FILOMAT (Niš) 9:2 (1995), 377–384

Filomat '94, Niš, October 22-24, 1994
Geometry. Computer Sciences

CODING FOR (5,13) CHANNEL CONSTRAINTS

Milan Simić and Rade Petrović

ABSTRACT. *Data Translation codes for the particular channel constraints are designed and presented in this paper. The encoding schemes belongs to the* $RLL(5,k)$ *codes family and can be used in digital recording and telecommunication practise.*

1. Introduction

Runlength limited codes, RLL, are used for digital storage, or as Translation codes for digital data transmission /1/. System for digital transmission can be defined as a system designed to best use a given channel, while the analog communication system is defined as the one designed to best fit a given signal source. Unconstrained data stream must be converted to constrained stream of symbols, (d,k), in order to solve the problems of spectral shaping, self-timing, and intersymbol interference (ISI). Lower bound of zero runs defined by parameter d is used to control ISI, while upper bound defined by parameter k is used to insure data stream selfclocking. Generally, parameters of any translation encoding scheme belong to the range of $0 \leq d < k \leq \infty$. Through the numberof already published papers, we have shown that channels with constraints in the range of $d = 5$ and $12 \leq k \leq 16$, are interesting for the future use in both areas of application. The channel $(5,13)$ have not been especially treated yet, and it is the purpose of this paper. The presented encoding schemes offer a great opportunities in choosing encoding rules, so that RLL codes can be combined with permutation codes, and the signal spectral density can be adjusted. Permutation codes are a class of error correction codes which have been suggested for use on the Gaussian channel.

2. Capacity and Coding Rate Consideration

Based on Schannon's FSM channel model (Finite State Machine), general algorithms are already developed for practical encoding schemes design.

Code designers try to enlarge the parameter d, and to shorten the parameter k, for the same coding rate $R = m/n$, defined as ratio of unconstrained input data symbols number, m, to the number of constrained signal symbols, n, at the coder output. The operation of string translators, named encoders and decoders, is to map input string of symbols from one alphabet into the output string of symbols from the other alphabet. Input string may possibly be of infinite length, but for the practical reasons it is devised into finite strings of fixed or variable lengths, so that we have FL or VL encoding schemes.

Codeword assignment is obviously the function of the incoming dataword, but also it can be dependent on the channel state, presented by the FSM, when we have state dependent coding. Finally, it can be the function of the future dataword, and in that case we have Future dependent coding - FD. Shannon proved that, as codeword length n grows, the number $N(n)$ of (d,k) sequences approaches the value 2^{Cn}, where C is called information capacity. Capacity C can be treated as the theoretically maximum achievable coding rate R for infinite value of codeword length n, according to the equation:

$$C = \lim_{n \to \infty} \frac{1}{n}(log_2 N(n)) \tag{1}$$

It is clear that coding rate R always satisfies inequality $R < C$. The code is called the efficient one if the coding rate R is close to the capacity C. Different (d,k) sequences information capacities are already given in the references, but we did some more calculations based on the solution of the characteristic equation given by $det(A - \lambda I) = O$, where A is FSM state-transition matrix. The calculated capacity C is the capacity of a discrete noiseless channel expressed in units of **bits per channel symbol**, although it can also be calculated in units of **bits per second, bps**. Capacity of the channel in bits per channel symbol differ only by a factor equal to the number of channel symbols per second. Considering channel characteristics there are four concepts related to one another:

Data rate in bps, at which data can be communicated,

Bandwidth of the transmitted signal and the nature of the transmission medium in hertz,

Noise or average level of noise over the communications path,

Error rate, the rate at which errors occur.

For the coding purposes, or alphabet conversion, it is convenient to consider capacity in bits per symbol. Our conclusion was that the codes with coding rate $R = 1/3$ and parameters $(d,k) = (5,k)$ could be of interest in recording, as well as telecommunication practice. Density ratio of these codes, given by $DR = R(d+1) = 2$, is valuable improvement over existing codes.

After the considerations described above, we have found out that it is possible to design new coding schemes defined by parameters $(d, k) = (5, 13)$ and $R = 1/3$, as the $(5, 13)$ constrained sequences information capacity is $C = 0.343 < 1/3 = R$. Clock rate, defined as $CLR = 1/RT$, is increased, $CLR = 3$, and thus compensates information rate loss caused by translation of unconstrained input data sequences to the constrained sequences. The original information bit time interval, which corresponds to NRZ clock signal, is called bit window, and is labelled with T.

3. Encoding Schemes for (5,13) Constraints

- *State and Future Dependent Coding*

Figure.1 illustrates State transition diagram or FSM, for general constraints (d, k). In our particular case when $d = 5$ and $k = 13$, it is a graph with 14 nodes, or channel states, where arrows directed edges represents state transitions, and are labelled with channel bits. In the terminology of synchronous Bounded Delay (BD), or FD coding /2-5/, set S_c is a set of coding, or terminal states, which are the states entered at the end of codewords. Codewords are the paths through the FSM graph.

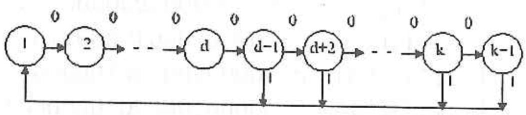

Fig. 1 State transition diagram for the (d, k) sequence

The existence of set $S_c = (S_c)$, as a subset of all FSM states set, $S = (S_i), i = 1, ..., 14$, is a necessary and sufficient condition for the existence of a code. FD RLL(5,13) code can be defined with 26 codewords, the lengths of which vary from 3 to 21 signal symbols, representing 1 to 7 data bits. Coding states set is $Sc = (S_c), c = 1, 2, 4, 5, 6, 7, 8$. The codeword choice is a function of the current state (State Dependent-SD), the information to be represented, and the future information. Code conversion rules are given in Table 1. Data bits in brackets indicate future bits in certain states, and are related to the states S_1, S_2, S_4, and S_5, where we have future dependency.

TABLE 1. Future Dependent RLL(5,13) Code

	Initial State	Input Data	Output Sequences	Final State
1	S_6, S_7, S_8	00	100000	S_6
2		010	010000000	S_8
3		011	001000000	S_7
4		100	000100000	S_6
5		1010	000010000000	S_8
6		1011	000001000000	S_7
7		1100	000000100000	S_6
8		11010	010000010000000	S_8
9		11011	010000001000000	S_7
10		11100	001000001000000	S_7
11		111010	000010000010000000	S_8
12		111011	000010000001000000	S_7
13		111100	000001000001000000	S_7
14		1111010	010000010000010000000	S_8
15		1111011	010000010000001000000	S_7
16		1111100	010000001000001000000	S_7
17		1111101	001000001000001000000	S_7
18		1111110(0)	000010000010000010000	S_5
19		1111110(1)	000010000010000001000	S_4
20		1111111(1)	000001000001000001000	S_4
21		1111111(00)	010000010000010000010	S_2
22		1111111(01)	010000010000010000001	S_1
23	S_1	0	000	S_4
24	S_2	0	000	S_5
25	S_4	1	000	S_7
26	S_5	0	000	S_8

- State Independent Coding

Using the same codepaths in the state transition diagram from Fig.1, state independent SI RLL(5,13) code can be defined with no look-ahead. In this case, coding states set is $S_c = (S_c), c = 6, 7, 8$. Code translation table can consists of 22 or 21 codewords, the lengths of which vary from 6 to 24 signal symbols, representing 2 to 8 data bits. Generally, future dependency can shorten codeword length, but in this case it does not affect the error propagation limiting (EPL), or codec complexity reduction. The next RLL(5,13) code is designed with given coding states set S_c, where the translation rules are defined for 21 VL codewords as presented in the Table2.

TABLE 2. State Independent RLL(5,13) Code

	Input Data	Output Sequences
1	00	100000
2	010	010000000
3	111	001000000
4	100	000100000
5	1010	000010000000
6	0111	000001000000
7	1011	000000100000
8	11001	010000010000000
9	11011	010000001000000
10	11000	001000001000000
11	011001	000010000010000000
12	011011	000010000001000000
13	011000	000001000001000000
14	1101001	010000010000010000000
15	1101011	010000010000001000000
16	1101000	010000001000001000000
17	1101010	001000001000001000000
18	01101001	000010000010000010000000
19	01101011	000010000010000001000000
20	01101000	000010000001000001000000
21	01101010	000001000001000001000000

The problem of EPL is directly related to the appropriate codeword to the data word assignment and the decoder design /6/. Based of that, similar data sequences are coded with similar symbol sequences. The encoder for the New code can be designed as any PAL SM (State Machine) encoder for RLL codes, as for example for RLL(2,7), or RLL(5,16) codes, but we propose sliding window decoder.

The decoder for New RLL(5,13) code has 26 bits shift register, as sliding window, with serial input-parallel output and PLA array architecture for combinatorial logic design. Programmable AND-OR array generates canonical form sum-of-products of the variables involved in a function. Variables are taken from the shift register positions. Sequential decoder generates one output data bit for each incoming 3 symbol bit pattern, after the time delay for 9 symbols, or 3 data bits. There is no internal feedback in the decoder as the output depends only on the input string in length of 26 bits, so that the error propagation is limited to only 9 data bits since $24 < 26 < 27$. If we denote the contents of the shift register positions by $x_i, i = 1, 2, ..., 26$, with shifts from x_1 to x_{26}, the decoded group occupies positions x_{10}, x_{11} and x_{12}, while the past $(x_i, i = 13 - 26)$, as well as, the future $(x_i, i = 1 - 9)$ bit groups, can affect the decoder decision. After a great deal of calculation, since the truth table has 116 rows and 26 columns, it can be shown that the decoder output, $d = f(x_1, ..., x_{26})$, is defined by Boolean expression:

$$d = \sum_{i=1}^{18} p_i$$

$$p_1 = x_7x_{13};\ p_2 = x_7x_{14};\ p_3 = x_8x_{14};\ p_4 = x_9x_{18};\ p_5 = x_9x_{19};$$

(2) $$p_6 = x_9x_{20};\ p_7 = x_{13}\bar{x}_{19};\ p_8 = x_{13}x_{25};\ p_9 = x_{14}\bar{x}_{20};\ p_{10} = x_4x_{11}\bar{x}_{17};$$

$$p_{11} = x_5x_{11}\bar{x}_{17};\ p_{12} = x_{10}\bar{x}_{16}\bar{x}_{17};\ p_{13} = \bar{x}_{10}x_{16}\bar{x}_{22};\ p_{14} = \bar{x}_{10}\bar{x}_{11}x_{17}x_{23};$$

$$p_{15} = \bar{x}_1\bar{x}_2x_8\bar{x}_{15}\bar{x}_{16}\bar{x}_{17};\ p_{16} = x_6\bar{x}_{12}\bar{x}_{13}\bar{x}_{14}\bar{x}_{15}\bar{x}_{16}\bar{x}_{17};$$

$$p_{17} = x_{12}\bar{x}_{18}\bar{x}_{19}\bar{x}_{20}\bar{x}_{21}\bar{x}_{22}\bar{x}_{23};\ p_{18} = x_{15}\bar{x}_{21}\bar{x}_{22}\bar{x}_{23}\bar{x}_{24}\bar{x}_{25}\bar{x}_{26};$$

The truth table used in the evaluation of decoder function is only a part of the whole table with $2^{26} = 67108864$ rows.

- *Improved ACH coding*

Referring to the FSM model, as in any other method, with ACH approach it is possible to derive encoder state transition table in systematic manner, for any channel constraints, if coding is realisable depending on R and C. Recently /7/ the novel method, or improved ACH, was presented. The same approach was used for the following scheme design. Constrained channel is described by 14-by-14 state transition matrix D:

$$D = (d_{ij});\quad i, j = 1, ..., 14$$
$$d_{i1} = 1 \quad \text{for} \quad i \geq (d+1) = 6$$
$$d_{ij} = 1 \quad \text{for} \quad j = i + 1$$
$$d_{ij} = 0 \quad \text{for the other cases}$$

Next step was to derive $B = D^3$ from D, and it should be for our channel the following matrix :

$$B = D^3 = (b_{ij});\quad i, j = 1, ..., 14$$
$$b_{i1} = 1 \quad \text{for} \quad 4 \leq i \leq 12$$
$$b_{i2} = 1 \quad \text{for} \quad 5 \leq i \leq 13$$
$$b_{i3} = 1 \quad \text{for} \quad 6 \leq i \leq 14$$
$$b_{ij} = 1 \quad \text{for} \quad j = i + 3$$
$$b_{ij} = 0 \quad \text{for the other cases}$$

After that we have found vector V with positive integer components such that:

$$D^3V \geq 2V$$

Two optimal, from many possible solutions, are the following:

V1 transposed = (4 5 6 8 10 12 12 11 9 9 7 3 3 3)
$m = v(i)$ for $i = 1$ to $14; m = 102$

V2 transposed = (4 5 6 8 10 12 12 11 9 9 7 3 3 1)
$m = v(i)$ for $i = 1$ to $14; m = 100$

where $v(i)$ are components of the vector V.

The number of encoder states, in the state splitting process, is given by the corresponding component of vector V so that the total number of encoder and decoder states is m. The Encoder matrix E is squared, 100x100 matrix in the other case which is the best one, and we can use Milan approach, with H matrix, to define the encoder /7/.

Since the number of states is $100 \leq 128 = 2^7$ states, the error propagation is limited to only 7 data bits, for this class of codes /7/.

4. Conclusion

Encoding schemes presented in this article are coding problem solutions for (5,13) channel constraints. Error propagation is limited in each case, as more precisely presented in the previous papers for similar codes, and the further analyse can be done in order to adjust signal spectres. In addition to that, RLL codes can be combined with permutation codes to improve the reliability, or Data Rate in the communication channel.

The last one scheme from this paper, can give us more freedom to make the appropriate codewords to datawords choice. Since the presented channel codes are selfclocking, and according to the existing standards they can be used for voice and all other source data transfer, so they are suitable to be data encoding schemes for ISDN, or BISDN via fibre optic media. Finally, the FDDI code is only RLL(0,3) encoding scheme.

References

[1] Richard, E.B., *Digital transmission of information*, Addison Wesley Publishing Company, 1990.

[2] Adler, R.L., Coppersmith, D., Hassner, M., *Algorithms for Sliding Block Codes*, IEEE Trans.Inf. Theory **IT-29, No. 1, January** (1983), 5–22.

[3] Simić, M., Petrović, R., *New RLL code for digital data storage*, Electron. Lett. **25, (15)** (1989), 951–954.

[4] Franaszek, P.A., *Synchronous bounded delay coding for input restricted channels*, IBM J. Res. and Develop. **24, No.1** (January 1980,), 43–48.

[5] Franaszek, P.A., *A general method for channel coding,* IBM J. Res. Develop. **24, No.5** (September 1980), 638–641.
[6] Simić, M., Petrović, R., *New EPL RLL(5,16) code*, Electron.Lett. **27, (23)** (1991), 2100–2102.
[7] Simić, M., *RLL(5,12) Coding,* ETRAN, Niš 1994.

Milan Simić, Faculty of Philosophy, Department of Mathematics, University of Niš, Yugoslavia

Rade Petrović, University of Mississippi, Centre for Telecommunication, PO Box 9031, University, MS, 38677