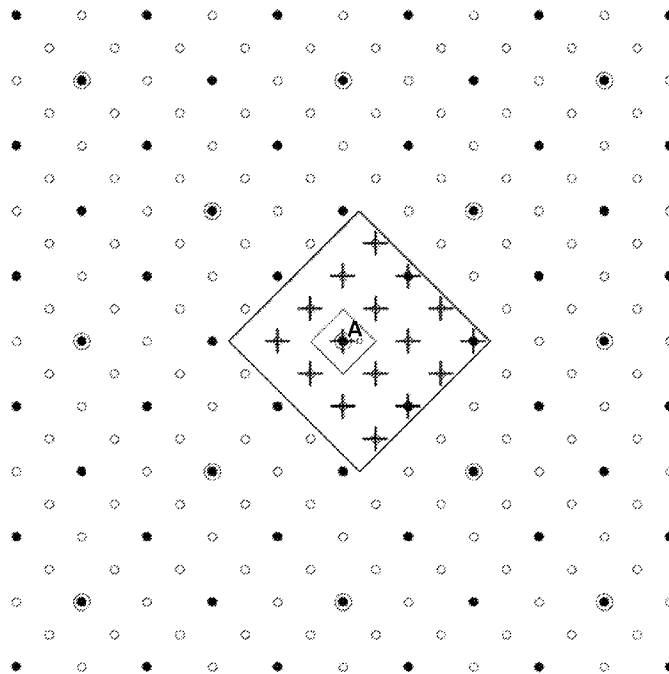




- (51) International Patent Classification:  
*H04L 27/34* (2006.01)      *H04L 27/38* (2006.01)
- (21) International Application Number:  
PCT/US2023/062953
- (22) International Filing Date:  
21 February 2023 (21.02.2023)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
63/312,496      22 February 2022 (22.02.2022)      US
- (71) Applicant: **RAMPART COMMUNICATIONS, INC.**  
[US/US]; 166 Defense Hwy, Suite 303, Annapolis, Maryland 21401 (US).
- (72) Inventors: **ROBINSON, Matthew Brandon**; c/o Rampart Communications, Inc., 166 Defense Hwy, Suite 303, Annapolis, Maryland 21401 (US). **MACKES, Stephen Douglas**; c/o Rampart Communications, Inc., 166 Defense Hwy, Suite 303, Annapolis, Maryland 21401 (US).

- (74) Agent: **HUTTER, Christopher R.** et al.; Cooley LLP, 1299 Pennsylvania Ave. NW, Suite 700, Washington, District of Columbia 20004 (US).
- (81) Designated States (*unless otherwise indicated, for every kind of 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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, 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, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.
- (84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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,

(54) Title: METHODS AND APPARATUS FOR SIGNAL MODULATION USING LATTICE-BASED SIGNAL CONSTELLATIONS



**FIG. 6**

(57) Abstract: A data modulation method includes receiving a bit string at a processor, and identifying a set of binary strings based on the bit string. Each binary string from the set of binary strings is mapped to an element from a set of elements of a lattice-based signal constellation, without using a lookup table. Real-valued points from the set of elements are identified based on the mapping. The method also includes causing transmission of a signal having a modulation based on the real-valued points.



LU, LV, MC, ME, 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).

**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

**Published:**

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

# METHODS AND APPARATUS FOR SIGNAL MODULATION USING LATTICE-BASED SIGNAL CONSTELLATIONS

## CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to and the benefit of U.S. Provisional Patent Application No. 63/312,496, filed February 22, 2022 and titled “METHODS AND APPARATUS FOR SIGNAL MODULATION USING LATTICE-BASED SIGNAL CONSTELLATIONS,” the contents of which are incorporated by reference herein in their entirety.

## FIELD

[0002] The present disclosure relates to signal processing for communication systems, and more specifically, to lattice-based data modulation methods.

## BACKGROUND

[0003] A lattice is a periodic arrangement of points in an  $n$ -dimensional space. Communications engineers and information theorists use lattices for quantization and modulation, for example to perform lossy compression (“source coding”) and/or noise immunity (“channel coding”).

## SUMMARY

[0004] In some embodiments, a data modulation method includes receiving a bit string at a processor, and identifying a set of binary strings based on the bit string. Each binary string from the set of binary strings is mapped to an element from a set of elements of a lattice-based signal constellation, without using a lookup table. Real-valued points from the set of elements are identified based on the mapping. The method also includes causing transmission of a signal having a modulation based on the real-valued points.

[0005] In some embodiments, a data demodulation method includes receiving at a processor and from a transmitter, a signal encoding a bit string. A point of an  $n$ -dimensional lattice associated with the signal is identified, via the processor, based on the signal and a closest vector algorithm. The point of the  $n$ -dimensional lattice is multiplied, via the processor, by an inverse of a basis of the  $n$ -dimensional lattice, to identify an element from a plurality of elements of an  $n$ -dimensional matrix  $\mathbb{Z}^n$  associated with the signal. Bits are recovered based on

the signal, via the processor, by reducing components of the signal modulo  $m$ , to produce an element of a quotient  $\mathbb{Z}^n/m\mathbb{Z}^n$ .

### BRIEF DESCRIPTION OF THE DRAWINGS

[0006] FIG. 1 is a diagram of a lattice-based data modulation system, according to an embodiment.

[0007] FIGS. 2-6 show various aspects of constellation diagrams, illustrating various aspects of the present disclosure, in accordance with some embodiments.

[0008] FIG. 7 is a flowchart illustrating a lattice-based data modulation method, according to an embodiment.

[0009] FIG. 8 is a flowchart illustrating a lattice-based data demodulation method, according to an embodiment.

[0010] FIG. 9 is a plot of probabilities for error bounds associated with Leech lattices, according to an embodiment.

[0011] FIG. 10 shows an example hexagonal lattice, for use in data modulation / demodulation, in accordance with one or more embodiments.

[0012] FIG. 11 shows shaping region vs. shape gain tradeoff curves, for different values of  $N$  associated with an  $N$ -dimensional shaping region, in accordance with one or more embodiments.

### DETAILED DESCRIPTION

[0013] Vast literature exists on modulation techniques for wireless communications. One related area of research is in using various types of dense packings and/or (generally regular) lattices to modulate data into higher dimensional spaces. These ideas go back to the original work of Claude Shannon (see, e.g., Shannon, Claude E., "Probability of Error for Optimal Codes in a Gaussian Channel," Bell System Tech. J. 38, 611-656) and the Shannon-Hartley theorem, and research continues through today.

[0014] One known idea within the wireless communications community is to use points in higher dimensional spaces that are more "dense" than the usual quadrature amplitude modulation / amplitude and phase-shift keying (QAM/PSK) constellations. This allows greater distance (e.g., Hamming distance or Euclidean distance) between the constellation points, which decreases the probability that channel noise or other channel distortions will

cause an error. There have also been many attempts to combine coding theory with modulation, such as trellis coded modulation and multilevel codes

**[0015]** Most of the foregoing approaches use either a coding theoretic construction (e.g., one of the approaches discussed in Conway, J and Sloane, N, “Sphere Packings, Lattices, and Groups, Springer, 1993) or a lattice constellation without underlying coding.

**[0016]** Known coding theoretic construction approaches typically include breaking a lattice, or a set of lattices, into a set of lattice cosets. A set of coded bits is then used to select a coset (as a subset of the lattice or set of lattices), and a set of uncoded bits is used to select a point within the subset. For example, a set of message bits may be run through a standard (binary) error correcting code (e.g., a convolutional encoder), and the output (encoded) bits are used to select a coset. The remaining message bits are then used to select a point within that coset.

**[0017]** Another known approach is to use a lattice constellation without underlying coding, but instead to use an elaborate series of lookup tables (which becomes intractable at higher throughputs/larger constellations), or to use more geometrically convenient shaping regions such as rectangles (which reduce the efficiency of the constellations).

**[0018]** In contrast to the foregoing known techniques, some embodiments of the present disclosure facilitate bit-to-symbol mapping and symbol-to-bit mapping for a lattice based constellation, for example in a modem / baseband processor and for mapping bits into complex baseband I/Q points, irrespective of any underlying coding theoretic schemes, and without singling out closest elements of the lattices. Moreover, embodiments of the present disclosure do not use lookup-tables to map bit strings to lattice points, or to subsets of lattice points. Still further, in some implementations, embodiments of the present disclosure do not use rectangular shaping regions.

**[0019]** As used herein, a lattice can refer to a set of points in an  $n$ -dimensional space given by all linear combinations with integer coefficients of a basis set of up to  $n$  linearly independent vectors. One example of a lattice is a Leech lattice, discussed further below. As used herein, a Voronoi region of a lattice point can refer to the region of the  $n$ -dimensional space closer to that lattice point than to all other lattice points. A code is a finite set of codewords having a specified length, a codeword being a sequence of symbols encoding a message to be transmitted within a communication system. Codewords are translated into signals (coded signals) via modulation to real and/or complex values. Coded signals can be represented as points within a

signal space. A lattice code is defined by a finite set of lattice points within a predefined region of a given lattice, the predefined region referred to herein as a “shaping region.”

**[0020]** Consider an ordered basis  $B$  for an  $n$ -dimensional lattice  $\Lambda^n$ . The basis is an isomorphism from  $\mathbb{Z}^n$  (an  $n$ -dimensional vector of integers) to points of  $\Lambda^n$ . Furthermore, note that  $B$  restricts to an isomorphism of subgroups  $\lambda\mathbb{Z}^n \rightarrow \lambda\Lambda^n$  for  $\lambda \in \mathbb{Z}$ . Illustrating this below, notice that since the square on the left side of the diagram commutes, the composition of the basis with the quotient map  $Q$  has kernel (“ker”)  $\lambda\mathbb{Z}^n$  (i.e.,  $\ker(Q \circ B) = \lambda\mathbb{Z}^n$ ). Now, by the universal property of cokernels, there exists a unique homomorphism,  $\sigma: \mathbb{Z}^n/\lambda\mathbb{Z}^n \rightarrow \Lambda^n/\lambda\Lambda^n$  that makes the square on the right commute.

$$\begin{array}{ccccc}
 \mathbb{Z}^n & \xleftarrow{\lambda} & \mathbb{Z}^n & \longrightarrow & \mathbb{Z}^n / \lambda\mathbb{Z}^n \\
 B \downarrow & & B \downarrow & & \sigma \downarrow \\
 \Lambda^n & \xleftarrow{\lambda} & \Lambda^n & \xrightarrow{Q} & \Lambda^n / \lambda\Lambda^n
 \end{array}$$

**[0021]** By employing the same strategy using  $B^{-1}$ , a map  $\tau: \Lambda^n/\lambda\Lambda^n \rightarrow \mathbb{Z}^n/\lambda\mathbb{Z}^n$  is obtained, which makes the right side commute. Therefore,  $\tau = \sigma^{-1}$ , making  $\sigma$  an isomorphism. So, the quotient  $\Lambda^n/\lambda\Lambda^n$  is isomorphic as an abelian group to the quotient  $\mathbb{Z}^n/\lambda\mathbb{Z}^n$ . The space  $\mathbb{Z}^n/\lambda\mathbb{Z}^n$  is the set of  $n$  copies of integers modulo  $\lambda$  (i.e., of the form  $(a_1, a_2, a_3, \dots, a_n)$ , where each  $a_i \in (0, \dots, \lambda-1)$ ). This size of this set is  $\lambda^n$ . And, because of the isomorphism implied by the above commutative diagram, the size of  $\Lambda^n/\lambda\Lambda^n$  is also  $\lambda^n$ .

**[0022]** If the value of  $\lambda$  is taken to be a power of 2 (such that  $\lambda = 2^k$ ), then the number of points in the quotient  $\Lambda^n/\lambda\Lambda^n$  is  $2^{nk}$ . This means that there is a bijection from binary strings (or sub-strings) of length  $nk$  to the elements of the quotient  $\Lambda^n/\lambda\Lambda^n$ .

**[0023]** As an explicit construction of this bijection, consider the set of  $2^{nk}$  strings of  $nk$  bits. If any given bit string is broken into blocks or sub-strings of  $k$  bits, each  $k$ -bit sub-string can be naturally mapped to the integers mod  $\lambda$  by treating that  $k$ -bit sub-string as the binary expansion of the integer. In this way, each of the length- $nk$  bit strings is mapped to  $\mathbb{Z}^n/\lambda\mathbb{Z}^n$  -  $n$  values taken from the integers mod  $\lambda = 2^k$ . So, given a length- $nk$  bit string, mapped to an element  $\alpha \in \mathbb{Z}^n/\lambda\mathbb{Z}^n$ , a point of the lattice  $\Lambda$  can be produced using the basis. Specifically, the point of the lattice will be  $p = B\alpha \in \Lambda^n$ .

**[0024]** The point  $p$  can be reduced to an enlarged Voronoi cell at the origin by subtracting the point of  $\lambda A^n$  that is closest to  $p$ , resulting in a point  $q \in A^n/\lambda A^n$ .

**[0025]** One subtlety that may arise is that the fundamental/Voronoi region of  $\lambda A^n$  may pass through some of the points of  $A^n$ . Suppose that a point  $x \in A^n$  intersects the fundamental region of  $\lambda A^n$ . By the group property of lattices, this means that  $-x$  also intersects the fundamental region of  $\lambda A^n$ . As used herein, a “fundamental region” refers to a Voronoi cell of the origin, or a lattice region in which every point is closer to the origin than to any other lattice point. If  $x = B\alpha$ , then by linearity, the point that produces  $-x$  is  $-\alpha$ . But both  $\alpha$  and  $-\alpha$  are both elements of  $Z^n/\lambda Z^n$ .

**[0026]** As an illustration of this, let  $n = 2$  and let  $k = 1$ , and let the lattice basis  $B$  have basis vectors  $(1,1)$  and  $(1,-1)$ , so the matrix is  $B =$

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

**[0027]** The set of valid  $\alpha$  vectors is  $(0,0)$ ,  $(0,1)$ ,  $(1,0)$ , and  $(1,1)$ .  $B$  maps these, respectively, to  $(0,0)$ ,  $(1,-1)$ ,  $(1,1)$ , and  $(2,0)$ . However, consider the additive inverse of the three non-zero  $\alpha$  vectors -  $(0,-1)$ ,  $(-1,0)$ , and  $(-1,-1)$ .  $B$  maps these, respectively, to  $(-1,1)$ ,  $(-1,-1)$ , and  $(-2,0)$ . However, the points  $(0,-1)$  and  $(0,1)$  are equivalent modulo 2, as are the points  $(-1,0)$  and  $(1,0)$ , and the points  $(-1,-1)$  and  $(1,1)$ . So, for the points that lie on the surface of the fundamental region of  $\lambda A^n$ , there is an ambiguity.

**[0028]** The ambiguity can be resolved by creating a convention for the selection of points to be used. This convention can include, for example, selecting a face of a fundamental region to be used, and then only using the point on that (selected) face. Alternatively, the convention can include slightly shifting the fundamental region away from the origin, to “break” the symmetry that resulted in points on the fundamental region surface. For example, slightly shifting the fundamental region (i.e., resulting in a “shifted fundamental region”) can refer to shifting the fundamental region by a smallest amount (and in an appropriate direction) that prevents any points from intersecting the enlarged fundamental region.

**[0029]** Turning now to the drawings, FIG. 1 is a diagram of a lattice-based data modulation system, according to an embodiment. The lattice-based data modulation system 100 can be used, for example, for remediating signal distortion by correcting timing and frequency offsets. As shown in FIG. 1, the lattice-based data modulation system 100 includes a signal transmitter

110 in communication (e.g., via a wired or wireless communications network “N”) with a signal receiver 130. Optionally, one or both of the signal transmitter 110 and signal receiver 130 is also in communication (e.g., via a wired or wireless communications network “N”) with one or more remote compute devices 120 (e.g., for remote storage of data). The signal transmitter 110 includes a processor 112 operably coupled to a communications interface 114 and to a memory 116. The memory 116 stores data and/or processor-executable instructions (e.g., to perform method 700 of FIG. 7, as discussed below). For example, as shown in FIG. 1, the memory 116 includes bit strings 116A, binary strings 116B, lattice-based signal constellations 116C (including lattice elements 116D), real-valued points 116E, symbols 116F, algorithms 116G (e.g., one or more closest vector algorithms), and optionally quotients 116H. Similarly, the signal receiver 130 includes a processor 132 operably coupled to a communications interface 134 and a memory 136. The memory 136 stores data and/or processor-executable instructions (e.g., to perform method 800 of FIG. 8, as discussed below). For example, as shown in FIG. 1, the memory 136 includes bit strings 136A, binary strings 136B, lattice-based signal constellations 136C (including lattice elements 136D), real-valued points 136E, symbols 136F, algorithms 136G (e.g., one or more closest vector algorithms), and optionally quotients 136H.

**[0030]** To illustrate the procedures discussed above, consider the constellation diagrams shown in FIGS. 2-6. of FIG. 2 shows the lattice generated by the B matrix given above. Every dot in this diagram represents a point of the lattice  $\Lambda^2$ . The empty circles represent points that are only in  $\Lambda^2$ . The filled in dots are points in both  $\Lambda^2$  and  $2\Lambda^2$ . The filled in dots with a circle around them are in  $\Lambda^2$ ,  $2\Lambda^2$ , and  $4\Lambda^2$ . The diamond at the center of the diagram represents the fundamental region of  $\Lambda^2$  (taking the point at the center of the diamond as the origin). In this example,  $n=2$  (i.e., a 2-dimensional lattice).

**[0031]** By taking  $k=1$ , the fundamental region of  $2\Lambda^2$  can be included, as shown in FIG. 3. Two of the sides of the fundamental region are drawn with solid lines, and the two opposite sides are drawn with dashed lines. The diagram of FIG. 3 illustrates the implementation of a convention in which only (1) points that are either entirely inside the fundamental region, and (2) points that are on the surface if and only if the points intersect a solid line, are included. The  $\lambda^n = (2^k)^n = (2^1)^2 = 4$  points that are selected by this convention are shown with plus (+) symbols drawn over them. These 4 points are the lattice points that will be used in bit-to-symbol mapping.



**[0032]** Similarly, for  $k=2$ , the fundamental region of  $4\Lambda^2$  can be drawn, and a convention similar to the convention applied in FIG. 3 can be used, as shown in FIG. 4. In the diagram of FIG. 4, the  $\lambda^n = (2^k)^n = (2^2)^2 = 16$  points that are selected for use in data modulation are represented by a plus (+). In other words, these 16 points are the lattice points that will be used in bit-to-symbol mapping.

**[0033]** In other embodiments, the fundamental region may be shifted, as shown in FIG. 5 for  $k=1$ . A small diamond (square rotated  $45^\circ$ ) is shown at the center point of the fundamental region of  $2\Lambda^2$ , and the 4 selected points are marked with a plus (+).

**[0034]** Similarly, as shown in the embodiment of FIG. 6, for  $k=2$ , the fundamental region may be shifted, and the 16 points selected for use in data modulation marked by a plus (+).

**[0035]** In some implementations, when using a small offset from the origin to produce the  $2^{nk}$  points, it can be advantageous to shift each point of a fundamental region to make the dot labelled "A" the actual center of the set of points, so that the set of all data points has zero mean. The offset is a single vector in Euclidean space ( $\mathbb{R}_n$ ), and is not made of lattice points.

**[0036]** In some embodiments, a data modulation method includes performing a bit-to-symbol mapping that uses a lattice constellation.

**[0037]** In some embodiments, a data demodulation method includes performing a symbol-to-bit mapping that uses a lattice constellation.

**[0038]** In some embodiments, a lattice-based data modulation method or demodulation method does not include lattice coding, and does not include the use of redundant lattice points.

**[0039]** In some embodiments, a data modulation method or demodulation method includes using a quotient of a lattice with a copy of that lattice scaled by a positive integer.

**[0040]** In some embodiments, a data modulation method includes mapping  $2^{nk}$  binary strings (of length  $nk$ ) to elements of a quotient  $\Lambda^n/2^k\Lambda^n$ , or to a shifted copy (e.g., as discussed and shown with reference to FIG. 5) of said quotient. This can be performed by first mapping each length  $nk$  binary string to an element  $\alpha \in \mathbb{Z}^n/\lambda\mathbb{Z}^n$ , then using the lattice basis  $B$  to produce a point  $B\alpha \in \Lambda^n$ . Then, the closest point to  $B\alpha$  in  $2^k\Lambda^n$  is determined via a selected closest vector algorithm (e.g., selected prior to use of the closest vector algorithm), and that closest point is subtracted from  $B\alpha$ . The resulting point is guaranteed to be in the fundamental region of  $2^k\Lambda^n$  (or, equivalently, in  $\Lambda^n/2^k\Lambda^n$ ). The real-valued points into which the bits are mapped can then be used as the in-phase and quadrature components of a signal at baseband. In some implementations, the real-valued points are the same as the closest points to  $B\alpha$  in  $2^k\Lambda^n$ .

Alternatively or in addition, given a set of points (e.g., the 16 points shown in FIG. 6), all of the points will be used in the modulation scheme. For example, 8 of the points may be used for the in-phase component of the signal and the remaining 8 points may be used for the quadrature component of the signal. Alternatively or in addition, the mapping of the  $2^{nk}$  binary strings to the elements of the quotient  $\Lambda^n/2^k\Lambda^n$  (or to the shifted copy of said quotient) is performed without using a rectangular shaping region. Alternatively or in addition, the mapping of the  $2^{nk}$  binary strings to the elements of the quotient  $\Lambda^n/2^k\Lambda^n$  (or to the shifted copy of said quotient) is not based on lattice coding.

**[0041]** The real-valued points can then be transmitted (optionally with any of: upsampling, filtering, driving to carrier, sending through a digital-to-analog converter (DAC), etc.), and received by a recipient. The receiver may then perform synchronization and/or equalization. To demodulate / recover the bits, the receiver will then use the pre-selected closest vector algorithm to recover the “closest” point of  $\Lambda^n$ , and this recovered closest point is either multiplied by the inverse  $B^{-1}$ , thereby producing an element of  $\mathbb{Z}^n$ , or it is used to recover soft information about the received value which can then be fed into an error correction scheme. The individual components (i.e., the elements of  $\mathbb{Z}^n$ ) are then reduced mod  $2^k$  (i.e., modular arithmetic is performed on the individual components), to produce an element of  $\mathbb{Z}^n/2^k\mathbb{Z}^n$  (or, stated more generally,  $\mathbb{Z}^n/m\mathbb{Z}^n$ , where  $m$  is a positive integer). If the quotient involves an ambiguity resulting from points of  $\Lambda^n$  that lie on the surface of the fundamental region of  $2^k\Lambda^n$ , the ambiguity may be resolved (e.g., as discussed above), for example, by enforcing a convention that selects a set of “faces” of the fundamental region to use. Alternatively, the ambiguity can be resolved by enforcing a convention that includes centering the fundamental region of  $2^k\Lambda^n$  around a point located away from the origin that does not result in elements of  $\Lambda^n$  that lie on a surface of its fundamental region (as discussed above in reference to FIGS. 5 and 6). If the shift away from the origin is used, then the resulting constellation may be shifted by a constant vector to ensure it has a zero mean.

**[0042]** FIG. 7 is a flowchart illustrating a lattice-based data modulation method, according to an embodiment. As shown in FIG. 7, the method 700 includes receiving, at 702, a bit string at a processor, and identifying, at 704, a set of binary strings based on the bit string. Each binary string from the set of binary strings is mapped, at 706, to an element from a set of elements of a lattice-based signal constellation, without using a lookup table. Mapping binary strings to the elements of the lattice-based signal constellation, without using a lookup table is advantageous

for at least the reason that, as discussed above, the use of lookup tables can become intractable at higher throughputs and/or for larger constellations. Real-valued points from the set of elements are identified at 708 based on the mapping. The method 700 also includes causing transmission of a signal having a modulation based on the real-valued points at 710.

**[0043]** In some implementations, the real-valued points represent in-phase/quadrature (I/Q) points or components.

**[0044]** In some implementations, the set of elements is associated with a quotient of a first lattice and a second lattice, the second lattice being a copy of the first lattice scaled by a positive integer.

**[0045]** FIG. 8 is a flowchart illustrating a lattice-based data demodulation method, according to an embodiment. As shown in FIG. 8, the method 800 includes receiving, at 802, at a processor and from a transmitter, a signal representing a bit string. A point of an  $n$ -dimensional lattice associated with the signal is identified at 804, via the processor, based on the signal and a closest vector algorithm. The point of the  $n$ -dimensional lattice is multiplied, at 806 and via the processor, by an inverse of a basis of the  $n$ -dimensional lattice, to identify an element from a plurality of elements of an  $n$ -dimensional matrix  $Z^n$  associated with the signal. Bits are recovered at 808 based on the signal, via the processor, by reducing the I/Q points of the signal modulo  $m$  (e.g.,  $2^k$ , where “k” represents a number of bits in the bit string), to produce an element of a quotient  $Z^n/mZ^n$  (e.g.,  $Z^n/2^kZ^n$ ).

**[0046]** In some implementations, the method also includes performing at least one of synchronization or equalization of the signal prior to identifying the point of an  $n$ -dimensional lattice associated with the signal.

**[0047]** In some implementations, the point is a first point, and the method also includes identifying, via the processor, an ambiguity associated with the quotient  $Z^n/mZ^n$ . In response to identifying the ambiguity, a set of faces of a fundamental region of the  $n$ -dimensional lattice can be selected via the processor, and a second point of the  $n$ -dimensional lattice associated with the signal can be identified via the processor based on the set of faces of a fundamental region of the  $n$ -dimensional lattice, the instructions to recover the bits including instructions to recover the bits further based on the second point.

**[0048]** In some implementations, the point is a first point, and the method also includes identifying, via the processor, an ambiguity associated with the quotient  $Z^n/mZ^n$ . In response to identifying the ambiguity, a fundamental region of the  $n$ -dimensional lattice is centered

around a second point of the  $n$ -dimensional lattice that is distanced from an origin of the  $n$ -dimensional lattice, such that the plurality of elements of the  $n$ -dimensional matrix  $\mathbb{Z}^n$  do not lie on a surface of the fundamental region.

### **Leech Lattices**

**[0049]** Some embodiments of the present disclosure improve upon known communications techniques using Leech lattices as the basis for a lattice-based signal constellation. The Leech lattice is an even unimodular lattice  $\Lambda_{24}$  in 24-dimensional Euclidean space, and is an attractive candidate. The following discussion quantifies the gain achievable by Leech lattices, as contrasted with some known constellations, assuming spherical shaping regions, which are approximated, in some instances, using a scaled Voronoi as described herein.

**[0050]** The Leech lattice is expected to have some interesting communication capabilities. Consider the set of points (also referred to herein as “centers”) of  $\Lambda$  contained within and on a sphere of radius  $4\sqrt{n}$  for some fixed positive integer  $n$ , and denote this set by  $\Lambda^n$  i.e.,  $\Lambda^n = \bigcup_{i=1}^n \Lambda_i$ . Each point can serve as a signal vector. Those signals in  $\Lambda_n$  are referred to herein as surface vectors, and the others are referred to herein as internal vectors of  $\Lambda^n$ . The decision region for an internal vector is bounded by 196,560 hyper-planes, which makes it unlikely that an exact expression for the probability of error will be found. An upper bound for the probability of error and/or a lower bound for the probability of error, however, can be found.

**[0051]** Assume that all signals are equally likely, and consider the probability of error given that an internal vector was sent. Further, assume that the channel has the effect of additively perturbing each signal component independently by a gaussian random variable of mean zero and variance  $N_0/2$ . The probability of a correct decision  $P(C)$  is greater than the probability that the perturbed vector remains within a sphere of radius  $2\sqrt{2}$  about the sent signal vector. To calculate this probability, denote the twenty-four dimensional gaussian distribution of independent variates, each with mean zero and variance  $N_0/2$ , by:

$$\rho(x) = \exp[-|x|^2/N_0]/(\pi N_0)^{12}.$$

**[0052]** A sphere of radius  $r$  in  $E_{24}$  has a surface area:

$$S(r) = \frac{2 \cdot \pi^{12}}{\Gamma(12)} r^{23}.$$

[0053] Thus, the probability of the signal vector remaining inside a sphere of radius  $2\sqrt{2}$  is

$$\begin{aligned} &= \int_0^{2\sqrt{2}} S(r)p(|x| = r) dr \\ &= \frac{2}{\Gamma(12)} \left(\frac{1}{N_0}\right)^{12} \int_0^{2\sqrt{2}} r^{33} \exp(-r^2/N_0) dr \end{aligned}$$

[0054] By the change of variable  $y = r^2/N_0$  this last expression may be written as:

$$\frac{1}{\Gamma(12)} \int_0^{8/N_0} y^{11} e^{-y} dy = P(12, 8/N_0),$$

where  $P(a, x)$  is the incomplete gamma function. The probability of a correct decision for a surface vector will be slightly greater than this quantity. Here, the region of integration is over the same sphere as above, plus a conical region exterior to the sphere of radius  $2\sqrt{n}$ . The integration over the conical region may be expressed in terms of a double integral that may be complex to compute, and that may be small except at very high noise levels. For simplicity, this term may be ignored. Hence, a bound on the probability of error  $P_L(E)$  for the subset of the Leech Lattice  $A^n$  is given by:

$$P_L(E) < 1 - P(12, 8/N_0),$$

which may be written as:

$$P_L(E) < e_{11}(8/N_0) \exp[-8/N_0], \quad (1)$$

where

$$e_{11}(x) = \sum_{i=0}^{11} \frac{x^i}{i!}.$$

[0055] The signal vectors of  $A^n$  are energy constrained by:

$$E_8 = 24E_N \leq 16n,$$

and so  $E_N = (16/24)n = 2n/3$ , where  $E_N$  is the energy per dimension and the subscript  $N$  is 24. Thus, in the upper bound (1),

$$\frac{8}{N_0} = \frac{E_N}{N_0} \frac{12}{n}.$$

**[0056]** Compare the bound (1) to the bound  $P_G(E)$ , for signals constrained in energy by  $24E_N$ :

$$P_G(E) < B \exp \left[ -N \left\{ \left( 1 - \beta + \frac{1}{2} 1n \left( \beta + \frac{E_N}{2N_0} \right) - R \right) \right\} \right], \quad (2)$$

$$R < \frac{1}{2} 1n \left( 1 + \frac{2E_N}{N_0} \right),$$

where

$$B = 24\pi e^2 (1 - \beta^2)$$

and

$$\beta = \frac{1}{2} \left[ 1 - \frac{E_N}{N_0} + \left( 1 + \left( \frac{E_N}{N_0} \right)^2 \right)^{1/2} \right].$$

**[0057]** The probabilities for the two error bounds (1) and (2) are plotted in FIG. 9 for  $n = 10$  or, equivalently, for  $R = 1.2597$ , where:

$$R = \frac{1}{24} \log_e M,$$

$M$  being the number of signals. The curves are labelled by their equation numbers. Notice that only certain rates are possible with the Leech Lattice. As can be seen in FIG. 9, the upper bound for the Leech Lattice is lower than Gallagers bound, for  $n = 10$ .

**[0058]** For large values of  $E_n/N_0$  ( $E_n/N_0 > 10$ ),  $\beta \rightarrow \frac{1}{2}$  and Gallagers bound behaves as:

$$K(E_N/N_0)^{-12}, \quad (3)$$

where  $K$  is a constant. The bound for  $A^n$  in (1) behaves as

$$K' \left( \frac{E_N}{N_0} \right)^{11} \exp \left[ - \frac{E_N}{N_0} \frac{12}{10} \right],$$

which gives exponential decrease rather than  $n$ -th power. This is reflected in the behaviour of the curves of FIG. 9.

**[0059]** The two bounds behave in approximately the same manner for other values of  $n$ . Consider the effect of the change from  $n = 10$  to  $n = 100$ . Using the asymptotic value for the number of points in  $A^{100}$ , one obtains:

$$R' = 2.38870.$$

**[0060]** The effect on the bound (1) is merely to shift the curve to the right by 10 db. The effect on bound (2) is to multiply values for  $n = 10$  by  $e^{N\Delta R} = e^{24 \times 1.1290} = e^{27.096} = 5.852 \times 10^{11}$ . For values of  $E_N/N_0$  such that  $10 \log_{10}(E_N/N_0) > 10$ , i.e.,  $E_N/N_0 > 10$ , the behaviour indicated by Eq. (3) becomes a very good approximation. On the logarithmic scales used in the figure, this implies that the bound (2) will be very close to a straight line. Using this straight line approximation to the bound, it may be seen that multiplication by  $5.852 \times 10^{11}$  is equivalent to a right shift of the bound by an amount slightly greater than 10 db.

**[0061]** Note that the bound (2) is valid only for  $R < \frac{1}{2} 1n(1 + 2E_N/N_0)$  which implies, for  $n = 10$  and  $R = 1.2597$ , that the bound is valid only for:

$$10 \log_{10}(E_N/N_0) > 7.5686.$$

**[0062]** Attempts were made, without success, to find a reliability or exponent function for the bound of (1).

### **Euclidean Codes**

**[0063]** A Euclidean code is a finite set of points  $x_1, \dots, x_M$  in  $n$ -dimensional real Euclidean space  $R^n$ . Euclidean codes can be used as signal sets for a Gaussian channel, as representative

points (or output vectors) in a vector quantizer, and in many other applications. There are a number of desirable properties that a Euclidean code should have:

- 1) the number  $M$  of code vectors should be large;
- 2) the total energy  $\sum \|x_i\|^2$  (or alternatively the peak energy  $\max \|x_i\|^2$ ) should be small;
- 3) the minimum distance between the  $x_i$  should be large (or alternatively the probability of incorrect decoding should be small, if the code is used on a Gaussian channel);
- 4) given  $k$ , it should be possible to readily find the  $k$ th code vector  $x_k$ ;
- 5) given  $x_k$ , it should be possible to readily find its index  $k$ ; and
- 6) given an arbitrary point  $z \in \mathbf{R}^n$ , it should be possible to readily find the closest code vector  $x_k$ .

**[0064]** The foregoing properties (also referred to herein as “problems” to solve) are not necessarily independent of one another, and there may be tradeoffs between / among them. Lattice codes include a subset of points of some lattice in  $\mathbf{R}^n$ , which is a set of points of a lattice packing of equal  $n$ -dimensional spheres. The code is defined by specifying a lattice  $\Lambda$  in  $\mathbf{R}^n$  and a certain region of the space  $\mathbf{R}^n$ , and consists of all lattice points inside this region.

**[0065]** For these codes, the minimum distance is the minimum distance between lattice points, and the number of code vectors is determined by the density of the lattice. Thus, properties 1) and 3) amount to saying that the lattice should have a high density, and property 2) states that the region of space defining the code should be as nearly spherical as possible.

**[0066]** Some known very fast algorithms have been used to satisfy property 6) for a large class of lattice codes. According to some embodiments of the present disclosure, similar lattices are used to satisfy properties 4) and 5). Since if the codes are used for a Gaussian channel 6) is the decoding problem while 4) and 5) are encoding problems, properties 4) and 5) might be expected to be more readily solved than property 6). For these codes, however, properties 4), 5), and 6) appear to be of comparable difficulty.

**[0067]** In accordance with some embodiments, to solve for properties 4) and 5), only lattice codes that are defined by certain very special regions of space, regions that can be called Voronoi-shaped, are considered, and such regions are referred to herein as Voronoi codes.

**[0068]** Voronoi codes can have the following two drawbacks: only certain rates can be attained, and because the Voronoi-shaped regions defining them are not spherical, they do not in general



have the lowest possible total energy (although the difference may be small). Under the notation set forth herein, the norm  $\|x\|^2$  of a vector  $x$  is its squared length  $x \cdot x$ .

Encoding Algorithm to Solve Problem 4): Finding a Code Vector from its Index

**[0069]** Given an index vector  $(k_1, \dots, k_n)$  with  $0 \leq k_i \leq r - 1$ , suppose one desires to find that vector  $x$  in the Voronoi code  $C_\Lambda(r, a)$  for which  $\text{index}(x) = (k_1, \dots, k_n)$ . To do so, first form  $x' = \sum k_i v_i$ , which has index equal to  $(k_1, \dots, k_n)$  but need not be in the code. The desired lattice point  $x$  is then the unique solution to:

$$x \equiv x' \pmod{r\Lambda}, \quad x \in a + V_r \tag{4}$$

(by the definition of  $C_\Lambda(r, a)$ ). Equation (4) is readily solved if there is an algorithm available for solving for property 6), i.e., for finding the closest point of  $\Lambda$  to an arbitrary point of the space. In fact, if one sets  $z = r^{-1}(x' - a)$ , and  $\lambda$  is the closest point of  $\Lambda$  to  $z$ , then  $x = x' - r\lambda$  is the desired lattice point. Thus, property 4) may be solved as follows.

**[0070]** *Given the index vector  $(k_1, \dots, k_n)$ , calculate  $x' = \sum k_i v_i$  and  $z = r^{-1}(x' - a)$ . Find the closest point  $\lambda \in \Lambda$  to  $z$  (using, for example, the algorithms in [3]). Then  $x = x' - r\lambda - a$  is in the Voronoi code  $C_\Lambda(r, a)$  and has index  $(k_1, \dots, k_n)$ .*

**[0071]** For example, consider a four-dimensional Voronoi code  $C_{D_4}(4, a)$  obtained from a  $D_4$  lattice. Suppose the given index vector is  $(2, 0, 0, 1)$ . Next, calculate:

$$\begin{aligned} x' &= (5, 0, 0, 1), \\ z &= \frac{1}{4} \left( 5, -\frac{3}{16}, -\frac{11}{32}, \frac{15}{32} \right) \\ &= (1.25, -0.05 \dots, -0.09 \dots, 0.12 \dots) \\ \lambda &= (2, 0, 0, 0), \text{ and} \\ x &= (-3, 0, 0, 1) - a. \end{aligned}$$

**Finding the Closest Point of an  $n$ -Dimensional Integer Lattice  $\mathbb{Z}^n$**

**[0072]** In accordance with some embodiments, an algorithm for finding the closest point of the integer lattice  $\mathbb{Z}^n$  to an arbitrary point  $x \in \mathbb{R}^n$  is used. For a real number  $x$ , let:

$$f(x) = \text{closest integer to } x.$$

[0073] In case of a tie, choose the integer with the smallest absolute value. For  $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ , let

$$f(x) = (f(x_1), \dots, f(x_n)).$$

[0074] Also define  $g(x)$ , which is the same as  $f(x)$  except that the *worst* component of  $x$ —that furthest from an integer—is rounded the *wrong way*. In case of a tie, the component with the lowest subscript is rounded the wrong way. More formally, for  $x \in \mathbb{R}$ ,  $f(x)$  and the function  $w(x)$ , which rounds the wrong way, are defined as follows. (Here,  $m$  is an integer.)

If $x = 0$ ,	then $f(x) = 0$ ,
	$w(x) = 1$ .
If $0 < m \leq x \leq m + \frac{1}{2}$ ,	then $f(x) = m$ ,
	$w(x) = m + 1$ .
If $0 < m + \frac{1}{2} < x < m + 1$ ,	then $f(x) = m + 1$ ,
	$w(x) = m$ .
If $-m - \frac{1}{2} \leq x \leq -m < 0$ ,	then $f(x) = -m$ ,
	$w(x) = -m - 1$ .
If $-m - 1 < x < -m - \frac{1}{2}$ ,	then $f(x) = -m - 1$ ,
	$w(x) = -m$ .

(5)

(Ties are handled so as to give preference to points of smaller norm.) Also:

$$x = f(x) + \delta(x),$$

so that  $\delta(x) \leq 1/2$  is the distance from  $x$  to the nearest integer.

[0075] Given  $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ , let  $k$  ( $1 \leq k \leq n$ ) be such that:

$$\delta(x_k) \leq \delta(x_i) \quad \text{for all } 1 \leq i \leq n$$

and

$$\delta(x_k) = \delta(x_i) \quad \forall k \leq i.$$

[0076] Then,  $g(x)$  is defined by:

$$g(x) = (f(x_1), f(x_2), \dots, w(x_k), \dots, f(x_n)).$$

[0077] *Algorithm 1 — To Find the Closest Point of  $\mathbb{Z}^n$  to  $x$ :* Given  $x \in \mathbb{R}^n$ , the closest point of  $\mathbb{Z}^n$  is  $f(x)$ . (If  $x$  is equidistant from two or more points of  $\mathbb{Z}^n$ , this procedure finds the one with the smallest norm.)

[0078] To see that the procedure works, let  $u = (u_1, \dots, u_n)$  be any point of  $\mathbb{Z}^n$ . Then,

$$\|u - x\|^2 = \sum_{i=1}^n (u_i - x_i)^2,$$

which is minimized by choosing  $u_i = f(x_i)$  for  $i = 1, \dots, n$ . Because of (5), ties are broken correctly, favoring the point with the smallest norm.

Finding the Closest Point of  $D_n$

[0079] *Algorithm 2—To Find the Closest Point of  $D_n$  to  $x$ :* Given  $x \in \mathbb{R}^n$ , the closest point of  $D_n$  is whichever of  $f(x)$  and  $g(x)$  has an even sum of components (one will have an even sum, the other an odd sum). If  $x$  is equidistant from two or more points of  $D_n$ , this procedure produces a nearest point having the smallest norm.

[0080] This procedure works because  $f(x)$  is the closest point of  $\mathbb{Z}^n$  to  $x$  and  $g(x)$  is the next closest.  $f(x)$  and  $g(x)$  differ by one in exactly one coordinate, and so precisely one of  $\sum f(x_i)$  and  $\sum g(x_i)$  is even and the other is odd. Again, (5) implies that ties are broken correctly.

[0081] *Example:* Find the closest point of  $D_4$  to  $x$  (0.6, -1.1, 1.7, 0.1). Compute:

$$f(x) = (1, -1, 2, 0)$$

and

$$g(x) = (0, -1, 2, 0),$$

since the first component of  $x$  is the furthest from an integer. The sum of the components of  $f(x)$  is  $1 - 1 + 2 + 0 = 2$ , which is even, while that of  $g(x)$  is  $0 - 1 + 2 + 0 = 1$ , which is odd. Therefore,  $f(x)$  is the point of  $D_4$  closest to  $x$ .

[0082] To illustrate how ties are handled, suppose:

$$x = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right).$$

[0083] In fact,  $x$  is now equidistant from eight points of  $D_4$ , namely (0, 0, 0, 0), any permutation of (1, 1, 0, 0), and (1, 1, 1, 1). The algorithm computes:

$$f(x) = (0, 0, 0, 0), \quad \text{sum} = 0, \text{ even,}$$

$$g(x) = (1, 0, 0, 0), \quad \text{sum} = 1, \text{ odd},$$

and selects  $f(x)$ . Indeed,  $f(x)$  does have the smallest norm of the eight neighboring points.

### Finding the Closest Point of $E_8$

**[0084]** Since  $E_8$  is the union of two cosets of  $D_8$ , the discussion in the previous section leads to the following procedure.

*Algorithm 3—To Find the Closest Point of  $E_8$  to  $x$ :* Given  $x = (x_1, \dots, x_8) \in \mathbb{R}^8$ .

**[0085]** Compute  $f(x)$  and  $g(x)$ , and select whichever has an even sum of components; call it  $y_0$ . Compute  $f(x - \frac{1}{2})$  and  $g(x - \frac{1}{2})$ , where:

$$\frac{1}{2} = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right),$$

and select whichever has an even sum of components; add  $\frac{1}{2}$  and call the result  $y_1$ .

**[0086]** Compare  $y_0$  and  $y_1$  with  $x$ , and choose the closest. For example, to find the closest point of  $E_8$  to

$$x = (0.1, 0.1, 0.8, 1.3, 2.2, -0.6, -0.7, 0.9),$$

compute

$$f(x) = (0, 0, 1, 1, 2, -1, -1, 1), \quad \text{sum} = 3, \quad \text{odd},$$

$$g(x) = (0, 0, 1, 1, 2, 0, -1, 1), \quad \text{sum} = 4, \quad \text{even},$$

and take  $y_0 = g(x)$ . Also,

$$x - \frac{1}{2} = (-0.4, -0.4, 0.3, 0.8, 1.7, -1.1, -1.2, 0.4),$$

$$f(x - \frac{1}{2}) = (0, 0, 0, 1, 2, -1, -1, 0), \quad \text{sum} = 1, \quad \text{odd},$$

$$g(x - \frac{1}{2}) = (-1, 0, 0, 1, 2, -1, -1, 0), \quad \text{sum} = 0, \text{ even},$$

and so

$$\begin{aligned} y_1 &= g(x - \frac{1}{2}) + \frac{1}{2} \\ &= (-0.5, 0.5, 0.5, 1.5, 2.5, -0.5, -0.5, 0.5). \end{aligned}$$

Finally,

$$\|x - y_0\|^2 = 0.65, \quad \|x - y_1\|^2 = 0.95,$$

and it can be concluded that  $y_0 = g(x)$  is the closest point to  $x$ .

### Finding the Closest Point of $A_n, A_n^*, E_7,$ and $E_7^*$

**[0087]** *Algorithm 4—To Find the Closest Point of  $A_n$  to  $x$ :*

*Step 1:* Given  $x \in \mathbb{R}^{n+1}$ , compute  $s = \sum x_i$  and replace  $x$  by

$$x' = x - \frac{s}{n+1} (1, 1, \dots, 1).$$

*Step 2:* Calculate  $f(x^i) = (f(x'_0), \dots, f(x'_n))$  and the *deficiency*  $\Delta = \sum f(x'_i)$ .

*Step 3:* Sort the  $x'_i$  in order of increasing value of  $\delta(x'_i)$  (defined in Section III). We obtain a rearrangement of numbers  $0, 1, \dots, n$ , say  $i_0, i_1, \dots, i_n$ , such that:

$$-\frac{1}{2} \leq \delta(x'_{i_0}) \leq \delta(x'_{i_1}) \dots \leq \delta(x'_{i_n}) \leq \frac{1}{2}.$$

*Step 4:* If  $\Delta = 0$ ,  $f(x')$  is the closest point of  $A_n$  to  $x$ .

If  $\Delta > 0$ , the closest point is obtained by subtracting 1 from the components  $f(x'_{i_0}), \dots, f(x'_{i_{\Delta-1}})$ .

If  $\Delta < 0$ , the closest point is obtained by adding 1 to the components  $f(x'_{i_n}), f(x'_{i_{n-1}}), \dots, f(x'_{i_{\Delta+1}})$ .

**[0088]** *Remarks:* As discussed above,  $f(x)$  is the closest point of  $\mathbb{Z}^{n+1}$  to  $x$ . The procedure described here finds the closest point of  $A_n$  because it makes the smallest changes to the norm of  $f(x)$  needed to make  $\sum f(x'_i)$  vanish.

**[0089]** Step 1 projects  $x$  onto  $x'$ , the closest point of the hyperplane  $\sum x_i = 0$ . Since  $A_n$  is by definition contained in this hyperplane, it may be possible to assume that  $x$  already lies there, in which case Step 1 can be omitted.

**[0090]** The only substantial amount of computation needed is for the sort in Step 3, which takes  $O(n \log n)$  steps. Step 3 can be omitted, however, if  $x$  is expected to be close to  $A_n$ . In this case,  $\Delta$  will be small, and Steps 3 and 4 can be replaced by the following:

**[0091]** *Step 3':* If  $\Delta = 0$ ,  $f(x')$  is the closest point of  $A_n$  to  $x$ . If  $\Delta > 0$ , find the  $\Delta$  components of  $x'$ , say  $x'_{i_n}, x'_{i_{n-1}}, \dots, x'_{i_{n-\Delta+1}}$ , for which  $\delta(x'_i)$  is as small (i.e., as close to  $-\frac{1}{2}$ ) as possible. The closest point of  $A_n$  is obtained by subtracting one from the components  $f(x'_{i_0}), \dots, f(x'_{i_{\Delta-1}})$  of  $f(x)$ .

**[0092]** If  $\Delta < 0$ , find the  $|\Delta|$  components of  $x'$ , say  $x'_{i_n}, x'_{i_{n-1}}, \dots, x'_{i_{n-\Delta+1}}$ , for which  $\delta(x'_i)$  is as large (i.e., as close to  $\frac{1}{2}$ ) as possible. The closest point of  $A_n$  is obtained by adding 1 to the components of  $f(x'_{i_n}), \dots, f(x'_{i_{n-\Delta+1}})$  of  $f(x)$ .

**[0093]** In any case,  $|\Delta|$  cannot exceed  $n/2$ . If  $\Delta$  is expected to be large, however, the first version of the algorithm is preferable. The closest point of  $A_n^*$  can be found using the fact that  $A_n^*$  is the union of  $n + 1$  cosets of  $A_n$ . For example, the hexagonal lattice  $A_2$  is shown in FIG. 10, together with ordinary two-dimensional coordinates  $(u_1, u_2)$  for the points. The three-dimensional coordinates  $(x_0, x_1, x_2)$  with  $x_0 + x_1 + x_2 = 0$  are obtained by multiplying  $(u_1, u_2)$  on the right by the matrix:

$$M = \begin{pmatrix} 1 & 0 & -1 \\ \frac{1}{\sqrt{3}} & \frac{-2}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix}.$$

**[0094]** Conversely, the  $u$ -coordinates may be obtained from the  $x$ -coordinates by:

$$(u_1, u_2) = (x_0, x_1, x_2) \cdot \frac{1}{2} M^{tr}.$$

**[0095]** For example, the points  $(0, 0)$ ,  $(1, 0)$ ,  $(1/2, \sqrt{3}/2)$ ,  $(-1/2, \sqrt{3}/2)$  have  $x$ -coordinates  $(0, 0, 0)$ ,  $(1, 0, -1)$ ,  $(1, -1, 0)$ ,  $(0, -1, 1)$ , respectively. To find the closest point of  $A_n$  to the point  $P$  with coordinates:

$$(u_1, u_2) = (0.4, -0.4),$$

first find the  $x$ -coordinates of  $P$ , which are  $x = (x_0, x_1, x_2) = (0.169, 0.462, -0.631)$ .

**[0096]** Step 1 of the algorithm can be omitted, since  $x_0 + x_1 + x_2 = 0$  holds automatically. Step 2 produces:

$$f(x) = (0, 0, -1),$$

with deficiency  $\Delta = -1$ . At Step 3, one obtains:

$$\delta(x_0) = 0.169 < \delta(x_2) = 0.369 < \delta(x_1) = 0.462.$$

**[0097]** At Step 4, 1 is added to  $f(x_1)$ , obtaining:

$$(0, 1, -1),$$

which is the closest point of  $A_2$ . The  $u$ -coordinates for this point are:

$$(0, 1, -1) \cdot \frac{1}{2} M^{tr} = \left( \frac{1}{2}, -\frac{\sqrt{3}}{2} \right)$$

(see FIG. 10).

[0098] Since  $A_3 \cong D_3$ , Algorithm 2 is preferable to Algorithm 4 for finding the closest point of the face-centered cubic lattice. Finally,  $E_7$  and  $E_7^*$  can be handled via the algorithm for  $A_7$ .

### Decoding the Leech Lattice

[0099] In one or more embodiments of the present disclosure, certain lattices can be defined by ‘code formulas’, as follows. Let the standard binary representation of an integer  $x$  be:

$$x = x_0 + 2x_1 + 4x_2 + \dots,$$

where  $x_k$  is called  $2^k$ 's-coefficient,  $k \geq 0$ , and the congruence relations

$$x = \sum_k x_k 2^k \pmod{2^n}, \quad n \geq 0$$

may be used to determine the coefficients recursively (particularly if  $x$  is negative). Let  $C_0, C_1, C_2, \dots$  be a set of binary block codes of length  $N$ , which must be nested and satisfy certain other conditions to produce a lattice. Then, define  $\Lambda$  as the set of all integer  $N$ -tuples  $x$  whose  $2^k$ 's-coefficient  $N$ -tuple is a code-word in the code  $C_k$ , for all  $k \geq 0$ . The foregoing can be written symbolically by the ‘code formula’:

$$\Lambda = C_0 + 2C_1 + 4C_2 + \dots.$$

[0100] When, as is always the case, the  $2^k$ 's-coefficients are allowed to take on any value for  $k \geq K$ —i.e., when  $C_k$  is the universe  $(N, N)$  code for  $k > K$ , one may write:

$$\Lambda = C_0 + 2C_1 + \dots + 2^{K-1}C_{2^{K-1}} + 2^K\mathbf{Z}^N,$$

where  $\mathbf{Z}^N$  is the  $N$ -dimensional integer lattice.

[0101] One standard definition for the Leech lattice  $\Lambda_{24}$  expresses it as the union of a sublattice  $H_{24}$  and a coset  $H_{24} + \mathbf{a}$  of  $H_{24}$ . Here,  $H_{24}$  is a lattice sometimes called the ‘Leech half-lattice,’ which may be defined by the code formula:

$$H_{24} = (24, 12) + 2(24, 23) + 4\mathbf{Z}^{24},$$

where  $C_0 = (24, 12)$  is the binary Golay code and  $C_1 = (24, 23)$  is a single-parity-check code. The translation 24-tuple  $\mathbf{a}$  may be taken as  $\mathbf{a} = (-3, 1^{23})/2$ .

[0102] The Golay code has minimum distance 8 and 759 weight-8 codewords. The minimum squared distance between points in  $\Lambda_{24}$  or  $H_{24}$  in this representation is  $d_{\min}^2 = 8$ .  $H_{24}$  has 98,256

lattice points of Euclidean norm 8, namely  $(24 \cdot 23/2) \cdot 4 = 1,104$  points with two coordinates of magnitude 2 and 22 of magnitude 0, plus  $759 \cdot 128 = 97,152$  points with eight coordinates of magnitude 1 and 16 of zero.  $\Lambda_{24}$  has 196,560 points of norm 8, namely those of  $H_{24}$  plus  $24 \cdot 4096 = 98,304$  points in  $H_{24} + \mathbf{a}$  with one coordinate of magnitude  $3/2$  and 23 coordinates of magnitude  $1/2$ .

**[0103]** The *Voronoi region*  $\mathbf{R}_v(\mathbf{0})$  of a lattice  $\Lambda$  is the set of points that are at least as close to  $\mathbf{0}$  as to any other point in  $\Lambda$ ; i.e., the Voronoi region is essentially the decision region of a maximum-likelihood decoding algorithm for  $\Lambda$  (up to the ambiguity involved in resolving ties on the boundary). The packing radius  $r_{\min}(\Lambda)$ , or error-correction radius, is the radius of the largest sphere that can be inscribed in  $\mathbf{R}_v(\mathbf{0})$ , and is equal to  $d_{\min}(\Lambda)/2$ , where  $d_{\min}^2(\Lambda)$  is the minimum squared distance between points in  $\Lambda$ . The kissing number  $N_0(\Lambda)$ , or error coefficient, is the number of points on the boundary of  $\mathbf{R}_v(\mathbf{0})$  with norm  $r_{\min}^2(\Lambda)$ , and is equal to the number of points in  $\Lambda$  of norm  $d_{\min}^2(\Lambda)$ . For  $H_{24}$ ,  $d_{\min}^2(H_{24}) = 8$ ,  $r_{\min}^2(H_{24}) = 2$ , and  $N_0(H_{24}) = 98256$  [2]. For  $\Lambda_{24}$ ,  $d_{\min}^2(\Lambda_{24}) = 8$ ,  $r_{\min}^2(\Lambda_{24}) = 2$ , and  $N_0(\Lambda_{24}) = 196,560$ .

**[0104]** Let  $G_{24}$  be the ‘Construction A’ lattice consisting of all integer 24-tuples whose ones-coefficient  $N$ -tuple is a codeword in the  $(24,12)$  binary Golay code; i.e.,  $G_{24}$  has the code formula  $G_{24} = (24,12) + 2\mathbf{Z}^{24}$ . Then,  $H_{24}$  is a sublattice of  $G_{24}$ , and in fact  $G_{24}$  is the union of  $H_{24}$  and a coset  $H_{24} + \mathbf{b}$  of  $H_{24}$ , where one may take  $\mathbf{b} = (2,0^{23})$ .  $H_{24}$  is thus the subset of  $G_{24}$  in which the twos-coefficient 24-tuple has even parity, and  $H_{24} + \mathbf{b}$  is the subset with odd parity.

**[0105]** Any soft-decision decoding algorithm for the Golay code may be used as a decoding algorithm for  $G_{24}$  as follows. Given any real 24-tuple  $\mathbf{r}$ , first find the closest even and odd integers  $k_{j0}$  and  $k_{j1}$  to each coordinate  $r_j$  of  $\mathbf{r}$ . The differences in squared distances,  $\pm [(r_j - k_{j0})^2 - (r_j - k_{j1})^2] / 2 = \pm [(r_j - (k_{j0} + k_{j1}) / 2)]$ , may be taken as the ‘metrics’ for 0 and 1, respectively, for that coordinate in any soft-decision decoding algorithm for the Golay code. The decoded Golay codeword is then mapped back to  $k_{j0}$  or  $k_{j1}$ , at the  $j$ th coordinate, depending on whether the decoded codeword is 0 or 1 in that coordinate. A maximum-likelihood decoding algorithm for the Golay code that uses between about 700 and 800 operations per 24-tuple on average can be used as a maximum-likelihood decoder for  $G_{24}$ ; i.e., this algorithm can be used to find the closest point in  $G_{24}$  to any given real 24-tuple  $\mathbf{r}$ .

**[0106]** A decoding algorithm for  $H_{24}$  can then be specified as follows. Decoding Algorithm 1 ( $H_{24}$ ): Given any real 24-tuple  $\mathbf{r}$  representing a received word, first find the closest point  $\mathbf{x}_0$  in  $G_{24}$  to  $\mathbf{r}$ . Check the parity of the twos-coefficient 24-tuple of  $\mathbf{x}_0$ ; if it is even, then it is in  $H_{24}$ ,



so accept it. If it is odd, then change one coordinate of  $\mathbf{x}_0$  by  $\pm 2$  in the coordinate  $x_{j_0}$  where such a change will increase the squared distance  $(r_j - x_{j_0})^2$  by the least possible amount—i.e., where  $|r_j - x_{j_0}|$  is greatest. The resulting point  $\mathbf{x}'_0$  has even twos-coefficient parity and is thus in  $H_{24}$ .

**[0107]** Additional complexity in Decoding Algorithm 1 beyond decoding  $G_{24}$  can include a parity check and/or a computation and comparison of 24 magnitudes  $|r_j - x_{j_0}|$ , however this complexity may be negligible.

**[0108]** Decoding Algorithm 1 always maps  $\mathbf{r}$  into a point in  $H_{24}$  by construction, but not necessarily into the closest point in  $H_{24}$ . For example, the 24-tuple  $\mathbf{x} = (-1, 1^7, 0^{16})$  is in  $G_{24}$  but not in  $H_{24}$ , and the point  $\mathbf{r} = \mathbf{x}/2$  is at squared distance 2 from both  $\mathbf{x}$  and the origin  $\mathbf{0}$ , which is in all lattices. If the  $G_{24}$  decoder resolves this tie by choosing  $\mathbf{x}$ , then the parity check will fail. Changing one coordinate by  $\pm 2$ , however, cannot result in the origin  $\mathbf{0}$ , which is the closest point in  $H_{24}$ , but rather must result in some other point in  $H_{24}$  of norm 8 that is at squared distance 4 from  $\mathbf{r}$ .

**[0109]** Decoding Algorithm 1 does, however, always map  $\mathbf{r}$  into the closest point  $\mathbf{x}$  in  $H_{24}$  when  $\mathbf{r} - \mathbf{x}$  is within the error-correction radius of  $H_{24}$ , as shall now be shown. In other words, Decoding Algorithm 1 is a bounded-distance decoding algorithm with the same error exponent as a maximum-likelihood decoder for  $H_{24}$ .

**[0110]** *Lemma 1:* Given a 24-tuple  $\mathbf{r}$ , if a point  $\mathbf{x}$  exists in  $H_{24}$  such that  $\|\mathbf{r} - \mathbf{x}\|^2 < 2$ , then Decoding Algorithm 1 decodes  $\mathbf{r}$  to  $\mathbf{x}$ .

**[0111]** *Proof:* Without loss of generality, let  $\mathbf{x} = \mathbf{0}$  since if Decoding Algorithm 1 maps  $\mathbf{r}$  to  $\mathbf{x}$ , then it maps  $\mathbf{r} + \mathbf{x}'$  to  $\mathbf{x} + \mathbf{x}'$ ; i.e., suppose that  $\|\mathbf{r}\|^2 < 2$ . Then, since the first step finds the closest point  $\mathbf{x}_0$  in  $G_{24}$ ,  $\mathbf{x}_0$  must either be  $\mathbf{0}$  or a point  $\mathbf{x}_0$  in  $G_{24}$  of norm  $\|\mathbf{x}_0\|^2 < 8$ . The only points in  $G_{24}$  with norm less than 8 are  $\mathbf{0}$  and the points with a single nonzero coordinate of magnitude 2. If  $\mathbf{x}_0$  is any of the latter points, however, then parity will fail, and  $\mathbf{0}$  will be one of the candidates for the modified point  $\mathbf{x}_0'$ . In fact,  $\mathbf{0}$  must then be chosen because any other candidate point is in  $H_{24}$ , therefore has norm at least 8, and thus cannot be closer to  $\mathbf{r}$  than  $\mathbf{0}$ . Hence all points  $\mathbf{r}$  with  $\|\mathbf{r}\|^2 < 2$  map to  $\mathbf{0}$ .

**[0112]** The set of all points  $\mathbf{r}$  that map to  $\mathbf{0}$  is called the decision region  $\mathbf{R}_1(\mathbf{0})$  of Decoding Algorithm 1. By the translation property, the set of all points that map to any lattice point  $\mathbf{x}$  is  $\mathbf{R}_1(\mathbf{x}) = \mathbf{R}_1(\mathbf{0}) + \mathbf{x}$ . Lemma 1 shows that  $\mathbf{R}_1(\mathbf{0})$  contains all points of norm less than 2. Since spheres of radius 2 drawn around the points of  $H_{24}$  must touch, there must be points of norm 2

on the boundary of  $\mathbf{R}_1(\mathbf{0})$ . The number of points on the boundary of  $\mathbf{R}_1(\mathbf{0})$  with norm  $\|\mathbf{r}\|^2 = 2$  is the effective error coefficient  $N_{0,\text{eff}}$  of Decoding Algorithm 1. Lemma 2 shows that Decoding Algorithm 1 approximately doubles the effective error coefficient of  $H_{24}$ .

**[0113]** *Lemma 2:* The effective error coefficient of Decoding Algorithm 1 is  $N_{0,\text{eff}} = 98,256 + 97,152 = 195,408$ .

**[0114]** *Proof:* In addition to the 98,256 points in  $H_{24}$  of norm 8, there are  $759 \cdot 128 = 97,152$  points of norm 8 with eight coordinates of magnitude 1 and 16 of 0 that are in  $G_{24}$  but not in  $H_{24}$  (those with odd twos-coefficient parity). If the  $G_{24}$  decoder decodes  $\mathbf{r}$  to any of these points in the first step of Decoding Algorithm 1, the second step cannot yield  $\mathbf{x}'_0 = \mathbf{0}$ .

**[0115]** To decode the Leech lattice  $\Lambda_{24}$ , Decoding Algorithm 1 may be applied twice to the two cosets of  $H_{24}$  of which  $\Lambda_{24}$  is the union.

**[0116]** *Decoding Algorithm 2 ( $\Lambda_{24}$ ):* Given any real 24-tuple  $\mathbf{r}$ , apply Decoding Algorithm 1 to  $\mathbf{r}$  to find a point  $\mathbf{x}_0$  in  $H_{24}$ ; also, Decoding Algorithm 1 to  $\mathbf{r} - \mathbf{a}$  to find a point  $\mathbf{x}_1$  in  $H_{24}$  whose translate  $\mathbf{x}_1 + \mathbf{a}$  is in the coset  $H_{24} + \mathbf{a}$ . Compute the squared distances  $\|\mathbf{r} - \mathbf{x}_0\|^2$  and  $\|\mathbf{r} - \mathbf{x}_1 + \mathbf{a}\|^2$ , and choose  $\mathbf{x}_1$  or  $\mathbf{x}_1 + \mathbf{a}$  according to which distance is smaller.

**[0117]** The complexity of Decoding Algorithm 2 is not much more than twice that of Decoding Algorithm 1 since the complexity of the translations of  $\mathbf{r}$  and  $\mathbf{x}_1$  by  $\mathbf{a}$  and the computation and comparison of the two squared distances is small compared to the complexity of Golay decoding.

**[0118]** It can be shown that Decoding Algorithm 2 is a bounded-distance decoding algorithm that achieves the error-correction radius of  $\Lambda_{24}$  and increases the effective error coefficient by a factor of only about 1.5. Thus, the effective signal-to-noise ratio required by Decoding Algorithm 2 is only about 0.1 dB worse than that of maximum-likelihood decoding if the noise is Gaussian and the desired error rate is of the order of  $10^{-6}$ .

**[0119]** *Theorem 1:* Given a 24-tuple  $\mathbf{r}$ , if there is a point  $\mathbf{x}$  in  $\Lambda_{24}$  such that  $\|\mathbf{r} - \mathbf{x}\|^2 < 2$ , then Decoding Algorithm 2 decodes  $\mathbf{r}$  to  $\mathbf{x}$ .

**[0120]** *Proof:* If  $\mathbf{x}$  is in  $H_{24}$ , then by Lemma 1, Decoding Algorithm 1 applied to  $\mathbf{r}$  yields  $\mathbf{x}$ ; if  $\mathbf{x}$  is in  $H_{24} + \mathbf{a}$ , then  $\mathbf{x} - \mathbf{a}$  is in  $H_{24}$  and thus by Lemma 1, Decoding Algorithm 1 applied to  $\mathbf{r} - \mathbf{a}$  yields  $\mathbf{x} - \mathbf{a}$ . Since  $d_{\min}^2(\Lambda_{24}) = 8$ , there can be only one point  $\mathbf{x}$  in  $\Lambda_{24}$  such that  $\|\mathbf{r} - \mathbf{x}\|^2 < 2$ . If either of the two trial decodings finds such a point, it must be chosen as the closest point.

**[0121]** *Theorem 2:* The effective error coefficient of Decoding Algorithm 2 is  $N_{0,\text{eff}} = 196,560 + 97,152 = 293,712$ .

[0122] *Proof:* The effective error coefficient is the number of points on the boundary of the decision region  $\mathbf{R}_2(\mathbf{0})$  of norm 2, which is the same as the number of points  $\mathbf{x}$  in  $G_{24}$  or  $G_{24} + \mathbf{a}$  of norm 8 that are in  $\Lambda_{24}$  or cannot be modified to  $\mathbf{0}$  by a change of  $\pm 2$  in one coordinate. This includes the 196,560 points in  $\Lambda_{24}$  of norm 8 and also the 97,152 points of norm 8 that are in  $G_{24}$  but not  $H_{24}$  that were mentioned in the proof of Lemma 2. Any point in  $G_{24} + \mathbf{a} = G_{24} + (1/2)^{24}$  can be modified to a point in  $H_{24} + \mathbf{a}$  by a change of  $\pm 2$  in the first coordinate, so that there are no further points of this type.

### **Shaping Region vs. Shape Gain Tradeoffs**

[0123] In some embodiments, when selecting the boundary of a signal constellation used for data transmission, one tries to minimize the average energy of the constellation for a given number of points from a given packing. The reduction in the average energy per two dimensions due to the use of a region  $\mathcal{E}$  as the boundary instead of a hypercube, is called the shaping gain  $\gamma_s$  of  $\mathcal{E}$ . The price to be paid for shaping involves an increase in the factor constellation-expansion ratio ( $CER_s$ ), an increase in the factor peak-to-average-power ratio (PAR), and an increase in the addressing complexity. There exists a tradeoff between  $\gamma_s$  and  $CER_s$ , PAR, however as discussed herein, an  $N$ -dimensional shaping region may be selected having a structure that simultaneously optimizes the tradeoffs, as discussed below.

[0124] In one or more embodiments of the present disclosure, the integral of a function of the general form  $F(X_0^2 + \dots + X_{N-1}^2)$  over a final region  $\mathcal{A}_N$  is calculated as:

$$\begin{aligned} & \int_{\mathcal{A}_N} F(X_0^2 + \dots + X_{N-1}^2) dX_0 \dots dX_{N-1} \\ &= (\pi R_2^2)^n \sum_{k=0}^{[\beta]} (-1)^k C_n^k \frac{(\beta - k)^n}{(n - 1)!} \\ & * \int_0^1 F(R_2^2 [(\beta - k)\tau + k]) \tau^{n-1} d\tau. \end{aligned}$$

[0125] This integral is used to calculate the volume and the second moment of the  $\mathcal{A}_N$  region. The results, together with  $V(\mathcal{E}_2) = \pi R_2^2$  and  $E_p(\mathcal{E}_2) = R_2^2$ , where  $V(\mathcal{E}_2)$  is the volume of shaping region  $\mathcal{E}_2$  and  $R_2$  is the radius, can be used to compute  $\gamma_s$ ,  $CER_s$ , and PAR. FIG. 11 shows the corresponding tradeoff curves for different values of  $N$ . As  $N \rightarrow \infty$ , the induced

probability distribution along 2-D subspaces of the  $\mathcal{A}_N$  region tends to a truncated Gaussian distribution; a consequence of the optimality of these regions.

**[0126]** The expression  $\psi = \beta/n(n = N/2)$  can be used as the normalized parameter for the  $\mathcal{A}_N$  region. The complete notation for the region is  $\mathcal{A}_N(\psi)$ . For  $\psi = 1/n(\beta = 1, R_N = R_2)$ , the spherical region  $\ell_2(R_N)$  is obtained. This case corresponds to the final point on the tradeoff curves. For  $1/n < \varphi < 1(1 < \beta < n, R_2 < R_N < \sqrt{nR_2})$ , by increasing  $\psi$ , one moves along the curves towards their initial parts. Finally, for  $\psi = 1(\beta = n, R_N = \sqrt{nR_2})$ , one has  $\mathcal{A}_N = \{\ell_2(R_2)\}^n$ . This results in the starting point on the tradeoff curves. The two cases of  $0 < \psi < 1/n$  and  $\psi > 1$  results in the regions  $\mathcal{S}_N(\sqrt{n\psi}R_2)$  and  $(\mathcal{S}_2(R_2))^n$ , respectively.

**[0127]** Referring to FIG. 11, it can be seen that, in general, the initial parts of the optimum tradeoff curves have a steep slope. This means that an appreciable portion of the maximum shaping gain, corresponding to a spherical region, can be achieved with a small value of  $CER_s$ , PAR. Table I (below) contains a set of points from the optimum tradeoff curves. These are the points marked on the curves in FIG. 11.

$N$	$A$			$B$			$L$			$K$			$S$		
	$CER_s$	PAR	$\gamma_s$ dB	$CER_s$	PAR	$\gamma_s$ dB	$CER_s$	PAR	$\gamma_s$ dB	$CER_s$	PAR	$\gamma_s$ dB	$CER_s$	PAR	$\gamma_s$ dB
4	1.41	3.00	0.46	—	—	—	—	—	—	1.41	3.00	0.46	1.41	3.0	0.46
8	1.19	2.60	0.60	1.68	3.76	0.72	1.19	2.60	0.60	1.41	3.19	0.70	2.21	5.0	0.73
12	1.12	2.47	0.61	1.41	3.26	0.82	1.19	2.68	0.70	1.41	3.26	0.82	2.99	7.0	0.88
16	1.09	2.39	0.60	1.30	3.04	0.85	1.19	2.71	0.76	1.41	3.33	0.90	3.76	9.0	0.98
24	1.06	2.31	0.57	1.19	2.76	0.84	1.19	2.76	0.84	1.41	3.42	1.00	5.29	13.0	1.10
32	1.04	2.26	0.55	1.14	2.62	0.81	1.19	2.80	0.89	1.41	3.45	1.06	6.80	17.0	1.17
48	1.03	2.22	0.52	1.09	2.48	0.76	1.19	2.83	0.96	1.41	3.51	1.14	9.80	25.0	1.26
64	1.02	2.18	0.48	1.07	2.41	0.72	1.19	2.86	1.00	1.41	3.53	1.18	12.04	33.0	1.31
128	1.01	2.12	0.41	1.03	2.27	0.61	1.19	2.93	1.08	1.41	3.65	1.27	24.67	65.0	1.40
$\infty$	1.00	2.00	0.20	1.00	2.00	0.20	1.19	3.00	1.20	1.41	3.75	1.40	$\infty$	$\infty$	1.53

Table I – A Set of Points from the Optimum Tradeoff Curves of FIG. 11

**[0128]** The  $S$ -points correspond to a spherical region and achieve the maximum shaping gain in a given dimensionality. The  $K$ -points correspond to  $r_s = N/4$  ( $CER_s = (2)^{1/2} = 1.41$ ). They achieve almost all of the shaping gain of the  $S$ -points, but with a much lower value of  $CER_s$ , PAR. The  $L$ -points correspond to  $r_s = N/8$  ( $CER_s = (2)^{1/4} = 1.19$ ). They achieve a significant  $\gamma_s$  with a very low  $CER_s$ , PAR. The  $B$ -points correspond to  $r_s = 3$  ( $CER_s = (8)^{2/N}$ ). The  $A$ -points correspond to the addressing scheme based on the lattice  $D_n^*$ , and result in  $r_s = 1$  ( $CER_s = (2)^{2/N}$ ). For  $N = 4$ , this point corresponds to a spherical region.

**[0129]** From FIG. 11, it can be seen that for  $N$  around 12, the  $A$ -points with  $r_s = 1$  are located near the knee of the optimum tradeoff curves. For larger dimensionalities, specifically for  $N > 16$ , they are closer to the initial parts of the curves. This means that for  $N > 16$ , one bit of

redundancy per  $N$  dimensions is too small. A solution in a space of dimensionality  $N = n' \times N'$  ( $N'$  even) is to use the lattice  $D_n^*$ ,  $n = N'/2$  to shape the  $N' - D$  subspaces and then achieve another level of shaping on the  $n' = N/N'$ -fold Cartesian product of these subspaces. This is one example of the application of a multilevel shaping/addressing scheme.

**[0130]** More generally, consider an  $\mathcal{A}_N(\psi)$  region. This region has an  $\mathcal{A}_N(N\psi/N')$  region along each of its constituent  $N' - D$  ( $N'$  even) subspaces. The basic idea is that the  $\mathcal{A}_N(N\psi/N')$  subregions can be modified such that the complexity of the addressing in the  $N$ - $D$  space is decreased while the overall suboptimality is small. Specifically, in some of our schemes, 1) the  $\mathcal{A}_N(N\psi/N')$  region is replaced by the region  $\mathcal{A}_N(1/2)$ , and/or 2) this region is partitioned into a finite number of energy shells, and then the Cartesian product of the  $N' - D$  subspaces can be shaped.

**[0131]** Additional details regarding multi-dimensional signal constellations, signal modulation, and other methods / techniques compatible with embodiments of the present disclosure can be found in the following references, the entire contents of each of which is incorporated by reference herein for all purposes and appended hereto as Appendices A-D, respectively: G. R. Lang and F. M. Longstaff, "A Leech lattice modem," in *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 968-973, Aug. 1989, doi: 10.1109/49.29618 (Appendix A); European Patent Number EP1608081, issued May 6, 2009 and titled "Apparatus and Method for Space-Frequency Block Coding/Decoding in a Communication System" (Appendix B); Zamir, Ram, "Lattice Coding for Signals and Networks," Cambridge University Press, 2014 (Appendix C); and Hao, W. & Zhang, J.-Q & Song, H.-B (2014), A lattice based approach to the construction of multi-dimensional signal constellations, Tien Tzu Hsueh Pao/Acta Electronica Sinica, 42, 1672-1679. 10.3969/j.issn.0372-2112.2014.09.002 (Appendix D).

**[0132]** Additional details regarding lattice constructions and lattice encoding compatible with embodiments of the present disclosure can be found in Blake, Ian, "The Leech Lattice as a Code for the Gaussian Channel," *Information and Control* 19, 66-74 (1971); J. Choi, Y. Nam and N. Lee, "Spatial Lattice Modulation for MIMO Systems," in *IEEE Transactions on Signal Processing*, vol. 66, no. 12, pp. 3185-3198, 15 June 15, 2018, doi: 10.1109/TSP.2018.2827325; G. D. Forney, "Coset codes. I. Introduction and geometrical classification," in *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1123-1151, Sept. 1988, doi: 10.1109/18.21245; G.D. Forney, "Coset codes. II. Binary lattices and related codes," in *IEEE*

Transactions on Information Theory, vol. 34, no. 5, pp. 1152-1187, Sept. 1988, doi: 10.1109/18.21246; J. Conway and N. Sloane, "A fast encoding method for lattice codes and quantizers," in IEEE Transactions on Information Theory, vol. 29, no. 6, pp. 820-824, November 1983, doi: 10.1109/TIT.1983.1056761; U.S. Patent Number 9,692,456, issued June 27, 2017 and titled "Product Coded Modulation Scheme Based on E8 Lattice and Binary and Nonbinary Codes"; U.S. Patent Number 7,036,071, issued April 25, 2006 and titled "Practical Coding and Metric Calculation for the Lattice Interfered Channel"; G. Forney, R. Gallager, G. Lang, F. Longstaff and S. Qureshi, "Efficient Modulation for Band-Limited Channels," in IEEE Journal on Selected Areas in Communications, vol. 2, no. 5, pp. 632-647, September 1984, doi: 10.1109/JSAC.1984.1146101; A. K. Khandani and P. Kabal, "An efficient block-based addressing scheme for the nearly optimum shaping of multidimensional signal spaces," in IEEE Transactions on Information Theory, vol. 41, no. 6, pp. 2026-2031, Nov. 1995, doi: 10.1109/18.476330; S. Stern and R. F. H. Fischer, "Lattice-Reduction-Aided Precoding for Coded Modulation over Algebraic Signal Constellations," WSA 2016; 20th International ITG Workshop on Smart Antennas, 2016, pp. 1-8; and G. D. Forney and G. Ungerboeck, "Modulation and coding for linear Gaussian channels," in IEEE Transactions on Information Theory, vol. 44, no. 6, pp. 2384-2415, Oct. 1998, doi: 10.1109/18.720542, the entire contents of each of which is incorporated by reference herein for all purposes.

**[0133]** Additional details regarding lattice decoding algorithms compatible with embodiments of the present disclosure can be found in: D. J. Costello and G. D. Forney, "Channel coding: The road to channel capacity," in *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150-1177, June 2007, doi: 10.1109/JPROC.2007.895188; J. Conway and N. Sloane, "Fast quantizing and decoding and algorithms for lattice quantizers and codes," in IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 227-232, March 1982, doi: 10.1109/TIT.1982.1056484; and G. D. Forney, "A bounded-distance decoding algorithm for the Leech lattice, with generalizations," in IEEE Transactions on Information Theory, vol. 35, no. 4, pp. 906-909, July 1989, doi: 10.1109/18.32173, the entire contents of each of which is incorporated by reference herein for all purposes.

**[0134]** Additional details regarding boundary selection for signal constellations in data transmission compatible with embodiments of the present disclosure can be found in: A. K. Khandani and P. Kabal, "Shaping multidimensional signal spaces. I. Optimum shaping, shell mapping," in IEEE Transactions on Information Theory, vol. 39, no. 6, pp. 1799-1808, Nov.

1993, doi: 10.1109/18.265491; and A. K. Khandani and P. Kabal, "Shaping multidimensional signal spaces. II. Shell-addressed constellations," in IEEE Transactions on Information Theory, vol. 39, no. 6, pp. 1809-1819, Nov. 1993, doi: 10.1109/18.265493, the entire contents of each of which is incorporated by reference herein for all purposes.

**[0135]** Example applications of lattice constellation compatible with embodiments of the present disclosure can be found in: U.S. Patent Number 9,172,578, issued October 27, 2015 and titled "High Speed Transceiver Based on Embedded Leech Lattice Constellation"; U.S. Patent Number 8,989,283, issued May 24, 2015 and titled "High Speed Transceiver Based on Concatenates of a Leech Lattice with Binary and Nonbinary Codes"; G. Ungerboeck, "Channel coding with multilevel/phase signals," in IEEE Transactions on Information Theory, vol. 28, no. 1, pp. 55-67, January 1982, doi: 10.1109/TIT.1982.1056454; and U. Wachsmann, R. F. H. Fischer and J. B. Huber, "Multilevel codes: theoretical concepts and practical design rules," in IEEE Transactions on Information Theory, vol. 45, no. 5, pp. 1361-1391, July 1999, doi: 10.1109/18.771140, the entire contents of each of which is incorporated by reference herein for all purposes.

**[0136]** Implementations of the various techniques described herein may be implemented in digital electronic circuitry, or in computer hardware, firmware, software (executed or stored in hardware), or in combinations of them. Implementations may be implemented as a computer program product, i.e., a computer program tangibly embodied, e.g., in a machine-readable storage device (computer-readable medium, a non-transitory computer-readable storage medium, a tangible computer-readable storage medium, etc.), for processing by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers. A computer program, such as the computer program(s) described above, can be written in any form of programming language, including compiled or interpreted languages, and can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program can be deployed to be processed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

**[0137]** Method steps may be performed by one or more programmable processors executing a computer program to perform functions by operating on input data and generating output. Method steps also may be performed by, and an apparatus may be implemented as, special

purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

**[0138]** Processors suitable for the processing of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. Elements of a computer may include at least one processor for executing instructions and one or more memory devices for storing instructions and data. Generally, a computer also may include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. Information carriers suitable for embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory may be supplemented by, or incorporated in, special purpose logic circuitry.

**[0139]** To provide for interaction with a user, implementations may be implemented on a computer having a display device, e.g., a liquid crystal display (LCD or LED) monitor, a touchscreen display, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input.

**[0140]** Implementations may be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation, or any combination of such back-end, middleware, or front-end components. Components may be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network (LAN) and a wide area network (WAN), e.g., the Internet.



[0141] While certain features of the described implementations have been illustrated as described herein, many modifications, substitutions, changes and equivalents will now occur to those skilled in the art. It is, therefore, to be understood that the appended claims are intended to cover all such modifications and changes as fall within the scope of the implementations. It should be understood that they have been presented by way of example only, not limitation, and various changes in form and details may be made. Any portion of the apparatus and/or methods described herein may be combined in any combination, except mutually exclusive combinations. The implementations described herein can include various combinations and/or sub-combinations of the functions, components and/or features of the different implementations described.

### *Claims*

1. A non-transitory, processor-readable medium storing instructions that, when executed by a processor, cause the processor to:
  - receive a bit string;
  - identify a plurality of binary strings based on the bit string;
  - map each binary string from the plurality of binary strings to an element from a plurality of elements of a lattice-based signal constellation, without using a lookup table;
  - identify a plurality of real-valued points from the plurality of elements based on the mapping; and
  - cause transmission of a signal having a modulation based on the plurality of real-valued points.
2. The non-transitory, processor-readable medium of claim 1, wherein the plurality of real-valued points represents a plurality of in-phase/quadrature (I/Q) points.
3. The non-transitory, processor-readable medium of claim 1, wherein each real-valued point from the plurality of real-valued points is associated with a point within a fundamental region of a Voronoi cell of the lattice-based signal constellation.
4. The non-transitory, processor-readable medium of claim 1, wherein each real-valued point from the plurality of real-valued points is associated with a point within a shifted fundamental region of a Voronoi cell of the lattice-based signal constellation.
5. The non-transitory, processor-readable medium of claim 1, wherein the plurality of elements is associated with a quotient of a first lattice and a second lattice, the second lattice being a copy of the first lattice scaled by a positive integer.
6. The non-transitory, processor-readable medium of claim 1, wherein each binary string from the plurality of binary strings is a binary expansion of an integer of the quotient  $\mathbb{Z}^n/m\mathbb{Z}^n$ , where  $\mathbb{Z}^n$  is an  $n$ -dimensional vector of integers,  $n$  is a positive integer, and  $m$  is a positive integer.

7. The non-transitory, processor-readable medium of claim 1, further storing instructions that, when executed by the processor, cause the processor to at least one of upsample the plurality of real-valued points, filter the plurality of real-valued points, drive the plurality of real-valued points to carrier, or send the plurality of real-valued points through a digital-to-analog converter, prior to causing transmission of the signal.
8. A method, comprising:
  - receiving, at a processor, a bit string;
  - identifying, via the processor, a plurality of binary strings based on the bit string;
  - mapping, via the processor, each binary string from the plurality of binary strings to an element from a plurality of elements of a lattice-based signal constellation, without using a rectangular shaping region;
  - identifying, via the processor, real-valued points from the plurality of elements based on the mapping; and
  - causing transmission of a signal having a modulation based on the real-valued points.
9. The method of claim 8, wherein the mapping of each binary string from the plurality of binary strings to an element from the plurality of elements of the lattice-based signal constellation is not based on lattice coding.
10. The method of claim 8, wherein the mapping of each binary string from the plurality of binary strings to an element from the plurality of elements of the lattice-based signal constellation is not based on redundant lattice points.
11. The method of claim 8, wherein the real-valued points represent in-phase/quadrature (I/Q) points.
12. The method of claim 8, wherein the plurality of elements is associated with a quotient of a first lattice and a second lattice, the second lattice being a copy of the first lattice scaled by a positive integer.

13. The method of claim 8, wherein each real-valued point from the real-valued points is associated with a point within a fundamental region of a Voronoi cell of the lattice-based signal constellation.

14. The method of claim 8, wherein each real-valued point from the real-valued points is associated with a point within a shifted fundamental region of a Voronoi cell of the lattice-based signal constellation.

15. The method of claim 8, wherein each binary string from the plurality of binary strings is a binary expansion of an integer of the quotient  $\mathbb{Z}^n/m\mathbb{Z}^n$ , where  $\mathbb{Z}^n$  is an  $n$ -dimensional vector of integers,  $n$  is a positive integer, and  $m$  is a positive integer.

16. The method of claim 8, further comprising at least one of upsampling the real-valued points, filtering the real-valued points, driving the real-valued points to carrier, or sending the real-valued points through a digital-to-analog converter, prior to causing transmission of the signal.

17. A non-transitory, processor-readable medium storing instructions that, when executed by a processor, cause the processor to:

receive, at a processor and from a transmitter, a signal encoding a bit string;

identify, via the processor and based on the signal and a closest vector algorithm, a point of an  $n$ -dimensional lattice associated with the signal;

multiply, via the processor, the point of the  $n$ -dimensional lattice by an inverse of a basis of the  $n$ -dimensional lattice, to identify an element from a plurality of elements of an  $n$ -dimensional matrix  $\mathbb{Z}^n$  associated with the signal; and

recover bits based on the signal, via the processor, by reducing components of the signal modulo  $m$ , to produce an element of a quotient  $\mathbb{Z}^n/m\mathbb{Z}^n$ , where  $n$  is a positive integer, and  $m$  is a positive integer.

18. The non-transitory, processor-readable medium of claim 17, further comprising instructions that, when executed by the processor, cause the processor to perform at least one

of synchronization or equalization of the signal prior to identifying the point of an  $n$ -dimensional lattice associated with the signal.

19. The non-transitory, processor-readable medium of claim 17, wherein the point is a first point, the media further comprising instructions that, when executed by the processor, cause the processor to:

identify, via the processor, an ambiguity associated with the quotient  $\mathbb{Z}^n/m\mathbb{Z}^n$ ; and  
in response to identifying the ambiguity:

select, via the processor, a set of faces of a fundamental region of the  $n$ -dimensional lattice, and

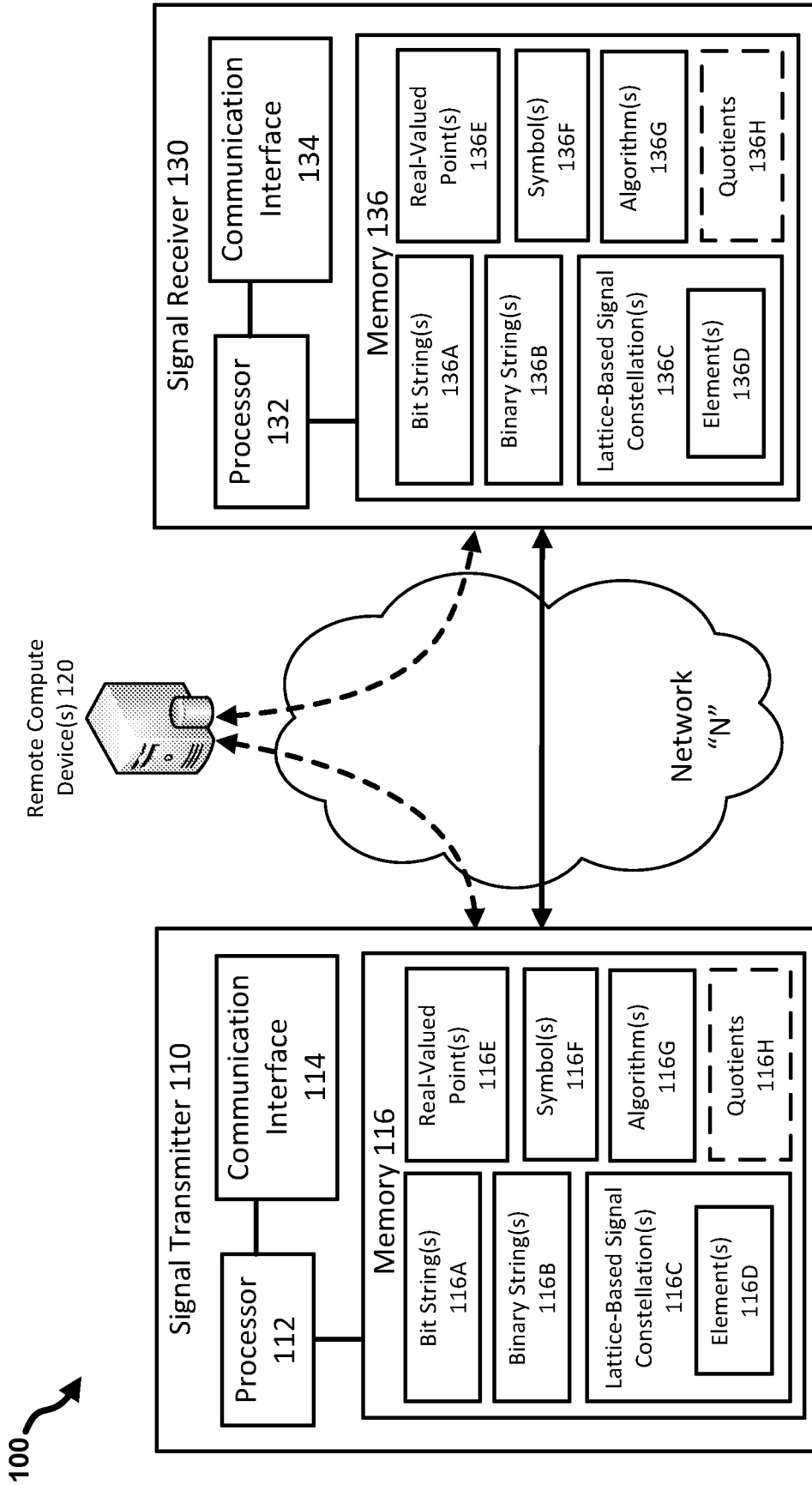
identify, via the processor and based on the set of faces of the fundamental region of the  $n$ -dimensional lattice, a second point of the  $n$ -dimensional lattice associated with the signal,

the instructions to recover the bits including instructions to recover the bits further based on the second point.

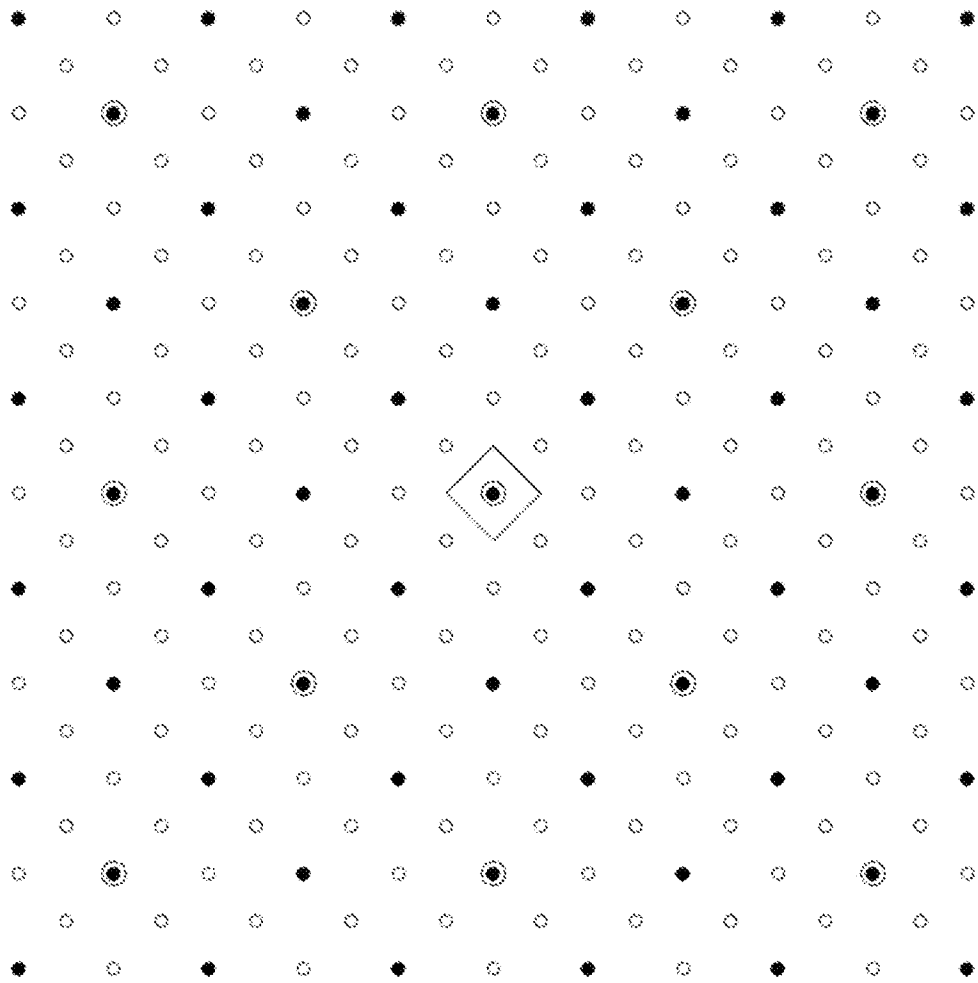
20. The non-transitory, processor-readable medium of claim 17, wherein the point is a first point, the media further comprising instructions that, when executed by the processor, cause the processor to:

identify, via the processor, an ambiguity associated with the quotient  $\mathbb{Z}^n/m\mathbb{Z}^n$ ; and  
in response to identifying the ambiguity:

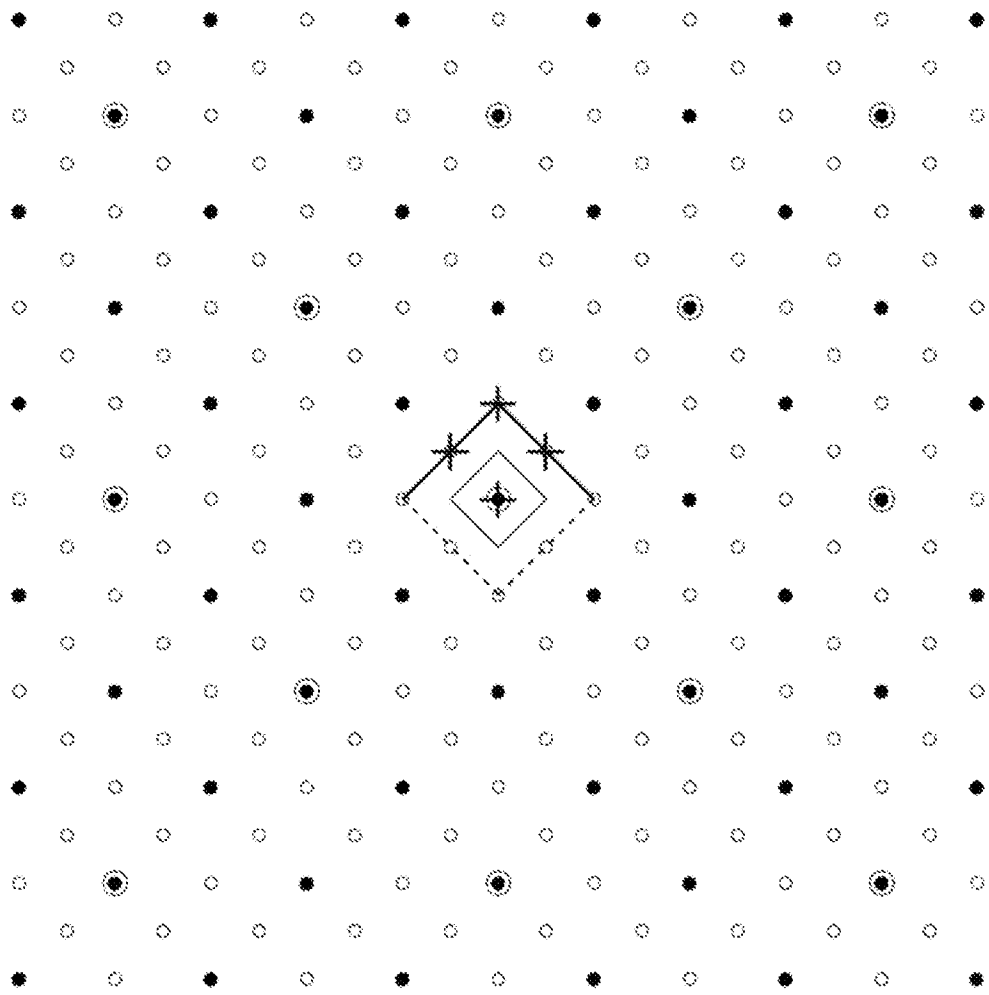
center a fundamental region of the  $n$ -dimensional lattice around a second point of the  $n$ -dimensional lattice that is distanced from an origin of the  $n$ -dimensional lattice, such that the plurality of elements of the  $n$ -dimensional matrix  $\mathbb{Z}^n$  do not lie on a surface of the fundamental region.



**FIG. 1**

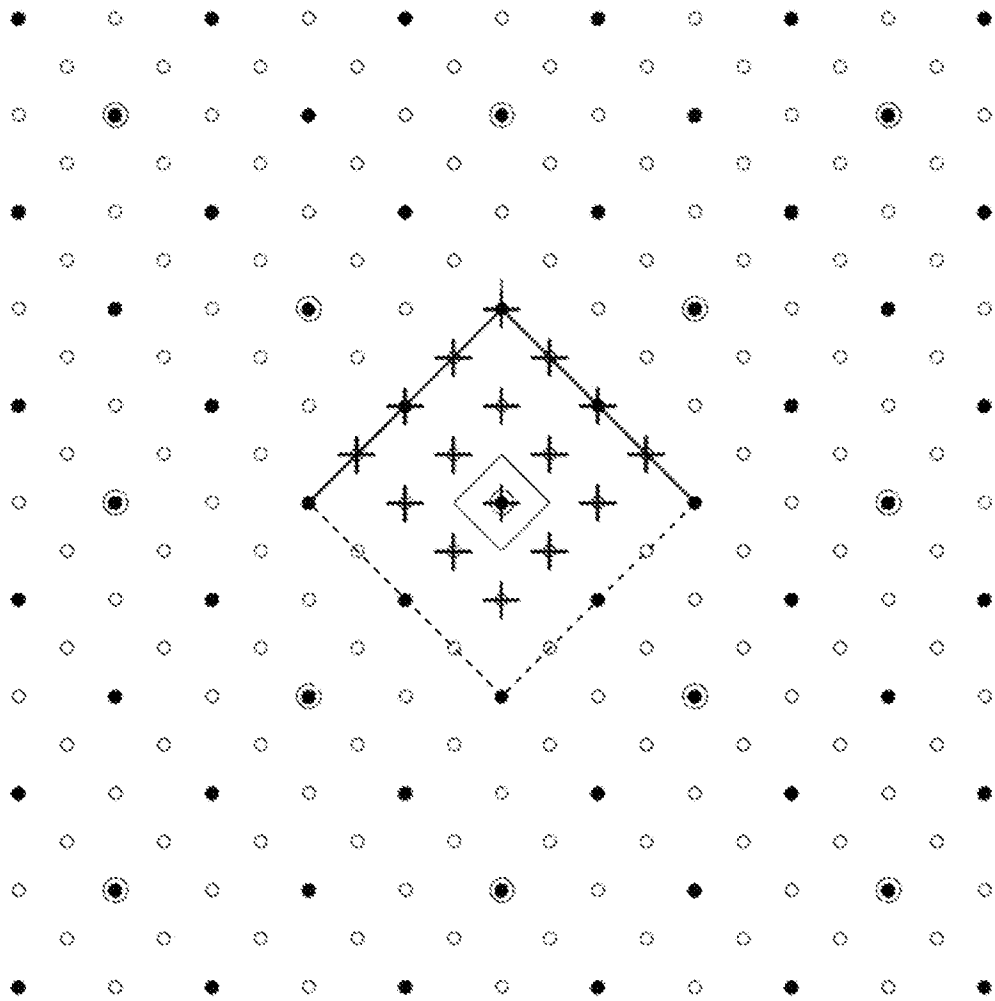


**FIG. 2**

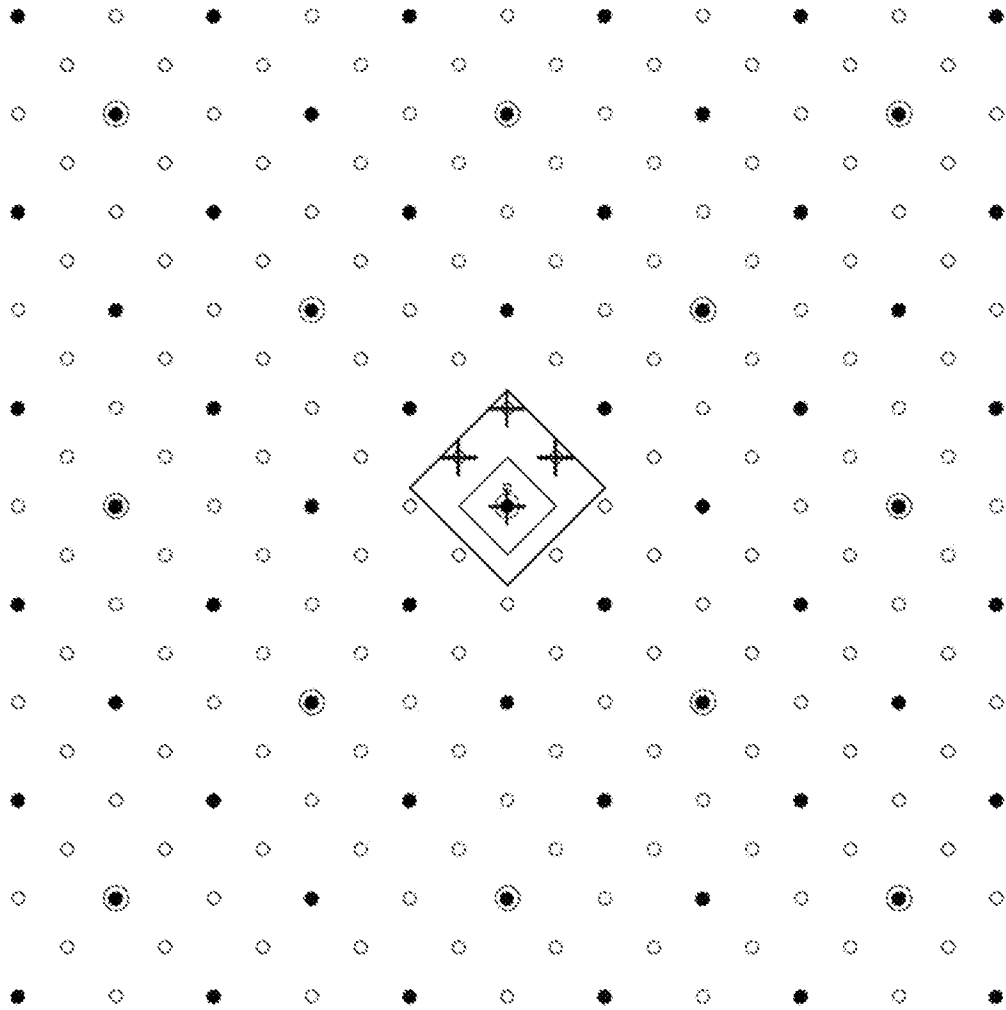


**FIG. 3**





**FIG. 4**



**FIG. 5**

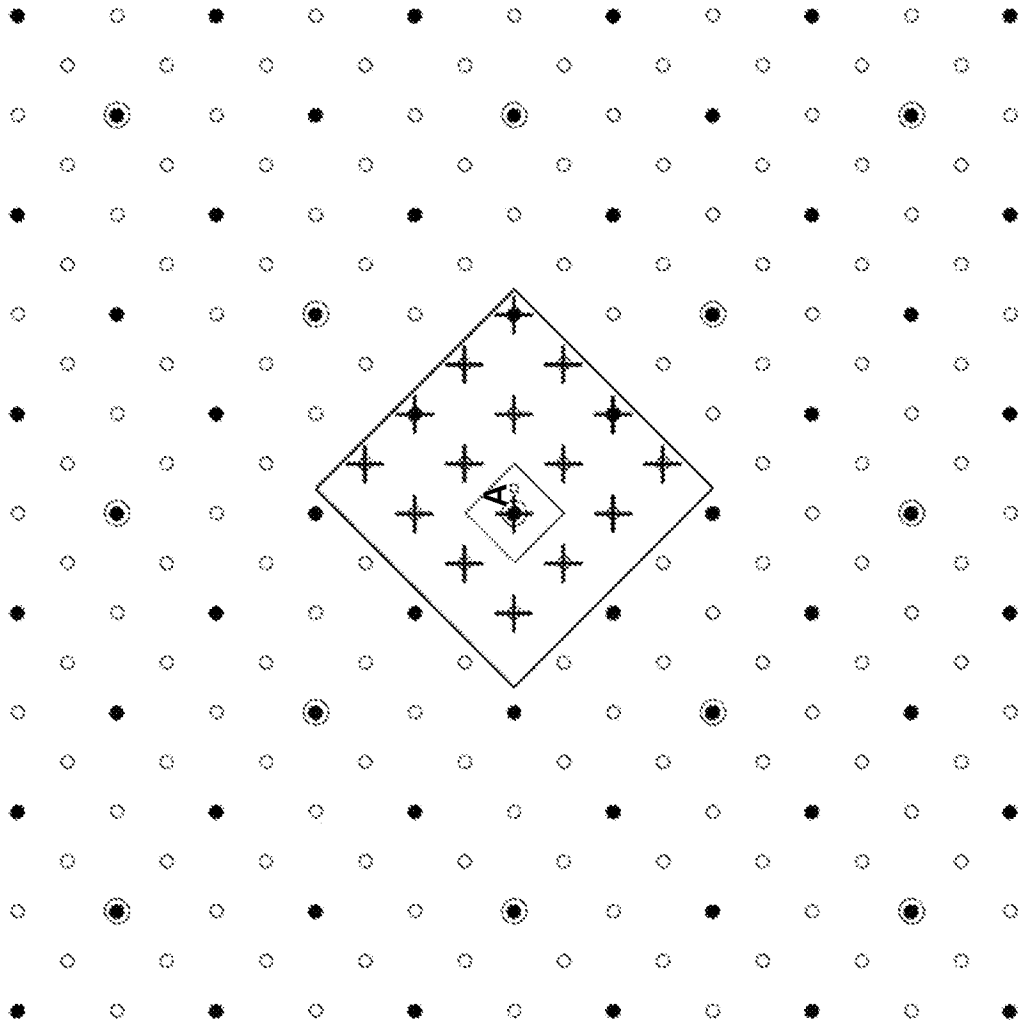
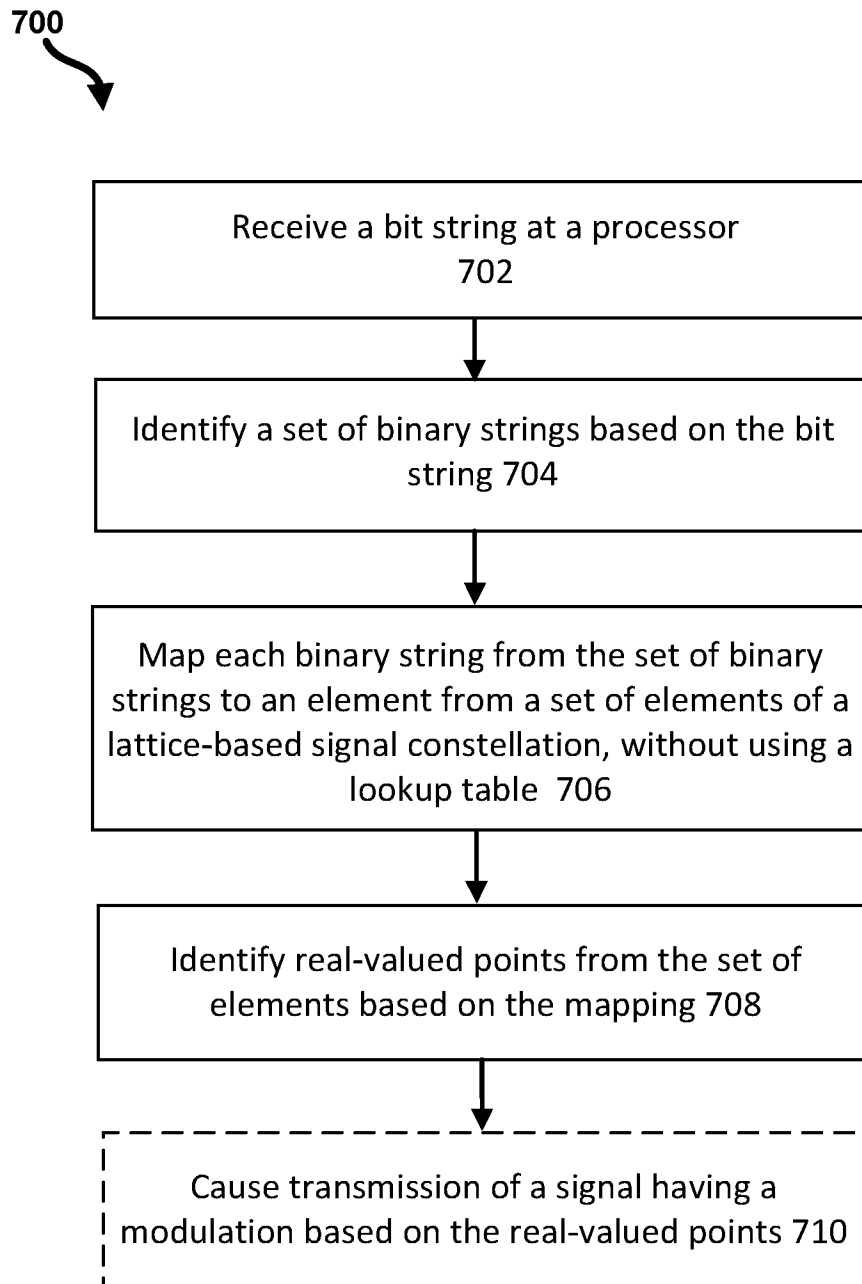
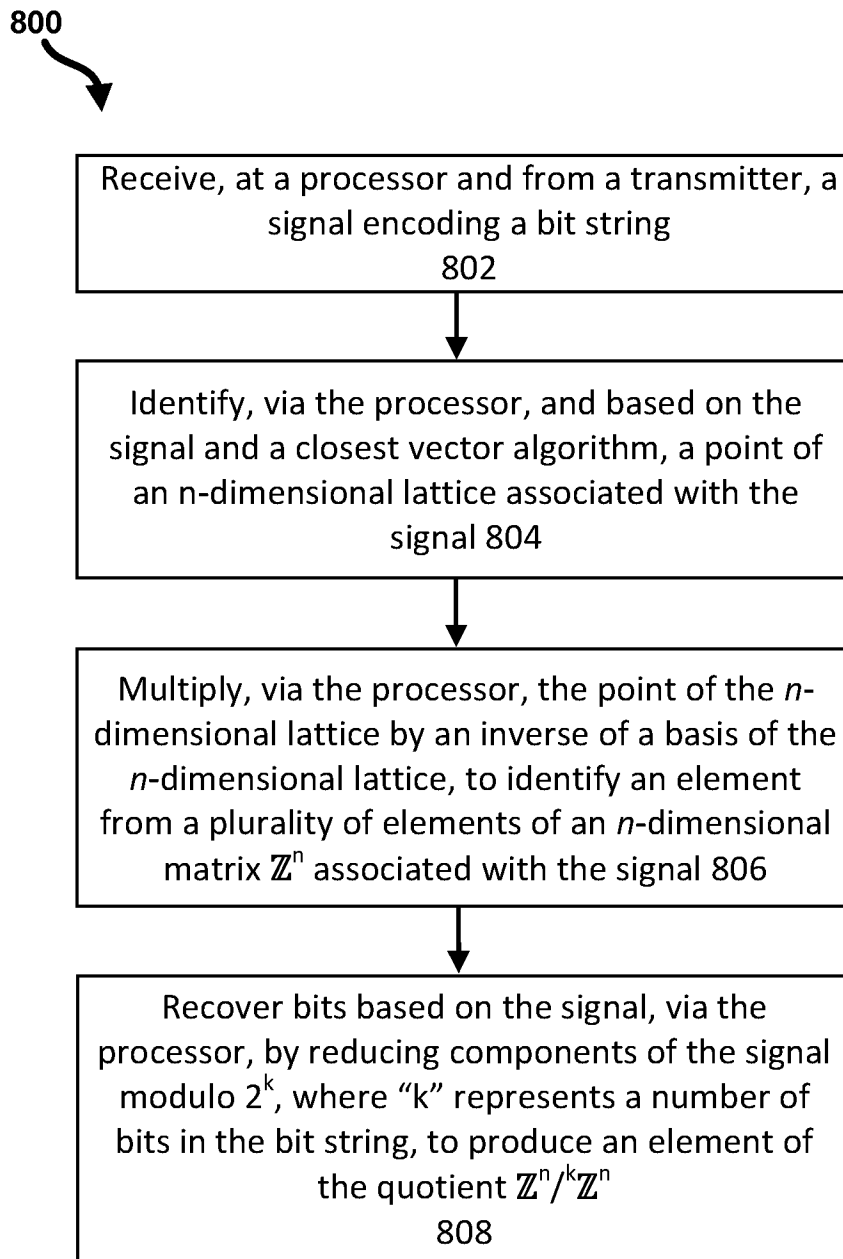


FIG. 6

**FIG. 7**

**FIG. 8**

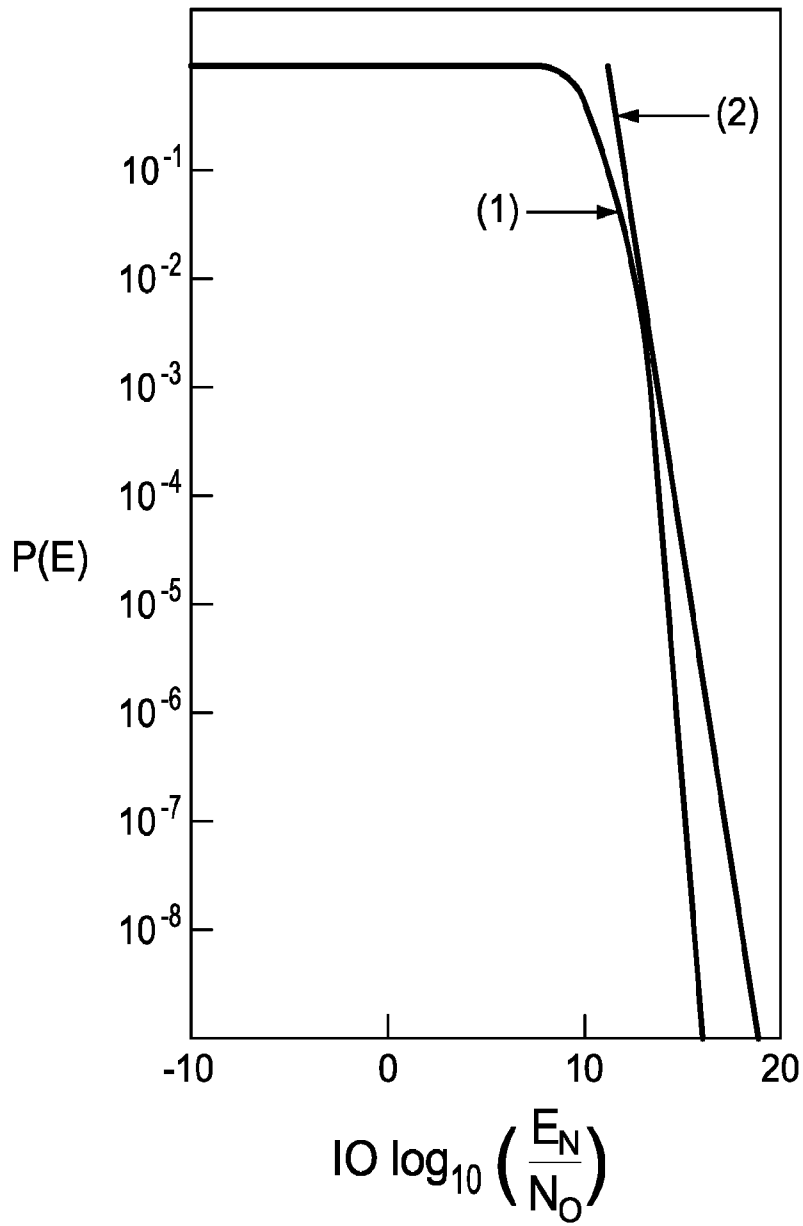


FIG. 9

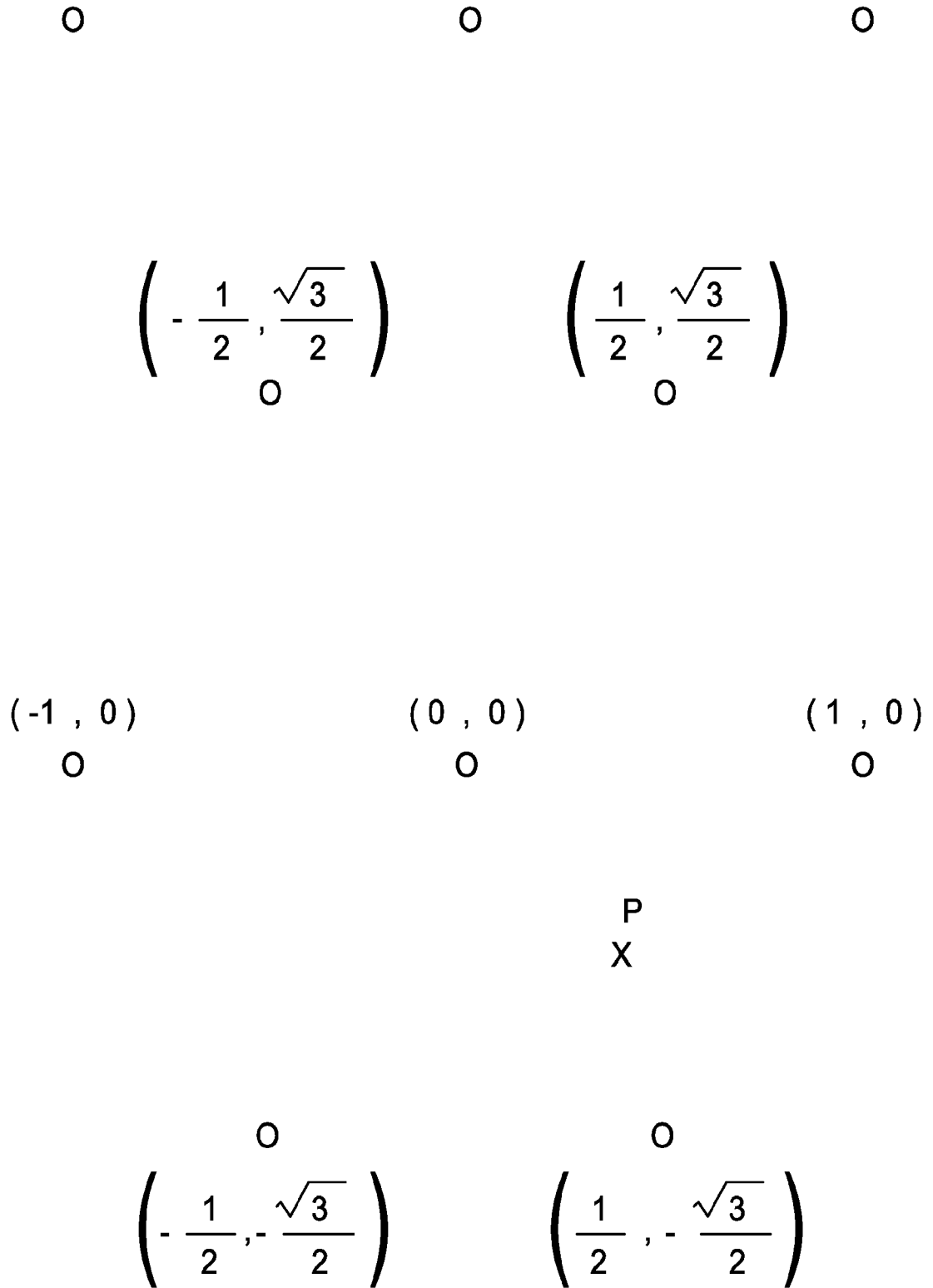


FIG. 10

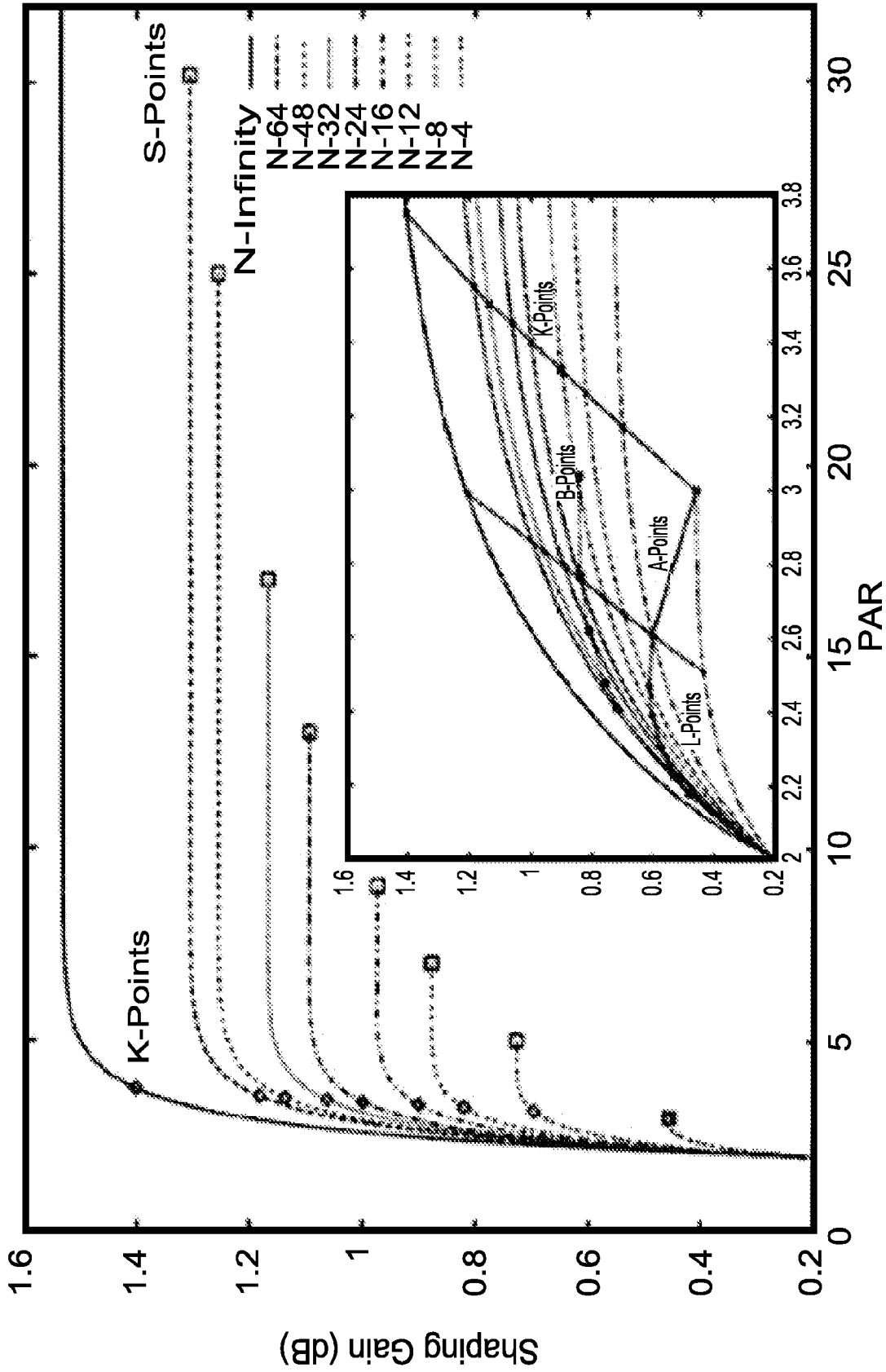


FIG. 11



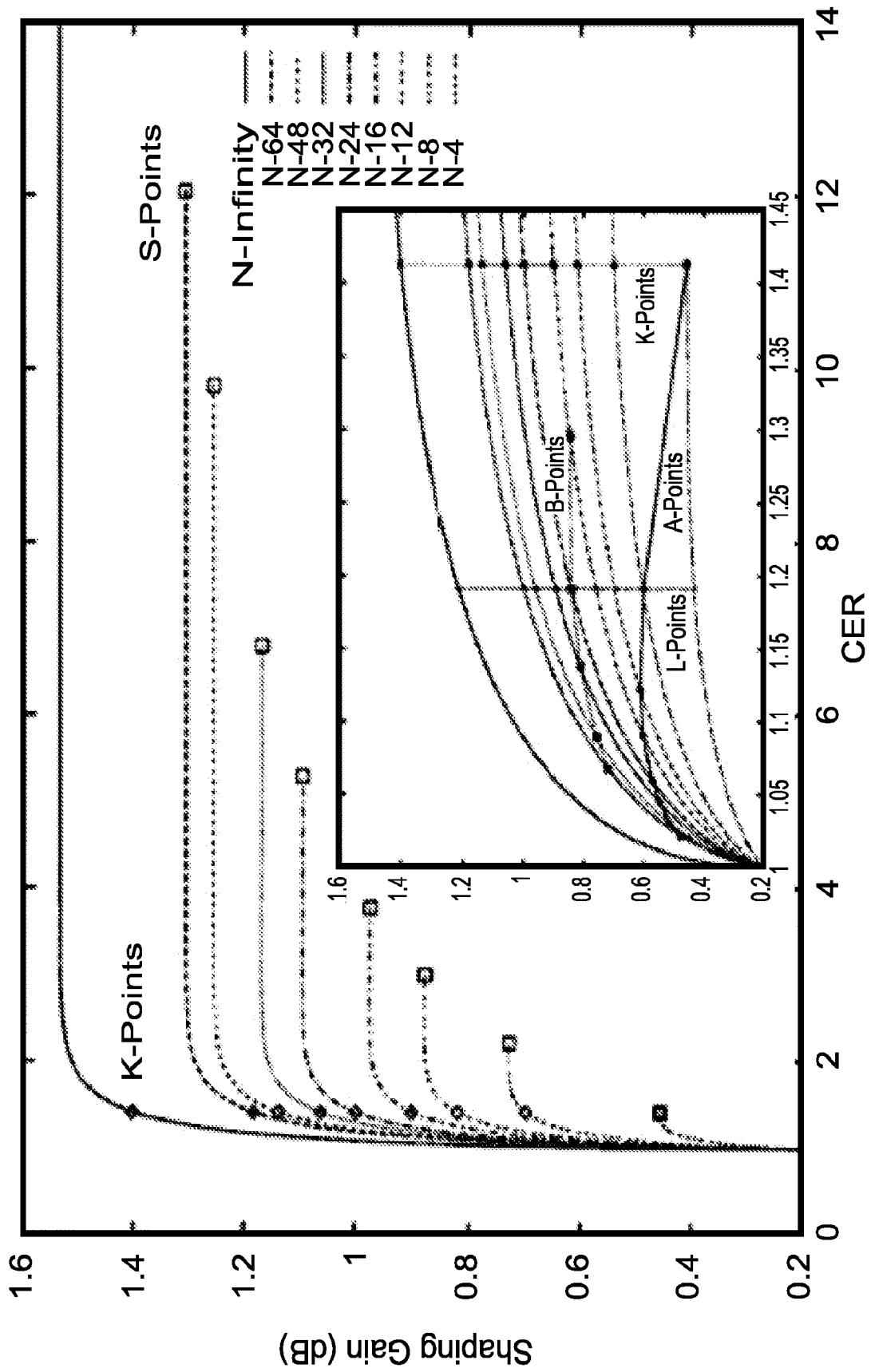


FIG. 11 (cont.)

# INTERNATIONAL SEARCH REPORT

International application No <b>PCT/US2023/062953</b>
--

**A. CLASSIFICATION OF SUBJECT MATTER**  
**INV. H04L27/34 H04L27/38**  
**ADD.**

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)  
**H04L**

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)  
**EPO-Internal**

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
<b>X</b>	<p><b>FORNEY G D ET AL: "MULTIDIMENSIONAL CONSTELLATIONS - PART I: INTRODUCTION, FIGURES OF MERIT, AND GENERALIZED CROSS CONSTELLATIONS", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE SERVICE CENTER, PISCATAWAY, US, vol. 7, no. 6, 1 August 1989 (1989-08-01), pages 877-892, XP000048613, ISSN: 0733-8716, DOI: 10.1109/49.29611 page 877 page 878 page 879 page 880, column 1 page 881 page 883 - page 884 page 889</b></p> <p style="text-align: center;">----- -/--</p>	<b>1-20</b>

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

\* Special categories of cited documents :

<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"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</p> <p>"X" 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</p> <p>"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</p> <p>"&amp;" document member of the same patent family</p>
---	---

Date of the actual completion of the international search <b>26 May 2023</b>	Date of mailing of the international search report <b>07/06/2023</b>
---	---

Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer  <b>Hanus, Pavol</b>
--	---

**INTERNATIONAL SEARCH REPORT**

International application No <b>PCT/US2023/062953</b>
--

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
<b>X</b>	<p><b>MIRANI ALI ET AL: "Low-Complexity Geometric Shaping",</b>  <b>JOURNAL OF LIGHTWAVE TECHNOLOGY, IEEE,</b>  <b>USA,</b>  <b>vol. 39, no. 2,</b>  <b>22 October 2020 (2020-10-22), pages</b>  <b>363-371, XP011828737,</b>  <b>ISSN: 0733-8724, DOI:</b>  <b>10.1109/JLT.2020.3033031</b>  <b>[retrieved on 2020-12-29]</b>  <b>figure 2b</b>  <b>page 365</b>  <b>page 368</b>  <b>page 369</b></p> <p align="center">-----</p>	<b>1-20</b>