

# Genetic Algorithm used in Search of good Tailbiting Codes

Piotr Remlein, Dawid Szłapka

Faculty of Electronics and Telecommunications  
Chair of Wireless Communication  
Poznan University of Technology  
Piotrowo 3a; 60-965 Poznan, Poland  
remlein@et.put.poznan.pl

**Abstract:** In a wireless transmission it is necessary to use the channel encoder in order to avoid losses due to transmission errors. This paper presents a method to design the tailbiting codes (TB) for packet data transmission over the additive white gaussian noise (AWGN) channel.

In this paper, genetic algorithms (GA) [1] are used in search of good tailbiting codes. The GA technique is applied in search of good binary tailbiting systematic, rate 1/2 and 2/3 convolutional codes concatenated with 4-PSK and 8-PSK modulation and in search of good non binary rate 1/2 convolutional codes over ring  $Z_4$  concatenated with 16-QAM modulation. The search is based on the maximum free Euclidean distance criterion. It is found that the new approach is very effective especially in case of encoders with large number of inputs or with large number of states. Finally, the search results are presented in the tables.

**Keywords:** tailbiting codes, codes over ring, genetic algorithm

## 1. Introduction

The wireless communication systems use many different encoding techniques for improving the error performance. In the paper [8] it is shown that for transmission of short blocks (under 150 bits) the tailbiting convolutional codes provide a better error performance than the turbo-codes.

In general case the problem of finding the optimal tailbiting codes concatenated with  $m$ -ary modulation has not been solved yet. The problem is especially difficult in case of encoders of large number of inputs or large number of encoder states.

Up to now, in the literature [5], [6], [7] there were published the best tailbiting codes with the greatest minimum Hamming distance. In case of 4-PSK, 8-PSK and 16-QAM signal transmission over the additive white Gaussian noise (AWGN) channel the quality criterion for the best convolutional codes is the Euclidean distance [9]. In this article we assumed the Euclidean distance as quality criterion for the tailbiting codes.

In general, a computer search for optimal codes is very time consuming and is far from being effective. In this paper we propose the computer search for good tailbiting codes based on the genetic algorithm usage.

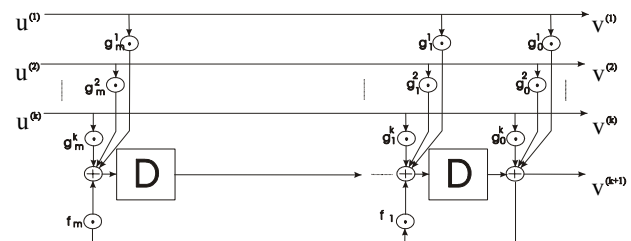
In Section 2, we describe the tailbiting encoder. In Section 3, we present the genetic algorithm. The tailbiting codes search results for 4-PSK, 8-PSK and 16-QAM modulation are presented in Section 4. Finally, some conclusions are drawn in Section 5.

## 2. Tailbiting encoding method

To be able to use a convolutional code (of rate  $R=k/n$  and  $m$  memory elements) in the packet transmission we must convert this code into a block code. In case of tailbiting method we transmit a block of convolutionally encoded data without any tail. We have to start the encoding process in the same encoder state in which the encoding process finishes. The encoder state is not known to the decoder. The decoder only knows that the starting and ending encoder states are the same.

The encoding procedure to achieve this is not difficult if the structure of the encoder is feedforward. In this case the starting state depends on the  $m$  last information symbols in the transmitted packet, where  $m$  is the number of memory cells in the encoder. In case of convolutional encoder with feedback (Fig. 1) the starting state depends on all of the information symbols in the packet. Finding the initial state, wherein the encoder should start the work and after  $N$  symbol intervals end encoding in the same state, is complex. One of methods to find this initial state was proposed in [10] and extended for multilevel codes in [11].

In Fig. 1, we show a realization of the systematic feedback convolutional encoder of rate  $R=k/n$ ,  $n=k+1$ .



**Figure 1.** Systematic feedback convolutional encoder over ring of integers modulo- $M$

Input to the encoder, at time  $t$ , is information vector  $U_t$  with  $M$ -ary elements  $u_t^{(i)}$  belonging to the ring  $Z_M = \{0, 1, 2, \dots, M-1\}$

$$U_t = (u_t^{(1)}, u_t^{(2)}, \dots, u_t^{(k)}) \quad (1)$$

The convolutional encoder produces a coded sequence of the symbols which belong to the same ring  $Z_M$

$$V_t = (v_t^{(1)}, v_t^{(2)}, \dots, v_t^{(n)}), \quad (2)$$

where  $n=k+1$ .

The coefficients of the encoder in Fig. 1 are taken from the set  $\{0, \dots, M-1\}$ . The memory cells are capable of storing the

ring elements. Multipliers and adders perform multiplication and addition respectively in the ring of integers modulo-M.

The encoder output  $V_t$  at time  $t$  due to the input  $U_t$  is:

$$V_t = U_t G, \quad (3)$$

where  $G$  denotes the generator matrix of the encoder [10].

The state of the encoder at time  $t$  is determined by the content of memory elements

$$X_t = (x_t^{(1)}, x_t^{(2)}, \dots, x_t^{(m)})^T, \quad (4)$$

where  $m$  is the number of encoder memory elements.

In case of using the convolutional encoder with feedback to packet transmission without tail we have to calculate the initial state  $X_0$  that is the same as the state  $X_N$  of the encoder after  $N$  cycles. To find the starting state we used the method proposed in [10]. The correct starting state can be calculated using the state space representation. The state of the encoder in time  $t+1$  is:

$$X_{t+1} = AX_t + BU_t^T, \quad (5)$$

where  $A$  is the  $(m \times m)$  state matrix which defines connections between memory elements,  $B$  is the  $(m \times k)$  control matrix which defines connections between encoder inputs and memory elements.

The output encoder vector  $V_t$  at time  $t$  is [10]

$$V_t^T = CX_t + DU_t^T, \quad (6)$$

where:  $C$  is the  $(n \times m)$  observation matrix which defines connections between encoder outputs and memory elements,  $D$  is the  $(n \times k)$  transition matrix which defines connections between encoder entries and outputs.

In the paper [10] it was shown that the state, at time  $t$  of the systematic convolutional encoder with feedback can be described as the superposition of two vectors  $X_t^{[zi]}$  and  $X_t^{[zs]}$  which define the ending state of the encoder

$$X_t = X_t^{[zi]} + X_t^{[zs]}, \quad (7)$$

where  $X_t^{[zi]}$  is the vector which defines the encoder state achieved after  $t$  cycles if the encoding process started in state  $X_0$  and all inputs symbols are zero,  $X_t^{[zs]}$  is the vector which defines the encoder state achieved after  $t$  cycles if the encoding started in the all zero state ( $X_0=0$ ) and the information sequence is encoded.

From the equations (5) and (7) we can write that:

$$X_t = X_t^{[zi]} + X_t^{[zs]} = A^t X_0 + \sum_{\tau=0}^{t-1} A^{(t-1)-\tau} B U_\tau^T \quad (8)$$

If we assume that the encoder state at time  $t=N$  is equal to the initial state  $X_0$ , we obtain from (8):

$$(I_m - A^N) X_0 = X_N^{[zs]}, \quad (9)$$

This equation can be described for convolutional encoders over ring  $Z_M$  as:

$$(I_m + A^N) X_0 = X_N^{[zs]}, \quad (10)$$

where  $I_m$  is the  $(m \times m)$  identity matrix.

As we can see from (10), we can calculate the correct initial state  $X_0$  of the encoder if the matrix  $(I_m + A^N)$  is invertible.

The matrix  $A$  in equation (10) for the systematic convolutional encoder with the feedback is described as [10], [11]:

$$A = \begin{bmatrix} 0 & \dots & 0 & f_m \\ 1 & & & f_{m-1} \\ & \ddots & & \vdots \\ & & 1 & f_1 \end{bmatrix} \quad (11)$$

With (9) and (10) we can describe the encoding process for TB codes as follows: at first we have to calculate the vector  $X_N^{[zs]}$  for a given data information packet. Accordingly, the encoder starts in the all zero state. All of the  $N-k$  information symbols are encoded but the output symbols are ignored. After  $N$  cycles the encoder will be in the state  $X_N^{[zs]}$ . Then we can calculate from (10) the correct initial state  $X_0$  and the encoder can start the proper encoding process and a valid codeword results. After  $N$  cycles the encoder ends its work in the same state as it started.

Following this description we show the example of TB encoding procedure with feedback systematic convolutional encoder over ring  $Z_4$ , modulo-4.

### EXAMPLE

A packet of four symbols is encoded. The symbols belong to the ring  $Z_4$ . The encoder is a systematic convolutional encoder over ring  $Z_4$  with feedback, with code rate  $R=1/2$  and two memory elements  $m=2$ . In Fig. 2, we show the structure of this encoder.

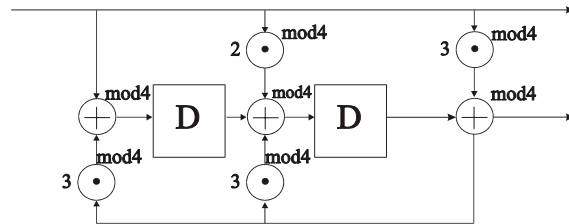


Figure 2. Encoder of the convolutional code

$$G(D) = \begin{bmatrix} 1 & 3 + 2D + D^2 \\ 1 & 1 + 3D + 3D^2 \end{bmatrix}$$

We encode the information block  $U=(U_0, U_1, U_2, U_3) = (1, 0,$

$3, 3)$ . The state matrix is given as  $A = \begin{bmatrix} 0 & 3 \\ 1 & 3 \end{bmatrix}$ . Therefore

$N=4, k=1$ , and from equation (9) we can calculate

$$\left( I_2 - \begin{bmatrix} 0 & 3 \\ 1 & 3 \end{bmatrix}^4 \right) X_0 = X_4^{[zs]}. \quad \text{From this formula we}$$

obtain:  $X_0 = \begin{bmatrix} 2 & 1 \\ 3 & 3 \end{bmatrix} X_4^{[zs]}$ . Therefore we have to calculate

the state  $X_4^{[zs]}$ . From Fig. 3 we can see that this state is equal to  $(3,1)^T$  and the correct state from which we must start

the encoding process is equal to  $X_0 = \begin{bmatrix} 2 & 1 \\ 3 & 3 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$ .

From Fig. 4 we can see that, if we start to encode the sequence  $U$  from state  $(3,0)^T$ , then after  $N=4$  cycles we end in the same state and obtain valid codeword

$V=(13,02,31,30)$ .

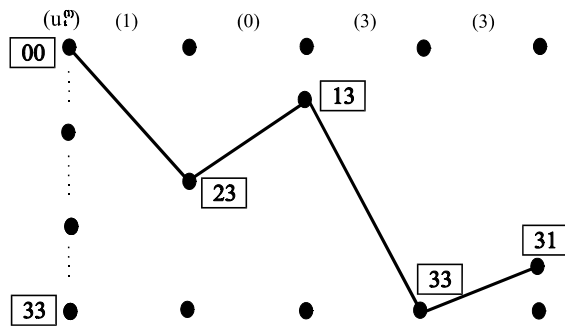


Figure 3. The tree diagram when the zero state response is obtained  $X_4^{[zs]}$

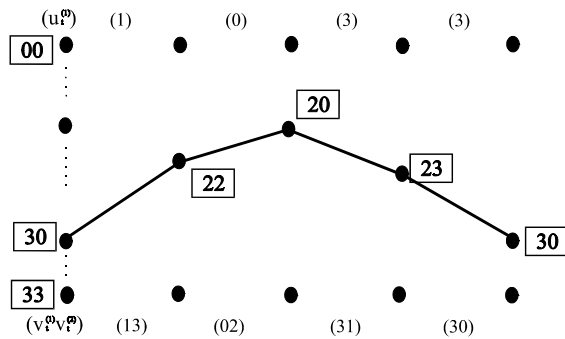


Figure 4. The tree diagram for proper encoding process for tailbiting convolutional codes over ring

### 3. Genetic Algorithm as a search method of good tailbiting codes

The design of the tailbiting codes is usually carried out by a computer search. To find the best tailbiting codes an efficient algorithm is necessary, especially for codes with large number of states.

The Larsen algorithm [3] is known as the most efficient algorithm for computing the minimum Hamming distance of binary convolutional codes. Ungerboeck [2] has used the Larsen algorithm to compute the minimum Euclidean distance of TCM codes by replacing the Hamming weight with the Euclidean weight. Du and Vucetic [4] used the same Larsen algorithm to compute minimum product distance of TCM codes for fading channels. All of these methods are very time consuming. The known search methods provide a systematic search of all possible codes, sometimes they use some rules to reduce the search. The number of all possible tailbiting codes is proportional to the value  $M^{mkN}$ .

The proposed, in this article, method transfers the code design problem to an optimization problem using the genetic algorithm. This method is time efficient and is especially suitable for codes with large number of states.

The genetic algorithm used in this paper utilizes three operations: reproduction, crossover, mutation. The tailbiting encoder from Figure 1. can be represented by the generator matrix  $G(D)$ , where  $D$  is a delay element and the entries of the matrix are polynomials with coefficients belonging to the ring of integers modulo- $M$ .

$$G(D) = \begin{bmatrix} \cdot & g^{(1)}(D) / f(D) \\ \cdot & g^{(2)}(D) / f(D) \\ \cdot & \cdot \\ \cdot & g^{(k)}(D) / f(D) \end{bmatrix} \quad (12)$$

$I_k$  is a  $k$  by  $k$  identity matrix and

$$g^i(D) = g_m^{(i)}D^V + \dots + g_2^{(i)}D^2 + g_1^{(i)}D + g_0^{(i)} \quad (13)$$

and

$$f(D) = f_m D^m + \dots + f_2 D^2 + f_1 D^1 + 1 \quad (14)$$

are the polynomials responsible for the feedforward and feedback connections, shown in Figure 1.

The coefficients of these polynomials determinate the codewords that are used by genetic algorithm. Each such codeword is an  $m$ -ary vector containing encoded coefficients. In the first step of the GA the set of codewords (so called main population) is generated. In the next step of the reproduction, the individual codewords are duplicated. The process depends on the code metrics (minimal Euclidean distance). After this the codewords are combined in the pairs. The point of the crossover is selected at random. The third operation - mutation is very unlikely and can change the value of one symbol in the codeword (Figure 5). After some steps of the GA we can see that the next population contains the codewords of the better metrics. In this way we find the best code which have the maximum value of the minimum Euclidean distance.

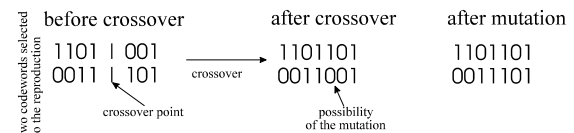


Figure 5. The GA operations

### 4. Search results

In this section we present the results of our computer search for the best tailbiting codes over ring modulo- $M$  ( $M=2$  and  $4$ ) for transmission over AWGN channel. The quality criterion is a minimum Euclidean distance  $d_{e, \min}$ . The codes were generated by the feedback systematic convolutional encoder over ring  $Z_2$  and  $Z_4$ . An object of search in this article were tailbiting convolutional codes over ring  $Z_2$  generated by concatenation of the systematic encoders with feedback with code rate  $R=1/2$  and  $R=2/3$  for 4-PSK and 8-PSK modulators, and tailbiting convolutional codes over ring  $Z_4$  generated by concatenation of the systematic encoders with feedback with code rate  $R=1/2$  for 16-QAM. The encoders found have  $m$  memory cells and the length of the information input symbol sequence equals  $N$ .

We consider two search methods. The first method is a full code search (FS). The second method uses genetic algorithm with population size 80, number of generations 40, crossover probability 0.9, and mutation probability 0.03. These parameters have great effect on GA and need to be selected carefully. For example, a larger number of generations gives a higher chance to find better solution, but

a larger number of generations would also lead to a longer search time. Then, with a larger population size it is possible that the GA algorithm finds a better solution in the smaller number of iterations. However, larger population size may lead to larger search time and additionally the memory usage will also increase. We chose these parameters after many experiments.

The new tailbiting codes over ring found by GA search are marked in tables by the bold characters. For codes over a ring the feedback coefficients  $f_0 \div f_m$  are written as a sequence of decimal numbers and the coefficients in the systematic branches  $g_0^k \div g_m^k$ . The coefficients equal to zero at the beginning of the sequence are skipped in the description. The new TS codes over ring found for 16-QAM are presented in Table 2. We found the best TS codes for encoders with 16, 64 and 256 states. All of these tailbiting codes marked by bold characters are the new codes and were not published yet. Only in three cases, the codes found by the GA method were worse than the codes found by FS method (marked).

It is possible because in FS method we search the best codes in the full set of codes as opposed to the GA method when we search only in the some code subsets. In this case it is possible that we will not find the optimal codes. The search results depend on the proper choice of GA algorithm parameters.

## 5. Conclusion

In the paper, we described how the systematic ring convolutional encoder with feedback can have the same starting and ending state.

We presented the search results of the best tailbiting convolutional codes over ring  $Z_2$  and  $Z_4$  for the transmission over AWGN channel. As a criterion of the optimization we took the Euclidean distance.

Tables of the best new tailbiting convolutional codes over ring  $Z_2$  and  $Z_4$  with rates  $R=1/2$  and  $R=2/3$  for 4-PSK, 8-PSK and 16-QAM modulation were obtained by computer search using genetic algorithm. All TB codes shown in tables and marked by bold characters were not presented in the literature the authors are aware of. The usefulness of the codes one ought to verify by means of computer simulations.

## References

- [1] D.E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley Publishing Company, Inc., 1989.
- [2] G. Ungerboeck, "Channel coding with multi-level/phase signals," IEEE Trans., IT-28, pp. 55-67, Jan. 1982.
- [3] K.J. Larsen, "Comments of an efficient algorithm for computing free distance," IEEE Trans., IT-19, pp. 577-579, July 1973.
- [4] F. Du, B. Vucetic, "Algorithm for computing minimum product distance of TCM codes," Electronics Letters, Vol. 28, No. 1, pp. 2-4, January 1992.
- [5] P. Ståhl, J.B. Anderson, R. Johannesson, "A note on tailbiting codes and their feedback encoders," IEEE Trans. Inform. Theory, Vol. 48, pp.529-534, February 2002.

- [6] H.H. Ma, J.K. Wolf, "On Tailbiting Convolutional Codes," IEEE Trans. Commun., Vol. 34, pp. 104-111, Feb. 1986.
- [7] I.E. Bocharova, R. Johannesson, B.D. Kudryashov, P. Ståhl, "Tail-biting codes: Bounds and search results," IEEE Trans. Inform. Theory, Vol. 48, pp.597-610, Apr. 2000.
- [8] S. Crozier, A. Hunt, K. Gracie, J. Lodge, "Performance and complexity comparison of block turbo-codes, hyper-codes and tail-biting convolutional codes," 19-th, Biennial Symposium on Communications, Kingston Ontario, Canada, pp. 84-88, May 31-June 3, 1998.
- [9] A. Dholakia "Introduction to Convolutional Codes with Applications," Kluwer Academic Publishers, 1994.
- [10] C. Weiß, C. Bettstetter, "Code Construction and Decoding of Parallel Concatenated Tail-Biting Codes," IEEE Trans. Inform. Theory, Vol. 47, pp.366-386, January 2001.
- [11] P. Remlein, "The Encoders with the feedback for the packed transmission without tail symbols," VIII-th Poznan Workshop on Telecommunication, PWT '03, Poznan, pp 165-169, (in polish), 11-12 December 2003.

m	4 FS GA				5 FS GA				6 FS GA			
	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	Q,g	d <sub>e,max</sub>
4	70,40	4,82	54,2	4,82	70,40	4,82	46,7	4,82	700,400	4,82	520,604	4,82
5	70,40	4,82	54,70	4,82	70,40	4,82	75,55	4,82	700,400	4,82	610,664	4,82
6	54,60	5,64	54,34	5,64	54,60	5,64	76,35	5,64	540,600	5,64	450,60	5,64
7	52,20	5,64	62,26	5,64	75,40	5,64	62,10	5,64	750,400	5,64	670,240	5,64
8	54,60	6,23	76,72	6,23	54,60	6,23	73,6	6,23	540,600	6,23	424,720	6,23
9	64,70	6,82	54,70	6,82	75,60	6,82	75,16	6,82	750,600	6,82	570,74	6,82
10	54,22	7,05	64,12	7,05	75,60	7,64	75,30	7,64	664,400	7,64	714,310	7,64
11	76,64	7,64	54,36	7,64	75,44	8,23	70,65	8,23	750,440	8,23	744,754	8,23
12	64,36	8,00	64,36	8,00	67,62	8,46	67,62	8,46	414,730	9,64	414,730	9,64
13	46,72	8,23	64,36	8,23	67,67	8,82	46,67	8,46	564,600	8,82	730,350	8,82

m	7 FS GA				8 GA		9 GA		10 GA		11 GA	
	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>
4	700,400	4,82	554,564	4,82	545,75	4,82	5414,5174	4,82	6266,5034	4,82	5370,7416	4,82
5	700,400	4,82	772,364	4,82	634,526	4,82	7420,5064	7,82	6130,2110	4,82	5346,3576	4,82
6	540,600	5,64	534,474	5,64	741,106	5,64	5114,6264	5,64	7216,4434	5,64	5456,4232	5,64
7	750,400	5,64	736,762	5,64	450,701	5,64	6034,7710	5,64	6520,1470	5,64	5604,5665	5,64
8	540,600	6,23	712,746	6,23	501,342	6,23	5364,6524	6,23	5634,2414	6,23	4251,2351	6,23
9	750,600	6,82	712,42	6,82	771,511	6,82	4500,5044	6,82	7126,3040	6,82	7512,500	6,82
10	664,400	7,64	406,714	7,64	717,545	7,64	7154,7560	7,64	6200,6146	7,64	7524,4327	7,64
11	662,600	8,23	422,262	8,23	414,116	8,23	5460,7144	8,23	4202,1222	8,23	6155,7545	8,23
12	436,620	9,64	510,256	8,82	510,256	8,82	7300,3010	9,64	5430,3662	9,64	5034,7254	8,46
13	564,600	8,82	742,164	8,82	465,121	8,46	7044,6030	8,82	4416,2416	8,46	7653,367	8,46
14	436,750	9,64	742,136	9,64	760,673	9,46	7300,714	9,64	7660,6034	9,64	5205,2312	9,64
15	626,700	9,64	570,726	9,64	455,310	9,64	7104,1434	9,64	7102,574	9,64	6531,766	9,64
16	676,344	9,87	442,772	9,64	765,540	9,64	5034,3660	9,87	4542,3344	9,87	6412,1161	9,64
17	436,260	10,64	474,466	10,64	646,31	10,64	6214,1320	9,87	4024,1676	10,23	7371,1336	10,46
18			436,552	10,64	714,651	10,64	6534,620	10,46			7452,6250	10,46
19			520,346	10,64	510,773	11,05	5020,3644	11,05			7030,5142	11,28
20			742,722	10,82	562,37	11,28	6534,2644	11,28				
21			652,362	11,64								

Table I. Tailbiting codes over ring  $Z_2$  with code rate  $R=1/2$  for 4-PSK modulation

m	2 FS GA				3 FS GA			
	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>
4	130,100	12,000	113,322	12,000	1300,1000	12,000	1323,32	12,000
5	113,210	14,128	113,210	14,128	1130,2100	14,128	1130,120	14,128
6	102,111	14,141	102,313	13,683	1121,1100	14,828	1123,33	14,828
7	111,123	16,944	111,123	16,944			1222,1131	16,969
8	111,221	16,970	111,322	16,970				

m	4 FS GA				5 FS GA			
	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>	q,g	d <sub>e,max</sub>
4	11100,10000	12,000	10002,3112	12,000	111000,100000	12,000	133132,20232	12,000
5	10132,10000	14,128	11302,1031	14,128	123100,100000	14,128	113002,112203	14,128
6	12330,11000	14,828	11232,21232	14,828			102310,30330	14,828
7	11331,20110	16,970	11201,11011	16,970			111013,322100	16,970
8			10332,2320	18,129			130012,3312	17,888

Table II. Tailbiting codes over ring  $Z_4$  with code rate  $R=1/2$  for 16-QAM modulation

M	4				5				6			
	FS		GA		FS		GA		FS		GA	
N	$G_1, g_2, q$	$d_{e\_max}$	$g_1, g_2, q$	$d_{e\_max}$	$g_1, g_2, q$	$d_{e\_max}$	$g_1, g_2, q$	$d_{e\_max}$	$g_1, g_2, q$	$d_{e\_max}$	$g_1, g_2, q$	$d_{e\_max}$
8	40,20,70	2,61	10,32,76	2,61	40,20,70	2,61	23,62,57	2,61	400,200,700	2,61	124,564,510	2,61
10	40,30,52	2,93	2,44,52	2,93	40,24,70	2,93	51,77,62	2,93	400,300,520	2,93	70,20,744	2,93
12	70,60,64	3,04	70,24,64	3,04	70,60,50	3,04	34,66,54	3,04	700,600,640	3,04	304,24,570	3,04
14	60,40,62	3,04	2,24,70	3,04	60,40,62	3,04	61,50,45	3,04			310,44,744	3,04
16	72,30,76	3,69	72,32,76	3,69			11,67,57	3,69			560,404,510	3,58

m	7		8		9		10		11	
	GA		GA		GA		GA		GA	
N	$G_1, g_2, q$	$d_{e\_max}$	$g_1, g_2, q$	$d_{e\_max}$	$g_1, g_2, q$	$d_{e\_max}$	$g_1, g_2, q$	$d_{e\_max}$	$g_1, g_2, q$	$d_{e\_max}$
8	174,76,422	2,61	730,224,733	2,61	3670,230,6654	2,61	3654,554,5160	2,61	4704,7143,7270	2,61
10	266,374,422	2,93	422,12,406	2,93	4350,7020,5510	2,93	7024,3440,6562	2,93	160,6307,4360	2,93
12	256,140,604	2,93	117,240,675	3,04	7770,7014,6160	2,93	2522,530,5326	3,04	465,1744,4536	2,93
14	76,624,724	3,04	136,132,573	3,04	64,4564,4030	3,04	250,3146,4746	3,04	302,3355,6474	3,04
16	514,662,742	3,69	225,776,411	3,69	7724,6554,7500	3,58	1352,274,7644	3,69	1554,1404,4773	3,69
18	34,320,576	4,02			5640,4630,4420	4,13	340,3354,5700	4,02	4311,2250,5625	4,02
20	400,260,766	4,13			4234,2154,6130	4,13	5516,2126,5160	4,13	3712,236,5016	4,13
22	562,12,460	4,28								

Table III. Tailbiting codes over ring  $Z_2$  with code rate  $R=2/3$  for 8-PSK modulation