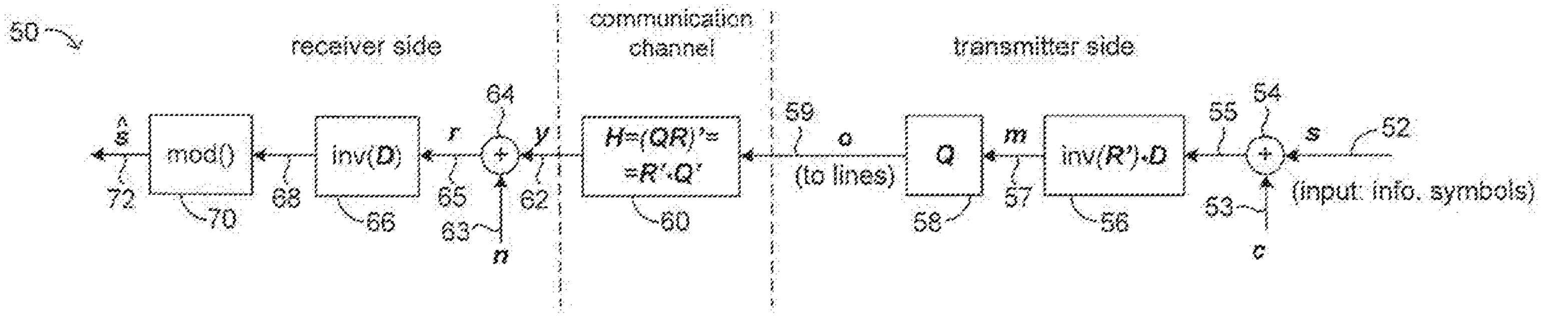


FIG. 2C
(PRIOR ART)

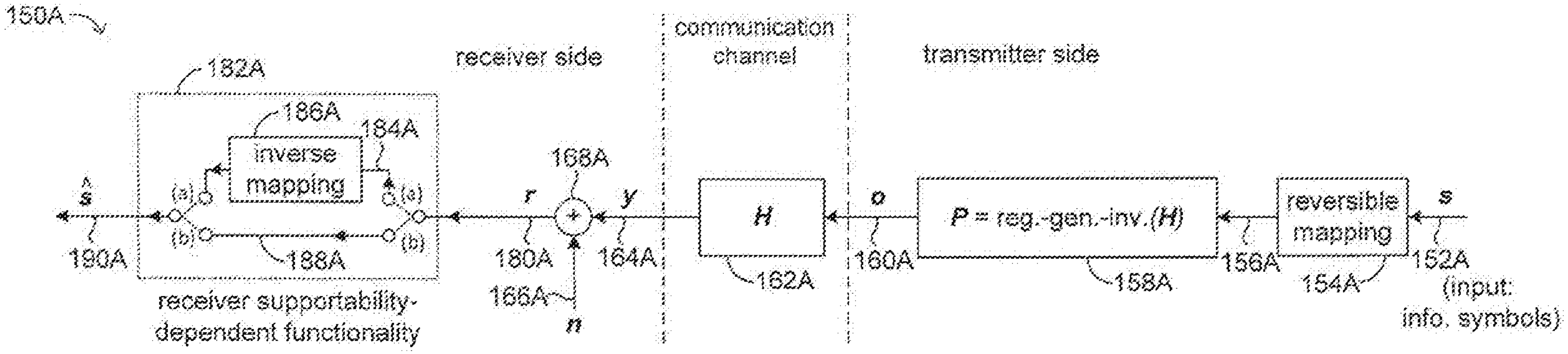


FIG. 3A

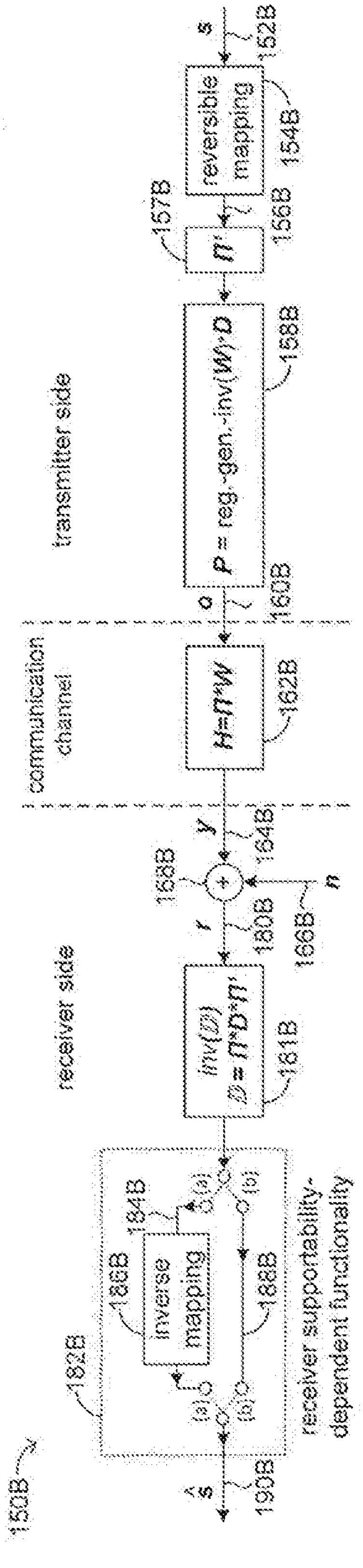

150B
receiver side
communication channel
transmitter side
ŝ
190B
182B
(a)
(b)
186B
inverse mapping
184B
(a)
(b)
188B
receiver supportability-dependent functionality
inv(𝔻)
𝔻 = Π*D*Π'
181B
r
180B
158B
+
y
164B
n
166B
H=Π*W
162B
o
160B
P = reg.-gen.-inv(W)·D
158B
157B
Π'
156B
reversible mapping
154B
s
152B

FIG. 3B

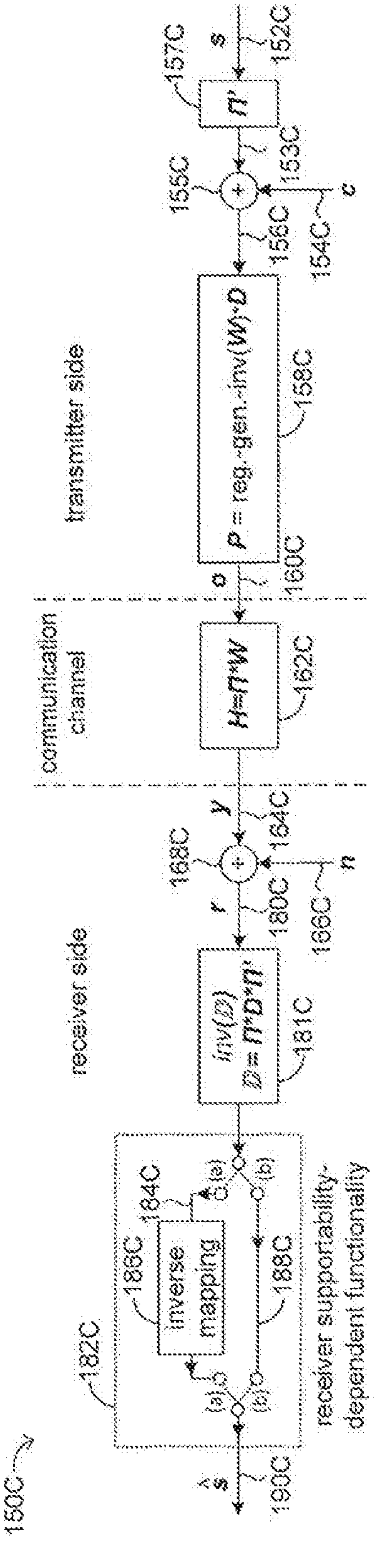

150C
receiver side
communication channel
transmitter side
ŝ
190C
182C
(a)
(b)
186C
inverse mapping
184C
(a)
(b)
188C
receiver supportability-dependent functionality
inv(𝔻)
𝔻 = Π*D*Π'
181C
r
180C
168C
+
y
164C
n
166C
H=Π*W
162C
o
160C
P = reg.-gen.-inv(W)·D
158C
155C
+
156C
c
154C
153C
157C
Π'
s
152C

FIG. 3C

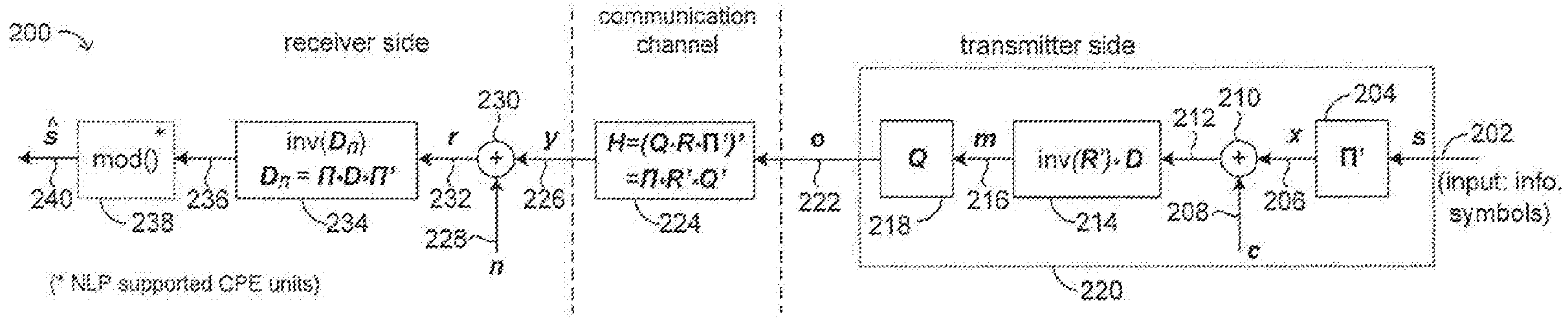

200
receiver side
communication channel
transmitter side
$\hat{s}$
240
mod() *
238
236
inv(D_n)
$D_n = \Pi \cdot D \cdot \Pi'$
234
r
232
230
+
228
n
y
226
$H=(Q \cdot R \cdot \Pi')'$
$=\Pi \cdot R' \cdot Q'$
224
o
222
Q
218
m
216
inv(R')·D
214
212
210
+
208
c
x
206
Π'
204
s
202
(input: info. symbols)
220
(* NLP supported CPE units)

FIG. 4

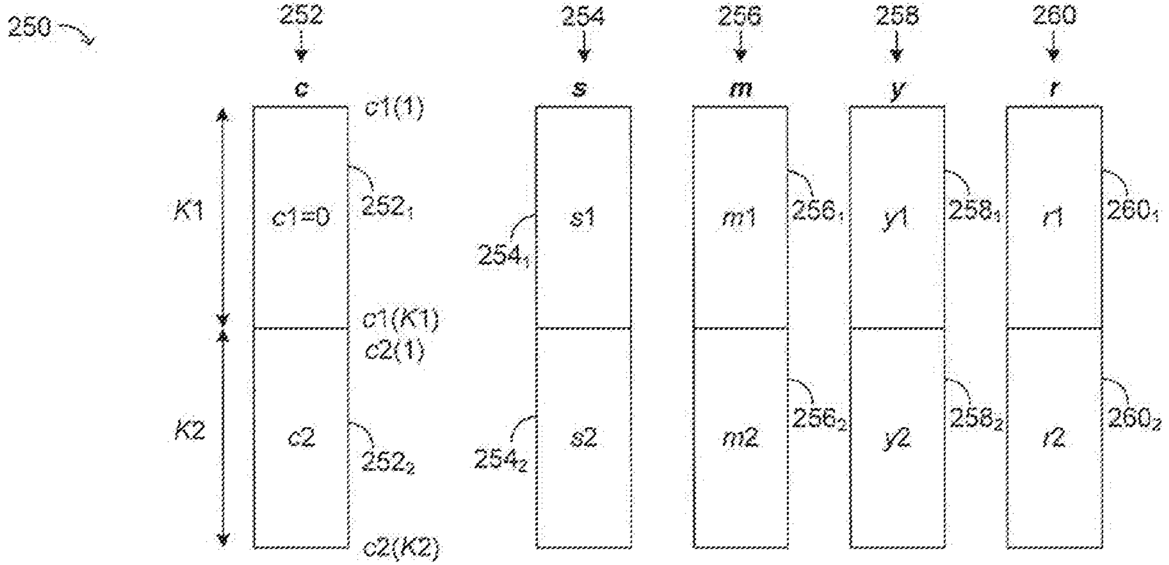

250
252
c
c1(1)
K1
c1=0
252₁
c1(K1)
c2(1)
K2
c2
252₂
c2(K2)
254
s
s1
254₁
s2
254₂
256
m
m1
256₁
m2
256₂
258
y
y1
258₁
y2
258₂
260
r
r1
260₁
r2
260₂

FIG. 5

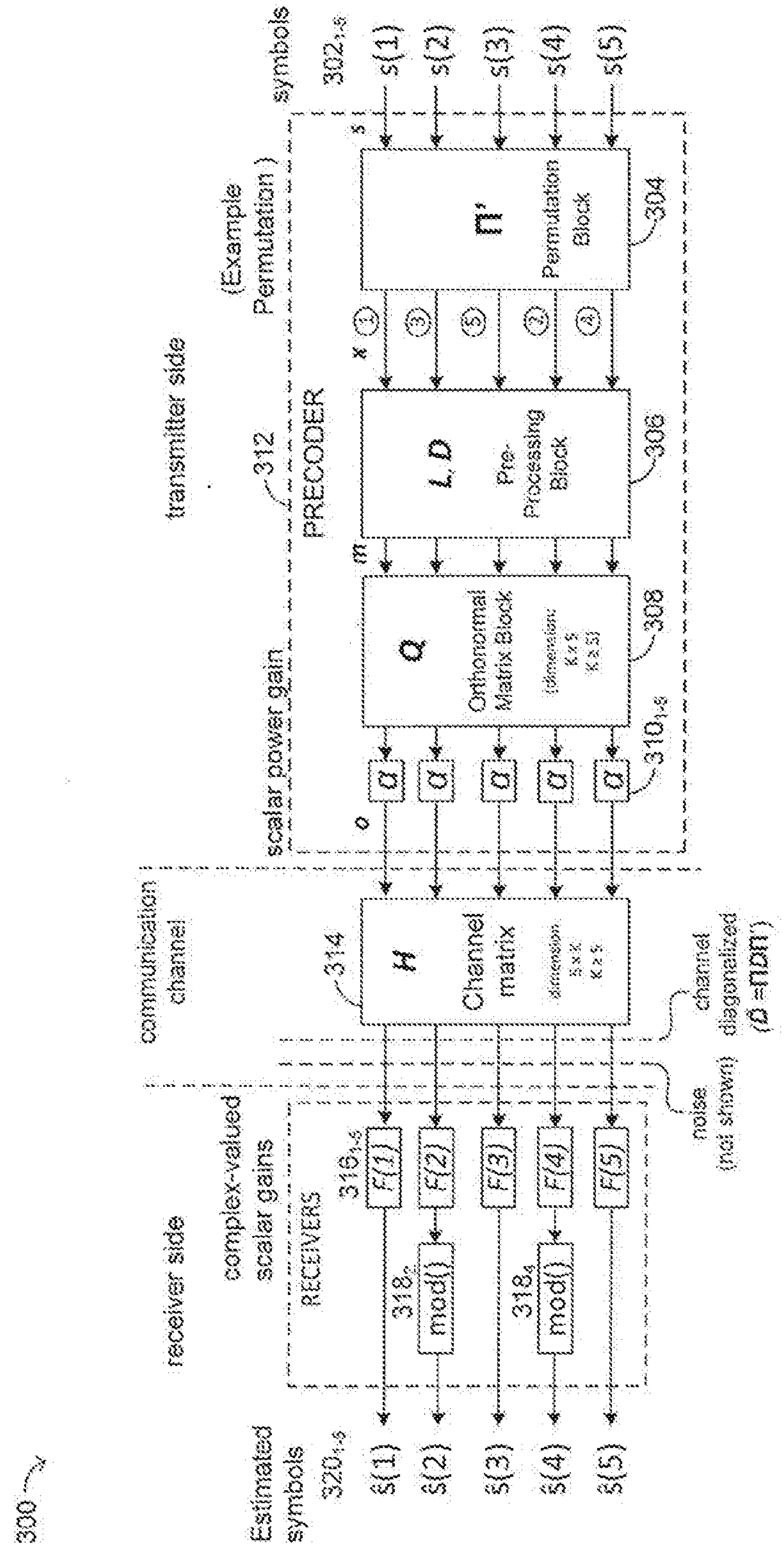

300
transmitter side
communication channel
receiver side
symbols
302$_{1-5}$
s(1)
s(2)
s(3)
s(4)
s(5)
s
(Example Permutation)
Π'
Permutation Block
304
x
1
3
5
2
4
PRECODER
312
L, D
Pre-Processing Block
306
m
Q
Orthonormal Matrix Block
(dimension: K x 5 K ≥ 5)
308
scalar power gain
α
310$_{1-5}$
o
H
Channel matrix
dimension 5 x K K ≥ 5
314
channel diagonalized (D̃ = ΠDΠ')
noise (not shown)
complex-valued scalar gains
RECEIVERS
316$_{1-5}$
F(1)
F(2)
F(3)
F(4)
F(5)
318$_2$
mod()
318$_4$
mod()
Estimated symbols
320$_{1-5}$
ŝ(1)
ŝ(2)
ŝ(3)
ŝ(4)
ŝ(5)

FIG. 6

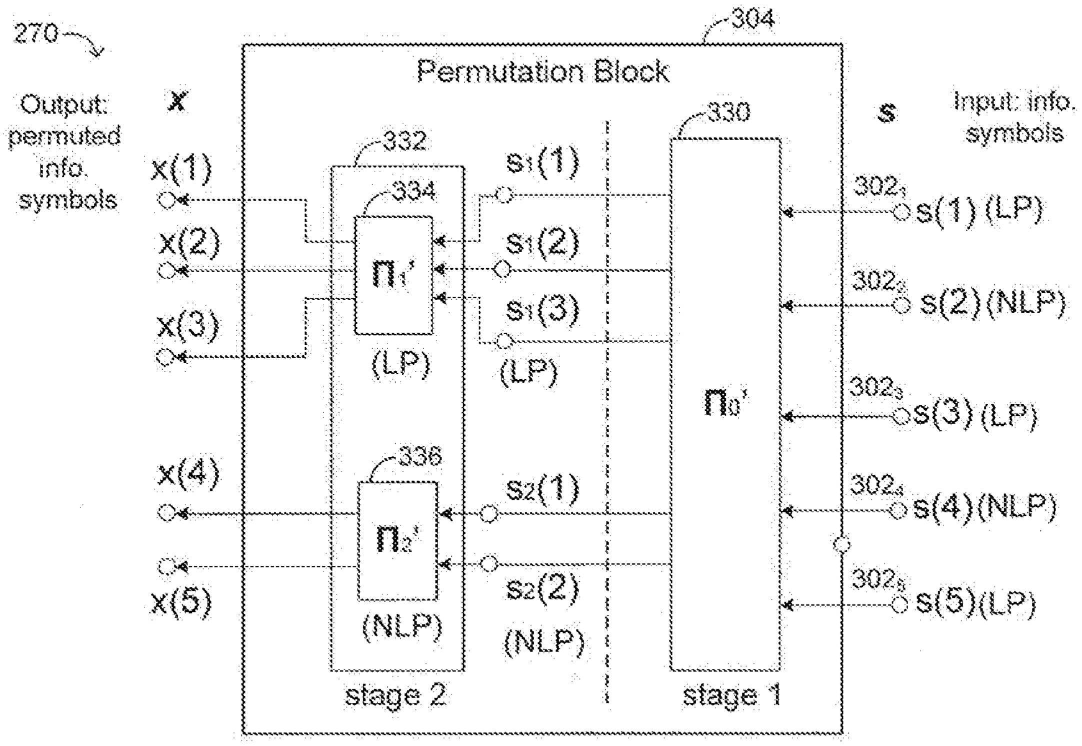

270
304
Permutation Block
Output: permuted info. symbols
x
330
s
Input: info. symbols
332
334
x(1)
x(2)
x(3)
Π_1'
(LP)
$s_1(1)$
$s_1(2)$
$s_1(3)$
(LP)
Π_0'
302_1 s(1) (LP)
302_2 s(2) (NLP)
302_3 s(3) (LP)
302_4 s(4) (NLP)
302_5 s(5) (LP)
336
x(4)
x(5)
Π_2'
(NLP)
$s_2(1)$
$s_2(2)$
(NLP)
stage 2
stage 1

FIG. 7

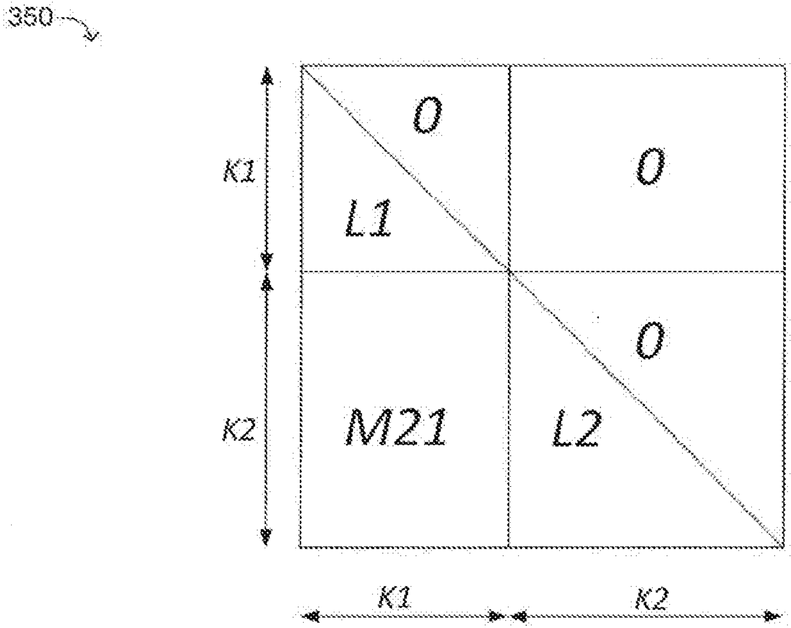

350
0
0
L1
0
M21
L2
K1
K2
K1
K2

FIG. 8

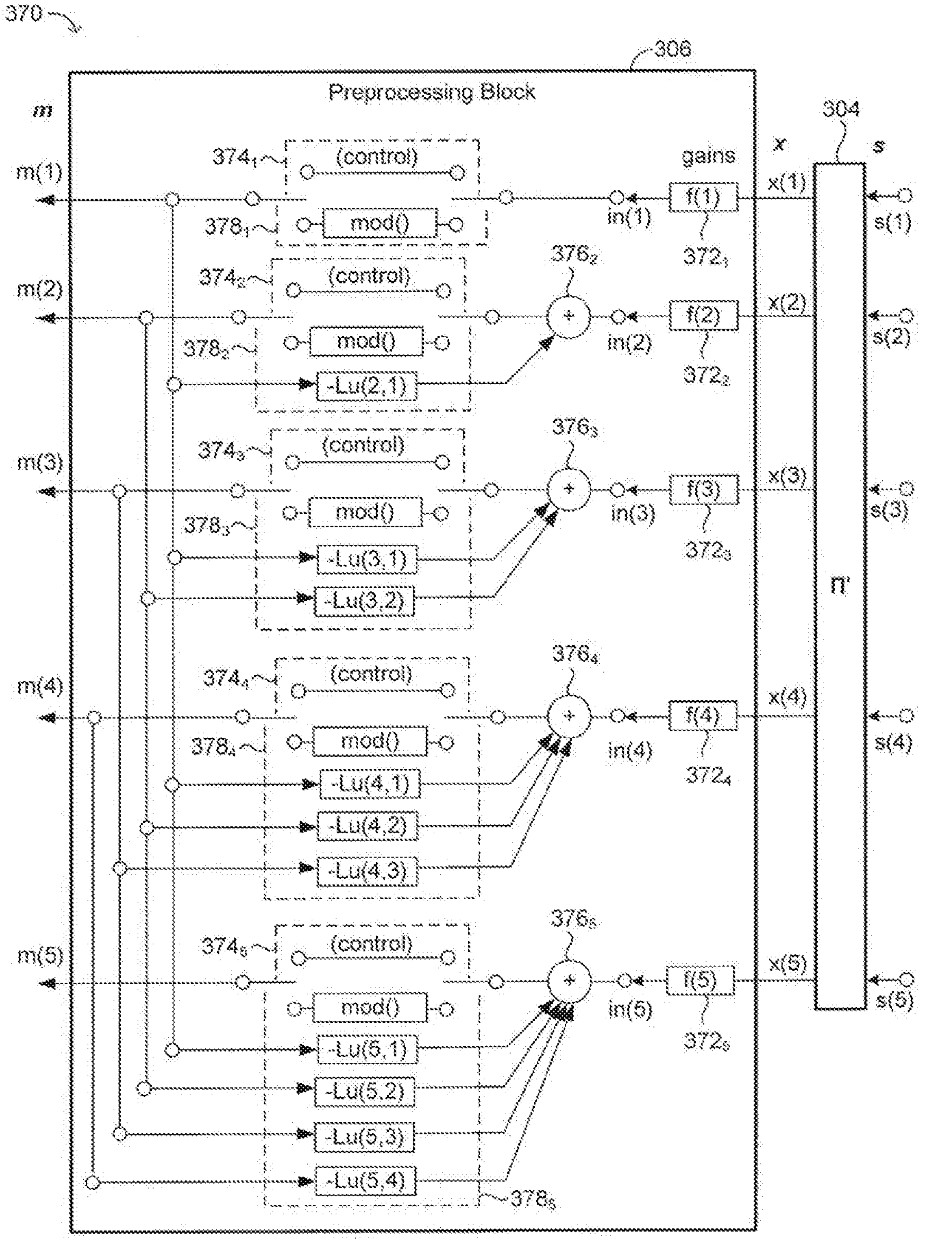

370
306
Preprocessing Block
304
m
x
s
gains
m(1)
m(2)
m(3)
m(4)
m(5)
374_1
374_2
374_3
374_4
374_5
(control)
mod()
378_1
378_2
378_3
378_4
378_5
-Lu(2,1)
-Lu(3,1)
-Lu(3,2)
-Lu(4,1)
-Lu(4,2)
-Lu(4,3)
-Lu(5,1)
-Lu(5,2)
-Lu(5,3)
-Lu(5,4)
376_2
376_3
376_4
376_5
+
in(1)
in(2)
in(3)
in(4)
in(5)
f(1)
f(2)
f(3)
f(4)
f(5)
372_1
372_2
372_3
372_4
372_5
x(1)
x(2)
x(3)
x(4)
x(5)
Π'
s(1)
s(2)
s(3)
s(4)
s(5)

FIG. 9

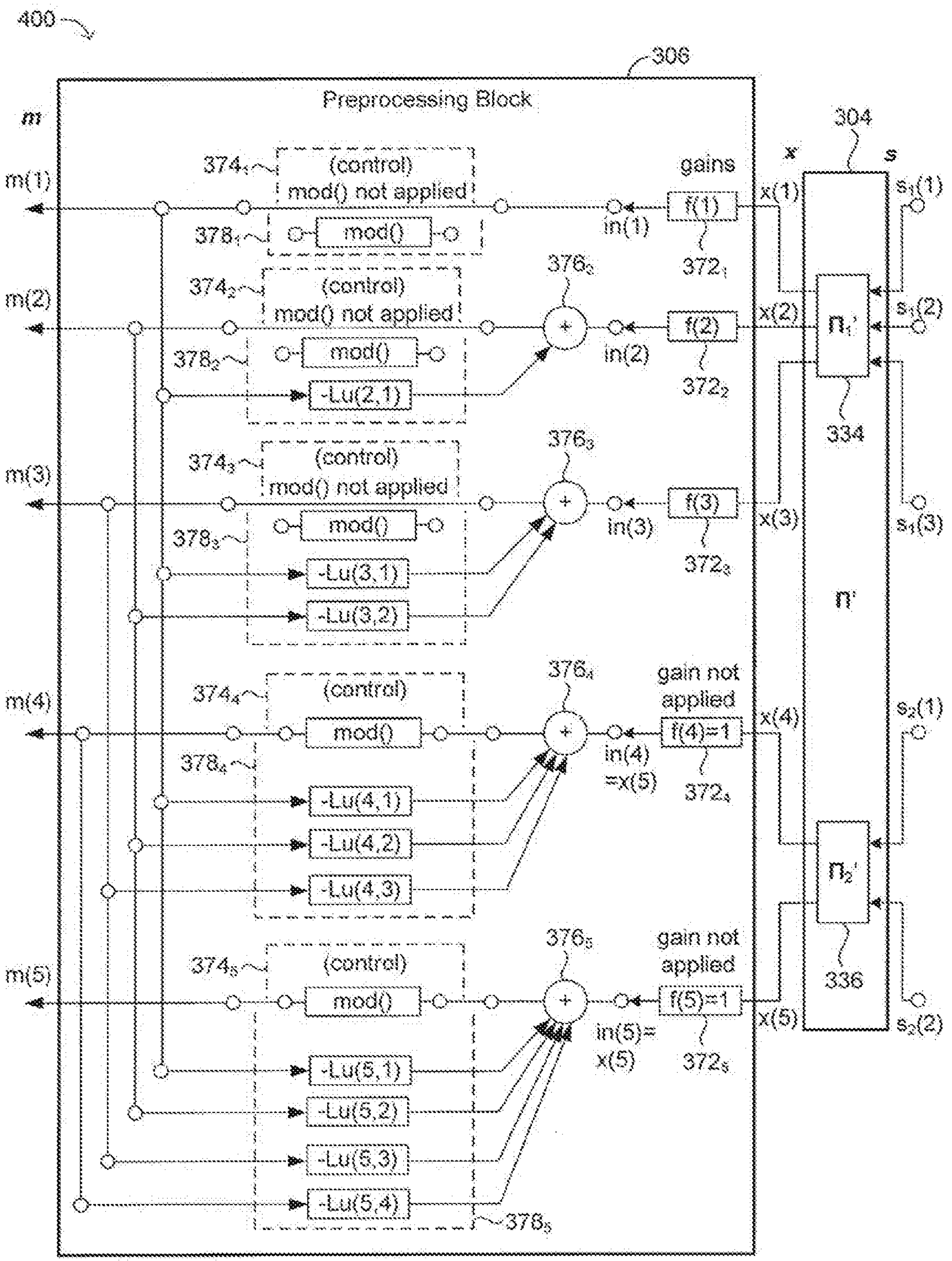

400
306
Preprocessing Block
m
304
x
s
gains
m(1)
374₁
(control)
mod() not applied
378₁
mod()
in(1)
f(1)
372₁
x(1)
s₁(1)
m(2)
374₂
(control)
mod() not applied
376₂
378₂
mod()
-Lu(2,1)
in(2)
f(2)
372₂
x(2)
n₁'
s₁(2)
334
m(3)
374₃
(control)
mod() not applied
376₃
378₃
mod()
-Lu(3,1)
-Lu(3,2)
in(3)
f(3)
372₃
x(3)
s₁(3)
n'
m(4)
374₄
(control)
mod()
376₄
gain not applied
378₄
in(4) =x(5)
f(4)=1
372₄
x(4)
s₂(1)
-Lu(4,1)
-Lu(4,2)
-Lu(4,3)
n₂'
m(5)
374₅
(control)
mod()
376₅
gain not applied
in(5)= x(5)
f(5)=1
372₅
x(5)
336
s₂(2)
-Lu(5,1)
-Lu(5,2)
-Lu(5,3)
-Lu(5,4)
378₅

FIG. 10

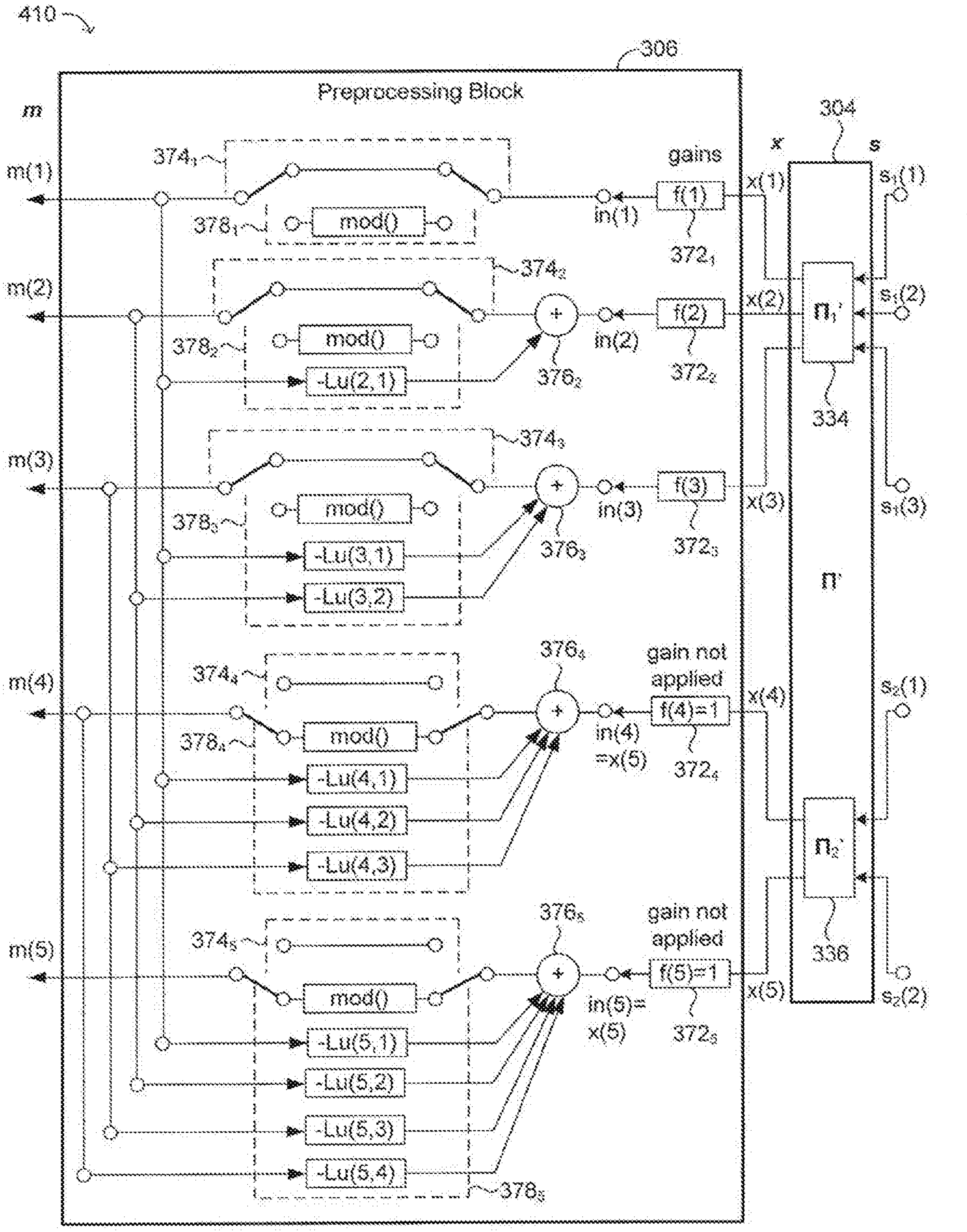

410
306
Preprocessing Block
m
x
s
304
m(1)
374₁
gains
x(1)
s₁(1)
f(1)
in(1)
378₁
mod()
372₁
m(2)
374₂
x(2)
s₁(2)
f(2)
in(2)
Π₁'
mod()
378₂
-Lu(2,1)
376₂
372₂
334
m(3)
374₃
f(3)
in(3)
x(3)
s₁(3)
mod()
378₃
376₃
372₃
-Lu(3,1)
-Lu(3,2)
Π'
376₄
gain not applied
374₄
m(4)
x(4)
s₂(1)
f(4)=1
in(4) =x(5)
378₄
mod()
372₄
-Lu(4,1)
-Lu(4,2)
Π₂'
-Lu(4,3)
376₅
gain not applied
374₅
m(5)
f(5)=1
x(5)
s₂(2)
336
mod()
in(5)= x(5)
372₅
-Lu(5,1)
-Lu(5,2)
-Lu(5,3)
-Lu(5,4)
378₅

FIG. 11

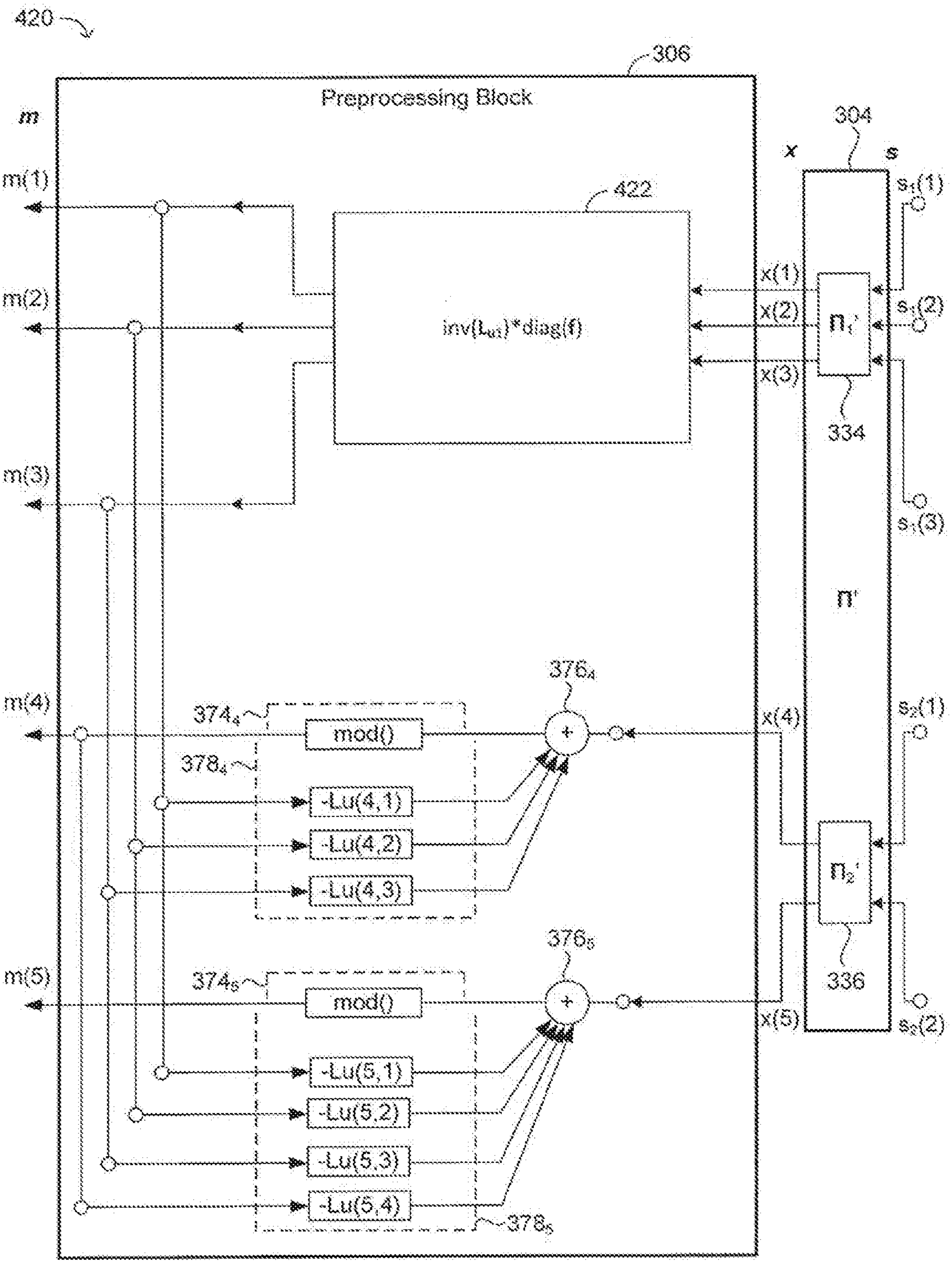

420
306
Preprocessing Block
m
m(1)
m(2)
m(3)
m(4)
m(5)
422
inv(L_{u1})*diag(f)
304
x
s
$s_1(1)$
$s_1(2)$
$s_1(3)$
$s_2(1)$
$s_2(2)$
x(1)
x(2)
x(3)
x(4)
x(5)
Π_1'
334
Π'
Π_2'
336
376_4
374_4
378_4
mod()
+
-Lu(4,1)
-Lu(4,2)
-Lu(4,3)
376_5
374_5
mod()
+
-Lu(5,1)
-Lu(5,2)
-Lu(5,3)
-Lu(5,4)
378_5

FIG. 12

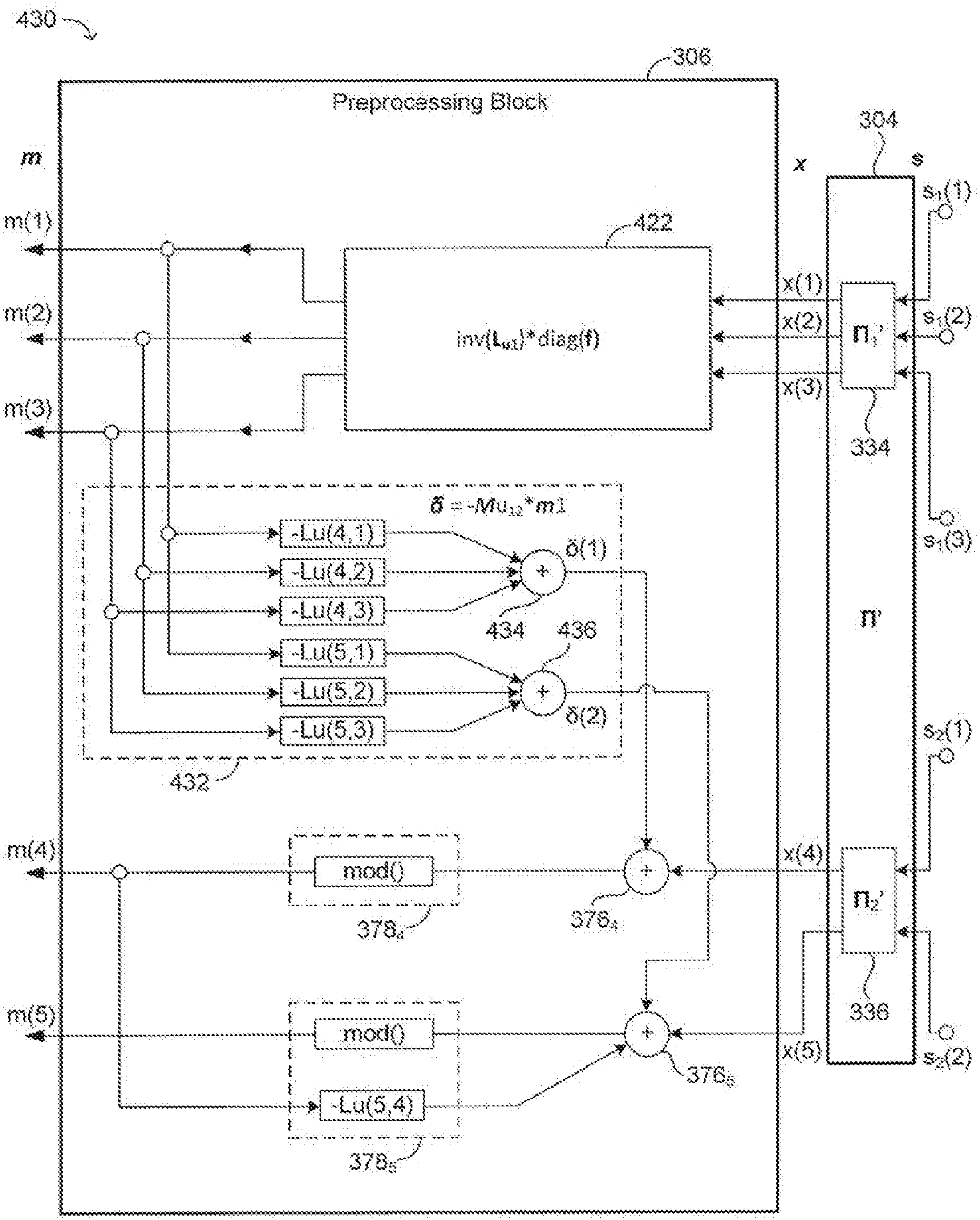
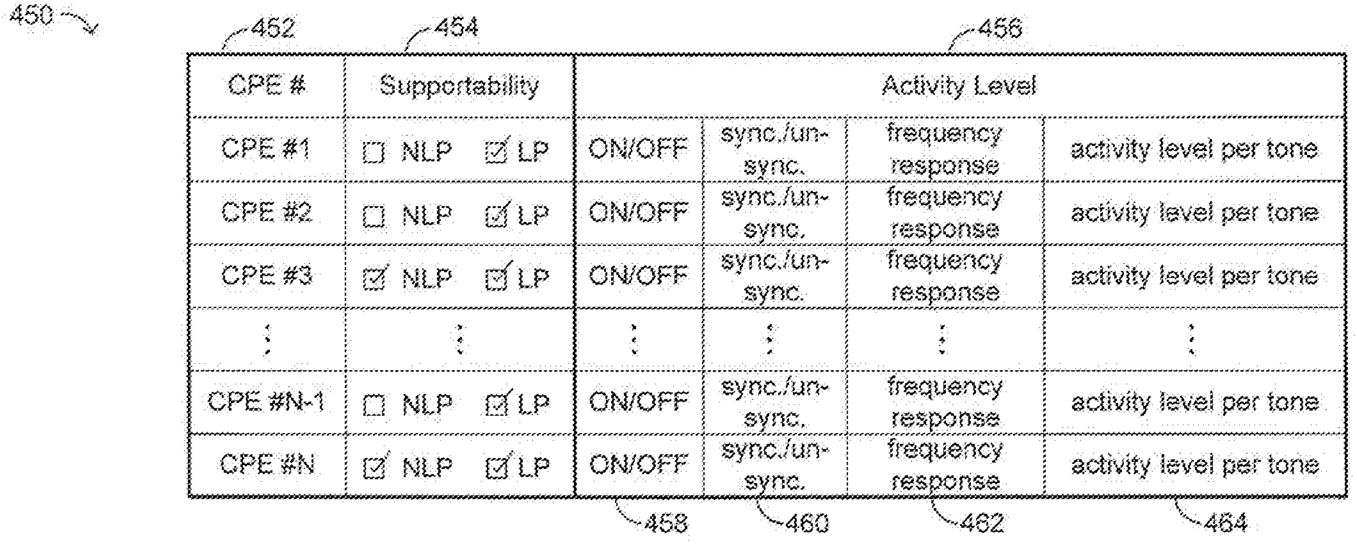

430
306
Preprocessing Block
m
x
304
s
m(1)
422
$s_1(1)$
m(2)
inv(L_{w1})*diag(f)
x(1)
x(2)
x(3)
Π_1'
$s_1(2)$
m(3)
334
$\boldsymbol{\delta}$ = -Mu_{12}*m1
-Lu(4,1)
-Lu(4,2)
-Lu(4,3)
δ(1)
+
434
436
-Lu(5,1)
-Lu(5,2)
-Lu(5,3)
+
δ(2)
$s_1(3)$
Π'
432
$s_2(1)$
m(4)
mod()
+
x(4)
Π_2'
378_4
376_4
336
m(5)
mod()
+
x(5)
$s_2(2)$
376_5
-Lu(5,4)
378_5

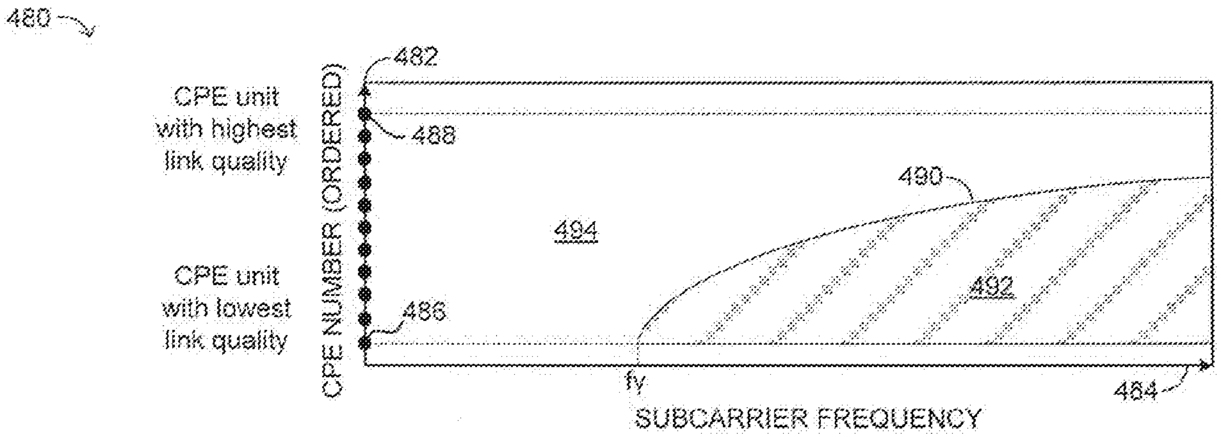
FIG. 13



450
452
454
456
458
460
462
464

CPE #	Supportability	Activity Level			
CPE #1	☐ NLP ☑ LP	ON/OFF	sync./un-sync.	frequency response	activity level per tone
CPE #2	☐ NLP ☑ LP	ON/OFF	sync./un-sync.	frequency response	activity level per tone
CPE #3	☑ NLP ☑ LP	ON/OFF	sync./un-sync.	frequency response	activity level per tone
⋮	⋮	⋮	⋮	⋮	⋮
CPE #N-1	☐ NLP ☑ LP	ON/OFF	sync./un-sync.	frequency response	activity level per tone
CPE #N	☑ NLP ☑ LP	ON/OFF	sync./un-sync.	frequency response	activity level per tone

FIG. 14A



480
482
488
CPE unit with highest link quality
CPE unit with lowest link quality
CPE NUMBER (ORDERED)
486
490
494
492
fy
484
SUBCARRIER FREQUENCY

FIG. 14B

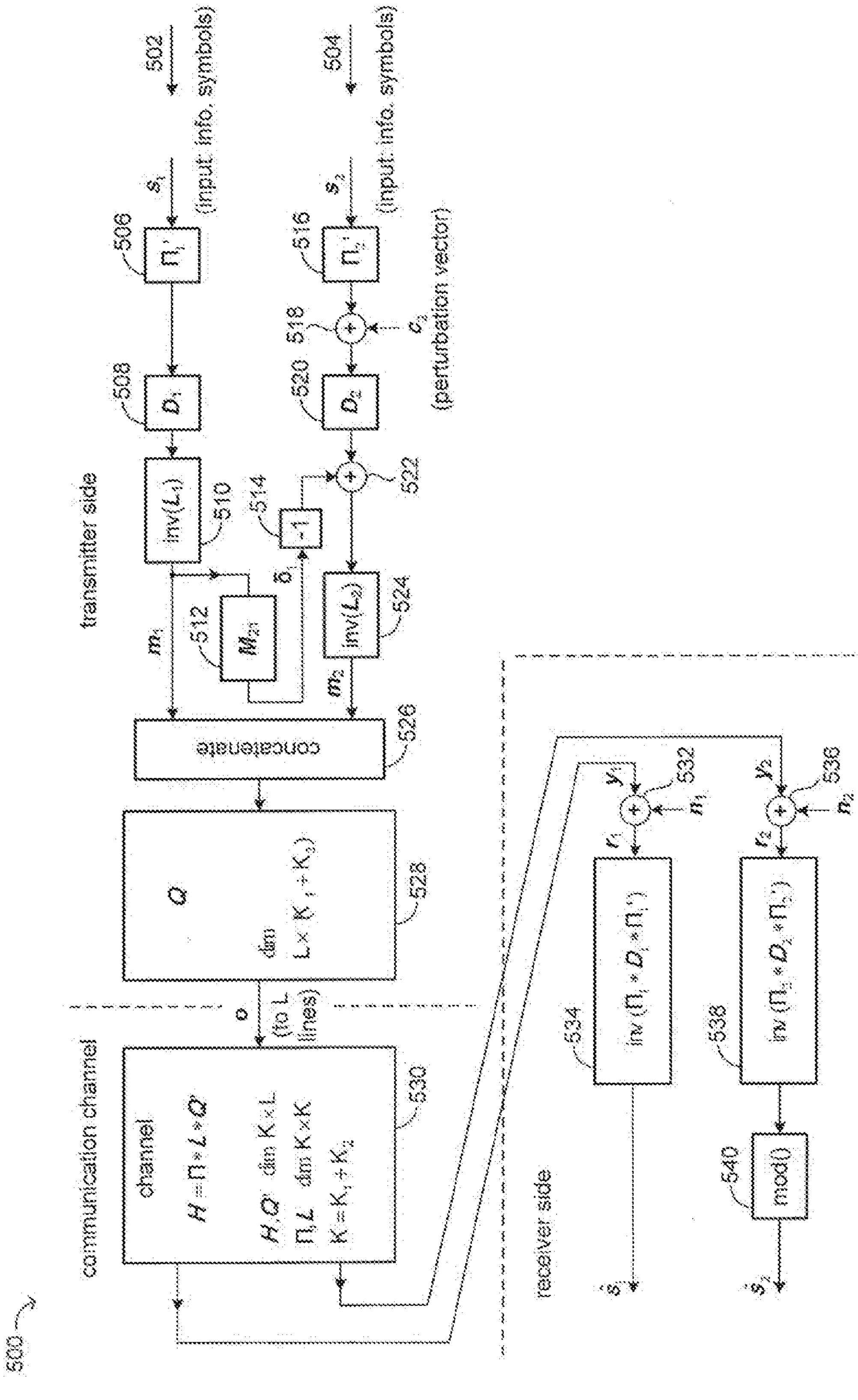

500
transmitter side
communication channel
receiver side
502
s_1
(input info. symbols)
504
s_2
(input info. symbols)
506
Π_1'
516
Π_2'
518
c_2
(perturbation vector)
508
D_1
520
D_2
510
inv(L_1)
514
-1
δ_1
522
512
M_{21}
m_1
524
inv(L_2)
m_2
526
concatenate
528
Q
dim
L × (K₁ + K₂)
o
(to L lines)
530
channel
H = Π * L * Q'
H, Q' dim K × L
Π, L dim K × K
K = K₁ + K₂
532
y_1
n_1
r_1
536
y_2
n_2
r_2
534
inv (Π₁ * D₁ * Π₁')
538
inv (Π₂ * D₂ * Π₂')
540
mod()
$\hat{s}_1$
$\hat{s}_2$

FIG. 15

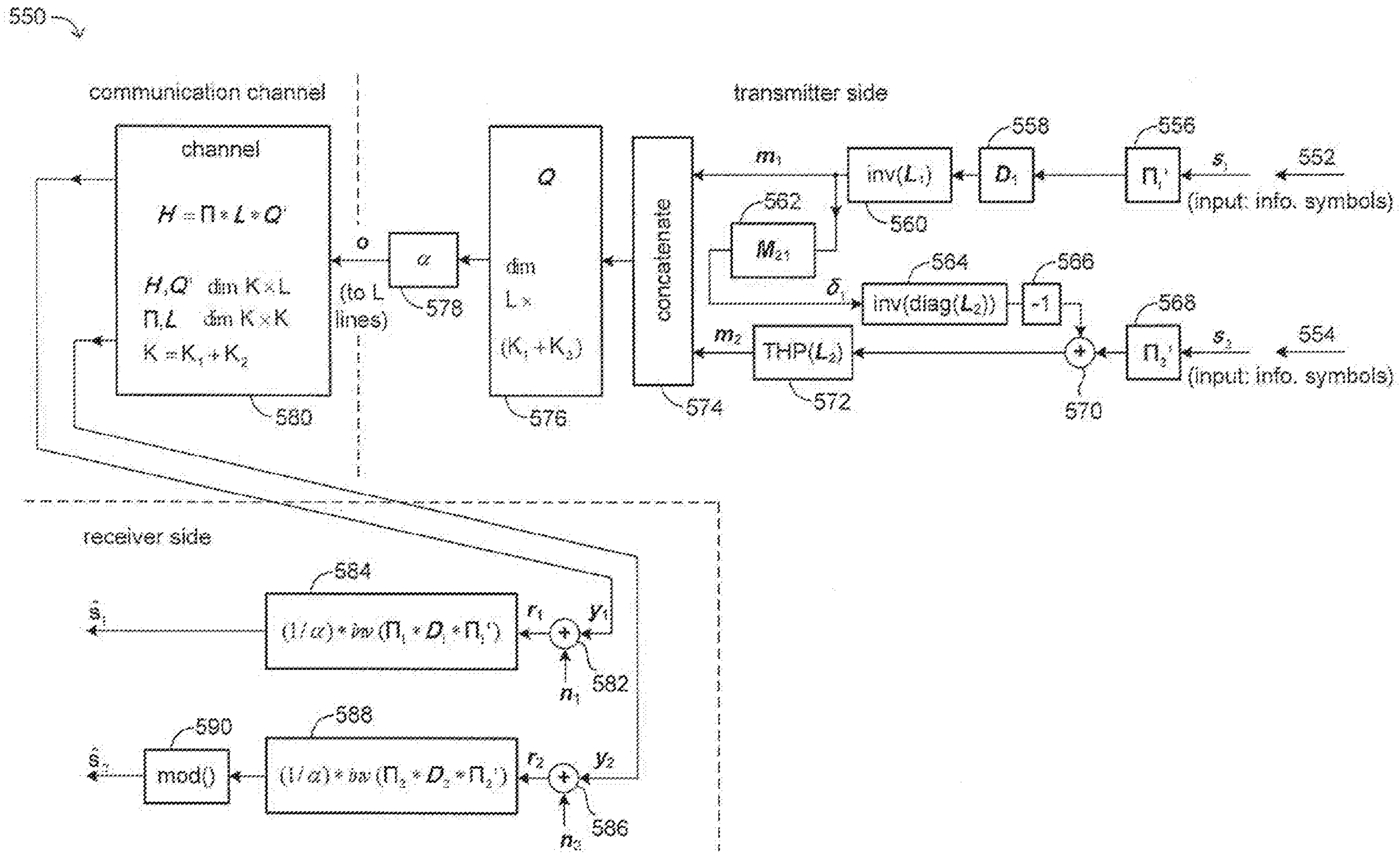

550
communication channel
transmitter side
channel
$H = \Pi * L * Q'$
H, Q' dim K × L
Π, L dim K × K
K = K₁ + K₂
580
o
(to L lines)
α
578
Q
dim
L ×
(K₁ + K₂)
576
concatenate
574
m_1
m_2
inv(L_1)
560
D_1
558
Π_1'
556
s_1
552
(input: info. symbols)
M_{21}
562
δ_1
inv(diag(L_2))
564
-1
566
THP(L_2)
572
+
570
Π_2'
568
s_2
554
(input: info. symbols)
receiver side
584
$(1/\alpha) * inv(\Pi_1 * D_1 * \Pi_1')$
$\hat{s}_1$
r_1
y_1
+
582
n_1
588
$(1/\alpha) * inv(\Pi_2 * D_2 * \Pi_2')$
mod()
590
$\hat{s}_2$
r_2
y_2
+
586
n_2

FIG. 16

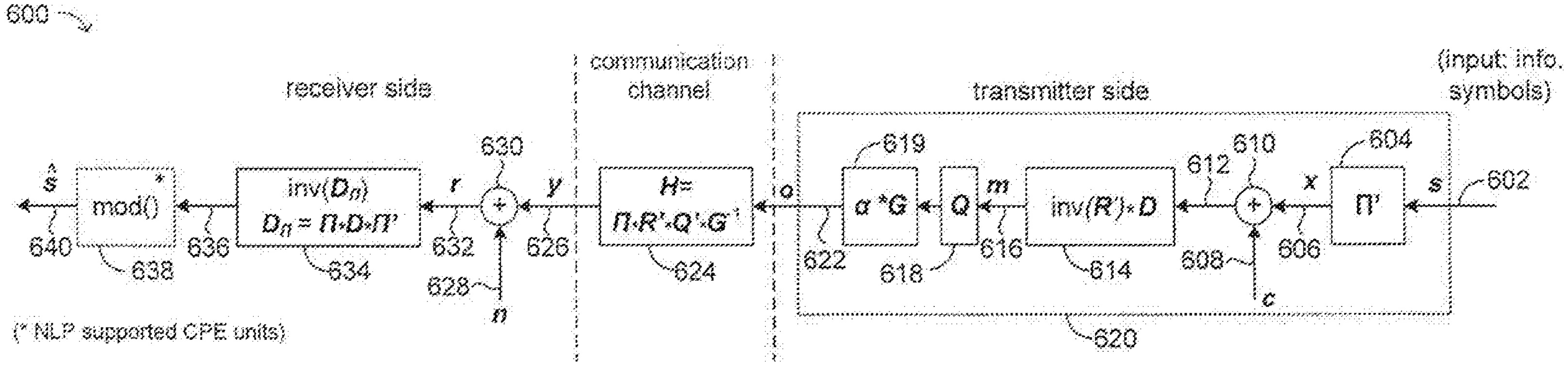


FIG. 17A

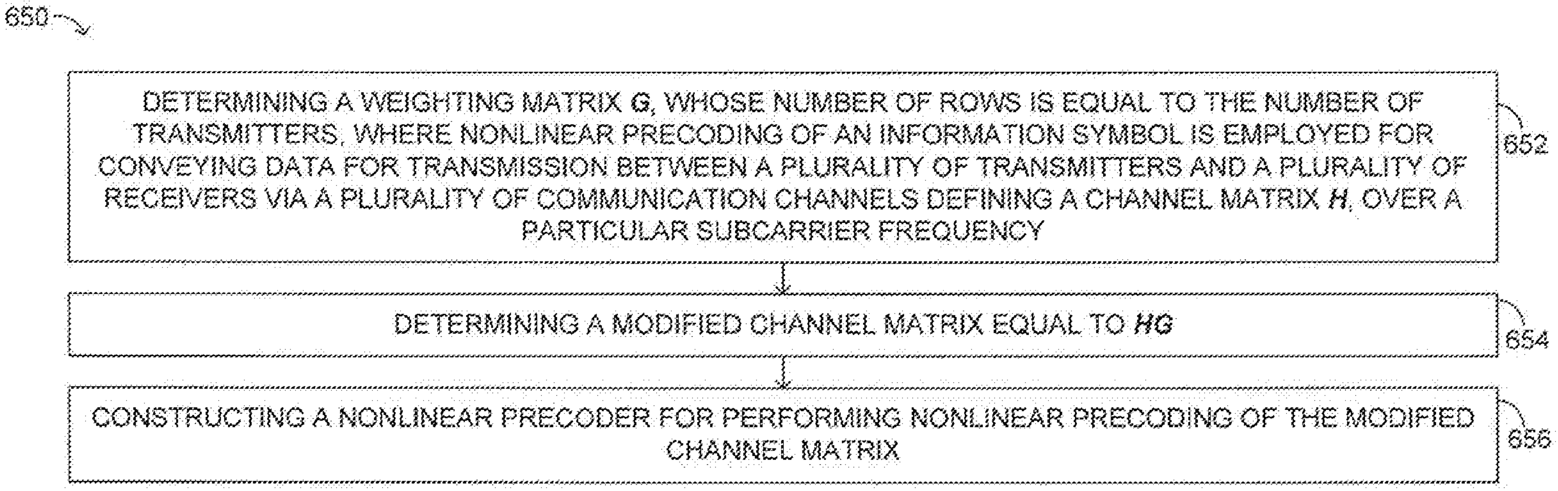
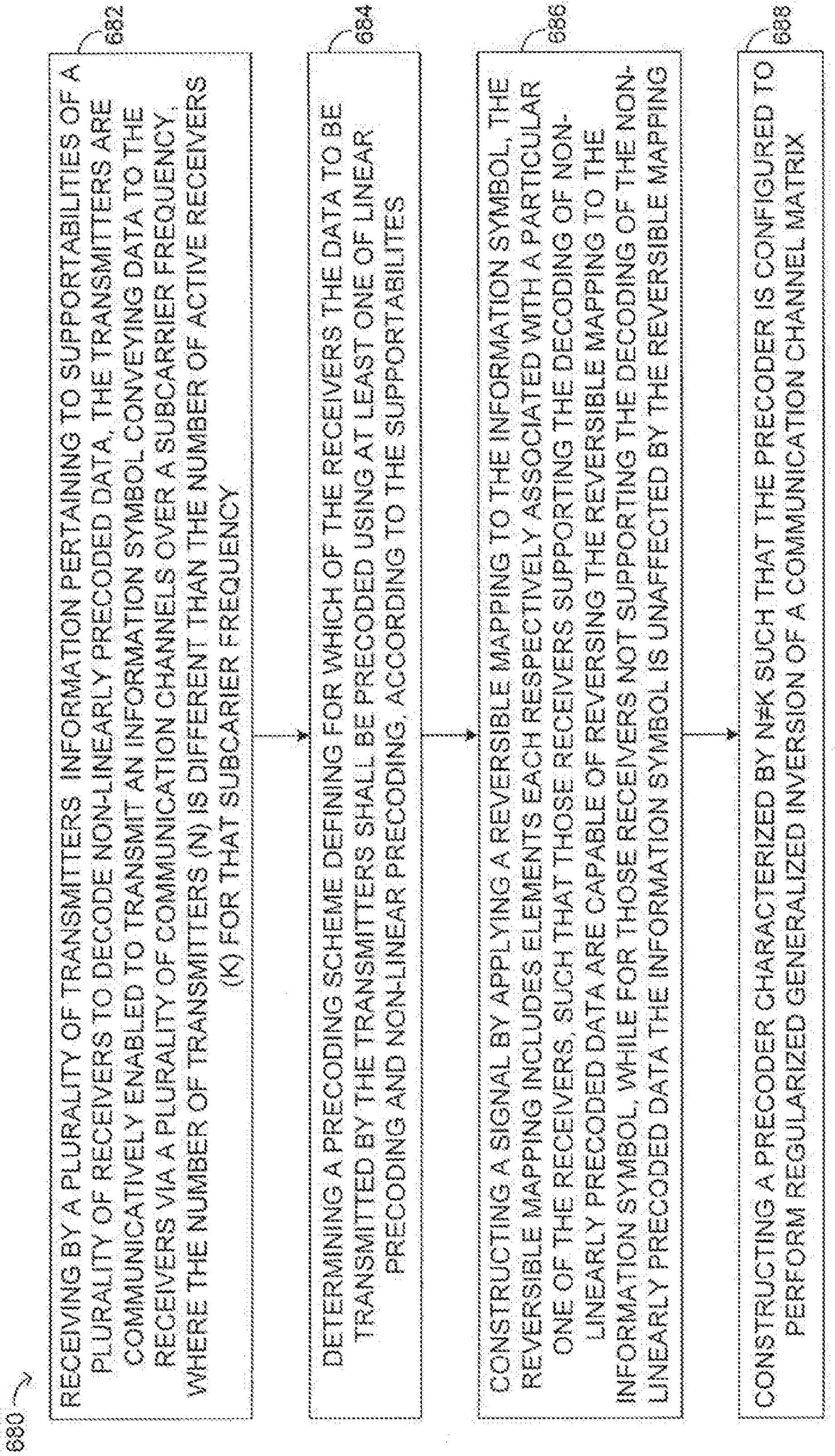

650
DETERMINING A WEIGHTING MATRIX G, WHOSE NUMBER OF ROWS IS EQUAL TO THE NUMBER OF TRANSMITTERS, WHERE NONLINEAR PRECODING OF AN INFORMATION SYMBOL IS EMPLOYED FOR CONVEYING DATA FOR TRANSMISSION BETWEEN A PLURALITY OF TRANSMITTERS AND A PLURALITY OF RECEIVERS VIA A PLURALITY OF COMMUNICATION CHANNELS DEFINING A CHANNEL MATRIX H, OVER A PARTICULAR SUBCARRIER FREQUENCY
652
DETERMINING A MODIFIED CHANNEL MATRIX EQUAL TO HG
654
CONSTRUCTING A NONLINEAR PRECODER FOR PERFORMING NONLINEAR PRECODING OF THE MODIFIED CHANNEL MATRIX
656

FIG. 17B



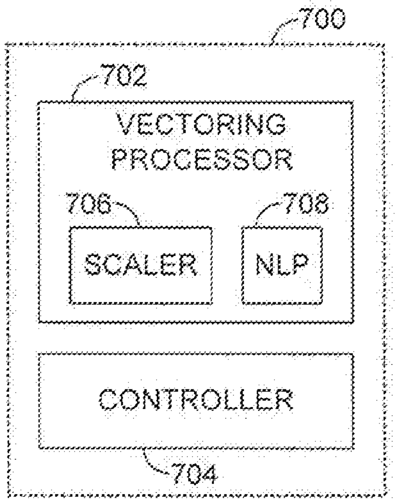
RECEIVING BY A PLURALITY OF TRANSMITTERS INFORMATION PERTAINING TO SUPPORTABILITIES OF A PLURALITY OF RECEIVERS TO DECODE NON-LINEARLY PRECODED DATA, THE TRANSMITTERS ARE COMMUNICATIVELY ENABLED TO TRANSMIT AN INFORMATION SYMBOL CONVEYING DATA TO THE RECEIVERS VIA A PLURALITY OF COMMUNICATION CHANNELS OVER A SUBCARRIER FREQUENCY, WHERE THE NUMBER OF TRANSMITTERS (N) IS DIFFERENT THAN THE NUMBER OF ACTIVE RECEIVERS (K) FOR THAT SUBCARIER FREQUENCY

DETERMINING A PRECODING SCHEME DEFINING FOR WHICH OF THE RECEIVERS THE DATA TO BE TRANSMITTED BY THE TRANSMITTERS SHALL BE PRECODED USING AT LEAST ONE OF LINEAR PRECODING AND NON-LINEAR PRECODING, ACCORDING TO THE SUPPORTABILITES

CONSTRUCTING A SIGNAL BY APPLYING A REVERSIBLE MAPPING TO THE INFORMATION SYMBOL, THE REVERSIBLE MAPPING INCLUDES ELEMENTS EACH RESPECTIVELY ASSOCIATED WITH A PARTICULAR ONE OF THE RECEIVERS, SUCH THAT THOSE RECEIVERS SUPPORTING THE DECODING OF NON-LINEARLY PRECODED DATA ARE CAPABLE OF REVERSING THE REVERSIBLE MAPPING TO THE INFORMATION SYMBOL, WHILE FOR THOSE RECEIVERS NOT SUPPORTING THE DECODING OF THE NON-LINEARLY PRECODED DATA THE INFORMATION SYMBOL IS UNAFFECTED BY THE REVERSIBLE MAPPING

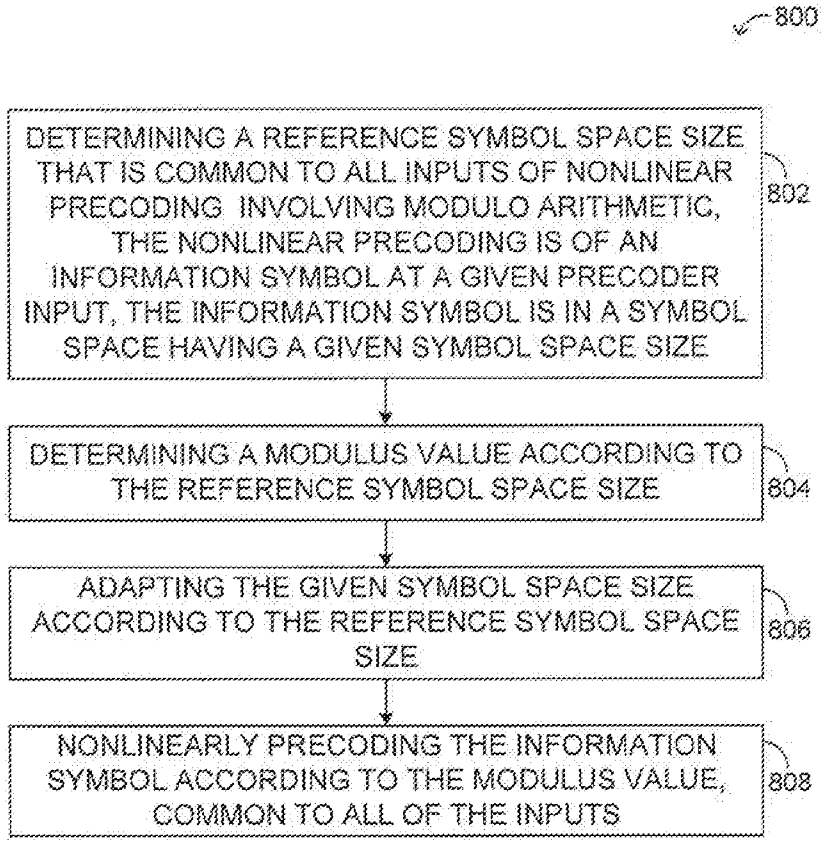
CONSTRUCTING A PRECODER CHARACTERIZED BY N≠K SUCH THAT THE PRECODER IS CONFIGURED TO PERFORM REGULARIZED GENERALIZED INVERSION OF A COMMUNICATION CHANNEL MATRIX

FIG. 18



700
702
VECTORING PROCESSOR
706
SCALER
708
NLP
CONTROLLER
704

FIG. 19



800
DETERMINING A REFERENCE SYMBOL SPACE SIZE THAT IS COMMON TO ALL INPUTS OF NONLINEAR PRECODING INVOLVING MODULO ARITHMETIC, THE NONLINEAR PRECODING IS OF AN INFORMATION SYMBOL AT A GIVEN PRECODER INPUT, THE INFORMATION SYMBOL IS IN A SYMBOL SPACE HAVING A GIVEN SYMBOL SPACE SIZE
802
DETERMINING A MODULUS VALUE ACCORDING TO THE REFERENCE SYMBOL SPACE SIZE
804
ADAPTING THE GIVEN SYMBOL SPACE SIZE ACCORDING TO THE REFERENCE SYMBOL SPACE SIZE
806
NONLINEARLY PRECODING THE INFORMATION SYMBOL ACCORDING TO THE MODULUS VALUE, COMMON TO ALL OF THE INPUTS
808

FIG. 23

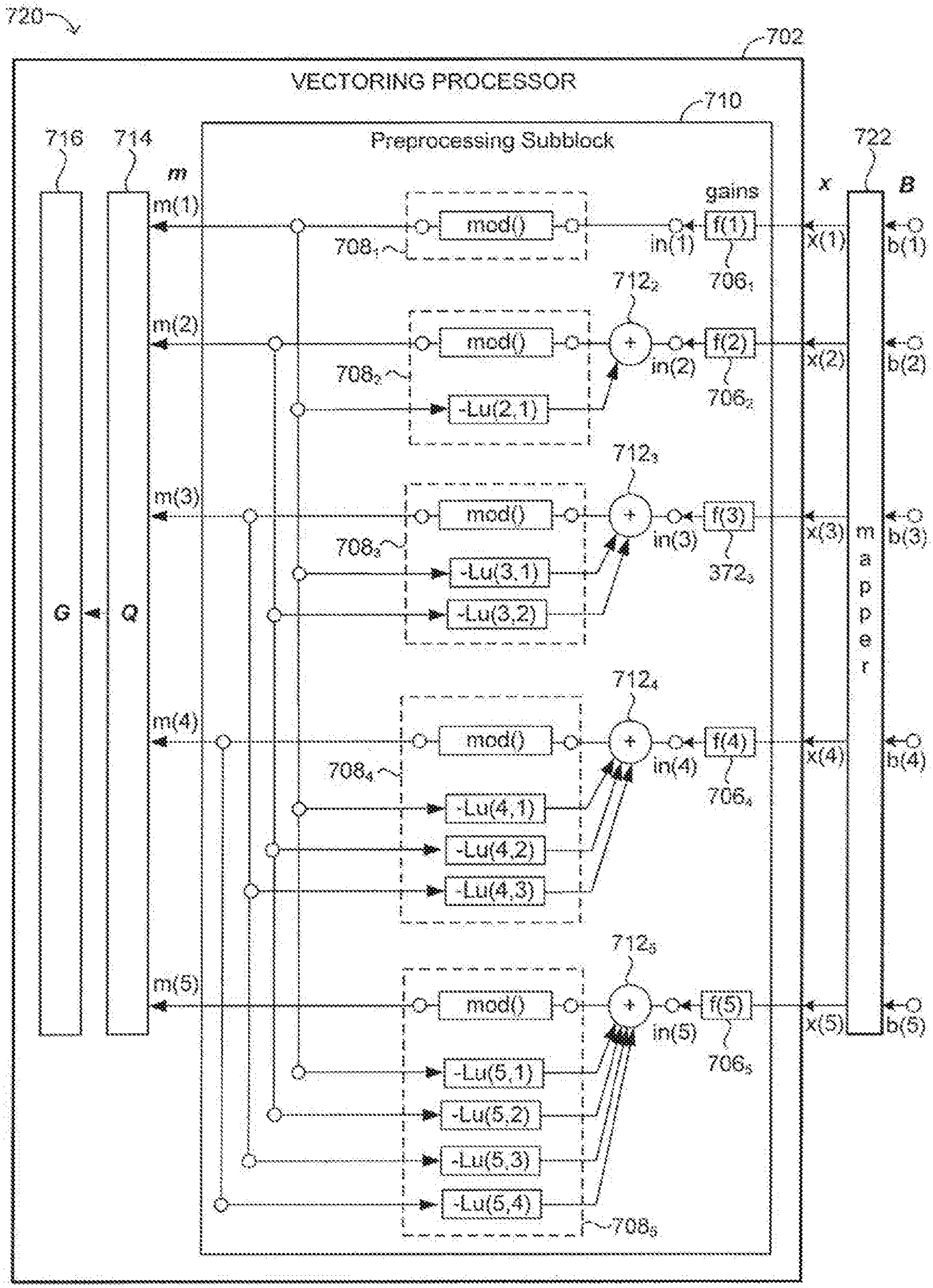

720
702
VECTORING PROCESSOR
710
Preprocessing Subblock
716
714
722
m
x
B
gains
m(1)
mod()
in(1)
f(1)
x(1)
b(1)
708$_1$
706$_1$
712$_2$
m(2)
mod()
in(2)
f(2)
x(2)
b(2)
708$_2$
-Lu(2,1)
706$_2$
712$_3$
m(3)
mod()
in(3)
f(3)
x(3)
b(3)
708$_3$
-Lu(3,1)
372$_3$
-Lu(3,2)
G
Q
mapper
712$_4$
m(4)
mod()
in(4)
f(4)
x(4)
b(4)
708$_4$
-Lu(4,1)
706$_4$
-Lu(4,2)
-Lu(4,3)
712$_5$
m(5)
mod()
in(5)
f(5)
x(5)
b(5)
706$_5$
-Lu(5,1)
-Lu(5,2)
-Lu(5,3)
-Lu(5,4)
708$_5$

FIG. 20

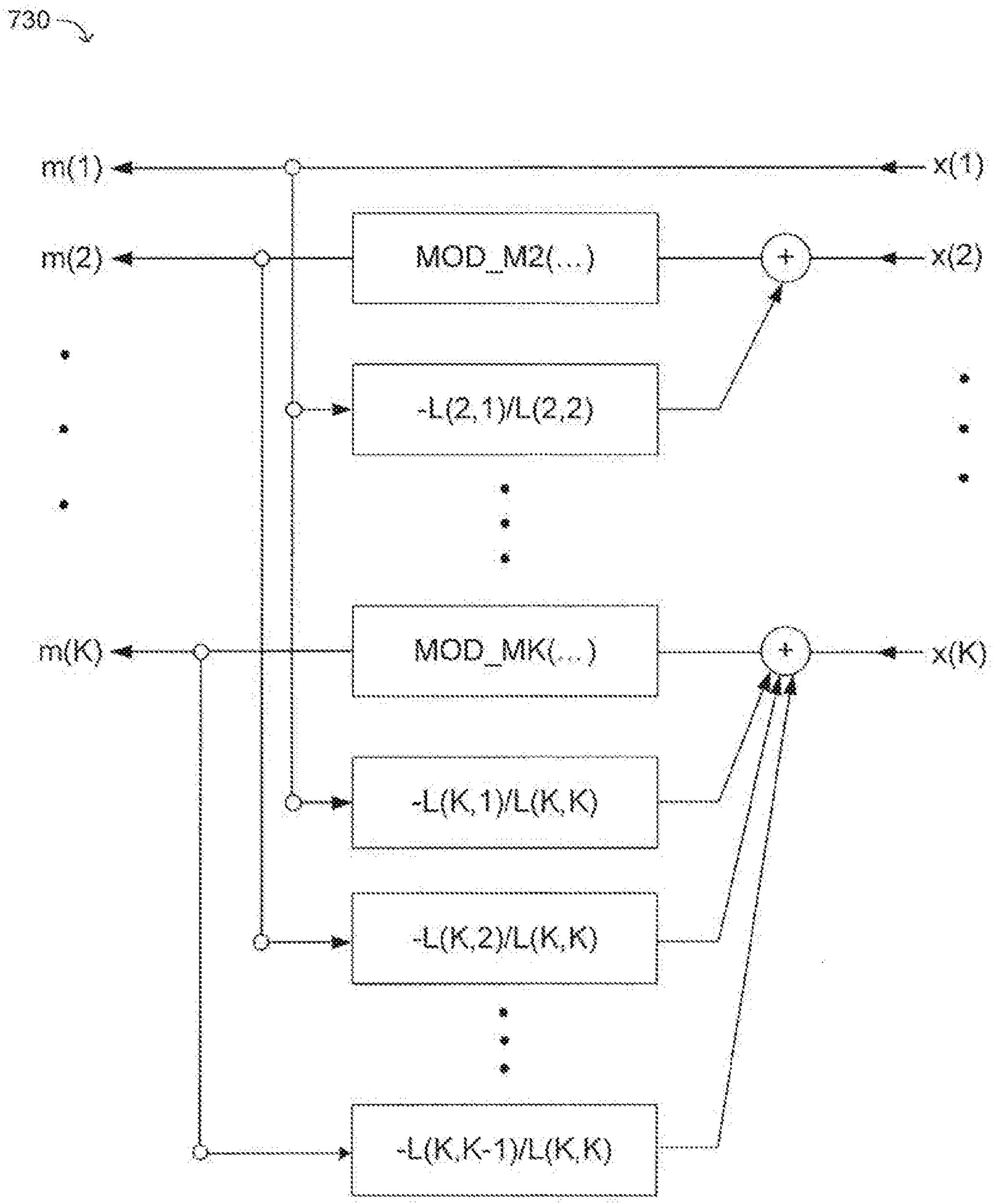

730
m(1)
x(1)
m(2)
MOD_M2(...)
+
x(2)
-L(2,1)/L(2,2)
m(K)
MOD_MK(...)
+
x(K)
-L(K,1)/L(K,K)
-L(K,2)/L(K,K)
-L(K,K-1)/L(K,K)

FIG. 21

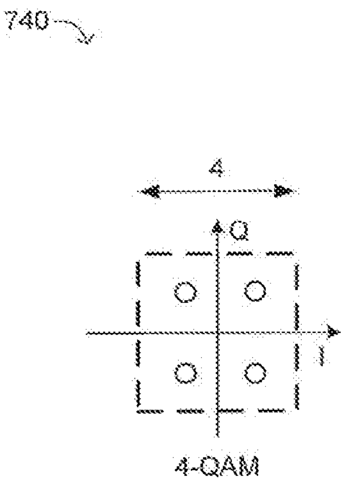


FIG. 22A

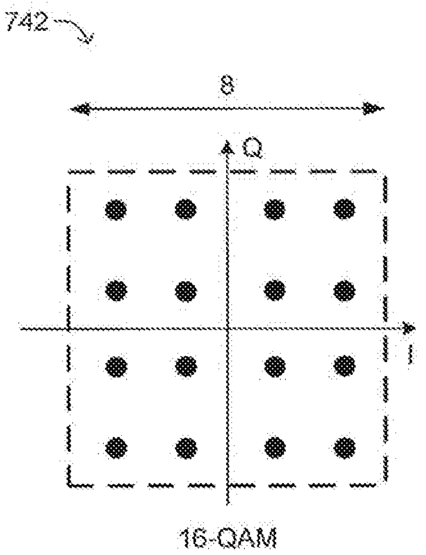


FIG. 22B

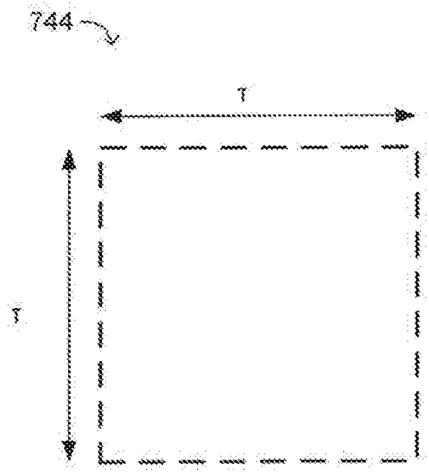


FIG. 22C

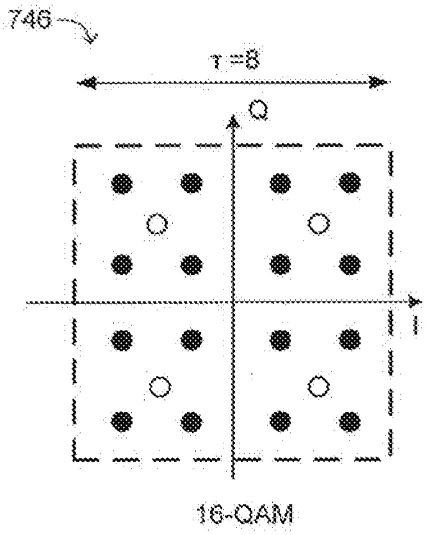


FIG. 22D

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL2017/051402

A. CLASSIFICATION OF SUBJECT MATTER

IPC (2018.01) H04B 3/32, H04J 11/00, H04L 1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification **symbols)**

IPC (2018.01) H04B 3/32, H04J 11/00, H04L 1/00

Documentation searched other than minimum documentation to the extent **that** such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, **where** practicable, search terms used)

Databases consulted: PATENTSCOPE, THOMSON INNOVATION, Google Patents, Orbit
Search terms used: Interference, FEXT,precoder, non-linear, MIMO, Thomlinson-harashima, modulo, arithmetic, costa, DPC, dirty paper, perturbation, beamforming, schedule, scheme,scaling, constellation, symbol, mapping, reversible

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2016043786 A1 MAES et al. 11 Feb 2016 (2016/02/11) ¶ 0004, 0005, 0015, 0019-0032, 0043, 0044, 0051, 0052, 0054, 0104-0111, 0131 fig. 4, fig. 5	58-60,63-68
Y	¶ 0004, 0005, 0015, 0019-0032, 0043, 0044, 0051, 0052, 0054, 0104-0111, 0131 fig. 4, fig. 5	61,62
A	CN 106060950 A UNIV CHONGQING POSTS & TELECOM 26 Oct 2016 (2016/10/26) Whole document	1-57
A	US 2015282144 A1 TOMEBA et al. 01 Oct 2015 (2015/10/01) Whole document	1-57
A	US 2015295621 A1 HUAWEI TECH CO LTD 15 Oct 2015 (2015/10/15) Whole document	1-57

□ Further documents are listed **in** the **continuation of** Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is ¾¾ to establish the publication date of another citation or other sp„c·al reason (as specified)

"0" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than tire priority date claimed

"T" later document published after the international filing date or priority date and not in conflict **with** the application but cited to understand the principle or theory underlying the invention

"χ" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y' document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document rnember of the same patent iaimiy

Dale **of** the actual completion **of** the International search 15 Apr 2018	Date of mailing of the international search report 22 Apr 2018
Name and mailing address of the ISA: Israel Patent Office Technology Park, Bldg.5, Malcha, Jerusalem, 9695101, Israel Facsimile No. 972-2-5651616	Authorized officer COPPENHAGEN Uri Telephone No. 972-2-5657811

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/IL2017/051402

Patent document cited search report	Publication date	Patent family member(s)	Publication Date
US 2016043786 A1	11 Feb 2016	US 2016043786 A1	11 Feb 2016
		US 9628157 B2	18 Apr 2017
		CN 105164928 A	16 Dec 2015
		CN 105164928 B	01 Dec 2017
		EP 2800283 A1	05 Nov 2014
		JP 2016522617 A	28 Jul 2016
		JP 6117429 B2	19 Apr 2017
		KR 20150135499 A	02 Dec 2015
		KR 101711321 B1	28 Feb 2017
		WO 2014177481 A1	06 Nov 2014
CN 106060950 A	26 Oct 2016	CN 106060950 A	26 Oct 2016
US 2015282144 A1	01 Oct 2015	US 2015282144 A1	01 Oct 2015
		JP WO2014069262 A1	08 Sep 2016
		Wo 2014069262 A1	08 May 2014
us 2015295621 A1	15 Oct 2015	us 2015295621 A1	15 Oct 2015
		us 9654172 B2	16 May 2017
		CN 103229445 A	31 Jul 2013
		CN 103229445 B	02 Dec 2015
		EP 2928167 A1	07 Oct 2015
		EP 2928167 A4	02 Mar 2016
		EP 2928167 B1	26 Apr 2017
		EP 3244544 A1	15 Nov 2017
		WO 2014100991 A1	03 Jul 2014