SYSTEM AND METHOD FOR MODULATION DIVERSITY

Claim of Priority under 35 U.S.C. §119

[0001] The present Application for Patent claims priority to Provisional Application No. 60/592,999 entitled "METHOD OF CHANNEL INTERLEAVING IN A OFDM WIRELESS COMMUNICATIONS SYSTEM" filed July 29, 2004, and assigned to the assignee hereof and hereby expressly incorporated by reference herein.

BACKGROUND

Field

[0002] The present disclosed embodiments relates generally to wireless communications, and more specifically to channel interleaving in a wireless communications system.

Background

[0003] Orthogonal frequency division multiplexing (OFDM) is a technique for broadcasting high rate digital signals. In OFDM systems, a single high rate data stream is divided into several parallel low rate substreams, with each substream being used to modulate a respective subcarrier frequency. It should be noted that although the present invention is described in terms of quadrature amplitude modulation, it is equally applicable to phase shift keyed modulation systems.

[0004] The modulation technique used in OFDM systems is referred to as quadrature amplitude modulation (QAM), in which both the phase and the amplitude of the carrier frequency are modulated. In QAM modulation, complex QAM symbols are generated from plural data bits, with each symbol including a real number term and an imaginary number term and with each symbol representing the plural data bits from which it was generated. A plurality of QAM bits are transmitted together in a pattern that can be graphically represented by a complex plane. Typically, the pattern is referred to as a "constellation". By using QAM modulation, an OFDM system can improve its efficiency.

[0005] It happens that when a signal is broadcast, it can propagate to a receiver by more than one path. For example, a signal from a single transmitter can propagate along a straight line to a receiver, and it can also be reflected off of physical objects to

propagate along a different path to the receiver. Moreover, it happens that when a system uses a so-called "cellular" broadcasting technique to increase spectral efficiency, a signal intended for a received might be broadcast by more than one transmitter. Hence, the same signal will be transmitted to the receiver along more than one path. Such parallel propagation of signals, whether man-made (i.e., caused by broadcasting the same signal from more than one transmitter) or natural (i.e., caused by echoes) is referred to as "multipath". It can be readily appreciated that while cellular digital broadcasting is spectrally efficient, provisions must be made to effectively address multipath considerations.

[0006] Fortunately, OFDM systems that use QAM modulation are more effective in the presence of multipath conditions (which, as stated above, must arise when cellular broadcasting techniques are used) than are QAM modulation techniques in which only a single carrier frequency is used. More particularly, in single carrier QAM systems, a complex equalizer must be used to equalize channels that have echoes as strong as the primary path, and such equalization is difficult to execute. In contrast, in OFDM systems the need for complex equalizers can be eliminated altogether simply by inserting a guard interval of appropriate length at the beginning of each symbol. Accordingly, OFDM systems that use QAM modulation are preferred when multipath conditions are expected.

[0007] In a typical trellis coding scheme, the data stream is encoded with a convolutional encoder and then successive bits are combined in a bit group that will become a QAM symbol. Several bits are in a group, with the number of bits per group being defined by an integer "m" (hence, each group is referred to as having an "m-ary" dimension). Typically, the value of "m" is four, five, six, or seven, although it can be more or less.

[0008] After grouping the bits into multi-bit symbols, the symbols are interleaved. By "interleaving" is meant that the symbol stream is rearranged in sequence, to thereby randomize potential errors caused by channel degradation. To illustrate, suppose five words are to be transmitted. If, during transmission of a non-interleaved signal, a temporary channel disturbance occurs. Under these circumstances, an entire word can be lost before the channel disturbance abates, and it can be difficult if not impossible to know what information had been conveyed by the lost word.

[0009] In contrast, if the letters of the five words are sequentially rearranged (i.e., "interleaved") prior to transmission and a channel disturbance occurs, several letters

might be lost, perhaps one letter per word. Upon decoding the rearranged letters, however, all five words would appear, albeit with several of the words missing letters. It will be readily appreciated that under these circumstances, it would be relatively easy for a digital decoder to recover the data substantially in its entirety. After interleaving the m-ary symbols, the symbols are mapped to complex symbols using QAM principles noted above, multiplexed into their respective sub-carrier channels, and transmitted.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1a shows a channel interleaver in accordance with an embodiment;

[0011] FIG. 1b shows a channel interleaver in accordance with another embodiment;

[0012] FIG. 2a shows code bits of a turbo packet placed into an interleaving buffer in accordance with an embodiment;

[0013] FIG. 2b shows an interleaver buffer arranged into an N/m rows by m columns matrix in accordance with an embodiment;

[0014] FIG. 3 illustrates an interleaved interlace table in accordance with an embodiment;

[0015] FIG. 4 shows a channelization diagram in accordance with an embodiment;

[0016] FIG. 5 shows a channelization diagram with all one's shifting sequence resulting in long runs of good and poor channel estimates for a particular slot, in accordance with an embodiment;

[0017] FIG. 6 shows a Channelization diagram with all two's shifting sequence resulting in evenly spread good and poor channel estimate interlaces; and

[0018] FIG. 7 shows a wireless device configured to implement interleaving in accordance with an embodiment.

DETAILED DESCRIPTION

[0019] In an embodiment, a channel interleaver comprises a bit interleaver and a symbol interleaver. Figure 1 shows two types of channel interleaving schemes. Both schemes use bit interleaving and interlacing to achieve maximum channel diversity.

[0020] Figure 1a shows a channel interleaver in accordance with an embodiment. Figure 1b shows a channel interleaver in accordance with another embodiment. The interleaver of figure 1b uses bit-interleaver solely to achieve m-ary modulation diversity and uses a two-dimension interleaved interlace table and run-time slot-to-interlace mapping to achieve frequency diversity which provides better interleaving performance without the need for explicit symbol interleaving.

[0021] Figure 1a shows Turbo coded bits 102 input into bit interleaving block 104. Bit interleaving block 104 outputs interleaved bits, which are input into constellation symbol mapping block 106. Constellation symbol mapping block 106 outputs constellation symbol mapped bits, which are input into constellation symbol interleaving block 108. Constellation symbol interleaving block 108 outputs constellation symbol interleaved bits into channelization block 110. Channelization block 110 interlaces the constellation symbol interleaved bits using an interlace table 112 and outputs OFDM symbols 114.

[0022] Figure 1b shows Turbo coded bits 152 input into bit interleaving block 154. Bit interleaving block 154 outputs interleaved bits, which are input into constellation symbol mapping block 156. Constellation symbol mapping block 15 outputs constellation symbol mapped bits, which are input into channelization block 158. Channelization block 158 channelizes the constellation symbol interleaved bits using an interleaved interlace table and dynamic slot-interlace mapping 160 and outputs OFDM symbols 162.

[0023] Bit Interleaving for modulation diversity

[0024] The interleaver of figure 1b uses bit interleaving 154 to achieve modulation diversity. The code bits 152 of a turbo packet are interleaved in such a pattern that adjacent code bits are mapped into different constellation symbols. For example, for 2m-Ary modulation, the N bit interleaver buffer are divided into N/m blocks. Adjacent code bits are written into adjacent blocks sequentially and then are read out one by one from the beginning of the buffer to the end in the sequential order, as shown in Figure 2a (Top). This guarantees that adjacent code bits be mapped to different constellation symbols. Equivalently, as is illustrated in Figure 2b (Bottom), the interleaver buffer is arranged into an N/m rows by m columns matrix. Code bits are written into the buffer column by column and are read out row by row. To avoid the adjacent code bit to be mapped to the same bit position of the constellation symbol due to the fact that certain bits of a constellation symbol are more reliable than the others for 16QAM depending on the mapping, for example, the first and third bits are more reliable than the second and fourth bits, rows shall be read out from left to right and right to left alternatively.

[0025] Figure 2a shows code bits of a turbo packet 202 placed into an interleaving buffer 204 in accordance with an embodiment. Figure 2b is an illustration of bit interleaving operation in accordance with an embodiment. Code bits of a Turbo packet 250 are placed into an interleaving buffer 252 as shown in figure 2b. The interleaving buffer 252 is transformed by swapping the second and third columns, thereby creating interleaving buffer 254, wherein m=4, in accordance with an embodiment. Interleaved code bits of a Turbo packet 256 are read from the interleaving buffer 254.

[0026] For simplicity, a fixed m=4 may be used, if the highest modulation level is 16 and if code bit length is always divisible by 4. In this case, to improve the separation for QPSK, the middle two columns are swapped before being read out. This procedure is depicted in Figure 2b (Bottom). It would be apparent to those skilled in the art that any two columns may be swapped. It would also be apparent to those skilled in the art that the columns may be placed in any order. It would also be apparent to those skilled in the art that the rows may be placed in any order.

[0027] In another embodiment, as a first step, the code bits of a turbo packet 202 are distributed into groups. Note that the embodiments of both figure 2a and figure 2b also distribute the code bits into groups. However, rather than simply swapping rows or columns, the code bits within each group are shuffled according to a group bit order for

each given group. Thus, the order of four groups of 16 code bits after being distributed into groups may be {1, 5, 9, 13} {2, 6, 10, 14} {3, 7, 11, 15} {4, 8, 12, 16} using a simple linear ordering of the groups and the order of the four groups of 16 code bits after shuffling may be {13, 9, 5, 1} {2, 10, 6, 14} {11, 7, 15, 3} {12, 8, 4, 16}. Note that swapping rows or columns would be a regressive case of this intra-group shuffling.

[0028] Interleaved Interlace

[0029] In accordance with an embodiment, the channel interleaver uses interleaved interlace for constellation symbol interleaving to achieve frequency diversity. This eliminates the need for explicit constellation symbol interleaving. The interleaving is performed at two levels:

[0030] · Within or Intra Interlace Interleaving: In an embodiment, 500 subcarriers of an interlace are interleaved in a bit-reversal fashion.

[0031] · Between or Inter Interlace Interleaving: In an embodiment, eight interlaces are interleaved in a bit-reversal fashion.

[0032] It would be apparent to those skilled in the art that the number of subcarriers can be other than 500. It would also be apparent to those skilled in the art that the number of interlaces can be other than eight.

[0033] Note that since 500 is not power of 2, a reduced-set bit reversal operation shall be used in accordance with an embodiment. The following code shows the operation:

```
[0034]  vector<int> reducedSetBitRev(int n)
[0035]  {
[0036]      int m=exponent(n);
[0037]      vector<int> y(n);
[0038]      for (int i=0, j=0; i<n; i++,j++)
[0039]      {
[0040]          int k;
[0041]          for (; (k=bitRev(j,m))>=n; j++);
[0042]          y[i]=k;
[0043]      }
[0044]      return y;
```

[0045] }

[0046] where n=500, m is the smallest integer such that $2^m > n$ which is 8, and bitRev is the regular bit reversal operation.

[0047] The symbols of the constellation symbol sequence of a data channel is mapped into the corresponding subcarriers in a sequential linear fashion according to the assigned slot index, determined by the Channelizer, using the interlace table as is depicted in Figure 3, in accordance with an embodiment.

[0048] Figure 3 illustrates an interleaved interlace table in accordance with an embodiment. Turbo packet 302, constellation symbols 304, and interleaved interlace table 306 are shown. Also shown are interlace 3 (308), interlace 4 (310), interlace 2 (312), interlace 6 (314), interlace 1 (316), interlace 5 (318), interlace 3 (320), and interlace 7 (322).

[0049] In an embodiment, one out of the eight interlaces is used for pilot, i.e., Interlace 2 and Interlace 6 is used alternatively for pilot. As a result, the Channelizer can use seven interlaces for scheduling. For convenience, the Channelizer uses *Slot* as a scheduling unit. A slot is defined as one interlace of an OFDM symbol. An *Interlace Table* is used to map a slot to a particular interlace. Since eight interlaces are used, there are then eight slots. Seven slots will be set aside for use for Channelization and one slot for Pilot. Without loss of generality, Slot 0 is used for the Pilot and Slots 1 to 7 are used for Channelization, as is shown in Figure 4 where the vertical axis is the slot index 402, the horizontal axis is the OFDM symbol index 404 and the bold-faced entry is the interlace index assigned to the corresponding slot at an OFDM symbol time.

[0050] Figure 4 shows a channelization diagram in accordance with an embodiment. Figure 4 shows the slot indices reserved for the scheduler 406 and the slot index reserved for the Pilot 408. The bold faced entries are interlace index numbers. The number with square is the interlace adjacent to pilot and consequently with good channel estimate.

[0051] The number surrounded with a square is the interlace adjacent to the pilot and consequently with good channel estimate. Since the Scheduler always assigns a chunk of contiguous slots and OFDM symbols to a data channel, it is clear that due to the inter-interlace interleaving, the contiguous slots that are assigned to a data channel will

be mapped to discontinuous interlaces. More frequency diversity gain can then be achieved.

[0052] However, this static assignment (i.e., the slot to physical interlace mapping table1 does not change over time) does suffer one problem. That is, if a data channel assignment block (assuming rectangular) occupies multiple OFDM symbols, the interlaces assigned to the data channel does not change over the time, resulting in loss of frequency diversity. The remedy is simply cyclically shifting the Scheduler interlace table (i.e., excluding the Pilot interlace) from OFDM symbol to OFDM symbol.

[0053] Figure 5 depicts the operation of shifting the Scheduler interlace table once per OFDM symbol. This scheme successfully destroys the static interlace assignment problem, i.e., a particular slot is mapped to different interlaces at different OFDM symbol time.

[0054] Figure 5 shows a channelization diagram with all one's shifting sequence resulting in long runs of good and poor channel estimates for a particular slot 502, in accordance with an embodiment. Figure 5 shows the slot indices reserved for the scheduler 506 and the slot index reserved for the Pilot 508. Slot symbol index 504 is shown on the horizontal axis.

[0055] However, it is noticed that slots are assigned four continuous interlaces with good channel estimates followed by long runs of interlaces with poor channel estimates in contrast to the preferred patterns of short runs of good channel estimate interlaces and short runs of interlaces with poor channel estimates. In the figure, the interlace that is adjacent to the pilot interlace is marked with a square. A solution to the long runs of good and poor channel estimates problem is to use a shifting sequence other than the all one's sequence. There are many sequences can be used to fulfill this task. The simplest sequence is the all two's sequence, i.e., the Scheduler interlace table is shifted twice instead of once per OFDM symbol. The result is shown in Figure 6 which significantly improves the Channelizer interlace pattern. Note that this pattern repeats every $2 \times 7 = 14$ OFDM symbols, where 2 is the Pilot interlace staggering period and 7 is the Channelizer interlace shifting period.

[0056] To simplify the operation at both transmitters and receivers, a simple formula can be used to determine the mapping from slot to interlace at a given OFDM symbol time

---

[1] The Scheduler slot table does not include the Pilot slot.

[0057] $i = \Re'\{(N - ((R \times t)\%N) + s - 1)\%N\}$

where

[0058] · $N = I - 1$ is the number of interlaces used for traffic data scheduling, where I is the total number of interlaces;

[0059] · $i \in \{0,1,\cdots,I-1\}$, excluding the pilot interlace, is the interlace index that Slot s at OFDM symbol t maps to;

[0060] · $t = 0,1,\cdots,T-1$ is the OFDM symbol index in a super frame, where T is the total number of OFDM symbols in a frame2;

[0061] · $s = 1,2,\cdots,S-1$ s is the slot index where S is the total number of slots;

[0062] · R is the number of shifts per OFDM symbol;

[0063] · $\Re'$ is the reduced-set bit-reversal operator. That is, the interlace used by the Pilot shall be excluded from the bit-reversal operation.

[0064] Example: In an embodiment, I=8, R=2. The corresponding Slot-Interlace mapping formula becomes

[0065] $i = \Re'\{(7 - ((2 \times t)\%7) + s - 1)\%7\}$

[0066] where $\Re'$ corresponds to the following table:

$x \Rightarrow \Re'\{x\}$
$0 \Rightarrow 0$
$1 \Rightarrow 4$
$2 \Rightarrow 2$ or 6
$3 \Rightarrow 1$
$4 \Rightarrow 5$
$5 \Rightarrow 3$
[0067] $6 \Rightarrow 7$

[0068] This table can be generated by the following code:

```
[0069] int reducedSetBitRev(int x, int exclude, int n)
[0070] {
[0071]         int m=exponent(n);
```

---

[2] OFDM symbol index in a superframe instead of in a frame gives additional diversity to frames since the number of OFDM symbols in a frame in the current design is not divisible by 14.

```
[0072]      int y;
[0073]      for (int i=0; j=0; i<=x; i++, j++)
[0074]      {
[0075]          for (; (y=bitRev(j, m))==exclude; j++);
[0076]      }
[0077]      return y;
[0078]  }
```

[0079] where m=3 and bitRev is the regular bit reversal operation.

[0080] For OFDM symbol t=11, Pilot uses Interlace 6. The mapping between Slot and Interlace becomes:

[0081] · Slot 1 maps to interlace of $\Re'\{(7-(2\times 11)\%7+1-1)\%7\}=\Re\{6\}=7$;

[0082] ·Slot 2 maps to interlace of $\Re'\{(7-(2\times 11)\%7+2-1)\%7\}=\Re\{0\}=0$;

[0083] ·Slot 3 maps to interlace of $\Re'\{(7-(2\times 11)\%7+3-1)\%7\}=\Re\{1\}=4$;

[0084] ·Slot 4 maps to interlace of $\Re'\{(7-(2\times 11)\%7+4-1)\%7\}=\Re\{2\}=2$;

[0085] ·Slot 5 maps to interlace of $\Re'\{(7-(2\times 11)\%7+5-1)\%7\}=\Re\{3\}=1$;

[0086] ·Slot 6 maps to interlace of $\Re'\{(7-(2\times 11)\%7+6-1)\%7\}=\Re\{4\}=5$;

[0087] ·Slot 7 maps to interlace of $\Re'\{(7-(2\times 11)\%7+7-1)\%7\}=\Re\{5\}=3$.

[0088] The resulting mapping agrees with the mapping in Figure 6. Figure 6 shows a Channelization diagram with all two's shifting sequence resulting in evenly spread good and poor channel estimate interlaces.

[0089] In accordance with an embodiment, an interleaver has the following features:

[0090] The bit interleaver is designed to taking advantage of m-Ary modulation diversity by interleaving the code bits into different modulation symbols;

[0091] The "symbol interleaving" designed to achieve frequency diversity by INTRA-interlace interleaving and INTER-interlace interleaving;

[0092] Additional frequency diversity gain and channel estimation gain are achieved by changing the slot-interlace mapping table from OFDM symbol to OFDM symbol. A simple rotation sequence is proposed to achieve this goal.

[0093] Figure 7 shows a wireless device configured to implement interleaving in accordance with an embodiment. Wireless device 702 comprises an antenna 704, duplexer 706, a receiver 708, a transmitter 710, processor 712, and memory 714. Processor 712 is capable of performing interleaving in accordance with an embodiment. The processor 712 uses memory 714 for buffers or data structures to perform its operations.

[0094] Those of skill in the art would understand that information and signals may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

[0095] Those of skill would further appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present invention.

[0096] The various illustrative logical blocks, modules, and circuits described in connection with the embodiments disclosed herein may be implemented or performed with a general purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general purpose processor may be a microprocessor, but in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of

microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[0097] The steps of a method or algorithm described in connection with the embodiments disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium is coupled to the processor such the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a user terminal.

[0098] The previous description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the present invention. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without departing from the spirit or scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

[0099] WHAT IS CLAIMED IS:

CLAIMS

1. A method for interleaving, comprising:

placing code bits into groups according to a bit order; and

shuffling the code bits within each group according to a group bit order for the given group.

2. The method of claim 1, wherein the group bit order of the code bits for each group is the same.

3. The method of claim 1, wherein the bit order for placing code bits into each group is linear.

4. The method of claim 2, wherein the bit order for placing code bits into each group is linear.

5. The method of claim 2, wherein the group bit order for the given groups amounts to swapping columns if the given groups are viewed as a matrix.

6. The method of claim 2, wherein the group bit order for the given groups amounts to swapping rows if the given groups are viewed as a matrix.

7. The method of claim 5 further comprising transmitting code bits from the groups according to the group bit order.

8. The method of claim 6 further comprising transmitting code bits from the groups according to the group bit order.

9. A processor configured to:

place code bits into groups according to a bit order; and

shuffle the code bits within each group according to a group bit order for the given group.

10. A processor, comprising:

means for placing code bits into groups; and

means for shuffling the code bits within each group according to an order.

11. A readable media embodying a method for interleaving, comprising:

placing code bits into groups according to a bit order; and

shuffling the code bits within each group according to a group bit order for the given group.