2014-01

Performance of BCH codes with $(1+x)^s$ error detection

M. Ambroze and M. Tomlinson

The performance is investigated of a combined error correction and detection decoder for BCH codes for which the generator polynomial $g(x)$ has been augmented by a $(1+x)^s$ term in order to make the informations bits an integral number of bytes. An ARQ retransmission scheme on an AWGN channel is assumed and a comparison with a FEC only BCH code is given in terms of probability of error against E_b/N_0. It is shown for a BCH(127,106,7) code that at high E_b/N_0 the performance improvement is of three orders of magnitude at the cost of a small rate degradation. Goppa codes, whose length is an integral number of bytes, have also been investigated producing similar results.

Introduction: There are practical situations in which the generator polynomial of a cyclic code is multiplied by a $(1+x)^s$ factor in order that the number of information or parity bits is an integral number of bytes [1]. The resulting code is has a generator polynomial $g'(x)=(1+x)^s g(x)$ and a codeword of this code is given by $c'(x)=g'(x)d(x)$. In [2] it was shown that such a term can be used to detect failure of the BCH decoder to correct transmission errors and that the detection process can be implemented as a fast and efficient algorithm which does not add a significant overhead to the decoder complexity. The paper did not study the improvement due to this additional error detection capability. In this letter we compare the performance of two communication systems using a BCH code with generator polynomial $g(x)$:

1 *Error correction only.* A scheme using a hard decision bounded distance decoder correction up to t errors, where t is the design parameter for the BCH code.
2 *Error correction followed by detection and ARQ.* A scheme using a code based on the same BCH code with generator polynomial $g'(x)=g(x)(1+x)^s$ where s is a small integer. The decoder proceeds by applying the hard decision bounded distance decoder for the BCH code. If the decoding algorithm detects a number of errors larger than t (the bounded distance decoder fails), a retransmission is triggered. If the decoding algorithm succeeds, the resulting BCH codeword $c'(x)$ is checked for divisibility by $(1+x)^s$ using the fast algorithm described in [1]. If the divisibility test fails, a retransmission is triggered. If it succeeds, the codeword is assumed error free.

Error correction only: We assume an AWGN channel with BPSK modulation. The probability of bit error after hard decision is given by:

$$p = Q\left(\sqrt{2RE_b/N_0}\right) \tag{1}$$

Where $R=k/n$ is the code rate.
The probability of correct decoding is given by:

$$P_c = \sum_{w=0}^{t} \binom{n}{w} p^w (1-p)^{n-w} \tag{2}$$

Where t is the design number of errors that the BCH code can correct. The probability of word error after decoding is $P_e = 1 - P_c$.

Error correction followed by detection and ARQ: For a BCH code, denote e_w an error pattern of weight w and $e_w(x)$ its polynomial form. Denote the set of undetectable error patterns of weight w as $U_{w,s}$ and its cardinality as $|U_{w,s}|$. An error pattern $e_w \in U_{w,s}$ if:

1 It has the same syndrome as a weight $w' \in \{0,1,..,t\}$ error pattern $e_{w'}$, $\exists e_{w'}$ such that $H(e_w^T + e_{w'}^T) = 0$,
and
2 The polynomial $e_w(x) + e_{w'}(x)$ is divisible by $(1+x)^s$.

Also denote $n_d(w,s) = \binom{n}{w} - |U_{w,s}|$ the number of detectable error patterns of weight $w > t$.

Detection probability: The probability of detecting an error pattern of weight $w > t$ is given by:

$$f_p = \sum_{w=t+1}^{n} n_d(w,s) p^w (1-p)^{n-w} \tag{3}$$

where p is the channel error probability. It is the probability of a detectable (but not correctable) pattern. In practice, this sum is truncated to a maximum weight less than n on the basis that high powers of p can be neglected. The average number of retransmissions is given by:

$$\tilde{r} = \sum_{i=1}^{\infty} i f_p^i (1-f_p) = \frac{f_p}{1-f_p} \tag{4}$$

The average number of channel bits transmitted per block is:

$$n + n\tilde{r} = n\left(1 + \frac{f_p}{1-f_p}\right) = \frac{n}{1-f_p} \tag{5}$$

The (average) code rate taking into account retransmissions and the additional s parity checks is given by:

$$R_r = \frac{k-s}{n+n\tilde{r}} = (k-s)\frac{1-f_p}{n} \tag{6}$$

The probability of error after hard decision is:

$$p = Q\left(\sqrt{2R_r E_b/N_0}\right) \tag{7}$$

By comparing equations 1 and 7 it can be seen that, for the same hard decision error probability p, the ARQ scheme incurs an E_b/N_0 penalty due to the additional s parity bits and retransmissions:

$$\Delta E_b/N_0 = 10\log_{10}\frac{R}{R_r} \tag{8}$$

$$= 10\log_{10}\frac{k}{n}\frac{n}{k-s}\frac{1}{1-f_p} \tag{9}$$

$$= 10\log_{10}\frac{k}{k-s} - 10\log_{10}(1-f_p) \tag{10}$$

The probability of correct decoding is given by:

$$P_c = \left(1 + f_p + f_p^2 + ...\right)\sum_{w=0}^{t}\binom{n}{w}p^w(1-p)^{n-w} \tag{11}$$

$$= \frac{\sum_{w=0}^{t}\binom{n}{w}p^w(1-p)^{n-w}}{1-f_p} \tag{12}$$

The probability of error is $P_e = 1 - P_c$.

Results: For the BCH(127,106,7) code the values of $|U_{w,s}|$ are given in Table 1. Note that there are no undetectable error patterns for $w=4, s>0$. This is because the weight of $e_w(x) + e_{w'}(x)$ is always 7 (odd) for an undetectable pattern e_w of $w=4$. If an error pattern of weight $w=4$ has the same syndrome as an error pattern of lower weight w', this weight is always $w'=3$. This is because $d_m = 7$ for this code so $He_w^T + He_{w'}^T = 0$ only if $w + w' \geq d_m = 7$. The corresponding values $n_d(w,s) = \binom{n}{w} - |U_{w,s}|$ are also given in Table 1. The plotted results are given in Figure 1(a). To investigate the impact of ARQ, we show the performance of three FEC only schemes, BCH(127,106,7), BCH(127,99,9) and BCH(127,92,11) in comparison with ARQ schemes based on BCH(127,106,7) only without $(1+x)$ factors ($s=0$) and with $(1+x)^s$ factors (up to $s=4$). The graphs show improvement due to retransmissions at the cost of a rate penalty which is expressed in terms of $\Delta E_b/N_0$ in Figure 1(b). This rate penalty has a maximum at an $E_b/N_0 \approx 2.5$ dB due to retransmissions then decreases asymptotically towards $10\log_{10}\frac{k}{k-s}$. This maximum actually causes a performance degradation for $E_b/N_0 < 4$ dB due to a large number of retransmissions. However, at $E_b/N_0 > 4$ dB there is a decrease in word error probability which quickly reaches several orders of magnitude.

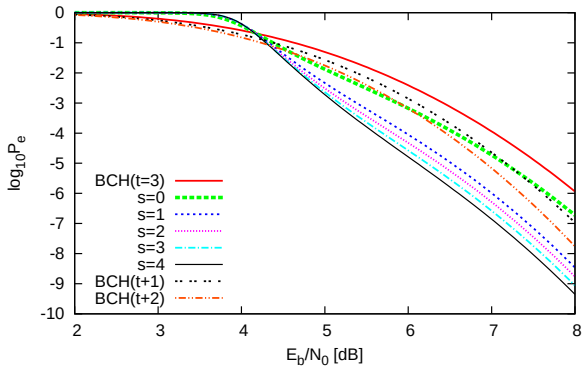
An interesting case in Figure 1(a) is that of $s=0$. This corresponds to the performance of the ARQ scheme without the additional $(1+x)^s$ term. As the BCH(127,106,7) is not a perfect code, we use a non-zero syndrome combined with the failure of the decoder to produce a solution if the number of errors is larger than t to detect errors. It can be seen in Figure 1(a) that this detection in combination with ARQ produces an improvement of one order of magnitude, similar to the improvement given

by using a BCH(127,99,9) code. This shows the power of ARQ over FEC alone and ties in with the well known result that, although feedback does not improve capacity, it can simplify code design. Adding the detection of non correctable syndromes using just the BCH(127,106,7) code with ARQ has virtually the same performance as using the BCH(127,99,9) code. Adding a single $1+x$ factor with ARQ produces over one order of magnitude improvement, better than FEC with BCH(127,92,11) code (also shown in Figure 1(a)). Adding additional $1+x$ factors produces smaller additional improvements.

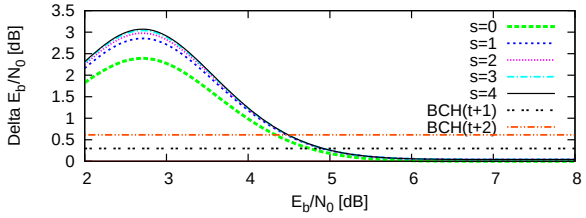
Table 1: The values of $|U_{w,s}|$ and $n_d(w,s)$ for BCH(127,106,7).

$\|U_{w,s}\|$			
	w		
s	4	5	6
0	1,693,545	41,661,207	835,589,769
1	0	40,645,800	20,322,540
2	0	20,393,688	10,196,844
3	0	10,267,992	5,133,996
4	0	5,217,688	2,608,844

$n_d(w,s) = \binom{127}{w} - \|U_{w,s}\|$			
	w		
s	4	5	6
0	8,641,080	212,570,568	4,333,789,656
1	10,334,545	213,586,695	5,149,056,885
2	10,334,545	233,838,087	5,159,182,581
3	10,334,545	243,963,783	5,164,245,429
4	10,334,545	249,014,087	5,166,770,581



(a)



(b)

Fig. 1 *BCH(127,106,7): (a) P_e improvement due to retransmissions for different values of s, (b) E_b/N_0 penalty due to lower rate and/or retransmissions.*

Extension to Goppa codes: The investigation has been extended to cover Goppa codes. As these are generally not cyclic codes, the s check bits corresponding to the $(1+x)^s$ factor are presented as data at the input of the Goppa encoder, together with the $k-s$ data bits. Written as a polynomial, the data bits at the input of the Goppa code encoder are given by $d(x) = (1+x)^s d'(x)$ where $d(x)$ has rank $k-1$ and $d'(s)$ has rank $k-s-1$. The decoding and detection proceeds in a similar manner as for BCH codes, with the only difference that only the decoded data, $d(x)$ is checked for divisibilty by $(1+x)^s$ as opposed to the whole decoded codeword $c(x)$ for the BCH codes. The results for the Goppa(256,232,7) are shown in Figure 2. They show a similar behaviour as the results for the BCH code, with around one order of magnitude improvement due to ARQ ($s=0$).

Conclusion: We have compared the performance of two communication systems, one using a BCH code with hard decision bounded distance

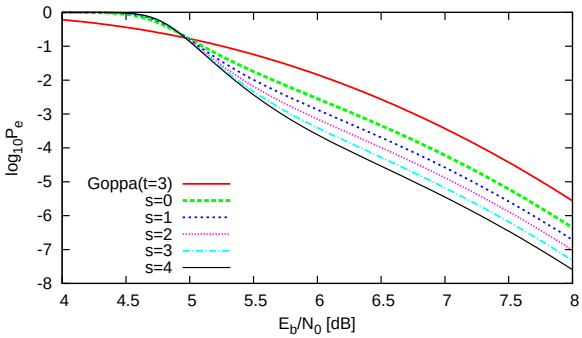


Fig. 2 *Goppa(256,232,7): P_e improvement due to retransmissions for different values of s.*

decoding and one using a code obtained from the same BCH code by appending a $(1+x)^s$ factor to the generator polynomial. The latter scheme is used in conjunction with retransmissions to exploit the improved error detection due to the additional factor. The improvement in performance on an AWGN channel and the error rate penalty due to the additional factor and retransmissions were investigated. A similar performance is exhibited by the scheme when the BCH code is replaced by a Goppa code, with the $(1+x)^s$ factor applied only to the data part of the codeword.The arrangement with Goppa codes has the practical advantage that both the code length and number of information bits is an integral multiple of bytes.

M. Ambroze and M. Tomlinson (*Plymouth University, UK*)

E-mail: mambroze@plymouth.ac.uk

References

1 Castagnoli, G., Brauer, S., and Herrmann, M.: 'Optimization of cyclic redundancy-check codes with 24 and 32 parity bits', IEEE Transactions on Communications, 41(6), pp. 883-892, 1993.

2 Vishal, A. and Frost, G. and Jung, D. and Newhart, D.: 'Results to improve the efficiency of BCH and CRC codes', *OnLine*, 2013. Retrieved from: http://www.math.psu.edu/mass/reu/2013/mathfest/writeup_august_6th.pdf