Determination of the best CRC codes with up to 10-bit redundancy

Tsonka S. Baicheva*

Abstract

All binary polynomials of degree up to 10 which are suitable to be used as generator polynomials of CRC codes are classified and all the necessary data for the evaluation of the error control performance of the CRC codes generated by the classified polynomials is calculated. A procedure, based on the computed data, for choosing the best CRC code is suggested.

Index Terms: CRC codes, undetected error probability performance

I Introduction

Cyclic codes form an important class of linear codes. They are interesting from theoretical point of view because of their rich algebraic structure and from practical one because they have very good error detecting capabilities and fast encoding and decoding implementation.

Very often in the applications one and the same cyclic code is used to protect different numbers of information symbols. In this case, shortened cyclic codes are used. Given an $[n, k]$ cyclic code C consider the set of code vectors for which the l leading high-order information digits are identical to zero. There are 2^{k-l} such code vectors and they form a linear subcode of C. If the l zero information digits are deleted from each of these code vectors, we obtain a set of 2^{k-l} vectors of length $n-l$. These 2^{k-l} shortened vectors form an $[n-l, k-l]$ linear code. This code is not cyclic, but has at least the same error-correcting capability as the code from which it is derived. The encoding and decoding of a shortened cyclic code can be accomplished in the same way as the original cyclic code. This is so because the deleted l leading zero information digits do not affect the parity-check. Shortened cyclic codes have extensive use, particularly in

*The author is with the Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, P.O.Box 323, 5000 Veliko Tarnovo, Bulgaria, e-mail: tsonka@moi.math.bas.bg

computer communications, under the label *Cyclic Redundancy Check (CRC)* codes. CRC codes are generally not cyclic, as noted above, but they are derived from cyclic codes and hence the name.

Because of the advantage to keep their good error-correcting performance and fast encoding and decoding for all possible shortenings, many CRC codes have been adopted as standards. Then it is usual in the practice a standardized CRC code to be selected with the hope that it is a good one. Unfortunately, many of the investigations (see for instance [1] - [11]) of standardized CRC codes show that some applications use CRCs that have far from the best for the given number of parity bits error detection performance. All these investigations, however, suggest best polynomials for given code lengths and bit error probabilities of the channel, but do not give the complete data which to allow the designers of the communication systems to compare the polynomials themselves and to choose the best polynomial for any particular application. Traditionally CRC codes are mostly implemented in error detecting procedures, but in some applications they are also used in combined error detection and error correction schemes. In such a case we need to evaluate the error correction performance of the codes too, in order to choose the best one for the application. Therefore complete investigations of all CRC codes with given redundancy will help designers in selecting the most effective one for the particular application.

In order to supply all the necessary information for comparing the error control performance of CRC codes with up to 10-bit redundancy, all polynomials suitable to be generator polynomials of such codes have been investigated in this work. These polynomials, omitting the reciprocal (they generate equivalent codes), have been listed and their orders have been determined in section III. In section IV, minimum distances, number of codewords of minimum weight, weight enumerators of the dual codes and weight enumerators of the coset leaders for all codes and for all its shortenings have been computed. Availability of these data allows computing and comparing the error detection or correction performance of the codes in a linear time. In order to avoid comparison between all polynomials of a given degree, a four step procedure for choosing the best one only by comparing a few of the best candidates has been suggested. In section V the performances of repeated CRC codes have been discussed.

II Some preliminary results

To present the way in which cyclic codes operate, it is useful to work in a framework in which binary information is represented as a polynomial. A CRC code of p check bits is defined in terms of a *generator polynomial* of degree p, whose leading and zero coefficients are nonzero. Let us juxtapose polynomials $i(x) = i_0 + i_1x + \ldots + i_{k-1}x^{k-1}$ to the message to be transmitted and $r(x) = r_0 + r_1x + \ldots + r_{p-1}x^{p-1}$ to the remainder obtained by dividing $x^p i(x)$ by $g(x)$. Then the codeword is $c(x) = x^p i(x) + r(x) = q(x)p(x) + r(x) + r(x) = q(x)p(x)$. Therefore, all codewords of the code are divisible by $g(x)$ and all polynomials divisible by $g(x)$ are code polynomials. According to this coding all the information symbols are before the check symbols in the codeword, thus, we have systematic encoding.

We will note that $g(x)$ is a polynomial dividing $x^{n_c} + 1$, where $n_c = min\{m|x^m \equiv 1 \bmod g(x)\}$. The number n_c is called *order of the polynomial* $g(x)$. So, we consider the binary cyclic $[n_c, n_c - p]$ code D generated by the polynomial $g(x)$ of degree p, where $n_c = x^m + 1$.

In practice, we often use some $[n, n-p]$ subcode C (a CRC code) of the code D obtained by shortening D in $n_c - n$ positions. Error detection at the receiving end is made by computing the check bits from the received information block i' and comparing these with the received check bits r'. It is possible, however, the transmission errors to change the codeword in such a way that they cannot be detected by this procedure, i.e. to obtain a codeword different from the sent one. As the code is linear, this may happen only when the error vector is also a codeword. Therefore, the error detection performance of a CRC code with weight enumerator $\{A_i\}$ and weight enumerator of the dual code $\{B_i\}$ used for communication through a binary symmetric channel with bit error rate ε is measured by the probability P_{ud} of undetectable channel errors occurring:

$$P_{ue} = \sum_{i=1}^{n} A_i \varepsilon^i (1-\varepsilon)^{n-i} = 2^{-p}[1 + \sum_{i=1}^{n} B_i (1-2\varepsilon)^i] - (1-\varepsilon)^n.$$

CRC codes are mostly used for error detection, but CRC codes with minimum distances greater than 2 can also be used for error correction. In this case we have to evaluate the error correction performance of the CRC code. This performance, when the decoding to the nearest codeword is applied, depends on the

weight enumerator of the coset leaders $\{\alpha_i\}$ of the code and is given by the probability of correct decoding

$$P_{corr} = \sum_{i=0}^{n} \alpha_i \varepsilon^i (1-\varepsilon)^{n-i}.$$

Thus to be able to calculate values of P_{ue} and P_{corr} we need to know the weight enumerator of the code $\{A_i\}$ or of its dual $\{B_i\}$ and the weight enumerator of the coset leaders of the code $\{\alpha_i\}$.

III CRC polynomials of degree up to 10

We will recall, that a binary polynomial must have at least two nonzero terms (x^p and 1) to be suitable for a generator polynomial of a CRC code with p redundancy bits. Therefore we have 2^{p-1} polynomials for each p. Some of them are reciprocal, which means that they generate equivalent codes and all shortenings of the codes are also equivalent codes. That is why we do not consider both polynomials but take only one of them. This way we form the list of all polynomials suitable to be generator polynomials of a CRC code with up to 10 redundancy. The list also includes factorization of each polynomial and its order. Factorization and computation of the order of polynomials was realized by a programme written in the programming language Maple and carried out by the software package. The programme is based on the following two facts from the theory of finite fields.

Theorem 1: [[12], Corollary 3.4] If $g \in GF(q)[x]$ is an irreducible polynomial over $GF(q)$ of degree p, then $ord(g)$ divides $q^p - 1$.

Theorem 2: [[12], Theorem 3.11] Let $GF(q)$ be the Galois field of q elements, q a power of a prime p, and let $f \in GF(q)[x]$ be a polynomial of positive degree and with $f(0) \neq 0$. Let $f = a f_1^{b_1} f_2^{b_2} \dots f_k^{b_k}$ and $f_1, \dots, f_k$ be distinct monic irreducible polynomials in $GF(q)[x]$, where $a \in F_q$, $b_1, \dots, b_k \in N$ is the canonical factorization of f in $GF(q)[x]$. Then $ord(f) = ep^t$, where e is the least common multiple of $ord(f_1), \dots, ord(f_k)$ and t is the smallest integer with $p^t \geq max(b_1, \dots, b_k)$.

As the full list of polynomials is too long (it contains 550 entries) as an example we present in the table below the list of the polynomials of degree up to 5. (The full list as well as all the data computed in this work are available at `http://www.moi.math.bas.bg/~tsonka`). Factors of the polynomials are presented by the numbers they occur in the table instead of the factors themselves. The sign ' after a number means

that the factor is reciprocal to the polynomial with this number.

Table. CRC polynomials of degree up to 5

N	Nickname	Polynomial	Factorization	Order
1	CRC-1	$x+1$		1
2		x^2+x+1		3
3		x^3+x^2+1		7
4		x^3+x^2+x+1	1^3	4
5	CCITT-4	x^4+x+1		15
6		x^4+x^2+x+1	1.3	7
7		x^4+x^3+x+1	$1^2.2$	6
8		x^4+x^2+1	2^2	6
9		$x^4+x^3+x^2+x+1$		5
10		$x^5+x^4+x^3+x^2+1$		31
11		$x^5+x^4+x^2+x+1$		31
12	USB-5	x^5+x^2+1		31
13		x^5+x+1	2.3	21
14	CCITT-5	x^5+x^3+x+1	$1.5'$	15
15		$x^5+x^4+x^3+1$	$1^2.3$	14
16		$x^5+x^3+x^2+1$	$1^3.2$	12
17		x^5+x^4+x+1	1^5	8
18		$x^5+x^4+x^3+x^2+x+1$	1.2^2	6

Then minimum distances, number of the codewords of minimum weight, weight enumerators of the dual codes and weight enumerators of the coset leaders for all CRC codes (full length and all its shortenings) from the list have been computed.

In order to determine the weight enumerator of an $[n,k]$ code or of its dual we have to generate all 2^k (2^{n-k} respectively) codewords. As the codes we have investigated have relatively small co-dimensions $n-k=p\leq 10$, we computed the weight enumerators of their dual codes $\{B_i\}$ that is a less time consuming operation. Then minimum distances and the number of the codewords of minimum weight A_2 were computed in a linear time via the MacWilliams' identities.

It has been proved in [13] that the determination of the covering radius R of a code is an NP complete problem. Since by definition, the covering radius of the code is the weight of the heaviest coset leader, we can conclude that the computation of the weight spectrum of the coset leaders $\{\alpha_i\}$ is also a very hard computational problem. To determine $\{\alpha_i\}$ we used the fact ([14, Theorem 2.1.9]) that if $H=(h_1,h_2,...,h_n)$ is a parity check matrix of the code C, then the covering radius of the code $R(C)$ is the smallest integer ρ such that every nonzero $n-k$-dimensional vector is a linear combination of not more that ρ columns of H.

Therefore, we have to make all combinations of $\rho = \lfloor (d-1)/2 \rfloor + 1, \ldots, R$ columns of H and for each value of ρ to count the different $n-k=p$-dimensional vectors which have been obtained. We will note that it is not necessary to make these computations for the values of ρ between 1 and $\lfloor (d-1)/2 \rfloor$, because it is well known that every p-dimensional vector of weight $1, \ldots, \lfloor (d-1)/2 \rfloor$ is a unique leader of the coset. Thus the number of the steps required to determine $\{\alpha_i\}$ is proportional to at most $\sum_{i=\lfloor (d-1)/2 \rfloor + 1}^{R} \binom{n}{i} 2^i$. All the computations about $\{B_i\}$, A_2, $\{\alpha_i\}$ and minimum distance have been done with programmes written in C by the author. In order to check the obtained results, some of them were recalculated by the computer package Q-EXTENTION [15].

IV Procedure for polynomial selection

We propose a four step procedure for selecting an appropriate candidate for a CRC polynomial for a particular application. This procedure is simple and very easy to implement and calculations can be done in a linear time thanks to the data (minimum distances, weight enumerators) computed in this investigation.

An important role in the procedure plays the order of the polynomial. This characteristic of the polynomial is not always considered in the investigations, but it gives an important information about the minimum distance performance of the CRC code. Let us suppose that $c(x) = q(x)g(x)$ is a codeword of a CRC code with a generator polynomial $g(x)$ of order n_c. As $g(x)$ has at least two nonzero coefficients, $c(x)$ also has at least two nonzero coefficients. Let us now suppose that $c(x) = x^i + x^j,\ i > j$. Thus $c(x) = q(x)g(x) = x^j(x^{i-j}+1)$ and $q'(x)g(x) = (x^{i-j}+1)$. This is impossible as $i - j < n_c$ and the smallest number m for which $x^m + 1 \equiv 0 \bmod g(x)$ is $ord\ g(x) = n_c$. Therefore every CRC code with a generator polynomial not divisible by x has minimum distance at least 3.

The steps of the procedure are the following:

1. The first limit which every application must confirm is the number of check symbols to be added to a data word in order to protect them. This number determines the degree p of the polynomials which will be considered.

2. Among all polynomials of the fixed degree p we choose only these with order greater or equal to the maximum length n at which the code will operate. If orders of all the polynomials are less than n we choose those of them with maximum order.

3. From the set of polynomials formed in the previous step, we consider the polynomials whose minimum distance is maximal. If there are too many such polynomials we select only those of them having the smallest number of codewords of minimum weight.

4. We compute and compare the values of P_{ue} for the selected codes and for the particular value(s) of the bit error probability ε at which the code will operate and choose the one with the smallest value. Additionally, if the code will be used in a combined error detection and error correction scheme we compare the probabilities of correct decoding P_{corr} of the codes and choose the one with the biggest value.

V Codes with lengths greater than the order of the generator polynomial

Many communication protocols impose no restrictions on the length of the protected message, i.e. the encoding algorithm of the CRC code is applied to a message of length many times greater than $n_c - p$, where n_c is the order of the CRC polynomial. Such an example is CCITT-5 (the polynomial with number 14 from the Table) where data message of length 3151 is protected by 5 check bits (ITU-T Recommendation G.704). It is clear, that there is no polynomial of degree 5 and order 3156, which is the necessary code length. In such a case we use a repeated version of the original code and will call this code *repeated* CRC code. The encoding and decoding procedures for these codes are the same, but its minimum distance is 2.

Let a message $i(x)$ of length $m > n_c - p$ be protected by p CRC bits. Encoding of this message word leads to a codeword $c(x) = x^p i(x) + r(x)$ of length $n = m + p$, where $r(x)$ is the residue modulo $g(x)$. Since $x^{n_c} + 1 \equiv 0 \bmod g(x)$, then

$$x^a \equiv x^b \bmod g(x),$$

where x^b is the residue of a modulo n_c, and

$$x^c + x^d \equiv 0 \bmod g(x),$$

where $c - d \equiv 0 \bmod n_c$. Therefore some messages of weight one or two lead to codewords of weight two that essentially worsens the error control performance of the code. The number of codewords of weight two

can be easily counted.

Proposition 1: Let $n = qn_c + r,\ 0 \leq r \leq n_c$. The number of codewords of weight two is given by

$$A_2 = \frac{q(n - n_c + r)}{2}.$$

Proof. The codeword can be divided into q blocks of n_c bits and one block of r bits, i.e.

$$[n_c][n_c] \dots [n_c][r].$$

The nonzero bits of any codeword of weight two are placed as follows:

a) One in the last block of r bits and one in the position with the same number in one of the first q blocks. Thus there are qr such vectors.

b) At positions with the same numbers in two of the first q blocks. Hence there are $\binom{q}{2} n_c$ such vectors.

Therefore

$$A_2 = qr + \frac{q(q-1)}{2} n_c = \frac{q(n - n_c + r)}{2}.$$

Remark. The number of the codewords of weight 2 in a repeated CRC code depends on the length of the code and the order of the generator polynomial. The bigger the order of the generator polynomial is, the smaller the number of the codewords of weight 2. Therefore it can be expected that better performance as repeated CRC codes will have CRC codes with the bigger orders.

We will note that when we would like to compare the P_{ue} performances of repeated CRC codes we can get rather precise information if we compute only the first addends of the expression for P_{ue}, i.e. $A_2\varepsilon^2(1-\varepsilon)^{n-2}$. The two graphics on Fig. 1 are a confirmation of our assertion. On the figure the undetected error probability performance of two standardized CRC codes CCITT-5 and USB-5 used at code length 3156 are plotted. On the left one only the first addends from the expression for P_{ue}, i.e. the number of the codewords of weight 2, were used to evaluate it. On the graphic on the right P_{ue} is computed via the first few nonzero elements of the weight enumerators of the codes. Namely A_2, A_4 for CCITT-5 and A_2, A_3, A_4 for USB-5 and the numbers are taken from [8]. It is clear, that though the comparison in the right graphic is more precise, the trend of indisputable better performance of USB-5 in comparison with CCITT-5 can be also seen from the left one.

To evaluate more precisely the probability of the undetected error of a long repeated code, where computing of the complete weight enumerator is an infeasible work even for the dual code, we could only use the first few elements of the weight enumerator instead of the whole one. We will get an acceptable accuracy, since the first addends in the expression for P_{ue} are most significant. Hence, we have to compute the number of the codewords of weights 3, 4, 5, ... To do this we have to generate all $\binom{n}{3}$, $\binom{n}{4}$, $\binom{n}{5}$, ... binary n-dimensional vectors (here n is the code length) of weights 3, 4, 5, ... respectively. Dividing by the generator polynomial (or equivalently multiplying by the parity-check matrix) we check for each of these vectors whether it is a codeword or not. This is a time consuming procedure at long code lengths, but a feasible work if we do it only for a few small weights. We can speed up this calculation by taking into consideration the following fact: if the basic cyclic code of length n_c has only codewords of even weight, then all its repetitions will have only even weight codewords. Therefore, for such codes, we have to check only even weight vectors in the above procedure.

Example. As an illustration of how we can use the results obtained in this work we will present a comparison (cf. Fig. 2) between a standardized code and not standardized ones. $x^5 + x^2 + 1$ (line 12 from the table) is the USB-5 polynomial used for error detection for Universal Serial Bus (USB) tokens to protect data words of length 11 bits. Therefore the length of the code must be $n = 16$ and we can see from the Table that the first four polynomials of degree 5 are suitable (their orders are greater than 16) to be used as generator polynomials of such a code. All of them have minimum distance 3 at this length and we look for the number of codewords of minimum weight. Codes with numbers 10 and 12 have $A_3 = 19$ and 11 and 13 $A_3 = 20$. In addition, at length 16 codes 10,12 and 11,13 have the same weight enumerators of the dual codes respectively. Thus we can compare only codes 11 and 12 (USB-5). The result of the comparison is presented on Fig. 2 and the superior performance of USB-5 for almost all bit error rates can be seen. Therefore USB-5 is optimal when used to protect 11 message bits, i.e. code length $n = 16$. Code with number 10 have the same as USB-5 weight enumerator of the dual code and coset leaders weight enumerator for $n = 16$. Thus these two codes have completely the same error control performance at length 16.

VI Conclusions

In this work all the necessary data for the evaluation of the error control performance of CRC codes of up to 10-bit redundancy is calculated. A fast and easy to implement procedure for choosing the best CRC code is proposed. Codes of lengths greater than the order of the generator polynomial are considered and formula for the determination of the number of the codewords of weight two is derived. We believe that the results obtained in this work will help designers of communication systems in selecting the most effective polynomial for any particular application.

References

[1] T. Baicheva, S. Dodunekov and P. Kazakov, "On the cyclic redundancy-check codes with 8-bit redundancy," *Computer Commun.*, vol.21, No 11, pp. 1030-1033, August, 1988.

[2] T. Baicheva, S. Dodunekov and P. Kazakov, "Undetected error probability performance of cyclic redundancy-check codes of 16-bit redundancy," *IEE Proc. Commun.*, vol. 147, No 5, pp. 253-256, October, 2000.

[3] G. Castagnoly, J. Ganz and P. Graber, "Optimum cyclic redundancy-check codes with 16-bit redundancy," *IEEE Trans. Commun.*, vol.38, pp. 111-114, January, 1990.

[4] G. Castagnoly, S. Bräuer and M. Herrmann, "Optimization of cyclic redundancy-check codes with 24 and 32 parity bits," *IEEE Trans. Commun.*, vol.41, pp. 883-892, June, 1993.

[5] T. Fujiwara, T. Kasami, A. Kitai and S. Lin, "On the undetected error probability for shorthened Hamming codes," *IEEE Trans. Commun.*, vol.33, pp. 570-574, June, 1985.

[6] G. Funk, "Determination of best shortened linear codes," *IEEE Trans. Commun.*, vol.44, pp. 1-6, January, 1996.

[7] P.Kazakov, "Fast Calculation on the Number of Minimum Weight Words of CRC Codes," *IEEE Trans. Inf. Theory*, vol. 49, pp. 1190-1195, March 2001.

[8] P. Koopman and T. Chakravarty, "Cyclic Redundancy Code (CRC) polynomial selection for embedded networks," in *Proc. of Internat. Conf. on Dependable Systems and Networks, DSN04* , 2004, pp. 145-154.

[9] P. Merkey and E. C. Posner, "Optimum cyclic redundacy codes for noisy channels," *IEEE Trans. Inf. Theory*, vol. 30, pp. 865-867, November, 1984.

[10] J. Ray and P. Koopman, "Efficient high hamming distance CRCs for embedded networks," in *Proc. of Dependable Systems and Networks*, 2006, Philadelphia PA, June 25-28.

[11] K.A. Witzke and C. Leung, "A comparison of some error detecting CRC code standarads," *IEEE Trans. Commun.*, vol.33, pp. 996-998, September, 1985.

[12] R. Lidl and H. Niederreiter, *Introduction to finite fields and their application,* Rev. Edition, Cambridge University Press, 1994.

[13] A. McLoughlin, "The complexity of computing the covering radius of a code," *IEEE Trans. Inf. Theory*, vol. 30, pp. 800-804, 1984.

[14] G. Cohen, I. Honkala, S. Litsyn and A. Lobstein, *Covering Codes,* North-Holland, Elsevier Science B.V., 1997.

[15] I. Bouyukliev, "What is Q-Extention," presented at 12th International Conference on Applications of Computer Algebra, Varna, Bulgaria, 2006.

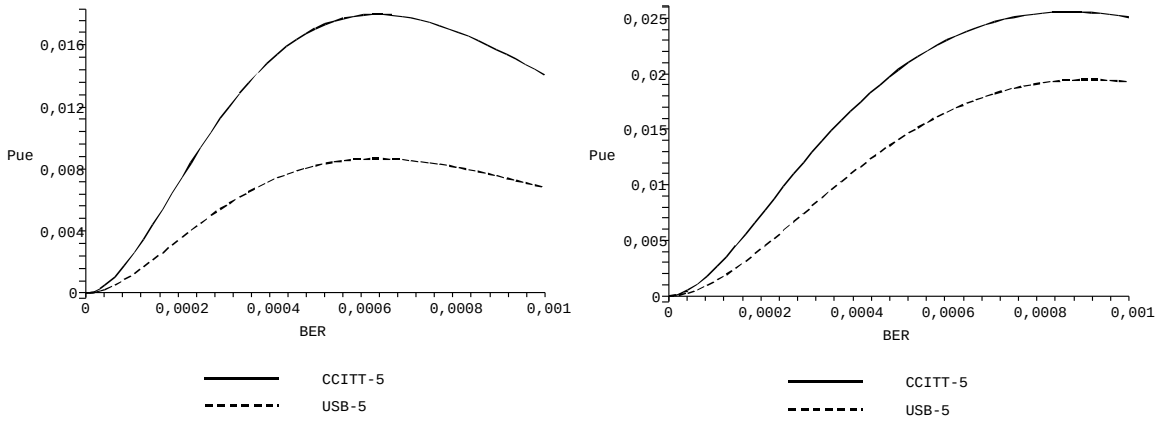

Figure 1: P_{ue} of CCITT-5 and USB-5 calculated via A_2 and via $\{A_2, A_3, A_4\}$.

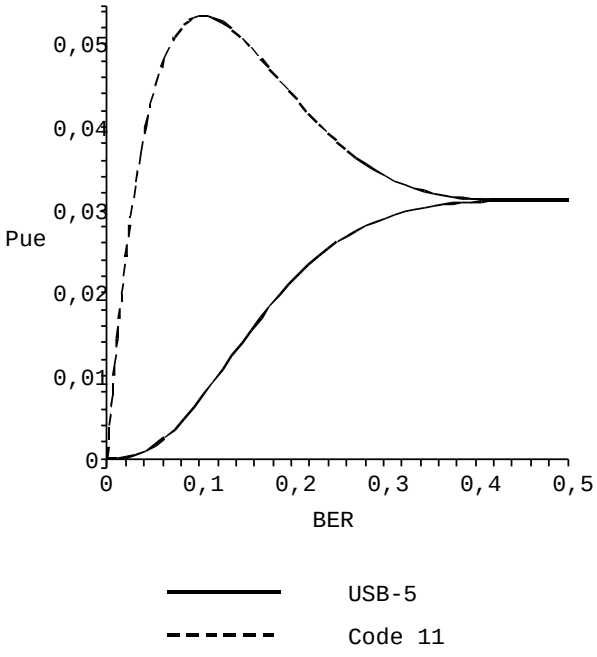

Figure 2: Comparison of P_{ue} of USB-5 and code with number 11 from the table.