

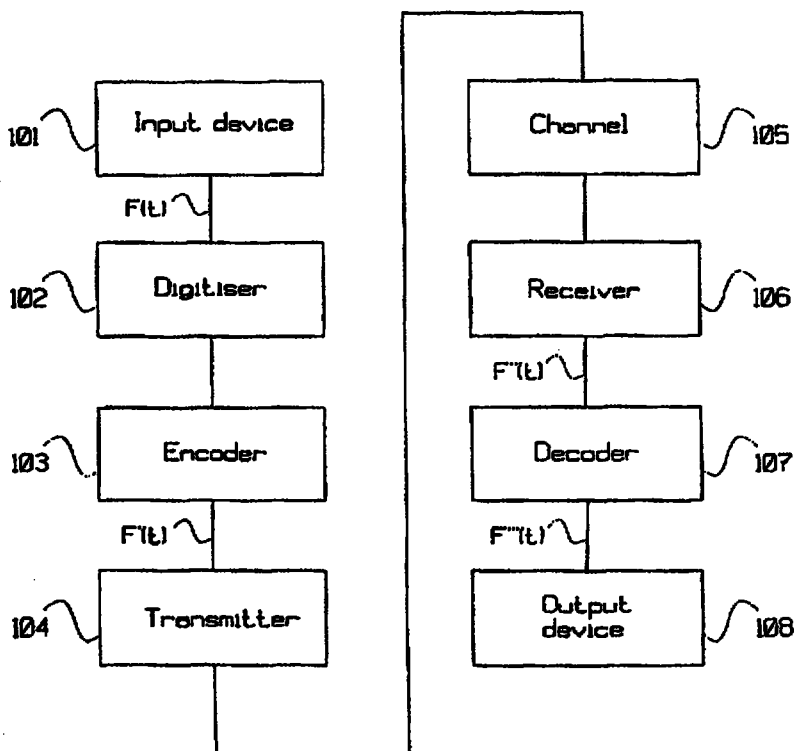
PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : H03M 7/16		A1	(11) International Publication Number: WO 97/20395
			(43) International Publication Date: 5 June 1997 (05.06.97)
(21) International Application Number: PCT/SE96/01555 (22) International Filing Date: 27 November 1996 (27.11.96) (30) Priority Data: 9504270-1 29 November 1995 (29.11.95) SE 60/007,614 29 November 1995 (29.11.95) US (71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON (publ) [SE/SE]; S-126 25 Stockholm (SE). (72) Inventor: BÄNG, Karl, Göran; Operagränd 20, S-175 48 Järfälla (SE). (74) Agents: BOHLIN, Björn et al.; Telefonaktiebolaget LM Ericsson, Patent & Trademark Dept., S-126 25 Stockholm (SE).		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG). Published <i>With international search report.</i>	
(54) Title: A METHOD AND AN ARRANGEMENT FOR MAKING FIXED LENGTH CODES			

(57) Abstract

The present invention relates to a method in a telecommunications system comprising transmission of information over a channel and encoding of the transmitted information wherein a fixed length code is designed according to the following steps: initialising of a code table, wherein a code word, hereafter called the kernel, is allocated to a position corresponding to a peak of a probability density function of the transmitted information and said kernel can be chosen arbitrarily as a n-bit code word; generating n new code words with the Hamming distance one to said kernel; packing said new code words as close as possible on either side of the kernel.



A METHOD AND AN ARRANGEMENT FOR MAKING FIXED LENGTH CODES

5

TECHNICAL FIELD

10 The present invention relates to a method of making fixed length codes more robust against bit errors. The invention also relates to an arrangement for carrying out the method.

DESCRIPTION OF THE BACKGROUND ART

15

Events can be coded in binary code words. A binary encoding method with substantially uniform rate is shown in the US patent US 5,300,930. Often one wants to code the information in as few bits as possible in order to minimise the bandwidth required. Depending on the probability distribution
20 of the events it is possible to code the information in such a way that events that occur rarely are coded with longer code words and events that occur frequently are coded with shorter code words and the encoding is done so that the decoder can distinguish between the different code words. Such codes are called instantaneous.

25

Very efficient codes can be designed in this way and the method of coding is called entropy coding.

30

However, assume that the transmission channel between the encoder and decoder introduces bit errors. In this case problems arise because the decoder cannot with certainty decode the different variable length code words since it is not possible to see where one code word starts and

another ends. Of course, this can have a devastating effect on the interpretation of the received information.

5 Usually this problem is solved by means of channel coding, where controlled redundancy is introduced in the channel encoder in such a way that the channel decoder can detect and correct erroneous bits.

10 Using this method more bits have to be transmitted over the channel and the channel bit rate increases as well as the complexity due to the channel encoder and decoder. Moreover, at bit error rates that are higher than the error correcting capacity of the channel coding, the decoded bit error rate often becomes worse than the uncoded bit error rate. Such a threshold exists for every code, either limited by the Shannon capacity formula or on a restriction in acceptable delay due to channel coding.

15 One way to overcome some of the drawbacks with a variable length code, in terms of bit error robustness, is to use a fixed length code. Here we lose some efficiency because all code words have the same length independent of the probability distribution of the events to be coded. However, once
20 synchronised, each code word can be decoded independent of any other code word and this makes the code rather robust.

25 If we assume that the information to be transmitted represents a 'level' monotonically increasing from a minimum value to a maximum value the events represent discrete points on this level curve and each point is coded with a fixed length code word.

30 A bit error in a code word causes a jump in the level curve and dependent on which bit in the code word that is erroneous, the level jump becomes smaller or larger. This means that some bit errors (at certain positions in the code word) may cause an acceptable decoded level (small level error) while other error bits in a code word cause a totally unacceptable level change.

The common way to improve the robustness to bit errors of a fixed length code table is by means of Gray coding is to be found in the following two books:

5 [1] John A.C. Bingham, "The Theory and Practice of Modem Design", John Wiley & Sons, 1988 (pp. 65-66).

[2] E.A.Lee, D.G. Messerschmitt, "Digital Communication", Kluwer Academic Publishers
10 Boston, 1988 (pp. 193).

A well known method to decrease the bit error sensitivity of a fixed length code is by means of Gray coding which is shown in figure 1. Gray codes have the property that nearest neighbours are encoded into code words that
15 differ by only one bit, i.e. the Hamming distance is 1. This works fine for a 2-bit code (4 words) in a 2-dimensional signal constellation, where also the Euclidean distance is at a minimum for code words with Hamming distance = 1.

20 In a 1-dimensional representation, when the code words represent a level, the situation is different, as illustrated with the following 3-bit codes with 8 levels.

A straight forward code and two Gray codes with associated bit error
25 distances are shown in Table 1. The distance d_1 is the sum of the distances obtained by changing one bit at a time in the code word. The mean distance per bit is then $d_1 / 3$ units.

Level	Straight Code		Gray Code 1		Gray Code 2	
	Code	Distance d1	Code	Distance d1	Code	Distance d1
0	000	4 + 2 + 1 = 7	000	5 + 1 + 3 = 9	000	3 + 1 + 7 = 11
1	001	4 + 2 + 1 = 7	010	5 + 1 + 1 = 7	010	1 + 1 + 5 = 7
2	010	4 + 2 + 1 = 7	011	5 + 1 + 1 = 7	110	1 + 1 + 3 = 5
3	011	4 + 2 + 1 = 7	001	1 + 1 + 3 = 5	100	3 + 1 + 1 = 5
4	100	4 + 2 + 1 = 7	101	1 + 3 + 1 = 5	101	3 + 1 + 1 = 5
5	101	4 + 2 + 1 = 7	100	5 + 1 + 1 = 7	111	1 + 1 + 3 = 5
6	110	4 + 2 + 1 = 7	110	5 + 1 + 1 = 7	011	1 + 1 + 5 = 7
7	111	4 + 2 + 1 = 7	111	5 + 3 + 1 = 9	001	3 + 1 + 7 = 11

Table 1.

Using a metric based upon the Euclidean distance, d1, or the squared Euclidean distance, d2, we can define the following distances

$$\epsilon_1 = \frac{1}{n} \sum_{m=0}^{M-1} p(m) d1(m)$$

and

$$\epsilon_2 = \frac{1}{n} \sum_{m=0}^{M-1} p(m) d2(m)$$

where

n is the number of bits in code word,

M is the number of code words,

p(m) is the pdf (probability density function) of the level, m = 0, 1, ..., M-1,

ϵ_1 is the mean error distance per bit and

ϵ_2 is the mean squared distance (mse) per bit.

The mean error distance per bit is $\epsilon_1 = 7/3 = 2.33$ units for all three codes.

Thus, for a *uniform pdf* for the level, nothing is gained with a Gray code compared with the straight code. However, if the levels 3 and 4 (2,3,4 and 5) are more common than the other levels a gain of the Gray code 1 (Gray code 2) in the mean error distance is achieved.

SUMMARY OF THE INVENTION

This invention is about how to make fixed length codes more robust against bit errors, in cases when the probability density function (pdf) has a peak (not uniformly distributed). A considerable gain in mean squared error (mse) can be achieved in these cases, and the actual gain depends on the number of bits in the code word and the actual pdf of the level.

PROBLEM DESCRIPTION

Introducing a bit error in the straight code in Table 1, the level jump may be 1, 2 or 4 units depending on the position of the erroneous bit. This is valid for each of the 8 code words and the mean error distance is $\epsilon_1 = 7/3 = 2.33$ units per bit. If the mse is used, the figure becomes $\epsilon_2 = (1^2 + 2^2 + 4^2)/3 = 7$ units² per bit. These values on ϵ_1 and ϵ_2 are independent of the actual pdf, and therefore this straight code may serve as a reference for other codes.

Assuming a uniform pdf for the Gray code 1 we get a mean error distance of $\epsilon_1 = 2 \cdot (9 + 7 + 7 + 5)/(8 \cdot 3) = 7/3 = 2.33$ units per bit which is equal to the straight coding. The mse becomes $\epsilon_2 = 25/3 = 8.33$ units² per bit.

The corresponding figures for the Gray code 2 are; mean error distance $\epsilon_1 = 7/3 = 2.33$ units per bit and mse $\epsilon_2 = 27/3 = 9$ units² per bit.

Thus, nothing is gained by using Gray coding compared to straight coding if the level pdf is uniform. On the contrary, if mse is chosen as the metric there is a performance loss.

However, if we have a pdf with a peak in the central part of the coding table a performance gain can be anticipated with the Gray code tables.

Still, the *minimum distance* and the *minimum squared distance* are for the Gray code tables $5/3 = 1.67$ units per bit and $11/3 = 3.67$ units² per bit, respectively, and these values may be too large to be acceptable and they do not represent the minimum values that can be achieved.

5

Sometimes the pdf does not have a peak at the central part but may have another shape, for example, two peaks on either side of the central part of the level curve. To cope with this situation another design method is required to build a bit error robust fixed length code table.

10

The chrominance (colour difference) components represented by the fixed length tables usually have a very pronounced peak, therefore the invented code should be suitable for improving the robustness to bit errors in this context.

15

BEST MODES OF CARRYING OUT THE INVENTION

20

A mobile telephone radio communication system is shown in figure 2. A radio base station BS transmits information to and from a mobile radio terminal MS. The radio connection is shown in the figure as a zig-zag symbol between the radio base station BS and the mobile radio terminal MS.

25

Figure 3 is a flow chart illustrating how information is transferred from the radio base station BS to the mobile radio terminal MS of figure 2. A block 101 illustrates an input device which in the embodiment is e.g. a video camera connected to the radio base station shown in figure 2. A block 102 illustrates a digitiser connected to the input device. An encoder 103 is in accordance with the invention connected to the digitiser. A transmitter is illustrated with block 104. The blocks 102-104 are parts of the radio base station BS. Block 105 illustrates the channel on which information is transmitted from the radio base station BS to the mobile radio terminal MS. The channel is a radio channel of 9.6 kbits/s disturbed by e.g. Rayleigh

30

fading. Block 106 illustrates a receiver situated in the mobile radio terminal MS and a decoder is illustrated by block 107. Block 108 symbolises e.g. a video monitor. The input device 101 comprises an information source that generates a variable that is a function of time $F(t)$, wherein a probability density function for values of $F(t)$ occurring in said information source has a maximum. The digitiser 102 and the encoder 103 of the function $F(t)$ perform digitisation and encoding of $F(t)$ into a function $F'(t)$ by using a table such that the Hamming distance between code words of $F'(t)$ that are adjacent to one another is lower for values of $F(t)$ that are near the maximum of the probability density function than for values of $F(t)$ that are further away from said maximum. The transmitter 104 transmits the information in the function $F'(t)$ over the channel 105. The information is received by the receiver 106 as $F''(t)$ because $F'(t)$ has been disturbed by Rayleigh fading in the channel 105. The decoder 107 decodes the received function $F''(t)$ back into a function $F'''(t)$ which is similar to the original function $F(t)$, by using a table which is the inverse of the table in the encoder 103.

The earlier mentioned Gray code is characterised by the fact that the Hamming distance for a code word to the closest neighbours is one. If the code represents a level monotonically increasing from a minimum value to a maximum value the Hamming distance may not be the proper metric to use for a robust code design. Rather, one should use a metric that reflects the *level change* when the different bits in a code word change, such as the Euclidean distance or the squared Euclidean distance (the mse).

The design method for the pdf with a peak is illustrated with a 3 bit code table, the so called Baang-1 code. We assume that we know the pdf or at least the position of the peak of the pdf representing the level curve.

The other code design method, suitable for more general pdf:s is illustrated by a 4 bit code table, the so called Baang-2 code.

7.1 Pdf with a peak (Baang-1 code)

Here we want to minimise the distances for code words taken from a particular code word, hereafter denoted the *kernel*, and outwards. The code words are taken in an order of increasing distance from the kernel.

7.1.1) Initialise the code table.

7.1.1.a) Pick the kernel, for example 111, and put it on that level position that corresponds to the peak of the level pdf, in our case, say level 3.

Level	Code Word
3	111 (kernel)

7.1.1.b) On either side of the kernel, insert two new code words with the Hamming distance one to the kernel. In this case we change the leftmost bit first and then the next bit in order. However, any order of bitchange can be used.

Level	Code Word	Distance d1
2	011	
3	111	$1 + 1 + ? = ?$
4	101	

7.1.1.c) The third bit, the rightmost bit of the kernel, is now changed and the new code word is put on level 1 (or level 5)

Level	Code Word	Distance d1	d2
1	110		
2	011		
3	111	$1 + 1 + 2 = 4$	$1^2 + 1^2 + 2^2 = 6$
4	101		

With this initialisation we have packed all code words with the Hamming distance one as close as possible (in the Euclidean distance sense) to the kernel.

5 Should there be more than 3 bits in the code words, say n bits, the same procedure is applied and the result is an initialisation of $(n + 1)$ code words.

7.1.2) Continue to build the table.

10

7.1.2.a) The next code word to consider is the one closest to the kernel and with the maximum degree of freedom to design new code words in a compact way.

15

Level 2 with the code word 011 has code words on each side, but level 4 with the code word 101 has a code word only on one side (111), and thus has the maximum degree of freedom. This code word (101) is called the 1:st reference.

20

Thus, changing the bits in the 1:st reference we get the following table: (here we use the same order for changing bits as was used in the initialisation, but any other order can be used as well)

25

Level	Code Word	Distance d1	d2
1	110		
2	011		
3	111	$1 + 1 + 2 = 4$	6
4 -->	101	$1 + 1 + 2 = 4$	6
5	001		
6	100		

30

If not the bit changed code word is in the table we add it as close as possible to the reference word.

5 7.1.2.b) Following the principle in 2a) we now consider level 2 and choose code word 011 as the 2:nd reference and get

	Level	Code Word	Distance d1	d2
	0	010		
	1	110		
10	2 -->	011	$1+3+2=6$	14
	3	111	$1+1+2=4$	6
	4	101	$1+1+2=4$	6
	5	001		
	6	100		

15

7.1.2.c) For the next reference to consider (110 or 001) there is no difference in the distance to the kernel and the number of freedom is the same and so 2a) does not give us the information how to proceed and any side can be used.

20 In this example we stick to the symmetry principle and process that code word that is on the opposite side of the kernel of the last processed code word.

In this case it means level 5 with the reference word 001, and the result is

	Level	Code Word	Distance d1	d2
	0	010		
	1	110		
	2	011	$1+3+2=6$	14
	3	111	$1+1+2=4$	6
30	4	101	$1+1+2=4$	6
	5 -->	001	$1+3+2=6$	14
	6	100		
	7	000		

7.1.3) All 8 code words are now associated with a level and we just calculate the distance $d1(m)$ and mse $d2(m)$ for the remaining 4 code words.

5

Table 2 shows the final result, the 3 bit Baang-1 code.

	Straight Code		Baang-1 Code		
	Code	Distance	Code	Distance	
		d1		d1	d2
0	000	$4+2+1=7$	010	$1+7+2=10$	54
1	001	$4+2+1=7$	110	$1+5+2=8$	30
2	010	$4+2+1=7$	011	$1+3+2=6$	14
3	011	$4+2+1=7$	111	$1+1+2=4$	6
4	100	$4+2+1=7$	101	$1+1+2=4$	6
5	101	$4+2+1=7$	001	$1+3+2=6$	14
6	110	$4+2+1=7$	100	$1+5+2=8$	30
7	111	$4+2+1=7$	000	$1+7+2=10$	54

Table 2.

10

As can be seen from the table, the distance $d2$ for the straight code is 21 for each code word.

7.1.3) Termination of the table.

15

If a new code word designed from a reference and allocated according to the principles outlined above falls outside one of the end levels, a reference word from the other side of the kernel is used. This may happen, for example, when the peak of the pdf is not in the central part of the level distribution. The following example illustrates the principle, where we assume that the peak of the pdf is located at level 5.

20

		Step 1	Step 2	Step 3	Step 4	
	Level	Code	Code	Code	Code	Distance d1
	0				000	1 + 1 + 4 = 6
5	1			100	100	1 + 6 + 5 = 12
	2		010	010	010	5 + 2 + 2 = 9
	3		001	001	--> 001	3 + 1 + 3 = 7
	4	-->	011	011	011	1 + 1 + 2 = 4
	5	111	111	111	111	1 + 1 + 2 = 4
10	6		101	--> 101	101	3 + 1 + 5 = 9
	7		110	110	110	5 + 6 + 2 = 13

7.2 More general pdf (Baang-2 code).

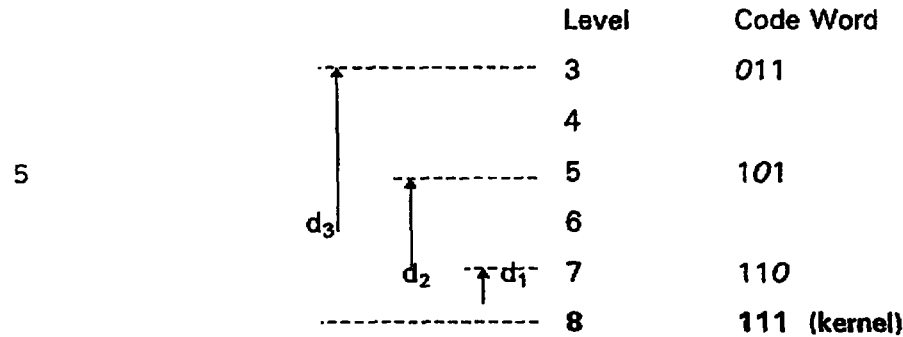
This technique may be used when the pdf has a more general shape and the design steps are as follows. Here we illustrate the method by a 4 bit code ($n=4$).

7.2.1) Initialise the code table.

7.2.1.a) Pick a $(n-1)$ bit code word, for example 111, denoted kernel, and put it on the level position that corresponds to the peak of the level pdf, say level 8.

Level	Code Word
8	111

7.2.1.b) Insert a maximum of $(n-1)$ new code words with Hamming distance one from the kernel and put them on the distances $d = [d_1, d_2, \dots, d_k]$ units from the kernel, in our example we choose $d_1 = 1$, $d_2 = 3$ and $d_3 = 5$ and we get,



10

7.2.2) Continue to build the table.

7.2.2.a) The code table is built from the kernel and upwards and the code word associated with the distance d_1 is used as the 1:st reference, i.e. 110. (The index in the distance measure is taken as the order in which the references are processed).

15

Changing the bits in the 1:st reference and packing the new words as close as possible to the kernel we get

20

Level	Code Word
3	011
4	100
5	101
6	010
7	110
8	111

25

Next we use the code word associated with d_2 , 101, as a reference, changing the bits and obtain

30

	Level	Code Word
	2	001
5	3	011
	4	100
	5 -->	101
	6	010
	7	110
10	8	111

Next reference corresponding to d_3 , 011, does not give any new code word, and all the references have now been used.

15

2.2.2.b) With the code word next to the kernel as a reference, 110, change the bits and fill up the table with new code words in positions as close as possible to the kernel. In this case 110 does not give any new code word and so continue with the next code word counted from the kernel, namely 010.

20

	Level	Code Word
	1	000
	2	001
25	3	011
	4	100
	5	101
	6 -->	010
	7	110
30	8	111

and we have all 8 code words.

7.2.2.c) To obtain the 16 possible code words for the 4 bit code we take the mirror picture of the 8 code words below the 8:th word (or above the 1:st word) and put a 1 (or 0) at the leftmost position (or rightmost position or any position) of the 8 original words and a 0 (or 1) at the same bit position in the reflected 8 words, i.e. one implementation gives

	Level	Code Word	Distance d1
	1	1000	24
	2	1001	18
10	3	1011	18
	4	1100	16
	5	1101	14
	6	1010	12
	7	1110	8
15	8	1111	10
	9	0111	10
	10	0110	8
	11	0010	12
	12	0101	14
20	13	0100	16
	14	0011	18
	15	0001	18
	16	0000	24

25 The design method ensures a symmetrical distance distribution $d1(m)$ and $d2(m)$ and it allows a flexibility to adapt the distance distribution to the actual pdf of the level in such a way that a small distance is allocated to the peak(s) of the pdf, thereby making the mean error distance per bit ϵ_1 and the squared distance per bit ϵ_2 small.

30

Forbidden code words are removed from the table.

To illustrate the flexibility of the Baang-2 code methodology, we generate 2 different 4 bit code tables based on the two initialisation vectors $d_a = [1 \ 3 \ 7]$ and $d_b = [1 \ 2 \ 3]$ and this time we show the metric $d_2(m)$ (mse). The kernel is chosen to 111. See Table 3.

Level m	Code d_a	$d_2(m)$	Code d_b	$d_2(m)$
1	1011	287	1000	239
2	1000	190	1001	195
3	1001	130	1010	145
4	1010	103	1100	103
5	1101	63	1011	71
6	1100	43	1101	49
7	1110	20	1110	35
8	1111	60	1111	15
9	0111	60	0111	15
10	0110	20	0110	35
11	0100	43	0101	49
12	0101	63	0011	71
13	0010	103	0100	103
14	0001	130	0010	145
15	0000	190	0001	195
16	0011	287	0000	239

Table 3.

5

As a comparison a straight code has $d_2(m) = 85$ for each code word.

7.3 Variations of code tables.

10

The bit-columns of the code tables can be changed in any order without changing the metric (distance).

Ones can be exchanged by zeros and vice versa without changing the metric.

15

Baang-1 code.

The Baang-1 code gives the *minimum distance* (or mse) to code words with Hamming distance one, i.e. for code words with single bit errors, to the kernel.

Alternative code words on each side of the kernel are designed in such a way that the levels associated with the Hamming distance = 1 to the code word considered are minimised.

The kernel has the least possible distance (or mse) and the distance measure (or mse) increases further away from the kernel.

The Baang-1 code is particularly useful for representing information that has a peak in its pdf.

Although the design method has been described for a short 3-bit code, for the purpose of illustration, the same design principle is applicable for longer codes. Furthermore, the maximum gain compared to a straight code increases with the length of the code words.

Baang-2 code.

With this design method we have a possibility to adapt the error distances to the peak(s) of the pdf in such a way that small error distances are allocated to the peak(s) of the pdf. The way to control this allocation is by means of the initialisation process i.e. how we set the reference code words related to the kernel - the distances $d_1, d_2, d_3, \dots d_k$, where $0 < k < n-1$ and where each d-component takes on distinct integer values in the interval $[1, 2^{(n-1)}-1]$. The d-values are put at positions where the pdf has high values.

Columns in the code table can be reordered in any order without changing the distance measures. Ones can be replaced by zeros and vice versa without changing the distance measures. The initialisation process can, of course, also generate new code words downwards (instead of upwards) and the code table can be generated in an apparent way based on the steps described above (a symmetry principle).

The arrangement in accordance with the invention comprises means for allocating a new code word to the other side of the kernel in cases where the end of the code table has been reached on the side where said code word should be allocated according to previous steps. The arrangement also comprises means to stop new code word generation when all 2^n code words have been generated and means for removing not allowed code word(s), which with a suitable choice of the kernel as previously described, is (are) allocated towards the end(s) of the code table. A number of variations of the above mentioned embodiments are of course possible. Bit-columns in the code table can e.g. be reordered in any order without changing the distance measures. Binary "ones" can be replaced by "zeros" and vice versa without changing the distance measures.

**CLAIMS**

5 **1. Method in a telecommunication system for making robust fixed length codes, which system comprises an input device (101) containing an information source that generates a variable that is a function of time $F(t)$, wherein a probability density function for values occurring in said function of time $F(t)$ has a maximum and which system comprises a digitiser (102) and an encoder (103) for digitising and encoding said values into code**
10 **words such that the Hamming distance between those among said code words that are adjacent to one another is lower for those among said values that are near the maximum of the probability density function than for values that are further away from said maximum.**

15 **2. Method in a telecommunications system according to claim 1 wherein when encoding said values of said function of time $F(t)$ a fixed length code is used designed according to the following steps:**
- initialising of a code table, wherein a code word, hereinafter called the
20 kernel, is allocated to a value the position of which corresponds to a peak of a probability density function of the transmitted information and wherein said kernel can be chosen arbitrarily as a n-bit code word;
- generating n new code words with the Hamming distance one to said kernel;
- packing said new code words as close as possible on either side of the
25 kernel in order to build said code table.

3. The method for claim 2, comprising the following further steps:
- continuing to build the table, where a first code word to consider, hereafter called the first reference word, is the code word closest to the
30 kernel that either, when n is odd, is located on that side of the kernel that has the least number of initialised code words
or, when n is even, is located on any side of the kernel;



- generating n new code words with Hamming distance one to said first reference word and packing said n new code words, if not already in the table, as close as possible to the said first reference word.

5 4. The method of claim 3, comprising the following further steps:

- selecting a second reference word from the not yet selected side of the kernel and as close as possible to the kernel;
- generating n new code words with Hamming distance one to said second reference word and packing said n new code words, if not already in the table, as close as possible to the said second reference word.

10

5. The method of claim 4, comprising the following further steps:

- repetitively selecting new reference words as close as possible to the kernel in an alternating order and increasing distance from the kernel;
- generating n new code words with Hamming distance one to current reference word and packing said n new code words, if not already in the table, as close as possible to the said current reference word.

15

6. In a telecommunications system that comprises transmission of

20

information over a channel a method to encode the transmitted information wherein the fixed length code is designed according to the following steps,

initialising of a code table, wherein a code word, hereafter called the kernel, is allocated to a position corresponding to the peak of the probability density function of the transmitted information and wherein said kernel can be chosen arbitrarily as a n -bit code word but in cases where there are less than 2^n allowed code words in the table the kernel can be favourably selected as the code word with the largest Hamming distance to the not allowed code word(s),

25

30

generating n new code words with the Hamming distance one to said kernel and packing said new code words as close as possible on either side of the kernel.



7. The method of claim 6 further comprising the following steps

continuing to build the table, where the first code word to consider,
5 hereafter called the first reference word, is the code word closest to the
kernel that

either, when n is odd, is located on that side of the kernel that has the least
number of initialised code words

or, when n is even, is located on any side of the kernel,

10 generating n new code words with Hamming distance one to said first
reference word and packing said n new code words, if not already in the
table, as close as possible to the said first reference word.

8. The method of claim 7, comprising the following further steps:

- selecting a second reference word from the not yet selected side of the
kernel and as close as possible to the kernel;
- generating n new code words with Hamming distance one to said
15 second reference word and packing said n new code words, if not already in
20 the table, as close as possible to the said second reference word.

9. The method of claim 8, comprising the following further steps:

- repetitively selecting new reference words as close as possible to the
kernel in an alternating order and increasing distance from the kernel;
- 25 - generating n new code words with Hamming distance one to current
reference word and packing said n new code words, if not already in the
table, as close as possible to the said current reference word.

10. In a telecommunications system that comprises transmission of
30 information over a radio channel a method to encode the transmitted
information wherein the fixed length code is designed according to the
steps described in claim 1.



11. In a telecommunications system that comprises transmission of video information over a radio channel a method to encode the transmitted video information wherein the fixed length code is designed according to the steps described in claim 1.

5

12. In a telecommunications system that comprises transmission of information over a channel a method for making robust fixed length codes, which system comprises an input device (101) containing an information source that generates a variable that is a function of time $F(t)$, wherein a probability density function for values occurring in said function of time $F(t)$ has at least one maximum and which system comprises a digitiser (102) and an encoder (103) for digitising and encoding said values into code words such that the Hamming distance between those among said code words that are adjacent to one another is lower for those among said values that are near the at least one maximum of the probability density function than for values that are further away from said at least one maximum wherein the fixed length code is designed according to the following steps:

10

15

20

25

30

- initiating a code table, wherein a code word, hereafter called the kernel, is allocated to a position corresponding to an estimated or real symmetry line of the probability density function of the transmitted information, said kernel can be chosen arbitrarily as a $(n-1)$ bit word, where n is the number of bits in a code word, but in cases where there are less than 2^n allowed code words in the table, the kernel can be favourably selected as the $(n-1)$ bit word with the largest Hamming distance to $(n-1)$ bits of the not allowed code word(s);
- generating k new words, where $1 \leq k \leq (n-1)$, with the Hamming distance one to said kernel and said new words are put at level distances $d = [d_1, d_2 \dots d_k]$ units from the kernel, and each said level distance is distinct from any other level distance but can be of any integer value in the level range $[1, 2^{(n-1)} - 1]$ units, and where the



said level distance index determines the order in which the corresponding words are processed,

and by selecting the said level distances $d = [d_1, d_2 \dots d_k]$ affect the error distances $d1(m)$ and $d2(m)$ to match the probability density function in such a way that,

choosing $d = [1, 2, \dots, (n-1)]$ units, the minimum error distances $d1(m)$ and $d2(m)$ are allocated towards the kernel with a peak at the kernel, and

choosing $d = [(n-1), (n-2), \dots, 1]$ units, the minimum error distances $d1(m)$ and $d2(m)$ have a wider distribution compared to previous case, and

choosing $d = x$ where x is a vector of k distinct elements in the range $[1, 2^{(n-1)}-1]$ units, with the design flexibility to move the minimum error distances $d1(m)$ and $d2(m)$ away from the kernel and where the minimum error distances occur in the region where the elements in x are compactly positioned;

- continuing to build the table, where the first word to consider is the one pointed to by d_1 , hereafter called the first reference word;

- generating $(n-1)$ new words with Hamming distance one to said first reference word and packing said $(n-1)$ new words, if not already in the table, as close as possible to the kernel;

- selecting a second reference word pointed to by d_2 ;

- generating $(n-1)$ new words with Hamming distance one to said second reference word and packing said $(n-1)$ new words, if not already in the table, as close as possible to the kernel;

- repetitively selecting new reference words pointed to by successive elements in d , until the last element d_k is reached, and;



- generating (n-1) new words with Hamming distance one to current reference word and packing said (n-1) new words, if not already in the table, as close as possible to the kernel;
 - 5 - completing the table in cases the table is not full (with $2^{(n-1)}$ words) after said initialisation process, by which repetitively the word closest to the kernel is taken as a reference word and means for generating (n-1) new words with Hamming distance one to current reference word packing said (n-1) new words, if not already in the
10 table, as close as possible to the kernel;
 - stopping new word generation when $2^{(n-1)}$ words have been generated;
 - generating $2^{(n-1)}$ words by flipping the existing $2^{(n-1)}$ words up-down
15 below the kernel;
 - putting a one (or zero) in any column position in the first $2^{(n-1)}$ words and a zero (or one) in the same column position in the next $2^{(n-1)}$ words, whereby a code table with 2^n code words is generated, and;
20
 - removing not allowed code word(s), which with a suitable choice of the kernel as previously described, is (are) allocated towards the end(s) of the code table.
- 25 **13.** In a telecommunications system that comprises transmission of information over a radio channel a method to encode the transmitted information wherein the fixed length code is designed according to the steps described in 6.
- 30 **14.** In a telecommunications system that comprises transmission of video information over a radio channel a method to encode the transmitted video information wherein the fixed length code is designed according to the steps described in 6.

1/3

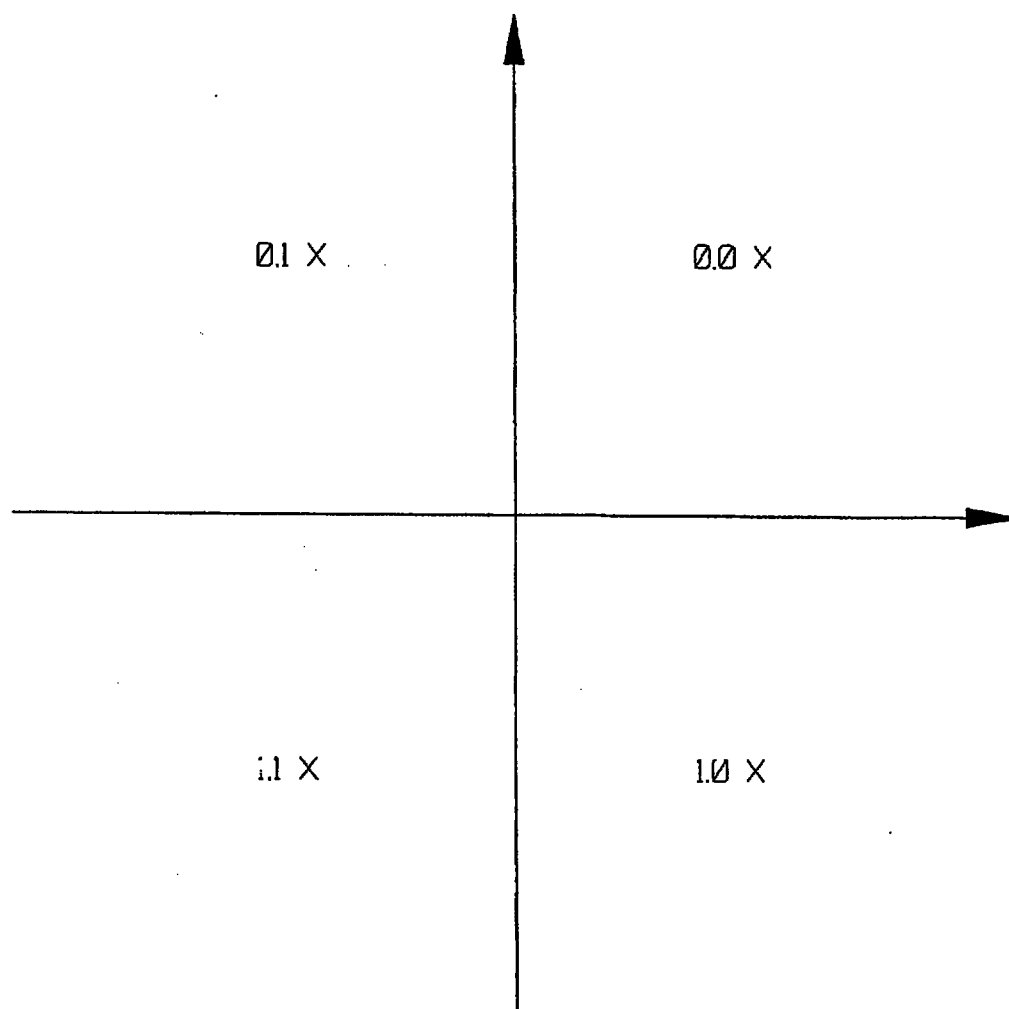


FIG. 1

2/3

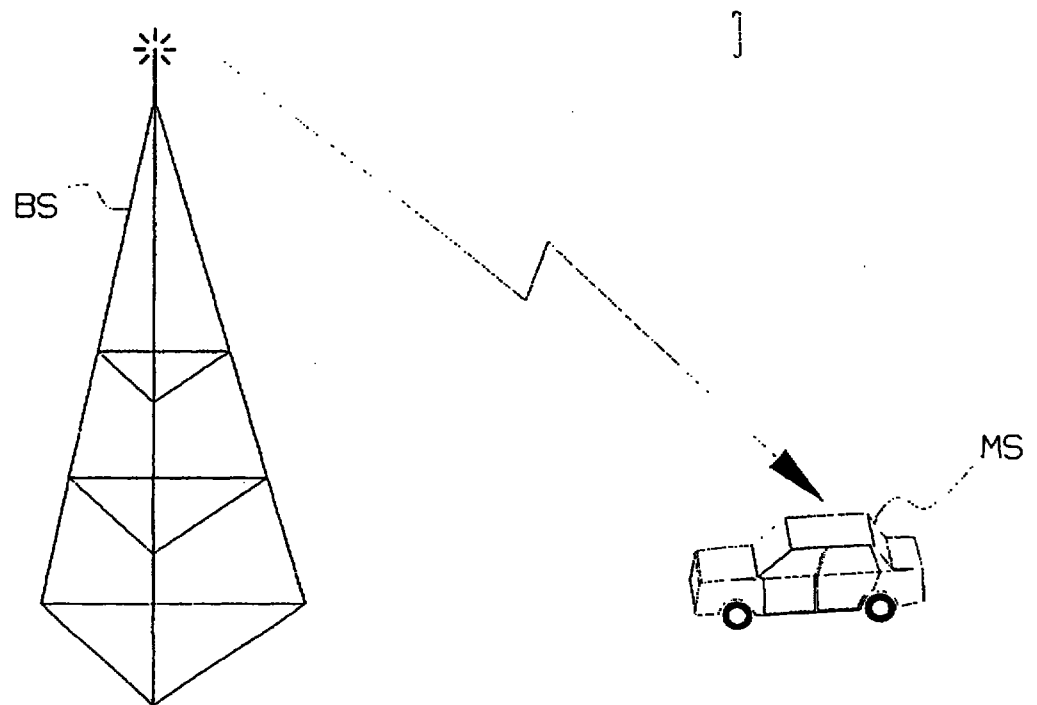


FIG. 2

3/3

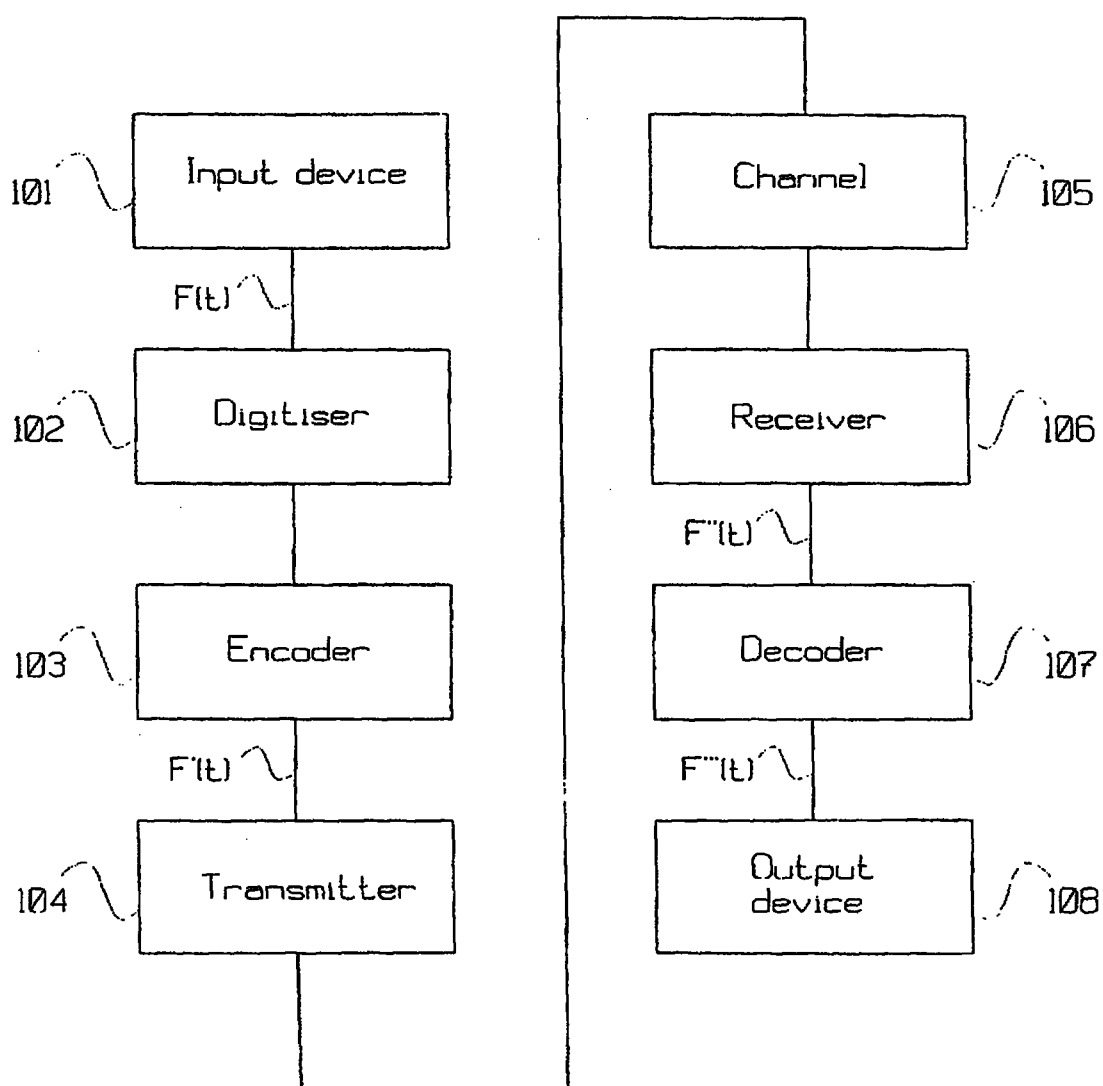


FIG. 3