US Patent & Trademark Office
Patent Public Search | Text View

United States Patent	12388526
Kind Code	B2
Date of Patent	August 12, 2025
Inventor(s)	Ke; Jianhong et al.

Method and system for compressing and employing pattern dependent look-up table

Abstract

The disclosed systems, and methods are directed to compressing a Pattern Dependent Look-up Table (PDLUT) including distortion correction values corresponding to data symbol sequences, including: i) symmetrically rearranging the data symbol sequences and the associated distortion correction values; ii) folding and amplitude reversing a second half of the rearranged data symbol sequences and the associated distortion correction values; iii) determining approximate distortion correction values for a first half of the rearranged data symbol sequences based on the distortion correction values associated with the first half of the rearranged data symbol sequences and the folded and amplitude reversed distortion correction values associated with the second half of the data symbol sequences; and iv) updating the PDLUT based on the approximate distortion values corresponding to the first half of the rearranged data symbol sequences.

Inventors: **Ke; Jianhong (Stittsville, CA), Li; Hongliang (Dollard-Des-Ormeaux, CA), Tang; Xuefeng (Kanata, CA), Li; Chuandong (Ottawa, CA)**

Applicant: **HUAWEI TECHNOLOGIES CO., LTD.** (Guangdong, CN)

Family ID: **1000008748991**

Assignee: **HUAWEI TECHNOLOGIES CO., LTD. (Shenzhen, CN)**

Appl. No.: **18/181819**

Filed: **March 10, 2023**

Prior Publication Data

Document Identifier	Publication Date
US 20240305375 A1	Sep. 12, 2024

Publication Classification

Int. Cl.: **H04B10/00** (20130101); **G06N20/00** (20190101); **H04B10/2507** (20130101); H04J14/00 (20060101)

U.S. Cl.:

CPC **H04B10/2507** (20130101); **G06N20/00** (20190101);

Field of Classification Search

CPC: H03F (2201/3233); H04B (1/62)

References Cited

U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
2004/0252783	12/2003	Kim	375/296	H04L 27/368
2013/0243125	12/2012	Matsubara	375/297	H03F 1/3247
2019/0149393	12/2018	Yu	375/296	H04B 10/0799

FOREIGN PATENT DOCUMENTS

Patent No.	Application Date	Country	CPC
3249819	12/2016	EP	N/A

Primary Examiner: Dobson; Daniel G

Attorney, Agent or Firm: BCF LLP

Background/Summary

CROSS-REFERENCE TO RELATED APPLICATIONS

(1) This is the first application filed for the instantly disclosed technology.

FIELD OF THE INVENTION

(2) The present disclosure generally relates to the field of communication systems and, in particular, to methods and systems for compressing and employing a Pattern Dependent Look-up Table (PDLUT).

BACKGROUND

(3) Generally, communication signals transmitted over wireless, wired, or optical networks are subject to various impairments such as non-linear distortions, especially at high baud speed (or high baud rate). The non-linear distortion includes pattern-dependent distortions with memory introduced by transmitter components such as digital-to-analog convertor (DAC), Radio Frequency (RF) amplifiers, and Mach-Zender modulators.

(4) The non-linear distortions are caused by combined non-linear effects from the components of the transmitter, which interacts with the frequency responses of the individual components. The non-linear distortions degrade an amplitude and a phase that are based on a symbol sequence in a signal received by a receiver.

(5) Such non-linear distortions have been recognized as one of the major impairments that limits the transceiver performance, especially for high-order modulation formats at high baud rates. The non-linear distortion typically exhibits a strong dependence on data patterns with a considerable

memory length.

(6) Generally, a lookup table (LUT) is generated by calculating distortion correction values corresponding to each symbol sequence in a present sequence of N symbols, and storing each calculated distortion correction value with an associated lookup table index. The distortion correction values are dependent upon the specific pattern of the symbol sequence, thus the resulting LUT is also referred to as a Pattern Dependent Lookup Table (PDLUT).

(7) Pattern-dependent lookup table (PDLUT) is an efficient approach for mitigating pattern dependent distortion. The distortion correction values in the PDLUT are obtained based on a calibration process by comparing the data received by the receiver and the transmitted data at the transmitter. Pattern-dependent distortion can be mitigated based on the distortion correction. The PDLUT has achieved a good performance of compensating the transmitter non-linearity at a lower modulation format (for example, 16QAM), however, with higher modulation (for example, 64QAM) the PDLUT size increases exponentially. This results in complex hardware implementation of the non-linearity compensation system.

(8) With this said, there is an interest in developing a compressed PDLUT that may reduce hardware complexity and improve power consumption of the non-linearity compensation system.

SUMMARY

(9) The embodiments of the present disclosure have been developed based on developers' appreciation of the limitations associated with the prior art: namely, an increasing Pattern-dependent lookup table (PDLUT) size for a higher modulation index.

(10) The developers have devised a system and a method of compressing a PDLUT including distortion correction values corresponding to data symbol sequences. More particularly, various embodiments of the present disclosure explore the symmetry and unification property of the PDLUT to compress the size of the PDLUT.

(11) In accordance with a first broad aspect of the present disclosure, there is provided a method of compressing a Pattern Dependent Look-up Table (PDLUT) including distortion correction values corresponding to data symbol sequences, the method comprising: symmetrically rearranging the data symbol sequences and the associated distortion correction values; folding and amplitude reversing a second half of the rearranged data symbol sequences and the associated distortion correction values; determining approximate distortion correction values for a first half of the rearranged data symbol sequences based on the distortion correction values associated with the first half of the rearranged data symbol sequences and the folded and amplitude reversed distortion correction values associated with the second half of the data symbol sequences; and updating the PDLUT based on the approximate distortion values corresponding to the first half of the rearranged data symbol sequences.

(12) In accordance with any embodiments of the present disclosure, the method further comprises: dividing the updated PDLUT into sub-PDLUTs; symmetrically rearranging the data symbol sequences and the approximate distortion correction values associated with the sub-PDLUTs; folding and amplitude reversing a second half of the rearranged data symbol sequences and the approximate distortion correction values associated with the sub-PDLUTs; determining updated approximate distortion correction values for a first half of the rearranged data symbol sequences associated with the sub-PDLUTs based on the approximate distortion correction values associated with the first half of the rearranged data symbol sequences and the folded and amplitude reversed approximate distortion correction values associated with the second half of the rearranged data symbol sequences associated with the sub-PDLUTs; and updating the PDLUT based on the updated approximate distortion values corresponding to the first half of the rearranged data symbol sequences associated with the sub-PDLUTs.

(13) In accordance with any embodiments of the present disclosure, the method further comprises: segregating the approximate distortion values into subsets; identifying similarities between the subsets and clustering the subsets into groups based on the similarities; selecting base subsets from

the groups, wherein the base subsets represent remaining subsets in the groups; generating a mapping table including mapping coefficients to map the base subsets to the remaining subsets in the groups; and updating the PDLUT with the base subsets and the mapping table

(14) In accordance with any embodiments of the present disclosure, identifying similarities and clustering the subsets is based on a principal component analysis (PCA) based artificial intelligence (AI) clustering technique.

(15) In accordance with any embodiments of the present disclosure, the mapping coefficients are generated based on a linear regression technique.

(16) In accordance with any embodiments of the present disclosure, the method further comprises providing the updated PDLUT to a receiver for pre-compensating under-correct symbols in the data symbol sequences.

(17) In accordance with any embodiments of the present disclosure, the PDLUT is a cascaded PDLUT, and the cascaded PDLUT includes a first PDLUT and a second PDLUT.

(18) In accordance with any embodiments of the present disclosure, determining approximate distortion correction values comprises one or more of: selecting one of the distortion correction values associated with the first half of the rearranged data symbol sequences or the folded and amplitude reversed distortion correction values associated with the second half of the data symbol sequences; averaging the distortion correction values associated with the first half of the rearranged data symbol sequences and the folded and amplitude reversed distortion correction values associated with the second half of the data symbol sequences; and adding a predetermined offset value to one of the distortion correction values associated with the first half of the rearranged data symbol sequences or the folded and amplitude reversed distortion correction values associated with the second half of the data symbol sequences.

(19) In accordance with a second broad aspect of the present disclosure, there is provided a method for pre-compensating an under-correct symbol, the method comprising: based on a first data symbol sequence, selecting a base subset from a plurality of base subsets included in a compressed Pattern Dependent Look-up Table (PDLUT); in the event that the first data symbol sequence is included in the base subset, selecting an approximate distortion correction value corresponding to the first data symbol sequence from the base subset; in the event that the first data symbol sequence is not included in the base subset, extracting mapping coefficients from a mapping table included in the compressed PDLUT, based on the mapping coefficients, mapping the base subset to generate a desired subset, and selecting an approximate distortion correction value corresponding to the first data symbol sequence from the desired subset; and applying the approximate distortion correction value to the under-correct symbol

(20) In accordance with any embodiments of the present disclosure, the method further comprises: in the event that the first data symbol sequence is not included in the desired subset, determining a data symbol sequence in the desired subset, which is opposite to the first data symbol sequence, selecting an approximate distortion correction value corresponding to the opposite data symbol sequence from the desired subset, and negating the approximate distortion correction value corresponding to the opposite data symbol sequence.

(21) In accordance with any embodiments of the present disclosure, the method further comprises: identifying a second data symbol sequence associated with the under-correct symbol in the PDLUT; in the event that the identified second data symbol sequence is included in the PDLUT, selecting an approximate distortion correction value corresponding to the identified second data symbol from the PDLUT; in the event that the identified second data symbol sequence is not included in the compressed PDLUT, determining a data symbol sequence in the compressed PDLUT which is opposite to the identified second data symbol sequence, selecting an approximate distortion correction value corresponding to the opposite data symbol sequence from the PDLUT, and negating the approximate distortion correction value corresponding to the opposite data symbol sequence; and applying the approximate distortion correction value to the under-correct symbol.

(22) In accordance with a third broad aspect of the present disclosure, there is provided a system for compressing a Pattern Dependent Look-up Table (PDLUT) including distortion correction values corresponding to data symbol sequences, the system comprising: a non-transitory memory element having instructions thereon; a processor coupled to the non-transitory memory element and which execute the instructions to cause the processor to: symmetrically rearrange the data symbol sequences and the associated distortion correction values; fold and amplitude reverse a second half of the rearranged data symbol sequences and the associated distortion correction values; determine approximate distortion correction values for a first half of the rearranged data symbol sequences based on the distortion correction values associated with the first half of the rearranged data symbol sequences and the folded and amplitude reversed distortion correction values associated with the second half of the data symbol sequences; and update the PDLUT based on the approximate distortion values corresponding to the first half of the rearranged data symbol sequences.

(23) In accordance with any embodiments of the present disclosure, the processor is further configured to: divide the updated PDLUT into sub-PDLUTs; symmetrically rearrange the data symbol sequences and the approximate distortion correction values associated with the sub-PDLUTs; fold and amplitude reverse a second half of the rearranged data symbol sequences and the associated approximate distortion correction values with the sub-PDLUTs; determine updated approximate distortion correction values for a first half of the rearranged data symbol sequences associated with the sub-PDLUTs based on the approximate distortion correction values associated with the first half of the rearranged data symbol sequences and the folded and amplitude reversed approximate distortion correction values associated with the second half of the rearranged data symbol sequences associated with the sub-PDLUTs; and update the PDLUT based on the updated approximate distortion values corresponding to the first half of the rearranged data symbol sequences associated with the sub-PDLUTs.

(24) In accordance with any embodiments of the present disclosure, the processor is further configured to: segregate the approximate distortion values into subsets; identify similarities between the subsets and clustering the subsets into groups based on the similarities; select base subsets from the groups, wherein the base subsets represent remaining subsets in the groups; generate a mapping table including mapping coefficients to map the base subsets to the remaining subsets in the groups; and update the PDLUT with the base subsets and the mapping table.

(25) In accordance with any embodiments of the present disclosure, the processor is further configured to provide the updated PDLUT to a receiver for pre-compensating under-correct symbols in the data symbol sequences.

(26) In accordance with a fourth broad aspect of the present disclosure, there is provided a system for pre-compensating an under-correct symbol, the method comprising: a non-transitory memory element having instructions thereon; a processor coupled to the non-transitory memory element and which execute the instructions to cause the processor to: based on a first data symbol sequence, select a base subset from a plurality of base subsets included in a compressed Pattern Dependent Look-up Table (PDLUT); in the event that the first data symbol sequence is included in the base subset, select an approximate distortion correction value corresponding to the first data symbol sequence from the base subset; in the event that the first data symbol sequence is not included in the base subset, extract mapping coefficients from a mapping table included in the compressed PDLUT, based on the mapping coefficients, map the base subset to generate a desired subset, and select an approximate distortion correction value corresponding to the first data symbol sequence from the desired subset; and apply the approximate distortion correction value to the under-correct symbol.

(27) In accordance with any embodiments of the present disclosure, the processor is further configured to: in the event that the first data symbol sequence is not included in the desired subset, determine a data symbol sequence in the desired subset, which is opposite to the first data symbol sequence, select an approximate distortion correction value corresponding to the opposite data

symbol sequence from the desired subset, and negate the approximate distortion correction value corresponding to the opposite data symbol sequence.

(28) In accordance with any embodiments of the present disclosure, the processor is further configured to: identify a second data symbol sequence associated with the under-correct symbol in the PDLUT; in the event that the identified second data symbol sequence is included in the PDLUT, select an approximate distortion correction value corresponding to the identified second data symbol from the PDLUT; in the event that the identified second data symbol sequence is not included in the compressed PDLUT, determine a data symbol sequence in the compressed PDLUT which is opposite to the identified second data symbol sequence, select an approximate distortion correction value corresponding to the opposite data symbol sequence from the PDLUT, negate the approximate distortion correction value corresponding to the opposite data symbol sequence; and apply the approximate distortion correction value to the under-correct symbol.

Description

BRIEF DESCRIPTION OF THE FIGURES

(1) The features and advantages of the present disclosure will become apparent from the following detailed description, taken in combination with the appended drawings, in which:

(2) FIG. **1** (Prior Art) depicts a block diagram of conventional transceiver;

(3) FIG. **2** (Prior Art) illustrates a conventional data symbol correction technique implemented over a transmitter digital signal processor (DSP);

(4) FIG. **3** (Prior Art) illustrates a conventional pattern-dependent lookup table (PDLUT) method and a cascaded-PDLUT method;

(5) FIG. **4** (Prior Art) illustrates a signal-to-noise ratio (SNR) based performance comparison of 16QAM based modulation format;

(6) FIG. **5** illustrates a transceiver configured to pre-compensate the data symbols based on compressed PDLUT, in accordance with various non-limiting embodiments of the present disclosure;

(7) FIG. **6** illustrates a representative rearranged PDLUT in accordance with various non-limiting embodiments;

(8) FIG. **7** illustrates a representation of the distortion correction values plotted against the index, in accordance with various embodiments of the present disclosure;

(9) FIG. **8** illustrates a representation of a folded PDLUT, in accordance with various non-limiting embodiments of the present disclosure;

(10) FIGS. **9-12** illustrate sub-PDLUTs, in accordance with various non-limiting embodiments of the present disclosure;

(11) FIG. **13** illustrates representations of the distortion correction values associated with the sub-PDLUTs respectively plotted against the corresponding indexes, in accordance with various embodiments of the present disclosure;

(12) FIG. **14** illustrates representations of the folded sub-PDLUTs, in accordance with various non-limiting embodiments of the present disclosure;

(13) FIG. **15** illustrates a representation of the approximate distortion correction values corresponding to the simplified PDLUT, in accordance with various non-limiting embodiments;

(14) FIG. **16** illustrates compressed features corresponding to the 16 subsets, in accordance with various non-liming embodiments of the present disclosure;

(15) FIG. **17** illustrates classified groups, in accordance with various non-limiting embodiments of the present disclosure;

(16) FIG. **18** illustrates a representation of the selected base subsets from the classified groups respectively, in accordance with various non-limiting embodiments of the present disclosure;

(17) FIG. **19** illustrates a mapping table generated to map the subsets in a group to the corresponding base subset, in accordance with various non-limiting embodiments of the present disclosure;
(18) FIG. **20** illustrates a representative rearranged PDLUT (with memory length 2) in accordance with various non-limiting embodiments;
(19) FIG. **21** illustrates a representative rearranged PDLUT (with memory length 3), in accordance with various non-limiting embodiments;
(20) FIG. **22** illustrates a representation of the distortion correction values associated with the rearranged PDLUT (with memory length 2) plotted against the index, in accordance with various embodiments of the present disclosure;
(21) FIG. **23** illustrates a representation of the distortion correction values associated with the rearranged PDLUT (with memory length 3) plotted against the index, in accordance with various embodiments of the present disclosure;
(22) FIGS. **24**-**25** illustrate representations of folded PDLUTs corresponding to the rearranged PDLUT (with memory length 2) and the rearranged PDLUT (with memory length 3) respectively, in accordance with various non-limiting embodiments of the present disclosure;
(23) FIG. **26** illustrates a representation of the selected base subsets 2, 4, 15, 14, and 25 from the five groups respectively, in accordance with various non-limiting embodiments of the present disclosure;
(24) FIG. **27** illustrates a mapping table generated to map the subsets in a group to the corresponding base subset, in accordance with various non-limiting embodiments of the present disclosure;
(25) FIG. **28** illustrates a first implementation of a symbol corrector, where the compressed PDLUT is generated based on the symmetry and unification of the PDLUT, in accordance with various embodiments of the present disclosure;
(26) FIG. **29** illustrates a second implementation of the symbol corrector, where the compressed PDLUT is generated based on the symmetry of the PDLUT, in accordance with various embodiments of the present disclosure;
(27) FIG. **30** illustrates a performance comparison table, in accordance with various non-limiting embodiments of the present disclosure;
(28) FIG. **31** illustrates a third implementation of the symbol corrector where the compressed PDLUT is generated based cascaded PDLUT using the symmetry and unification of the PDLUT, in accordance with various embodiments of the present disclosure;
(29) FIG. **32** illustrates a performance comparison table corresponding to the cascaded PDLUT, in accordance with various non-limiting embodiments of the present disclosure;
(30) FIG. **33** depicts a high-level block diagram of representative components for the compressed PDLUT generator, in accordance with various embodiments of the present disclosure; and
(31) FIG. **34** illustrates a process representing a method of compressing a Pattern Dependent Look-up Table (PDLUT) including distortion correction values corresponding to data symbol sequences, in accordance with various embodiments of the present disclosure; and
(32) FIG. **35** illustrates a process representing a method for pre-compensating an under-correct symbol, in accordance with various embodiments of the present disclosure.
(33) It is to be understood that throughout the appended drawings and corresponding descriptions, like features are identified by like reference characters. Furthermore, it is also to be understood that the drawings and ensuing descriptions are intended for illustrative purposes only and that such disclosures are not intended to limit the scope of the claims.

DETAILED DESCRIPTION

(34) The instant disclosure is directed to addressing at least some of the deficiencies of the conventional techniques used to compress and employ Pattern Dependent Lookup Table (PDLUT) to compensate non-linearity in a communication system. In particular, the instant disclosure

describes systems and methods to efficiently compressing and employing the PDLUT in the communication system.

(35) Unless otherwise defined or indicated by context, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which the described embodiments appertain to.

(36) In the context of the present specification, “controller” is any computer hardware that is capable of running software appropriate to the relevant task at hand. In the context of the present specification, in general the term “client device” is associated with a user of the client device. Thus, some (non-limiting) examples of client devices include personal computers (desktops, laptops, netbooks, etc.), smartphones, and tablets, as well as network equipment such as routers, switches, and gateways. It should be noted that a device acting as a client device in the present context is not precluded from acting as a server to other client devices. The use of the expression “a client device” does not preclude multiple client devices being used in receiving/sending, carrying out or causing to be carried out any task or request, or the consequences of any task or request, or steps of any method described herein.

(37) In the context of the present specification, unless provided expressly otherwise, the words “first”, “second”, “third”, etc. have been used as adjectives only for the purpose of allowing for distinction between the nouns that they modify from one another, and not for the purpose of describing any particular relationship between those nouns. Thus, for example, it should be understood that, the use of the terms “first processor” and “third processor” is not intended to imply any particular order, type, chronology, hierarchy or ranking (for example) of/between the server, nor is their use (by itself) intended to imply that any “second processor” must necessarily exist in any given situation. Further, as is discussed herein in other contexts, reference to a “first” element and a “second” element does not preclude the two elements from being the same actual real-world element. Thus, for example, in some instances, a “first” processor and a “second” processor may be the same software and/or hardware, in other cases they may be different software and/or hardware.

(38) It will be understood that when an element is referred to as being “connected” or “coupled” to another element, it can be directly or indirectly connected or coupled to the other element or intervening elements that may be present. In contrast, when an element is referred to as being “directly connected” or “directly coupled” to another element, there are no intervening elements present. Other words used to describe the relationship between elements should be interpreted in a like fashion (e.g., “between” versus “directly between,” “adjacent” versus “directly adjacent,” etc.).

(39) In the context of the present specification, when an element is referred to as being “associated with” another element, in certain embodiments, the two elements can be directly or indirectly linked, related, connected, coupled, the second element employs the first element, or the like without limiting the scope of present disclosure.

(40) The terminology used herein is only intended to describe particular representative embodiments and is not intended to be limiting of the present technology. As used herein, the singular forms “a,” “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises” and/or “comprising”, when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

(41) Implementations of the present technology each have at least one of the above-mentioned objects and/or aspects, but do not necessarily have all of them. It should be understood that some aspects of the present technology that have resulted from attempting to attain the above-mentioned object may not satisfy this object and/or may satisfy other objects not specifically recited herein.

(42) The examples and conditional language recited herein are principally intended to aid the reader in understanding the principles of the present technology and not to limit its scope to such specifically recited examples and conditions. It will be appreciated that those skilled in the art may

devise various arrangements which, although not explicitly described or shown herein, nonetheless embody the principles of the present technology and are included within its spirit and scope.

(43) Furthermore, as an aid to understanding, the following description may describe relatively simplified implementations of the present technology. As persons skilled in the art would understand, various implementations of the present technology may be of a greater complexity.

(44) In some cases, what are believed to be helpful examples of modifications to the present technology may also be set forth. This is done merely as an aid to understanding, and, again, not to define the scope or set forth the bounds of the present technology. These modifications are not an exhaustive list, and a person skilled in the art may make other modifications while nonetheless remaining within the scope of the present technology. Further, where no examples of modifications have been set forth, it should not be interpreted that no modifications are possible and/or that what is described is the sole manner of implementing that element of the present technology.

(45) Moreover, all statements herein reciting principles, aspects, and implementations of the present technology, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof, whether they are currently known or developed in the future. Thus, for example, it will be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative circuitry embodying the principles of the present technology. Similarly, it will be appreciated that any flowcharts, flow diagrams, state transition diagrams, pseudo-code, and the like represent various processes which may be substantially represented in computer-readable media and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

(46) The functions of the various elements shown in the figures, including any functional block labeled as a “processor” or a “processing unit”, may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. In some embodiments of the present technology, the processor may be a general-purpose processor, such as a central processing unit (CPU) or a processor dedicated to a specific purpose, such as a graphics processing unit (GPU). Moreover, explicit use of the term “processor” or “controller” should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital signal processor (DSP) hardware, network processor, application specific integrated circuit (ASIC), field programmable gate array (FPGA), read-only memory (ROM) for storing software, random access memory (RAM), and non-volatile storage. Other hardware, conventional and/or custom, may also be included.

(47) Software modules, modules, or units which are implied to be software, may be represented herein as any combination of flowchart elements or other elements indicating performance of process steps and/or textual description. Such modules may be executed by hardware that is expressly or implicitly shown.

(48) As used herein, the term “about” or “approximately” refers to a +/−15% variation from the nominal value. It is to be understood that such a variation is always included in a given value provided herein, whether or not it is specifically referred to.

(49) With these fundamentals in place, the instant disclosure is directed to address at least some of the deficiencies of the current technology. In particular, the instant disclosure describes systems and methods to efficiently compressing and employing the PDLUT in the communication system.

(50) Referring now to the drawings, FIG. **1** (Prior Art) depicts a block diagram of conventional transceiver **100**. The transceiver **100** includes a transmitter **102** and a receiver **106**. The transceiver **100** may be included in any type of communication system such as, for example, wired communication systems, wireless communication systems, and optical communication systems.

(51) The transmitter **102** is a coherent transmitter and includes 4 independent data paths. Each data path is associated with the xi, xq, yi and yq data. The data paths propagate through the transmitter

102 before being modulated on laser output. The transmitter **102** includes a transmitter digital signal processor (DSP) **104** configured to process the data symbols and transmit over the communication channel. The receiver **106** includes a receiver DSP **108** configured to process and recover the transmitted data symbols.

(52) It is contemplated that the conventional transceiver **100** includes components such as digital-to-analog convertor (DAC), Radio Frequency (RF) amplifiers, and Mach-Zender modulators, analog-to digital convertors (ADC) that may contribute to the non-linear distortions to the symbols propagating on different data paths. Generally, the non-linear distortions for different data paths are calibrated and compensated separately. For the purpose of simplicity, the xi, path has been taken as an example.

(53) The conventional transceiver **100** further includes a look up table (LUT) generator **110** configured to generate a LUT for pre-compensation of the data symbols to mitigate or reduce the effect of non-linear distortions. The LUT table may include distortion correction values corresponding to each data symbol. Typically, the distortion correction values are dependent upon the specific pattern of the symbol sequence, thus the resulting LUT is also referred to as a Pattern Dependent Lookup Table (PDLUT).

(54) The LUT generator **110** generates the PDLUT by comparing the ideal transmitted symbols Tx_xi with the received samples Rx_xi after linear compensation by the receiver DSP **108**. The PDLUT contains the required amplitude correction ΔS for the center symbol within each possible symbol sequence Tx_xi (k−n: k+n) of length m=2n+1. By way of example, if the symbol sequence length is 5, ΔS corresponds to the third symbol in the symbol sequence.

(55) A sliding window is used to identify the 2n+1 ideal symbol sequence to form the address of the table. The amplitude correction for the center symbol ΔS(k) is the difference between the symbol value Tx_xi (k) and the actual sample value Rx_xi(k) (i.e., ΔS(k)=Tx_xi (k)−Rx_xi (k)). As the window moves forward, each time a specific symbol sequence appears, the amplitude correction is accumulated at the corresponding PDLUT entry. A PDLUT counter (C) is used to track the number of updates for each specific entry. The average correction is saved for the final PDLUT.

(56) Once the PDLUT is generated, the data symbol values may be corrected based on the distortion correction values. FIG. **2** (Prior Art) illustrates a conventional data symbol correction technique **200** implemented over the transmitter DSP **104**. The transmitter **102** includes a data symbol corrector **202** configured to select the appropriate distortion correction values from the PDLUT and apply the distortion correction to the data symbols.

(57) More particularly, the transmitter DSP **104** relies on the same sliding window as used by the LUT generator **110** to identify the 2n+1 ideal symbol sequence to get the address (ii). The distortion correction Δ(ii) in the specific address is applied to the center symbol of the 2n+1 ideal symbol sequence Tx_xi (k−n: k+n).

(58) Even though, PDLUT has been illustrated a good performance of compensating the transmitter non-linearity, the PDLUT size increases exponentially with modulation formats. With a memory length of m, PDLUT size of 16QAM is 4.sup.m, while for 64QAM, the size is increased to 8.sup.m. PDLUT is quite large to be implemented for 64QAM and higher modulation format, thereby increases the hardware complexity and the associated power consumption.

(59) Conventionally, to reduce a size of the PDLUT, a cascaded-PDLUT method is employed. In the cascaded-PDLUT method two small LUTs replace one large LUT. The under-correct symbol is not in the middle, but at the left (CL) or right (CR) end. FIG. **3** (Prior Art) illustrates a conventional PDLUT method **302** and a cascaded-PDLUT method **304**. It is to be noted that a memory length of the conventional PDLUT may be odd or even. In the event that the memory length is odd, the symbol in the center is treated as under-correct symbol that needs correction, In the event that the memory length is even, the under-correct symbol could be in the center left (CL) or center right (CR). The performance of ‘CR’ and ‘CL’ could be different because of the non-linear asymmetry,

and the one with a better performance is selected. In the cascaded-PDLUT method **304**, the under-correct symbol could be at the extreme right or at the extreme left of the data symbol sequence.

(60) FIG. **4** (Prior Art) illustrates a signal-to-noise ratio (SNR) based performance comparison of 16QAM based modulation format. Conventional PDLUT with memory of 4 (also referred to as PDLUT4) and under-correct symbol on the central left (CL) shows a good performance with a considerable complexity. Cascaded PDLUT (3, 3) (also referred to as PDLUT 3+3) has the same performance as that of PDLUT4, but the LUT size is reduced to half. Therefore, considering the trade-off between LUT size and performance, PDLUT3+3 is preferred for 16QAM. Even though, the longer memory length (for example, with length 4, 4) illustrates a better performance, however, the hardware complexity increases with the increase in the size of PDLUT. For example, for 64QAM, the PDLUT size of PDLUT3+3 is increased from 128 to 1024, which is still too large to be implemented.

(61) With this said, there is an interest in developing systems and methods for developing low complexity based PDLUT.

(62) FIG. **5** illustrates a transceiver **500** configured to pre-compensate the data symbols based on compressed PDLUT, in accordance with various non-limiting embodiments of the present disclosure. As shown, the transceiver **500** may include a symbol generator **502**, a symbol corrector **504**, a transmitter DSP **506** on the transmitter side **501** and may include a receiver DSP **508** and a post processor **510** on the receiver side **507**. In addition, the transceiver **500** may include a compressed PDLUT generator **512**. It is to be noted that a location of the compressed PDLUT generator **512** should not limit the scope of present disclosure. In some embodiments, the compressed PDLUT generator may be located in close vicinity either to the transmitter side **501** or receiver side **507**. In other embodiments, the compressed PDLUT generator may be located at a distant location from the transmitter side **501** and the receiver side **507**. Also, the communication medium between the transmitter side **501**, the receiver side **507**, and the compressed PDLUT generator **512** may any suitable medium such as, for example, a wired medium, a wireless medium, optical medium or the like.

(63) The compressed PDLUT generator **512** may generate the initial PDLUT in a conventional manner as previously noted. The compressed PDLUT generator **512** may explore the symmetry property and unification property of the initial PDLUT for compression purposes. Even though, the initial PDLUT may be based on any suitable modulation format for example, PAM4, PAM8, 16QAM, 16QAMCS, 64QAM, 64QAMCS or higher modulation format and of any suitable memory length, such as 3, 4, 5, or 6, for the purpose of simplicity, 64QAM with a memory length of 3 is assumed.

(64) Each value in the initial PDLUT may be corresponding to a data symbol sequence of length 3 (also referred to a sequence of memory length 3). The middle data symbol in the sequence may the under-correct symbol, and the other two symbols may be the left and right symbols adjacent to the under-correct symbol. Since the modulation format assumed is 64QAM and the memory length is 3, the initial PDLUT may include a total of 8.sup.3=512 distortion correction values. The compressed PDLUT generator **512** may rearrange the data symbol sequences and the associated distortion correction values in a manner that the data symbol sequences are symmetrical. By way of example, symbol values in the first symbol data sequence in the rearranged PDLUT may be negative of the symbol values in the last symbol data sequence in the rearranged PDLUT. Similarly, symbol values in the second symbol data sequence in the rearranged PDLUT may be negative of the symbol values in the second last symbol data sequence in the rearranged PDLUT.

(65) FIG. **6** illustrates a representative rearranged PDLUT **600** in accordance with various non-limiting embodiments. The rearranged PDLUT **600** may include an index **602**, data symbol sequences **604**, and associated distortion correction values **606**. The data symbols may be any one of [−7, −5, −3, −1, 1, 3, 5, 7]. The data symbol sequences **604** may be arranged as follows: i) First, the left symbol is fixed at −7, the right symbol is fixed at −7, and the center symbol varies from −7

to 7; and ii) The right symbol is changed from −7 to −5, and the center symbol varies from −7 to 7, and so on.

(66) FIG. **7** illustrates a representation **700** of the distortion correction values **606** plotted against the index **602**, in accordance with various embodiments of the present disclosure. As shown, the distortion correction values **606** are approximately symmetrical about the center point of the index **602**. In other words, the distortion correction values **606** may be symmetrical about the index value 256. The distortion correction values **606** are approximately symmetrical as the sequences are opposite about the center point. For example, the first data symbol sequence is [−7 −7 −7] and the last data symbol sequence [7 7 7] is opposite to the first data symbol sequence, therefore the distortion may be approximately equal and opposite to each other.

(67) To compress or reduce the size of the rearranged PDLUT **600**, in certain embodiments, the compressed PDLUT generator **512** may be configured to fold, and amplitude reverse the second half of the rearranged data symbol sequences and the associated distortion correction values in the rearranged PDLUT **600**. The process of amplitude reversal may be achieved by multiplying the second half of the rearranged data symbol sequences and the associated distortion correction values by −1. In so doing, the distortion correction value associated with the last data symbol sequence may be overlapped with the distortion correction value associated with the first data symbol sequence. The distortion correction value associated with the second last data symbol sequence may be overlapped with the distortion correction value associated with the second data symbol sequence and so on.

(68) FIG. **8** illustrates a representation **800** of a folded PDLUT, in accordance with various non-limiting embodiments of the present disclosure. It is to be noted that the two overlapped distortion correction values may be approximately same. The compressed PDLUT generator **512** may determine an approximate distortion correction from the two overlapped distortion correction values. In some embodiments, the compressed PDLUT generator **512** may take an average of the two overlapped distortion correction values. In other embodiments, the compressed PDLUT generator **512** may select any one of the two overlapped distortion correction values. In order to determine an approximate distortion correction value corresponding to the first data symbol sequence is [−7 −7 −7], the compressed PDLUT generator **512** may either take an average of the reversed distortion correction value −1.309202 corresponding to the symbol sequence of [7 7 7] and the distortion correction value −1.17239 or select one of the distortion correction value −1.309202 and the distortion correction value −1.17239.

(69) In yet other embodiments, the compressed PDLUT generator **512** may add an offset value to the distortion correction values to obtain the approximate distortion correction values. The compressed PDLUT generator **512** may determine a value of the offset, for example, by blind searching, and the determined offset value may be the one that may correspond to a smallest RMS value of the errors between the distortion correction values of the first half of the rearranged data symbol sequences and reversed values of the second half of the rearranged data symbol sequences.

(70) By way of example, based on the blind searching technique, the compressed PDLUT generator **512** may determine that a representative value of the offset is 0.13. In order to determine an approximate distortion correction value corresponding to first data symbol sequence [−7 −7 −7], the compressed PDLUT generator **512** may multiply the distortion correction value 1.309202 associated with the last data symbol sequence [7 7 7] by −1 and subtract the representative offset 0.13 to determine the approximate distortion correction value. Similarly, to determine an approximate distortion correction value corresponding to second data symbol sequence [−7 −5 −7], the compressed PDLUT generator **512** may multiply the distortion correction value 0.532123 associated with the second last data symbol sequence [7 5 7] by −1 and subtract the representative offset 0.13 to determine the approximate distortion correction value and so on.

(71) In yet other embodiments, where the compressed PDLUT generator **512** determines the approximate distortion correction values by taking an average of the two overlapped distortion

correction values, the compressed PDLUT generator **512** may still add an offset value to the averaged distortion correction value. Similar to the previous embodiments, the compressed PDLUT generator **512** may determine a value of the offset, for example, by blind searching, and the determined offset value may be the one that may correspond to a smallest RMS value of the errors between the distortion correction values of the first half of the rearranged data symbol sequences and reversed values of the second half of the rearranged data symbol sequences.

(72) By way of example, based on the blind searching technique, the compressed PDLUT generator **512** may determine that a representative value of the offset corresponding to the averaging technique is 0.065. In order to determine an approximate distortion correction value corresponding to first data symbol sequence [−7 −7 −7], the compressed PDLUT generator **512** may multiply the distortion correction value 1.309202 associated with the last data symbol sequence [7 7 7] by −1, take an average of the distortion correction value −1.17239 associated with the first data symbol sequence [−7 −7 −7] and the reversed distortion correction value −1.309202 corresponding to the last data symbol sequence [7 7 7], and add the representative offset 0.065 to determine the approximate distortion correction value. Similarly, to determine an approximate distortion correction value corresponding to second data symbol sequence [−7 −5 −7], the compressed PDLUT generator **512** may multiply the distortion correction value 0.532123 associated with the second last data symbol sequence [7 5 7] by −1, take an average of the distortion correction value −0.71254 associated with the second data symbol sequence [−7 −5 −7] and the reversed distortion correction value −0.532123 corresponding to the second last data symbol sequence [7 5 7], and add the representative offset 0.065 to determine the approximate distortion correction value and so on.

(73) How the compressed PDLUT generator **512** determines the approximate distortion correction value should not limit the scope of present disclosure. Also, in the above embodiments, it was assumed that second half of the rearranged PDLUT **600** was folded, and amplitude reversed over the first half of the rearranged PDLUT **600**. In various embodiments, the first half of the rearranged PDLUT **600** may be folded, and amplitude reversed over the second half of the rearranged PDLUT **600**.

(74) The compressed PDLUT generator **512** may reduce the size of the PDLUT from 512 to and generate a compressed PDLUT having 256 indexes. In certain non-limiting embodiments, to further reduce the size of the compressed PDLUT having 256 indexes, the compressed PDLUT generator **512** may divide the compressed PDLUT having 256 indexes into four sub-PDLUTs, each including 64 indexes, 64 data symbol patterns and 64 distortion correction value.

(75) FIGS. **9-12** illustrate sub-PDLUTs, in accordance with various non-limiting embodiments of the present disclosure. The first sub-PDLUT **900** may include an index **902**, data symbol sequences **904**, and associated distortion correction values **906**. The index **902** in the first sub-PDLUT **900** is from 1 to 64. The left symbols in the data symbol sequences **904** are fixed to −7, the right symbol is fixed at −7, and the center symbol varies from −7 to 7. After that the right symbol is changed from −7 to −5, and the center symbol varies from −7 to 7, and so on. Since the center symbols and the right symbols are symmetrical and opposite to each other about the center point, the corresponding distortion correction values **906** are approximately same (with some offset) and opposite to each other about the center point. By way of example, the center and the right symbols [−7 −7] in the first data symbol sequence are opposite to the center and the right symbols [7 7] in the last data symbol sequence and the left symbols in the first data symbol sequence and the last data symbol sequence are the same. Therefore, the distortion may be approximately equal and opposite to each other.

(76) The second sub-PDLUT **1000** may include an index **1002**, data symbol sequences **1004**, and associated distortion correction values **1006**. The index **1002** in the second sub-PDLUT **1000** is from 65 to 128. The left symbols in the data symbol sequences **1004** are fixed to −5, the right symbol is fixed at −7, and the center symbol varies from −7 to 7. After that the right symbol is changed from −7 to −5, and the center symbol varies from −7 to 7, and so on. Similar to the first

sub-PDLUT **900**, the center symbols and the right symbols in the second sub-PDLUT **1000** are symmetrical and opposite to each other about the center point, the corresponding distortion correction values **1006** are approximately same (with some offset) and opposite to each other about the center point.

(77) The third sub-PDLUT **1100** may include an index **1102**, data symbol sequences **1104**, and associated distortion correction values **1106**. The index **1102** in the third sub-PDLUT **1100** is from 129 to 192. The left symbols in the data symbol sequences **1104** are fixed to −3, the right symbol is fixed at −7, and the center symbol varies from −7 to 7. After that the right symbol is changed from −7 to −5, and the center symbol varies from −7 to 7, and so on. Similar to the first sub-PDLUT **900** and the second sub-PDLUT **1000**, the center symbols and the right symbols in the third sub-PDLUT **1100** are symmetrical and opposite to each other about the center point, the corresponding distortion correction values **1106** are approximately same (with some offset) and opposite to each other about the center point.

(78) The fourth sub-PDLUT **1200** may include an index **1202**, data symbol sequences **1204**, and associated distortion correction values **1206**. The index **1202** in the fourth sub-PDLUT **1200** is from 193 to 256. The left symbols in the data symbol sequences **1204** are fixed to −1, the right symbol is fixed at −7, and the center symbol varies from −7 to 7. After that the right symbol is changed from −7 to −5, and the center symbol varies from −7 to 7, and so on. Similar to the first sub-PDLUT **900**, the second sub-PDLUT **1000** and the third sub-PDLUT **1100**, the center symbols and the right symbols in the third sub-PDLUT **1200** are symmetrical and opposite to each other about the center point, the corresponding distortion correction values **1106** are approximately same (with some offset) and opposite to each other about the center point.

(79) FIG. **13** illustrates representations **1302-1308** of the distortion correction values associated with the sub-PDLUTs **900**, **1000**, **1100**, and **1200** respectively plotted against the corresponding indexes, in accordance with various embodiments of the present disclosure. As shown, the distortion correction values are approximately symmetrical about the center point of the indexes. For the sub-PDLUTs **900**, **1000**, **1100**, and **1200**, the distortion correction values may be symmetrical about the index value 32, 96, 160, and 224 respectively.

(80) To further compress or reduce the size of the sub-PDLUTs **900**, **1000**, **1100**, and **1200**, in certain embodiments, the compressed PDLUT generator **512** may be configured to fold the sub-PDLUTs **900**, **1000**, **1100**, and **1200** around the respective center points and multiply the folded distortion correction values by −1. In so doing, the distortion correction value associated with the last data symbol sequence in each of the sub-PDLUTs **900**, **1000**, **1100**, and **1200** may be overlapped with the distortion correction value associated with the first data symbol sequence of each of the sub-PDLUTs **900**, **1000**, **1100**, and **1200**. The distortion correction value associated with the second last data symbol sequence in each of the sub-PDLUTs **900**, **1000**, **1100**, and **1200** may be overlapped with the distortion correction value associated with the second data symbol sequence of each of the sub-PDLUTs **900**, **1000**, **1100**, and **1200**. and so on.

(81) FIG. **14** illustrates representations **1402-1408** of the folded sub-PDLUTs **900**, **1000**, **1100**, and **1200**, in accordance with various non-limiting embodiments of the present disclosure. As previously discussed, the two overlapped distortion correction values may be approximately same with some offset. The compressed PDLUT generator **512** may determine an approximate distortion correction from the two overlapped distortion correction values in a similar manner as discussed above.

(82) Further, the compressed PDLUT generator **512** may combine the approximate distortion correction values corresponding to the folded sub-PDLUTs **900**, **1000**, **1100**, and **1200** to generate a simplified PDLUT such that the size of the simplified PDLUT may be reduced to ¼ of the original.

(83) FIG. **15** illustrates a representation **1500** of the approximate distortion correction values corresponding to the simplified PDLUT, in accordance with various non-limiting embodiments. The compressed PDLUT generator **512** may segregate the approximate distortion correction values

in the representation **1500** into subsets represented as 0, 1, 2 . . . 15 in FIG. **15**. Each subset may include 8 approximate distortion correction values corresponding to the under-correct symbol varying from −7 to 7. It may be observed from the representation **1500** that at least some of the subsets may be similar to each other, and therefore the subsets may be classified into several groups based on the respective similarity. For each group, one subset may be selected as a base subset while the other subsets may be represented in terms of the base subset. In other words, the compressed PDLUT generator **512** may further reduce a size of the simplified PDLUT by exploring the unification property of the approximate distortion correction values.

(84) In certain non-limiting embodiments, the unification process may be realized by artificial intelligence (AI) clustering technique. As noted above, each subset may include 8 approximate distortion correction values and for AI clustering, each approximate distortion correction value may represent a feature. The compressed PDLUT generator **512** may rely on principal component analysis (PCA) based AI clustering technique for reducing the dimensionality of the subsets, increasing interpretability and at the same time minimizing the information loss in the subsets. By using the PCA technique may reduce the features from 8 to 2 in each subset. FIG. **16** illustrates compressed features **1600** corresponding to the 16 subsets, in accordance with various non-liming embodiments of the present disclosure.

(85) After being clustered, the compressed PDLUT generator **512** may classify the 16 subsets into 4 groups. FIG. **17** illustrates classified groups **1700**, in accordance with various non-limiting embodiments of the present disclosure. The first group **1702** may include subsets 11, 14, and 15, the second group **1704** may include subsets 5, 6, 7, 9, 10, and 13, the third group **1706** may include subsets 4, 8, and 12, and the fourth group **1708** may include subsets 0, 1, 2, 3, and 5. It is contemplated that the number of classified groups may be increased or decreased. For a greater number of classified groups, the performance, in terms of pre-compensation of the symbols, may be slightly better. However, the size of the PDLUT may increase. The number of groups could be selected based on the trade-off between performance and complexity.

(86) From each of the classified groups **1702**, **1704**, **1706**, and **1708**, compressed PDLUT generator **512** may select a base subset and represent other subsets in terms of the selected base subset. Without limiting the scope of present, in one example, the compressed PDLUT generator **512** may select the base subsets by performing blind search. In other example, the compressed PDLUT generator **512** may determine the base subset by taking an average of the subsets in a given classified group.

(87) In one example, the compressed PDLUT generator **512** may perform mapping of the other subsets in a given classified group using a linear regression technique. A non-limiting example of the linear regression technique may be ax+b, where a and b may be the regression coefficient and x may represent approximate distortion correction values corresponding to the symbols −7 to 7 for the base subset. The compressed PDLUT generator **512** may generate a mapping table that may include the values of the regression coefficients a and b.

(88) FIG. **18** illustrates a representation **1800** of the selected base subsets 12, 1, 6, and 11 from the groups **1702**, **1704**, **1706**, and **1708** respectively, in accordance with various non-limiting embodiments of the present disclosure. Each subset may represent approximate distortion correction values corresponding to the symbols −7 to 7.

(89) FIG. **19** illustrates a mapping table **1900** generated to map the subsets in a group to the corresponding base subset, in accordance with various non-limiting embodiments of the present disclosure. As shown, the table may include four classified groups. From the first group **1702**, using the blind search technique, the compressed PDLUT generator **512** may select the subset 12 as the base subset. Further, using the linear regression technique, the compressed PDLUT generator **512** may determine the regression coefficients a and b corresponding to which the subsets 4, and 8 may be represented in terms of the subset 12 by using the relation ax+b. By way of example, to determine the approximate distortion correction value corresponding to the symbol −7 in the subset

4, the compressed PDLUT generator **512** may determine the approximate distortion correction value corresponding to the symbol −7 in the subset 12, multiply the determined value by 0.8511433 and add-0.075676. In this manner, the compressed PDLUT generator **512** may determine any approximate distortion correction value associated with the subsets 4 and 8 in terms of the regression coefficients a and b and the approximate distortion correction value associated with base subset 12.

(90) Similarly, from the second group **1704**, using the blind search technique, the compressed PDLUT generator **512** may select the subset 1 as the base subset. Further, using the linear regression technique, the compressed PDLUT generator **512** may determine the regression coefficients a and b corresponding to which the subsets 0, 2, 3, and 5 may be represented in terms of the subset 1 by using the relation ax+b. By way of example, to determine the approximate distortion correction value corresponding to the symbol 7 in the subset 0, the compressed PDLUT generator **512** may determine the approximate distortion correction value corresponding to the symbol 7 in the subset 1, multiply the determined value by 1.2105033 and add −0.204368. In this manner, the compressed PDLUT generator **512** may determine any approximate distortion correction value associated with the subsets 0, 2, 3, and 5 in terms of the regression coefficients a and b and the approximate distortion correction value associated with base subset 1.

(91) It is clear from FIGS. **18** and **19** that the four base subsets may include 32 approximate distortion correction values, and the mapping table **1900** may include 24 values corresponding to the regression coefficients a and b. The four base subsets and the mapping table **1900** may further reduce the size of the compressed PDLUT from 128 to 56 value PDLUT.

(92) Generally, the mapping table **1900** may include the base subsets and reference to the associated approximate distortion correction values, the subsets associated with each base subset, and the corresponding regression coefficients a and b. By virtue of the mapping table **1900**, only the value of a and b corresponding to the subsets may be required to be stored instead of storing the associated approximate distortion correction values, thereby further reducing the size of the compressed PDLUT.

(93) In addition to compressing the PDLUT, various embodiments of the present may be applicable to compress the cascaded PDLUTs as well. As an example, PDLUT2+3 may be considered in the following embodiments, however, it is contemplated that various embodiments of the present disclosure may be equally applicable to the other suitable cascade combinations of the PDLUT. Generally, the PDLUT2+3 may be referred to as cascaded PDLUT including a first PDLUT in which the length of the data symbol sequence is 2 and the under-correct symbol is at the right center of data symbol sequence and a second PDLUT in which the length of the data symbol sequence is 3 and the under-correct symbol is at the left center of data symbol sequence.

(94) The compressed PDLUT generator **512** may generate two initial PDLUTs in a conventional manner as previously noted. For the purpose of simplicity, 64QAM with a memory length of 2 and 3 are assumed. Each value in the two initial PDLUTs may be corresponding to a data symbol sequence of lengths 2 and 3 (also referred to a sequence of memory lengths 2 and 3). For the PDLUT with the memory length 2, the right symbol in the data symbol sequence may be the under-correct symbol. For the PDLUT with the memory length 3, the left most symbol in the data symbol sequence may be the under-correct symbol. Since the modulation format assumed is 64QAM and the memory lengths are 2 and 3, the two PDLUTs may include a total of 8.sup.2=64, and 8.sup.3=512 distortion correction values respectively.

(95) The compressed PDLUT generator **512** may rearrange the data symbol sequences in and the associated distortion correction values in the two initial PDLUTs in a manner that the data symbol sequences are symmetrical.

(96) FIG. **20** illustrates a representative rearranged PDLUT (with memory length 2) **2000** in accordance with various non-limiting embodiments. The rearranged PDLUT **2000** may include an index **2002**, data symbol sequences **2004**, and associated distortion correction values **2006**. The

data symbols may be any one of [−7, −5, −3, −1, 1, 3, 5, 7]. The data symbol sequences **2004** may be arranged as follows: i) First, the left symbol is fixed at −7; ii) the right symbol is varied from −7 to 7; iii) The left symbol is changed from −7 to −5, the right symbol is varied from −7 to 7 and so on.

(97) FIG. **21** illustrates a representative rearranged PDLUT (with memory length 3) **2100**, in accordance with various non-limiting embodiments. The rearranged PDLUT **2100** may include an index **2102**, data symbol sequences **2104**, and associated distortion correction values **2106**. The data symbols may be any one of [−7, −5, −3, −1, 1, 3, 5, 7]. The data symbol sequences **2104** may be arranged as follows: i) First, the right symbol is fixed at −7, the center symbol is fixed at −7, and the left symbol varies from −7 to 7; and ii) The center symbol is changed from −7 to −5, and the left symbol varies from −7 to 7, and so on.

(98) FIG. **22** illustrates a representation **2200** of the distortion correction values **2006** plotted against the index **2002**, in accordance with various embodiments of the present disclosure. As shown, the distortion correction values **2006** are approximately symmetrical about the center point of the index **2002**. In other words, the distortion correction values **2006** may be symmetrical about the index value 32. The distortion correction values **2006** are approximately symmetrical as the sequences are opposite about the center point. For example, the first data symbol sequence is [−7 −7] and the last data symbol sequence [7 7] is opposite to the first data symbol sequence, therefore the distortion may be approximately equal and opposite to each other.

(99) FIG. **23** illustrates a representation **2300** of the distortion correction values **2106** plotted against the index **2102**, in accordance with various embodiments of the present disclosure. As shown, the distortion correction values **2106** are approximately symmetrical about the center point of the index **2102**. In other words, the distortion correction values **2106** may be symmetrical about the index value 256. The distortion correction values **2006** are approximately symmetrical as the sequences are opposite about the center point. For example, the first data symbol sequence is [−7 −7 −7] and the last data symbol sequence [7 7 7] is opposite to the first data symbol sequence, therefore the distortion may be approximately equal and opposite to each other.

(100) To compress or reduce the size of the rearranged PDLUTs **2000** and **2100**, in certain embodiments, the compressed PDLUT generator **512** may be configured to fold the rearranged PDLUTs **2000** and **2100** around the center point and multiply the folded distortion correction values by −1. In so doing, the distortion correction value associated with the last data symbol sequence may be overlapped with the distortion correction value associated with the first data symbol sequence. The distortion correction value associated with the second last data symbol sequence may be overlapped with the distortion correction value associated with the second data symbol sequence and so on. As previously noted that the two overlapped distortion correction values may be approximately same. The compressed PDLUT generator **512** may determine an approximate distortion correction from the two overlapped distortion correction values. In some embodiments, the compressed PDLUT generator **512** may take an average of the two overlapped distortion correction values. In other embodiments, the compressed PDLUT generator **512** may select any one of the two overlapped distortion correction values. In yet other embodiments, the compressed PDLUT generator **512** may add an offset value to the distortion correction values in a similar manner as discussed previously.

(101) FIGS. **24-25** illustrate representations **2400** and **2500** of folded PDLUTs corresponding to the PDLUTs **2000** and **2100** respectively, in accordance with various non-limiting embodiments of the present disclosure. It is to be noted that the two overlapped distortion correction values may be approximately same. The compressed PDLUT generator **512** may determine an approximate distortion correction from the two overlapped distortion correction values by any suitable technique as previously discussed.

(102) The compressed PDLUT generator **512** may reduce the size of the PDLUT **2000** from 64 and generate a compressed PDLUT having 32 indexes. Also, the compressed PDLUT generator **512**

may reduce the size of the PDLUT **2100** from 512 and generate a compressed PDLUT having 256 indexes. Thus, by using this symmetry the size of the PDLUT **2000** and **2100** may be reduced to half of the original (i.e., from 64+512 to 32+256).

(103) In certain non-limiting embodiments, the compressed PDLUT generator **512** may further reduce the size of the PDLUT by applying the unification technique as previously discussed over the compressed PDLUT (with memory length 3). Similar to the previous embodiments, the compressed PDLUT generator **512** may segregate the approximate distortion correction values in the compressed PDLUT (with memory length 3) into 32 subsets 0, 1, 2 . . . 31. Each subset may include 8 approximate distortion correction values corresponding to the under-correct symbol varying from −7 to 7. As previously discussed, at least some of the subsets may be similar to each other, and therefore the subsets may be classified into several groups based on the respective similarity. For each group, one subset may be selected as a base subset while the other subsets may be represented in terms of the base subset.

(104) Based on the AI clustering, the compressed PDLUT generator **512** may classify the 32 subsets into five groups. The compressed PDLUT generator **512** may select one base subset from each of the five groups. FIG. **26** illustrates a representation **2600** of the selected base subsets 2, 4, 15, 14, and 25 from the five groups respectively, in accordance with various non-limiting embodiments of the present disclosure. Each subset may represent approximate distortion correction values corresponding to the symbols −7 to 7.

(105) FIG. **27** illustrates a mapping table **2700** generated to map the subsets in a group to the corresponding base subset, in accordance with various non-limiting embodiments of the present disclosure. As shown, the table may include five classified groups **2702**, **2704**, **2706**, **2708**, and **2708**. The first group **2702** may include subsets 2, 10, 11, 18, 19, 20, 27, and 28, out which the compressed PDLUT generator **512** may select subset 2 as the base subset. The second group **2704** may include subsets 3, 4, 12, 13, 21, 29, and 30, out which the compressed PDLUT generator **512** may select subset 4 as the base subset. The third group **2704** may include subsets 6, 7, 15, and 23, out which the compressed PDLUT generator **512** may select subset 15 as the base subset. The fourth group **2706** may include subsets 5, 14, 22, and 31, out which the compressed PDLUT generator **512** may select subset 14 as the base subset. The fifth group **2710** may include subsets 0, 1, 8, 9, 16, 17, 24, 25, and 26, out which the compressed PDLUT generator **512** may select subset 25 as a base subset.

(106) Further, using the linear regression technique, the compressed PDLUT generator **512** may determine the regression coefficients a and b corresponding to which the subsets may be represented in terms of the base subset by using the relation ax+b. It is understood that the five base subsets may include 40 approximate distortion correction values, and the mapping table may include 54 values corresponding to the regression coefficients a and b. Thus, by virtue of the mapping table **2700**, a size of the compressed PDLUT (with memory length 3) may be significantly reduced from the 256-values PDLUT to 94-values PDLUT.

(107) Returning to FIG. **5**, once the compressed PDLUT generator **512** may provide the compressed PDLUT to the symbol corrector **504**. The symbol corrector **504** may pre-compensate the incoming data symbols to reduce the effect non-linear distortions. The symbol corrector **504** may be implemented in any suitable manner. Depending on how the compressed PDLUT has been implemented, the implementation of the symbol corrector **504** may vary.

(108) FIG. **28** illustrates a first implementation **2800** of the symbol corrector **504** where the compressed PDLUT is generated based on the symmetry and unification of the PDLUT, in accordance with various embodiments of the present disclosure. The first implementation **2800** may include a unification module **2802** and a symmetry compensation module **2812**. The unification module **2802** may include base subsets **2804**, a mapping table **2806**, a first multiplier **2808**, and a first adder **2810**. The base subsets **2804** and the mapping table **2806** may be stored in a memory element (not illustrated). The symmetry module **2812** may include a second multiplier

2814, a second adder **2816**, a negator **2818**, an offset compensator **2820**, and a third adder **2822**. It is contemplated that various modules and components of the first implementation **2800** may be implemented on one or more processors and instructions for the execution of various associated functionalities may be stored in one or more memory elements.

(109) The unification module **2802** may be configured to receive a data symbol sequence, for example, [−7 −5 7]. The middle symbol in the sequence may be the under-correct symbol and may require pre-compensation. The unification module **2802** may identify the data symbol sequence and accordingly may select a suitable base subset from the base subsets **2802**. In case, the data symbol sequence is included in the selected base subset, the unification module **2802** may select the corresponding approximate distortion correction value. In case, the data symbol sequence is related to one of the subsets associated with the selected base subset, the unification module **2802** may select the closest approximate distortion correction value from the selected base subset and may fetch the suitable regression coefficients a and b from the mapping table **2806**.

(110) The first multiplier **2808** may be configured to multiply the regression coefficient a with the closest approximate distortion correction value and the first adder **2810** may add the regression coefficient b to the output of the first multiplier **2808** to generate the required approximate distortion correction value.

(111) There may be certain scenarios in which the exact data symbol sequence may not be included in the subsets, however, the subsets may include a data symbol sequence which may be opposite to the data symbol sequence received by the unification module **2802**. The generated required approximate distortion correction value may be negative of the actual required approximate distortion correction value. In such scenarios, based on an input from the negator **2818**, the second multiplier **2814** may be configured to multiply the generated required approximate distortion correction value with −1 and generate the corrected approximate distortion correction value.

(112) In certain non-liming embodiments, the offset compensator **2820** may be configured to provide certain adjustments to the approximate distortion value generated by the second multiplier **2814**. The offset value may be added by the second adder **2816**. Finally, the approximate distortion correction value may be added to the under-correct symbol by the third adder **2822**.

(113) FIG. **29** illustrates a second implementation **2900** of the symbol corrector **504** where the compressed PDLUT is generated based on the symmetry of the PDLUT, in accordance with various embodiments of the present disclosure. As shown, the second implementation **2900** may include a compressed PDLUT **2902**, a multiplier **2904**, a negator **2818**, a first adder **2908**, and offset compensator **2910**, and a second adder **2912**. It is contemplated that various modules and components of the first implementation **2800** may be implemented on one or more processors and instructions for the execution of various associated functionalities may be stored in one or more memory elements.

(114) The compressed PDLUT **2902** may be configured to receive a data symbol sequence, for example, [−7 −5 7]. The middle symbol in the sequence may be the under-correct symbol and may require pre-compensation. The compressed PDLUT **2902** may identify the data symbol sequence and accordingly may select an approximate distortion correction value from the compressed PDLUT.

(115) There may be certain scenarios in which the exact data symbol sequence may not be included in the compressed PDLUT **2902**, however, the compressed PDLUT **2902** may include a data symbol sequence which may be opposite to the data symbol sequence received by the compressed PDLUT **2902**. The generated required approximate distortion correction value may be negative of the actual required approximate distortion correction value. In such scenarios, based on an input from the negator **2906**, the multiplier **2904** may be configured to multiply the generated required approximate distortion correction value with −1 and generate the corrected approximate distortion correction value.

(116) In certain non-liming embodiments, the offset compensator **2910** may be configured to

provide certain adjustments to the approximate distortion value generated by the multiplier **2904**. The offset value may be added by the adder **2908**. Finally, the approximate distortion correction value may be added to the under-correct symbol by the second adder **2912**.

(117) FIG. **30** illustrates a performance comparison table **3000**, in accordance with various non-limiting embodiments of the present disclosure. As shown, for 64QAM modulation format, the BER without any pre-compensation technique may be approximately equal to 0.0445. PDLUT3 (i.e., PDLUT with memory length 3 and without compression) improves the BER performance largely as compared to that without any pre-compensation techniques. The BER with PDLUT3 may be around 0.025986. However, the size of the PDLUT is significantly large, it may require 512 entries in the PDLUT. With symmetry simplification only, even though the BER slightly degrades to approximately 0.026439 but the size of PDLUT may be reduced from 512 to 128. With the combined simplification of symmetry and unification, the BER performance further slightly degrades to 0.026953, which is still very close to the performance of original PDLUT3. However, the size of PDLUT may be significantly reduced from 512 to 56.

(118) FIG. **31** illustrates a third implementation **3100** of the symbol corrector **504** where the compressed PDLUT is generated based cascaded PDLUT using the symmetry and unification of the PDLUT, in accordance with various embodiments of the present disclosure. The third implementation **3100** may be an extension of the first implementation **2800**. The third implementation **3100** may include the components of the first implementation and additionally, the third implementation **3100** may include base subsets **3102** corresponding to the compressed CR PDLUT, a multiplier **3104**, a negator **3106**, a first adder **3108**, an offset compensator **3110**, and a second adder **3112**.

(119) The third implementation **3100** may be configured to process the data symbol sequence using the compressed cascaded PDLUT. While the base subsets **3102**, multiplier **3104**, negator **3106**, first adder **3108**, and the offset compensator **3110** process the CR data symbol sequence, the second implementation **2800** may process the CL data symbol sequence. Finally, the approximate distortion correction value may be added to the under-correct symbol by the second adder **3112**.

(120) FIG. **32** illustrates a performance comparison table **3200** corresponding to the cascaded PDLUT, in accordance with various non-limiting embodiments of the present disclosure. As shown, for 64 QAM modulation format, the BER performance without any pre-compensation technique may be approximately equal to 0.0445. PDLUT3+3 improves the BER performance largely as compared to that without any pre-compensation techniques. The BER performance with PDLUT3+3 may be around 0.023906. However, the size of the PDLUT3+3 is significantly large, it may require 1024 entries in the PDLUT. The BER performance with PDLUT3+2 may be around 0.027265 which is slightly higher than the BER performance corresponding to the PDLUT3+3. However, the size of the PDLUT3+2 is smaller, and it may require 576 entries only. The BER performance with PDLUT2+3 may be around 0.023969, which is significantly less as compared to the BER performance corresponding to the PDLUT3+2 for the same PDLUT size. With symmetry simplification only, even though the BER performance with PDLUT2+3 slightly degrades to approximately 0.024288 but the size of PDLUT may be reduced from 576 to 288. With the combined simplification of symmetry and unification, the BER performance further slightly degrades to 0.024666, which is still very close to the performance of original PDLUT2+3. However, the size of PDLUT may be significantly reduced from 576 to 126. Moreover, the hardware complexity may be reduced significantly.

(121) FIG. **33** depicts a high-level block diagram of representative components for the compressed PDLUT generator **512**, in accordance with various embodiments of the present disclosure. It should be appreciated that FIG. **33** provides only an illustration of one implementation of the compressed PDLUT generator **512** and does not imply any limitations with regard to the environments in which different embodiments may be implemented. Many modifications to the depicted environment can be done to implement the compressed PDLUT generator **512** without departing from the principles

presented herein.

(122) As shown, the compressed PDLUT generator **512** employs one or more processors **3302**, one or more computer-readable random access memories (RAMs) **3304**, one or more computer-readable read only memories (ROMs) **3306**, one or more computer-readable storage media **3308**, device drivers **3314**, a read/write (R/W) interface **3316**, a network interface **3318**, all interconnected over a communications fabric **3320**. Communication fabric **3320** may be implemented with any architecture designed for passing data and/or control information between processors (such as microprocessors, communications and network processors, etc.), memory, peripheral devices, and any other hardware components within a system.

(123) One or more operating system(s) **3310** and one or more application program(s) **3312** are stored on one or more of computer-readable storage media **3308** for execution by one or more of the processors **3302** via one or more of the respective RAMs **3304** (which typically include a cache memory). In the illustrated embodiment, each of computer-readable storage media **3308** may be a magnetic disk storage device of an internal hard drive, CD-ROM, DVD, memory stick, magnetic tape, magnetic disk, optical disc, a semiconductor storage device such as RAM, ROM, EPROM, flash memory or any other computer-readable tangible storage device that can store a computer program and digital information.

(124) The compressed PDLUT generator **512** may also include the R/W drive or interface **3316** to read from and write to one or more portable computer readable storage media **3328**. Application programs **3312** on said devices may be stored on one or more of the portable computer readable storage media **3328**, read via the respective R/W drive or interface **3316** and loaded into the respective computer readable storage media **3308**.

(125) It will be appreciated that in certain embodiments the application programs **3312** stored on one or more of the portable computer readable storage media **3328** may configure the compressed PDLUT generator **512** to provide various functionalities, in accordance with various embodiments of the present disclosure.

(126) The application programs **3312** on the said the compressed PDLUT generator **512** may be downloaded to the compressed PDLUT generator **512** from an external computer or external storage device via a communication network (for example, the Internet, a local area network or other wide area network or wireless network) and network interface **3318**. From network interface **3318**, the programs may be loaded onto computer-readable storage media **3308**.

(127) The compressed PDLUT generator **512** may also include a display screen **3322**, a keyboard or keypad **3324**, and a computer mouse or touchpad **3324**. The device drivers **3314** may interface to display screen **3322** for imaging, to keyboard or keypad **3324**, to a computer mouse or touchpad **3324**, and/or to display screen **3322** (in case of touch-screen display) for pressure sensing of alphanumeric character entry and user selections. The device drivers **3314**, R/W interface **3316** and network interface **3318** may comprise hardware and software (stored on computer-readable storage media **3308** and/or ROM **3306**).

(128) The programs described herein are identified based upon the application for which they are implemented in a particular embodiment of the present disclosure. However, it should be appreciated that any particular program nomenclature herein is used merely for convenience, and thus the disclosure should not be limited to use solely in any specific application identified and/or implied by such nomenclature.

(129) It will be appreciated that the compressed PDLUT generator **512** may be a server, a desktop computer, a laptop computer, a tablet, a smartphone, a personal digital assistant or any device that may be configured to implement the present technology, as should be understood by a person skilled in the art.

(130) FIG. **34** illustrates a process **3400** representing a method of compressing a Pattern Dependent Look-up Table (PDLUT) including distortion correction values corresponding to data symbol sequences, in accordance with various embodiments of the present disclosure. As shown, the

process **3400** commences at step **3402** where the compressed PDLUT generator **512** symmetrically rearranges the data symbol sequence and the associated distortion correction values.

(131) The process **3400** advances to step **3404** where the compressed PDLUT generator **512** folds and amplitude reverses second half of the rearranged data symbol sequences and the associated distortion correction values.

(132) The process **3400** proceeds to step **3406** where the compressed PDLUT generator **512** determines approximate distortion correction values for the first half of the rearranged data symbol sequences based on the distortion correction values associated with the first half of the rearranged data symbol sequences and the folded and amplitude reversed distortion correction values associated with the second half of the data symbol sequences.

(133) Finally ate step **3408**, the compressed PDLUT generator **512** updates the PDLUT based on the approximate distortion values corresponding to the first half of the rearranged data symbol sequences.

(134) FIG. **35** illustrates a process **3500** representing a method for pre-compensating an under-correct symbol, in accordance with various embodiments of the present disclosure. As shown, the process **3500** commences at step **3502**, where based on a first data symbol sequence, the symbol corrector **504** selects a base subset from a plurality of base subsets included in a compressed Pattern Dependent Look-up Table (PDLUT).

(135) The process **3500** advances to step **3504** where in the event that the first data symbol sequence is included in the base subset, the symbol corrector **504** selects an approximate distortion correction value corresponding to the first data symbol sequence from the base subset.

(136) The process **3500** advances to step **3506** where in the event that the first data symbol sequence is not included in the base subset, the symbol corrector **504** extracts mapping coefficients from a mapping table included in the compressed PDLUT, based on the mapping coefficients, maps the base subset to generate a desired subset, and selects an approximate distortion correction value corresponding to the first data symbol sequence from the desired subset.

(137) Finally at step **3408**, the symbol corrector **504** applies the approximate distortion correction value to the under-correct symbol.

(138) It is to be understood that the operations and functionality of various systems and methods, constituent components, and associated processes may be achieved by any one or more of hardware-based, software-based, and firmware-based elements. Such operational alternatives do not, in any way, limit the scope of the present disclosure.

(139) It will also be understood that, although the embodiments presented herein have been described with reference to specific features and structures, it is clear that various modifications and combinations may be made without departing from such disclosures. The specification and drawings are, accordingly, to be regarded simply as an illustration of the discussed implementations or embodiments and their principles as defined by the appended claims, and are contemplated to cover any and all modifications, variations, combinations or equivalents that fall within the scope of the present disclosure.

Claims

1. A method for pre-compensating an under-correct symbol, the method comprising: based on a first data symbol sequence, selecting a base subset from a plurality of base subsets included in a compressed Pattern Dependent Look-up Table (PDLUT); in the event that the first data symbol sequence is included in the base subset, selecting an approximate distortion correction value corresponding to the first data symbol sequence from the base subset; in the event that the first data symbol sequence is not included in the base subset, extracting mapping coefficients from a mapping table included in the compressed PDLUT, based on the mapping coefficients, mapping the base subset to generate a desired subset, and selecting an approximate distortion correction value

corresponding to the first data symbol sequence from the desired subset; and applying the approximate distortion correction value to the under-correct symbol.

2. The method of claim 1 further comprising: in the event that the first data symbol sequence is not included in the desired subset, determining a data symbol sequence in the desired subset, which is opposite to the first data symbol sequence, selecting an approximate distortion correction value corresponding to the opposite data symbol sequence from the desired subset, and negating the approximate distortion correction value corresponding to the opposite data symbol sequence.

3. The method of claim 1 further comprising: identifying a second data symbol sequence associated with the under-correct symbol in the PDLUT; in the event that the identified second data symbol sequence is included in the PDLUT, selecting an approximate distortion correction value corresponding to the identified second data symbol from the PDLUT; in the event that the identified second data symbol sequence is not included in the compressed PDLUT, determining a data symbol sequence in the compressed PDLUT which is opposite to the identified second data symbol sequence, selecting an approximate distortion correction value corresponding to the opposite data symbol sequence from the PDLUT, and negating the approximate distortion correction value corresponding to the opposite data symbol sequence; and applying the approximate distortion correction value to the under-correct symbol.

4. A system for pre-compensating an under-correct symbol, the system comprising: a non-transitory memory element having instructions thereon; a processor coupled to the non-transitory memory element and which execute the instructions to cause the processor to: based on a first data symbol sequence, select a base subset from a plurality of base subsets included in a compressed Pattern Dependent Look-up Table (PDLUT); in the event that the first data symbol sequence is included in the base subset, select an approximate distortion correction value corresponding to the first data symbol sequence from the base subset; in the event that the first data symbol sequence is not included in the base subset, extract mapping coefficients from a mapping table included in the compressed PDLUT, based on the mapping coefficients, map the base subset to generate a desired subset, and select an approximate distortion correction value corresponding to the first data symbol sequence from the desired subset; and apply the approximate distortion correction value to the under-correct symbol.

5. The system of claim 4, wherein the processor is further configured to: in the event that the first data symbol sequence is not included in the desired subset, determine a data symbol sequence in the desired subset, which is opposite to the first data symbol sequence, select an approximate distortion correction value corresponding to the opposite data symbol sequence from the desired subset, and negate the approximate distortion correction value corresponding to the opposite data symbol sequence.

6. The system of claim 4, wherein the processor is further configured to: identify a second data symbol sequence associated with the under-correct symbol in the PDLUT; in the event that the identified second data symbol sequence is included in the PDLUT, select an approximate distortion correction value corresponding to the identified second data symbol from the PDLUT; in the event that the identified second data symbol sequence is not included in the compressed PDLUT, determine a data symbol sequence in the compressed PDLUT which is opposite to the identified second data symbol sequence, select an approximate distortion correction value corresponding to the opposite data symbol sequence from the PDLUT, negate the approximate distortion correction value corresponding to the opposite data symbol sequence; and apply the approximate distortion correction value to the under-correct symbol.