The Error-Locator Polynomial of Two-Error Correcting Binary Goppa-Codes

K. Huber

The error locator polynomial of two-error correcting binary Goppa codes is given.

Introduction: Two-error correcting binary Goppa codes are very useful for practical applications. Some of them (e.g. the byte oriented [16,8,5] code) have parameters which are better than any linear cyclic codes. Unfortunately they are rarely used. One reason seems to be that Goppa codes demand some algebraic knowledge which many designers do not have. It is therefore of interest to directly give the error locator polynomial for Goppa codes to facilitate the decoding of such codes. Having an explicit expression for the error locator polynomial should also be of interest for theoretical reasons. The error locator polynomial follows immediately from variants of the Patterson algorithm for decoding Goppa codes. Although this is a rather simple task it has not been done to the best knowledge of the author. For the basics of Goppa codes see [2], [5], [6].

Goppa Codes: As usual for linear codes, the parameter triple $[n,k,d]$ designates length n, information rate k, and minimum distance d of the Goppa codes. Goppa Codes over the field $GF(q)$ are defined by its codewords $\mathbf{c}=(c_0,c_1,c_2,\ldots,c_{n-1})$ for which the following equation holds:

$$\sum_{i=0}^{n-1}\frac{c_i}{z-\alpha_i}\equiv 0 \bmod G(z)\,.$$

In the above equation $G(z)$ is a polynomial of degree s over the field $GF(q^m)$ called Goppa Polynomial and α_i are n elements of the field $GF(q^m)$ which are not zeros of the Goppa Polynomial, i.e. $G(\alpha_i)\neq 0$. If the codeword is transmitted over a channel it may be corrupted by an error vector $\mathbf{e}=(e_0,e_1,e_2,\ldots,e_{n-1})$ and the receiver gets the vector $\mathbf{r}=(r_0,r_1,r_2,\ldots,r_{n-1})$. For this vector the receiver can compute the syndrome

$$S(z)=\sum_{i=0}^{n-1}\frac{r_i}{z-\alpha_i}=\sum_{i=0}^{n-1}\frac{c_i}{z-\alpha_i}=0 \bmod G(z)$$

The error locator polynomial $\sigma(z)$ contains the roots which identify the error locations, i.e. if at positions $j\in F\subset\{0,1,\ldots n-1\}$ errors e_j occured the polynomial reads $\sigma(z)=\prod_{j\in F}(z-\alpha_j)$. The decoding problem is now reduced to solving the so called key equation

$$\omega(z)=\sigma(z)\cdot S(z) \bmod G(z)$$

for Goppa codes. Finding the polynomials σ,ω of lowest degree which solve the key equation makes it possible to find the positions (from the roots of σ) and the error values (using ω). Here we only consider binary codes ($q=2$) and the error values are equal to one. In the binary case we get

$$\sigma'(z)=\sigma(z)\cdot S(z) \bmod G(z)$$

where the dash denotes the derivative. Letting the inverse polynomial of $S(z)$ modulo $G(z)$ be denoted by $T(z)$ and setting $\sigma(z)=a^2(z)+zb^2(z)$ i.e. $\sigma'=b^2(z)$ we arrive at the new key equation for binary Goppa codes

$$a(z)=b(z)\cdot R(z) \bmod G(z)$$

where $R(z)$ is the square root of $T(z) \bmod G(z)$. This equation is valid for $T(z)\neq z$, for $T(z)=z$ one gets $\sigma(z)=z$. Further details are given in [6].

The procedure for solving the above equation for a,b which leads to σ is called the Patterson algorithm. The new key equation can be solved e.g. by the Berlekamp-Massey or the Euclidean algorithm (see [9], [6]). For the degrees of a,b we have $\deg\{a(z)\}\leq s/2$ and $\deg\{b(z)\}\leq(s-1)/2$. Finding the square root of $T(z)$ modulo $G(z)$ can be done by matrix-methods or better by the algorithm given in ([3], [4]). Computationally the Patterson algorithm is faster than solving the key equation using a Goppa polynomial having twice the degree than the Goppa polynomial $G(z)$ used here (see [1], p.237). This is particularly important for applications of Goppa codes for the McEliece cryptosystem [7]. Using the Patterson algorithm we can start with the initial degree of the polynomials having half the degree of the conventional approach.

Error Locator Polynomial for Two-Error Case: To explicitly determine the error locator polynomial, we now consider the above decoding algorithm for the two error case. To achieve the maximal length of Goppa codes we demand that the degree two Goppa polynomial $G(z)$ has no roots in $GF(2^m)$. To keep the polynomial as simple as possible one sets $G(z)=z^2+z+\gamma$. For γ an element is used which has trace equal to one, i.e.

$$\mathrm{tr}(\gamma)=\gamma^{2^0}+\gamma^{2^1}+\gamma^{2^2}+\ldots\gamma^{2^{m-1}}\stackrel{!}{=}1\,.$$

Then $G(z)$ has no roots in $GF(2^m)$. As half of the elements of $GF(2^m)$ have trace equal to one and half have trace equal to zero it is easy to rapidly find an element γ of trace one. For odd m we may take $\gamma=1$.

From $S(z)\cdot T(z)\equiv 1 \bmod G(z)$ we obtain $T(z)=T_0+T_1z$ with

$$\begin{aligned}T_1 &= \frac{S_1}{S_0^2+S_1^2\gamma+S_0S_1}\\ T_0 &= \frac{S_0+S_1}{S_0^2+S_1^2\gamma+S_0S_1}\end{aligned}$$

and finishing the Patterson algorithm leads — up to a multiplicative constant — to the error locator polynomial

$$T_0+\gamma(T_1+1)+z+(T_1+1)z^2\,.$$

This formula holds if two error happened. Luckily, it also gives the error locator polynomial for the one error case. This can be shown as follows. For a single error at position j the syndrome leads to

$$\sum_{i=0}^{n-1}\frac{1}{z-\alpha_j} \bmod G(z)=\frac{G(z)+G(\alpha_j)}{G(\alpha_j)(z-\alpha_j)}=\frac{1+\alpha_j}{\alpha_j^2+\alpha_j+\gamma}+\frac{z}{\alpha_j^2+\alpha_j+\gamma}$$

Plugging $S_0=\frac{1+\alpha_j}{\alpha_j^2+\alpha_j+\gamma}$ and $S_1=\frac{1}{\alpha_j^2+\alpha_j+\gamma}$ into the equation for T_1 we get $T_1=1$ and the error locator polynomial reduces to $T_0+z=z-\alpha_j$ i.e. the error locator polynomial for the one error case.

Hence the decoding for errors of weight up to two is straight-forward: Compute the syndrome $S(z)$: If $S(z)=0$ the received vector is a codeword. If $S(z)$ is non-zero compute T_1. The one-error case occurs for $T_1=1$ and the degree one polynomial $\sigma(z)=z-\alpha_j$ immediately gives the value α_j from which the error position j follows. For $T_1\neq 1$ the error locator polynomial has degree two and if it has two roots in $GF(2^m)$ (see below for the condition) these roots deliver the positions at which the two errors occured.

For the solution of a quadratic equation over $GF(2^m)$, one best uses a well-known (but perhaps not widely known) formula which can be traced back to Hilbert (see [8], pp.104-108). First using $z=\frac{x}{T_1+1}$ the locator polynomial above is transformed to

$$x^2+x+\kappa \qquad \text{where} \qquad \kappa=(T_0+1)(T_0+\gamma(T_1+1))\,.$$

If x_1 is a root then x_1+1 is the second. Using any element u of $GF(2^m)$ having trace equal to one (e.g. $u=\gamma$) it is easy to verify that x_1 given by

$$x_1=\kappa\cdot u^2+(\kappa+\kappa^2)\cdot u^{2^2}+\ldots+(\kappa+\kappa^2+\ldots\kappa^{2^{m-2}})\cdot u^{2^{m-1}}$$

is root of $x^2+x+\kappa$. For odd m, setting $u=1$, the formula for x_1 simplifies to $x_1=\kappa^2+\kappa^{2^3}+\ldots+\kappa^{2^{m-2}}$. Thus the roots of the error locator polynomial are given by $z_1=x_1/(T_1+1)$ and $z_2=x_2/(T_1+1)$. The condition that the quadratic $x^2+x+\kappa$ has two zeros in $GF(2^m)$ is $\mathrm{tr}(\kappa)=0$.

Conclusion: The error location polynomial for two error correcting Goppa codes has been determined explicitly. This result simplifies decoding such codes and may be useful for theoretical investigations.

K. Huber (*Huber Consult, Berlin, Germany*)

E-mail: coding@klaus-huber.net

Web: http://www.klaus-huber.net

References

1 Blahut, R.E.: 'Theory and Practice of Error Control Codes', *Addison-Wesley 1983*, Reading, Massachuchetts, reprinted with corrections 1984

2 Goppa,V.D.: 'A New Class of Linear Error-Correcting Codes', *Probl. Peredach. Inform.*, Vol.6, No.3, 1970, pp.24-30
3 Huber, K.: 'Note on decoding binary Goppa Codes', *Electronics Letters*, 18th Jan. 1996, Vol.32 No.2, pp.102-103
4 Huber, K.: 'Taking pth Roots Modulo Polynomials over Finite Fields', *Designs, Codes and Cryptography*, 28, 303-311, 2003
5 MacWilliams,F.J. Sloane, N.J.A.: 'The Theory of Error-Correcting Codes', *North Holland*, Amsterdam,1977
6 McEliece, R.J.:'The Theory of Information and Coding', *Addison-Wesley*, Reading, 1977
7 McEliece, R.J.: 'A public-key cryptosystem based on algebraic coding theory', *DSN Progress Rept. 42-44*, Jet Propulsion Laboratory, pp.114-116, 1978
8 McEliece, R.J.: 'Finite Fields for Computer Scientists and Engineers', *Kluwer Academic Publishers*, Boston Dordrecht Lancaster, 1987
9 Patterson, N.J.: 'The Algebraic Decoding of Goppa Codes', *IEEE Trans. on Inf. Theory*, Vol.IT-21, No.2, March 1975, pp.203-207