List decoding performance of algebraic geometric codes

L. Chen, R.A. Carrasco and M. Johnston

An efficient list decoder for algebraic geometric (AG) codes has been developed based on the mathematical framework of the Guruswami-Sudan algorithm. New simulation results presented show that coding gains of up to 1.5 dB over unique AG decoding algorithms are possible on the AWGN and Rayleigh fading channels.

Introduction: Algebraic geometric (AG) codes were introduced by Goppa [1] in 1981. Compared with Reed Solomon (RS) codes defined over the same Galois field (GF), they have longer code lengths, larger Hamming distances and hence better error correction capability. There are also more AG codes available. This suggests that AG codes could replace RS codes, especially in recording systems and mobile communications. Therefore, the authors have investigated practical decoding algorithms for AG codes. Johnston and Carrasco [2] constructed new AG codes from Hermitian curves, called Hermitian codes, and presented the first simulation results evaluating their performance. They employed the Sakata algorithm with majority voting [3] to determine the error locations, and inverse discrete Fourier transform (IDFT) to calculate the error magnitudes. This algorithm results in one uniquely decoded code word that cannot correct errors beyond the half distance boundary ($d/2$), which limits its performance over deeply corruptive channels.

However, there is an alternative decoding algorithm which can correct errors beyond $d/2$, called list decoding. In 1997, Sudan [4] introduced this algorithm to decode low rate RS codes beyond $d/2$. Later, Guruswami and Sudan [5] improved the algorithm to decode most RS codes beyond $d/2$ and extended the algorithm to decode the family of AG codes, called the GS algorithm. Hoholdt and Nielsen [6] presented a mathematical framework of this algorithm to decode Hermitian codes. To improve the GS algorithm's efficiency, Chen *et al.* [7] presented a reduced complexity GS algorithm for RS codes.

Recently, the authors have investigated the list decoding of Hermitian codes based on the GS algorithm and found this complexity reduced scheme can also be applied. This Letter presents the list decoding of Hermitian codes and new simulation results to evaluate their performance over the AWGN and Rayleigh fading channels. After an extensive literature survey, we believe that these are the first simulation results showing the list decoding performance of AG codes.

Construction of Hermitian codes: The Hermitian curves are defined over GF(w^2) as:

$$H_w(x, y) = x^{w+1} + y^w + y = 0 \tag{1}$$

where $w \geq 2$, with genus [8]:

$$g = \frac{w(w-1)}{2} \tag{2}$$

There are $n = w^3$ affine points [8] $P = (p_1, p_2, \ldots, p_n)$ that satisfy the curve $H_w(x, y)$ giving a block length of n. A Hermitian code $C(n, k)$ with length n and dimension k can be generated by evaluating the n affine points over the transmitted message polynomial $f(x, y)$ as:

$$C(n, k) = \{(f(p_1), f(p_2), \ldots, f(p_n)),\ p_i \in P\} \tag{3}$$

where the transmitted message polynomial $f(x, y)$ can be written as:

$$f(x, y) = f_0\phi_0(x, y) + f_1\phi_1(x, y) + \cdots + f_{k-1}\phi_{k-1}(x, y) \tag{4}$$

where $f_0, f_1, \ldots, f_{k-1} \in \mathrm{GF}(w^2)$ are the message symbols and $\phi_0(x, y)$, $\phi_1(x, y), \ldots, \phi_{k-1}(x, y)$ are the first k functions of the basis B containing rational functions with increasing pole order v_{p_∞} at the point of infinity p_∞ of the curve [8]:

$$B = \{\phi_i(x, y) | v_{p_\infty}(\phi_i(x, y)^{-1}) < v_{p_\infty}(\phi_{i+1}(x, y)^{-1}),\quad i \in N\} \tag{5}$$

List decoding: The system model of the list decoder is shown in Fig. 1. $R = (r_1, r_2, \ldots, r_n) \in \mathrm{GF}(w^2)$ is the received word after corruption by the channel. To obtain the correct transmitted message, there are two key steps: interpolation and factorisation. Combining the received word with the respective affine point used in code construction, there are n interpolated units: $\{(p_1, r_1), (p_2, r_2), \ldots, (p_n, r_n)\}$. The aim of interpolation is to construct a trivariate polynomial [5]:

$$Q(x, y, z) = \sum_{i,j \in N} q_{i,j}\phi_i(x, y)z^j,\quad q_{i,j} \in \mathrm{GF}(w^2) \tag{6}$$

which has a zero of multiplicity m over the n units. Based on the interpolated polynomial $Q(x, y, z)$, the factorisation process finds the list L of polynomials $h(x, y)$ in the form of (4) that satisfy:

$$L = \{h(x, y) | z - h(x, y) | Q(x, y, z) \quad \text{or} \quad Q(x, y, h(x, y)) = 0\} \tag{7}$$

Every candidate polynomial $h(x, y)$ is the z-root of $Q(x, y, z)$ and it has the possibility of being the transmitted message polynomial $f(x, y)$. The one that has the minimal distance to the received word R after re-encoding using (3) is chosen to be the decoded message polynomial. The GS algorithm's error correction capability τ grows with multiplicity m:

$$\tau_{m_1} \leq \tau_{m_2}, \quad \text{if} \quad m_1 < m_2 \tag{8}$$

According to [5], the GS algorithm has an error correction upper bound for decoding a (n, k) AG code given by:

$$\tau_{\max} = n - \lfloor \sqrt{n(k + g - 1)} \rfloor - 1 \tag{9}$$

It is obvious that $\tau_{\max}$ is always greater or equal to $d/2$, with designed distance $d = n - k - g + 1$. Therefore, the list decoder can outperform the unique decoder, especially for low rate codes.

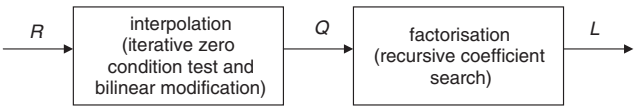

Fig. 1 *System model of list decoder*

When implementing interpolation, a list of polynomials is initialised. They are tested at every interpolated unit with every possible zero condition [6] and updated using bilinear modification iteratively. For those polynomials that do not satisfy a certain zero condition the minimal one under the trivariate lexicographic order [6] is chosen and updates itself with an increase in the leading order [7]. Other polynomials are updated without an increase in the leading order. After $n\binom{m+1}{2}$ iterative interpolations, the minimal polynomial is chosen to be the interpolated polynomial (6) and is factorised. During the iterative interpolation, those polynomials with leading order over $n\binom{m+1}{2}$ will not be chosen or contribute to the interpolated polynomial (6) and they can be eliminated in order to improve the algorithm's efficiency [7]. From (7) we can see that the candidate polynomials $h(x, y)$ are the z-roots of $Q(x, y, z)$ and they can be found by factorising $Q(x, y, z)$. Since $h(x, y)$ can be written in the form of (4) and the rational functions $\phi_0(x, y)$, $\phi_1(x, y)$, $\ldots$, $\phi_{k-1}(x, y)$ are predetermined in the list decoder, finding $h(x, y)$ is equivalent to finding out its coefficients $h_0, h_1, \ldots, h_{k-1} \in \mathrm{GF}(w^2)$, respectively. Therefore, we have developed a recursive coefficient search algorithm to implement factorisation. The polynomial $Q(x, y, h(x, y))$'s leading monomial is equivalent to $Q(x, y, h_{k-1}\phi_{k-1}(x, y))$'s leading monomial and so for $Q(x, y, h(x, y)) = 0$ we need its leading monomial's coefficient to equal zero. Therefore, h_{k-1} can be determined by solving $Q(x, y, h_{k-1}\phi_{k-1}(x, y))$'s leading monomial's coefficient with unknown h_{k-1}. Based on each h_{k-1}, $Q(x, y, z) = Q(x, y, z + h_{k-1}\phi_{k-1}(x, y))$ is updated and h_{k-2} is determined by solving $Q(x, y, h_{k-2}\phi_{k-2}(x, y))$'s leading monomial's coefficient. This coefficient search algorithm deduces the coefficients recursively until the last possible value of f_0 has been determined and finally outputs a list of candidate message polynomials.

Performance analysis: A software platform of the list decoder has been developed using the C programming language evaluating the performance of the (64, 29, 30) Hermitian code using the reduced complexity GS algorithm with different values of multiplicity m on the AWGN and Rayleigh fading channels, as shown by Figs. 2 and 3, respectively. In the simulations, QPSK modulation is used and channel estimation is assumed to be perfect. The Rayleigh fading

channel is fast fading with independent fading coefficients representing the effect of an ideal channel interleaver. The mean value and variance of the fading coefficients are 1.25 and 0.44, respectively. As mentioned previously, the transmitted message is chosen by comparing its corresponding code word's distance to the received word. However, in some situations there is more than one candidate message polynomial with the same minimal distance. In those situations we assume that the decoder always makes a correct decision, which means the one that matches $f(x, y)$ is to be chosen. The performance is compared with the unique decoding algorithm (the Sakata algorithm) used in [2]. From Figs. 2 and 3 we can see that when $m = 1$ the GS algorithm performs as well as the Sakata algorithm. However, when we increase the multiplicity to $m = 2$ the GS algorithm outperforms the Sakata algorithm with about 0.2 and 0.6 dB coding gains at $\mathrm{BER} = 10^{-5}$ over AWGN and Rayleigh fading channels, respectively. By assuming the GS algorithm is able to correct errors up to its upper bound given in (9), we have presented the algorithm's theoretical performance with the optimal multiplicity showing that the coding gain in AWGN and Rayleigh fading channels can be increased to 0.5 and 1.5 dB, respectively. Based on the complexity analysis in [7], the complexity reduced scheme can reduce the interpolation computation by up to 40%. However, the overall complexity of the GS algorithm is still high compared with the unique decoding algorithm, e.g. the GS algorithm with $m = 1$ has an average of 1.4×10^5 computations compared with 5×10^4 for the unique decoding algorithm [2]. Therefore, further research is required to address this problem.

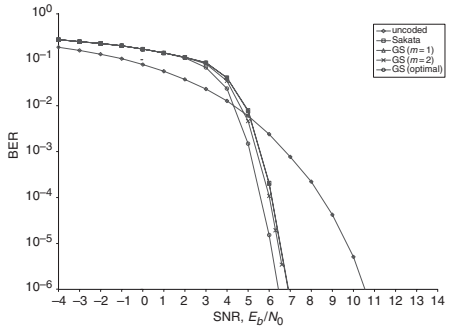

Fig. 2 *Performance of GS decode (64, 29, 30) Hermitian code over AWGN channels*

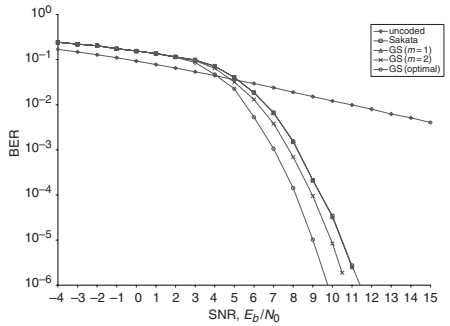

Fig. 3 *Performance of GS decode (64, 29, 30) Hermitian code over Rayleigh fading channels*

Conclusion: An efficient list decoder for AG codes has been developed with new simulation results presented. Compared with the conventional unique decoder, it can improve AG codes' performance by allowing it to correct errors beyond the half distance boundary. We believe the AG list decoder could be suitable for use in data storage and communication applications providing its complexity can be reduced.

27 June 2006
Electronics Letters online no: 20061999
doi: 10.1049/el:20061999

L. Chen, R.A. Carrasco and M. Johnston (*School of Electrical, Electronic and Computer Engineering, University of Newcastle-upon-Tyne, Newcastle-upon-Tyne, NE1 7RU, United Kingdom*)

E-mail: r.carrasco@ncl.ac.uk

References

1 Goppa, V.D.: 'Codes on algebraic curves', *Sov. Math.*, 1981, **Dok 24**, pp. 75–91
2 Johnston, M., and Carrasco, R.A.: 'Construction and performance of algebraic-geometric codes over AWGN and fading channels', *IEE Proc., Commun.*, 2005, **152**, (5), pp. 713–722
3 Sakata, S., *et al.*: 'Fast decoding of algebraic-geometric codes up to the designed minimum distance', *IEEE Trans. Inf. Theory*, 1995, **IT-41**, (5), pp. 1672–1677
4 Sudan, M.: 'Decoding of Reed Solomon codes beyond the error-correction bound', *J. Complexity*, 1997, **13**, (1), pp. 180–193
5 Guruswami, V., and Sudan, M.: 'Improved decoding of Reed-Solomon and algebraic-geometric codes', *IEEE Trans. Inf. Theory*, 1999, **45**, (6), pp. 1757–1767
6 Hoholdt, T., and Nielsen, R.R.: 'Decoding Hermitian codes with Sudan's algorithm, applied algebra, algebraic algorithms and error-correcting codes', *Lect. Notes Comput. Sci.*, 1719, Springer-Verlag, 1999, pp. 260–270
7 Chen, L., Carrasco, R.A., and Chester, E.G.: 'Performance of Reed-Solomon codes using the Guruswami-Sudan algorithm with improved interpolation efficiency', *IEE Proc., Commun.*, 2006, (accepted for publication)
8 Pretzel, O.: 'Codes and algebraic curves' (Clarendon Press, Oxford, 1998)