| | |
|---|---|
| Project | **IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>** |
| Title | **Method for using non squared constellation and how to decoded them with independent I and Q.** |
| Date Submitted | **2002-07-04** |
| Source(s) | Dr. Juan Alberto Torres<br>VOCAL Technologies, Ltd.<br>200 John James Audubon Parkway<br>Buffalo, NY 14228 (http://www.vocal.com)<br>Voice: 1 (716) 688-4675<br>Fax: 1 (716) 639-0713<br>mailto:jatorres@vocal.com |
| Re: | 802.16aD4 and 802.16cD1 |
| Abstract | This paper proposes a method for using non-squared constellations and how to decode them with independent I and Q. |
| Purpose | To be used by TGa and TGc for discussion and to help preparing the draft document. |
| Notice |  |
| Release |  |
| Patent Policy and Procedures |  |

**Juan Alberto Torres**

## Proposal

This paper proposes a method for using non-squared constellations and how to decode them with independent I and Q.

## Discussion

### Background

When designing a constellation, there is always a concern of the ease of decoding the received signals. For blind decisions, one must both find the nearest point and its numeric representation.

The easiest constellations to use are those that are square with an even power of two number of symbols. These square constellations are sometimes called even constellations because of this even power. The design of the mapping of square constellations is based upon one dimensional mapping techniques that are extended to two dimensions.

### One-Dimensional Constellation Mapping

Most one dimensional constellations attempt ease encoding and decoding by algebraically manipulating their numeric value directly to create the transmit signal value. For example, consider the follow one-dimensional mapping:

| numeric value | signal value |
|---|---|
| 0 | -3 |
| 1 | -1 |
| 2 | +1 |
| 3 | +3 |

signal value = (numeric value * 2) - 3

### One Dimensional Constellation Decisions

The efficient decoding of evenly spaced one-dimensional constellations relies upon the truncation operation inherent when masking binary numbers. By representing the received signal in M.N two complement integer mathematics, the decoder can mask all bits of a power of $2^0$ and lower. The decision can be accomplished as follows:

signal value = ((received value) & MASK) + 1

where MASK removes all bits with position $2^0$ or smaller.

**Direct Constellation Bit Mapping**

It can be directly observed that there are simpler numeric values to signal value translation than the one given above. The following mapping directly maps bits of the signal value to the numeric value. The decoder would have the values of:

| numeric value | bit value | signal value | bit value vv |
|---|---|---|---|
| 2 | 10 | -3 | 1111101.00 |
| 3 | 11 | -1 | 1111111.00 |
| 0 | 00 | +1 | 0000001.00 |
| 1 | 01 | +3 | 0000011.00 |

signal value = (sign extended numeric value * 2) + 1
numeric value = bit positions 2 and 1 of signal value

The decision/decoder now becomes:

signal value = ((received value) & MASK) + 1
numeric value = bit positions 2 and 1 of signal value

**Two-Dimension Even Square Constellation Mapping**

The extension to two-dimensional even square constellations is elementary. One can provide independent bit groupings in the numeric value and independent dimensional constellation mapping. For example, extending the previous example:

| numeric value | bit value | bit value(I Q) | signal value (I, Q) | vv | vv |
|---|---|---|---|---|---|
| 12 | 1100 | 10 10 | -3 , -3 | 1111101.00 | 1111101.00 |
| 13 | 1101 | 10 11 | -3 , -1 | 1111101.00 | 1111111.00 |
| 8 | 1000 | 10 00 | -3 , +1 | 1111101.00 | 0000001.00 |
| 9 | 1001 | 10 01 | -3 , +3 | 1111101.00 | 0000011.00 |
| 14 | 1110 | 11 10 | -1 , -3 | 1111111.00 | 1111101.00 |
| 15 | 1111 | 11 11 | -1 , -1 | 1111111.00 | 1111111.00 |
| 10 | 1010 | 11 00 | -1 , +1 | 1111111.00 | 0000001.00 |
| 11 | 1011 | 11 01 | -1 , +3 | 1111111.00 | 0000011.00 |
| 4 | 0100 | 00 10 | +1 , -3 | 0000001.00 | 1111101.00 |
| 5 | 0101 | 00 11 | +1 , -1 | 0000001.00 | 1111111.00 |
| 0 | 0000 | 00 00 | +1 , +1 | 0000001.00 | 0000001.00 |
| 1 | 0001 | 00 01 | +1 , +3 | 0000001.00 | 0000011.00 |
| 6 | 0110 | 01 10 | +3 , -3 | 0000011.00 | 1111101.00 |
| 7 | 0111 | 01 11 | +3 , -1 | 0000011.00 | 1111111.00 |
| 2 | 0010 | 01 00 | +3 , +1 | 0000011.00 | 0000001.00 |
| 3 | 0011 | 01 01 | +3 , +3 | 0000011.00 | 0000011.00 |

Notice the I and Q values are formed by reordering the bits of the numeric value.

**Two-Dimension Even Square Constellation Decoding**

The decision/decoder is created by using two one-dimensional decoders. This attribute allows an easy decoder to be created as:

signal value I = ((received value I) & MASK) + 1
signal value Q = ((received value Q) & MASK) + 1

numeric value = bit positions for $2^2$ and $2^1$ of signal value I and
bit positions for $2^2$ and $2^1$ of signal value Q

The numeric value is formed by moving the signal value bits to the appropriate numeric positions.

For each symbol, there are the following operations:

AND 2
OR 2

**Two-Dimension Odd Square Constellation Mapping**

While it is slightly more complicated, one can produce similar results for square with an odd power of two number of symbols. These constellations are produced from removing half the symbols of an even square constellation. Consider, for example, a square even constellation of $2^4$ points:

| numeric value | bit value (I Q) | bit value (I , Q) | signal value | vv | vv |
|---|---|---|---|---|---|
| 12 | 1100 | 10 10 | -3 , -3 | 1111101.00 | 1111101.00 |
| 13 | 1101 | 10 11 | -3 , -1 | 1111101.00 | 1111111.00 |
| 8 | 1000 | 10 00 | -3 , +1 | 1111101.00 | 0000001.00 |
| 9 | 1001 | 10 01 | -3 , +3 | 1111101.00 | 0000011.00 |
| 14 | 1110 | 11 10 | -1 , -3 | 1111111.00 | 1111101.00 |
| 15 | 1111 | 11 11 | -1 , -1 | 1111111.00 | 1111111.00 |
| 10 | 1010 | 11 00 | -1 , +1 | 1111111.00 | 0000001.00 |
| 11 | 1011 | 11 01 | -1 , +3 | 1111111.00 | 0000011.00 |
| 4 | 0100 | 00 10 | +1 , -3 | 0000001.00 | 1111101.00 |
| 5 | 0101 | 00 11 | +1 , -1 | 0000001.00 | 1111111.00 |
| 0 | 0000 | 00 00 | +1 , +1 | 0000001.00 | 0000001.00 |
| 1 | 0001 | 00 01 | +1 , +3 | 0000001.00 | 0000011.00 |
| 6 | 0110 | 01 10 | +3 , -3 | 0000011.00 | 1111101.00 |
| 7 | 0111 | 01 11 | +3 , -1 | 0000011.00 | 1111111.00 |
| 2 | 0010 | 01 00 | +3 , +1 | 0000011.00 | 0000001.00 |
| 3 | 0011 | 01 01 | +3 , +3 | 0000011.00 | 0000011.00 |

By removing every other symbols, in a two dimensional sense, in the I/Q plane, produces the following odd constellation of $2^3$ points.

| numeric value | bit value (I Q) | bit value (I,Q) | signal value | vv | vv |
|---|---|---|---|---|---|
| 12 | 1100 | 10 10 | -3 , -3 | 1111101.00 | 1111101.00 |
| 8 | 1000 | 10 00 | -3 , +1 | 1111101.00 | 0000001.00 |
| 15 | 1111 | 11 11 | -1 , -1 | 1111111.00 | 1111111.00 |
| 11 | 1011 | 11 01 | -1 , +3 | 1111111.00 | 0000011.00 |
| 4 | 0100 | 00 10 | +1 , -3 | 0000001.00 | 1111101.00 |
| 0 | 0000 | 00 00 | +1 , +1 | 0000001.00 | 0000001.00 |
| 7 | 0111 | 01 11 | +3 , -1 | 0000011.00 | 1111111.00 |
| 3 | 0011 | 01 01 | +3 , +3 | 0000011.00 | 0000011.00 |

The 4 data bits are combined using the least significant bit as a common two-dimensional mapping bit, namely:

| numeric value | bit value I Q common | bit value ( I, Q) | signal value | v\| | v\| |
|---|---|---|---|---|---|
| 6 | 110 | 1 1 0 | -3 , -3 | 1111101.00 | 1111101.00 |
| 4 | 100 | 1 0 0 | -3 , +1 | 1111101.00 | 0000001.00 |
| 7 | 111 | 1 1 1 | -1 , -1 | 1111111.00 | 1111111.00 |
| 5 | 101 | 1 0 1 | -1 , +3 | 1111111.00 | 0000011.00 |
| 2 | 010 | 0 1 0 | +1 , -3 | 0000001.00 | 1111101.00 |
| 0 | 000 | 0 0 0 | +1 , +1 | 0000001.00 | 0000001.00 |
| 3 | 011 | 0 1 1 | +3 , -1 | 0000011.00 | 1111111.00 |
| 1 | 001 | 0 0 1 | +3 , +3 | 0000011.00 | 0000011.00 |

**Two-Dimension Odd Square Constellation Decoding**

Though square, one cannot apply the techniques used for the even square constellations directly to the odd constellations. This is because the decision regions are shaped as diamonds (rotated squares). To combat this problem, the following decision/decoding strategy is used:

a. rotate the constellation forward by 45 degrees
b. make independent I and Q decisions
c. rotate the constellation back by 45 degrees

This following procedure illustrates this technique.

| | |
|---|---|
| rotated value I = | received value I - received value Q |
| rotated value Q = | received value Q + received value I |
| | |
| rotated value I = | (rotated value I / 2 + 1) & MASK |
| rotated value Q = | (rotated value Q / 2 ) & MASK |
| | |
| signal value I = | rotated value I + rotated value Q + 1 |
| signal value Q = | rotated value Q - rotated value I + 1 |
| | |
| numeric value = | bit positions for $2^2$ signal value I and |
| | bit positions for $2^2$ signal value Q and |
| | bit position $2^1$ of signal value I or Q |

The numeric value is formed by moving the signal value bits to the appropriate numeric positions.

For each symbol, there are the following operations:

| | |
|---|---|
| ADD | 5 |
| SHIFT | 2 |
| | |
| AND | 2 |
| OR | 2 |

Though almost three times as expensive in primitive operations as the even square constellation, this odd square constellation can be still be decoded in a very small number of primitive operations.