

# Simulation Results for Algebraic Soft-Decision Decoding of Reed-Solomon Codes

Warren J. Gross

Frank R. Kschischang

Ralf Koetter

P. Glenn Gulak

**Abstract**— The Koetter-Vardy algorithm is an algebraic soft-decision decoder for Reed-Solomon codes. The algorithm is based on an extension to the Guruswami-Sudan list-decoding algorithm with variable multiplicities that are assigned proportional to the reliabilities of the received symbols. There are three steps: (1) multiplicity calculation, (2) interpolation of a bivariate polynomial, and (3) finding the  $y$ -roots of this polynomial. A low-complexity algorithm for calculating the multiplicities is proposed. Simulation results indicate that the coding gain is dependent on the code rate and ranges from 0.25 dB to 4.25 dB with a practical upper limit of 1 – 1.5 dB, assuming binary phase shift keying and additive white Gaussian noise. Higher coding gains of between 2 dB and 6.8 dB can be achieved over a Rayleigh fading channel. The KV algorithm exhibits a performance-complexity tradeoff which is tunable by the choice of  $m_{max}$ ,  $n$  and  $k$ . The code parameters should be chosen carefully to take advantage of the “sweet spots” in the performance-complexity profile.

## I. INTRODUCTION

Reed-Solomon codes are powerful error-correcting codes that can be found in a wide variety of digital communications systems, from digital media to wireless communications and deep-space probes. The ubiquitous nature of these codes continues to fuel research into decoding algorithms some forty years after their introduction. A major challenge has been the development of soft-decision decoders; that is, decoders that can utilize the full information available from the channel in the decoding process.

Reed-Solomon codes are non-binary linear block codes whose symbols are chosen from a finite field, usually the binary extension field  $\text{GF}(2^q)$ . Algebraic decoders exploit the underlying algebraic structure of the code to generate a system of equations that is solved using the arithmetic operations of the finite field. This operation does not appear to be compatible with the real-valued, *soft* information available from the channel. Traditional decoders quantize soft information into *hard* decisions which can be utilized directly. It is well known that a performance penalty of approximately 2-3 dB in asymptotic coding gain on an additive Gaussian noise channel (AWGN) is paid when using a hard-decision decoder [1]. Even greater coding gains can be realized over Rayleigh fading channels.

In this paper we characterize the performance of a recently-introduced algebraic soft-decision decoding algorithm, the

Koetter-Vardy Algorithm [2, 3], which provides substantial coding gains while maintaining polynomial complexity in the length of the code.

## II. THE KOETTER-VARDY ALGORITHM

### A. Reed-Solomon Codes

Consider the finite field with  $Q$  elements,  $\text{GF}(Q)$ . The message to be transmitted,  $f$ , consists of  $k$  elements of  $\text{GF}(Q)$ .

$$f = (f_0, f_1, f_2, \dots, f_{k-1}), \quad f_i \in \text{GF}(Q). \quad (1)$$

The message symbols can be considered to be the coefficients of a degree  $k - 1$  message polynomial

$$f(x) = f_0 + f_1x + f_2x^2 + \dots + f_jx^j + \dots + f_{k-1}x^{k-1}. \quad (2)$$

An  $(n, k)$  Reed-Solomon (RS) code over  $\text{GF}(Q)$ , represents the  $k$ -symbol transmitted message  $f$  by an  $n$ -symbol codeword  $c$  formed by evaluating the message polynomial  $f(x)$  at  $n$  elements of  $\text{GF}(Q)$ . If the set of evaluation elements is  $D = \{x_1, x_2, \dots, x_n\}$  then the code is

$$RS_Q(n, k) = \{(f(x_1), f(x_2), \dots, f(x_n))\}, \quad x_i \in D, \quad (3)$$

for all possible message polynomials  $f(x)$ . The minimum distance of an  $(n, k)$  Reed-Solomon code is  $d_{min} = n - k + 1$ .

Usually,  $n = Q - 1$ , and the set of evaluation elements is the set of non-zero elements of  $\text{GF}(Q)$ . If  $n = Q$  then we call the code an *extended* Reed-Solomon code and  $D = \text{GF}(Q)$ . We call this method of generating a RS code the *evaluation map* method [4]. It is also possible to view Reed-Solomon codes as BCH codes. Then, the resulting cyclic code consists of codewords generated by multiplying  $f(x)$  by a *generator polynomial*  $g(x)$ ,

$$c(x) = f(x)g(x), \quad (4)$$

where  $c(x)$  is a degree  $n - 1$  codeword polynomial corresponding to the  $n$ -symbol codeword  $c$ . Other encodings are possible (e.g. systematic encodings).

The evaluation map method is useful because it provides insight leading to *interpolation*-based decoding algorithms.

### B. Decoding as an Interpolation Problem

In this section, we describe the Guruswami-Sudan (GS) algorithm for the hard-decision decoding of Reed-Solomon codes [5, 6] which is the basis for the Koetter-Vardy algorithm. For proofs, please see [5–7]. To formally state the algorithm, we need to define the *weighted degree* of a bivariate polynomial.

Warren J. Gross, Frank R. Kschischang and P. Glenn Gulak are with the Department of Electrical and Computer Engineering, University of Toronto, 10 King’s College Road, Toronto, Ontario, M5S 3G4, Canada. Email: wjgross@eecg.toronto.edu. This research was supported by NSERC and the Government of Ontario.

Ralf Koetter is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, U.S.A. This research was supported by the National Science Foundation under grant CCR-0073490.

Let  $P(x, y) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} p_{i,j} x^i y^j$  be a bivariate polynomial over  $\text{GF}(Q)$  and let  $w_x$  and  $w_y$  be nonnegative real numbers. The  $(w_x, w_y)$ -weighted degree of  $P(x, y)$ ,  $\deg^{(w_x, w_y)}(P)$ , is defined as the maximum over all the numbers  $iw_x + jw_y$  such that  $p_{i,j} \neq 0$ . The  $(1, 1)$ -weighted degree of a bivariate polynomial is the usual notion of degree.

A bivariate polynomial  $P(x, y)$  is said to pass through a point  $(x_i, y_i)$  if  $P(x_i, y_i) = 0$ . Consider the received word  $y = (y_1, y_2, \dots, y_n)$  where  $y = c + e$ . An element of  $\text{GF}(Q)$ ,  $x_i$ , can be uniquely associated with each  $y_i$  to form the list of points in  $\text{GF}(Q) \times \text{GF}(Q)$ ,

$$L_n = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}. \quad (5)$$

If there is no noise ( $e = 0$ ), then  $y_i = f(x_i)$  and the bivariate polynomial  $P(x, y) = y - f(x)$  passes through all of the points in  $L_n$ . To account for the effect of noise, introduce an *error locator polynomial* of the form  $\Lambda(x, y)$  so that  $\Lambda(x_i, y_i) = 0$  whenever  $e_i \neq 0$ . The decoding problem can be posed as the following interpolation problem [8]:

Given a set of  $n$  received points,  $L_n$ , find the bivariate polynomial with minimal  $(1, k-1)$ -weighted degree of the form  $P(x, y) = \Lambda(x, y)(y - f(x))$  with  $\deg f(x) < k$  such that  $P(x, y)$  passes through all the received points and  $y - f(x)$  passes through as many received points as possible.

The bivariate polynomial  $P(x, y)$  can be factored to find the *list* of factors of the form  $y - f(x)$ ,  $\deg f(x) < k$ . The decoded codeword can be found by re-encoding the decoded messages and then choosing the codeword with the minimum distance to the received word. However, there is no need to perform a complete factorization since we are just looking for the linear  $y$ -roots of  $P(x, y)$ . Roth and Ruckenstein give an appropriate root-finding algorithm for this problem [9]. This algorithm is a *bounded-distance* or *list decoder* for Reed-Solomon codes. The list decoding problem is to find the set of codewords at a distance of  $\tau$  from the received word where  $0 \leq \tau \leq n$ . Traditional decoders can only correct up to  $t = \lfloor d_{\min}/2 \rfloor$  errors. If we consider  $\tau > t$ , there may not be a unique codeword at a distance  $> d_{\min}/2$  from the received word. Therefore bounded-distance decoders with  $\tau > t$  return a list of candidate codewords.

An important concept for polynomials over a finite field of characteristic two is the *Hasse Derivative* [10]. The  $(\alpha, \beta)$ 'th Hasse derivative of a bivariate polynomial  $P(x, y)$  is defined for integers  $\alpha, \beta \geq 0$  as:

$$P^{[\alpha, \beta]}(x, y) = \sum_{a \geq \alpha \wedge b \geq \beta} \binom{a}{\alpha} \binom{b}{\beta} p_{a,b} x^{a-\alpha} y^{b-\beta}. \quad (6)$$

We say that a polynomial passes through a point with *multiplicity*  $m_i$  if the polynomial and its Hasse derivatives  $P^{[\alpha_i, \beta_i]}$ ,  $\alpha_i + \beta_i < m_i$ , all pass through the point. We can improve the error-correcting capability of the decoding algorithm by introducing *singularities* at each of the received points [6], forcing the polynomial to intersect itself multiple times at a point. The interpolation polynomial can be found by solving the system of equations implied by  $P^{[\alpha_i, \beta_i]}(x_i, y_i) = 0$  for all points  $(x_i, y_i) \in L_n$ ,  $\alpha_i, \beta_i \geq 0$ , such that  $\alpha_i + \beta_i < m_i$ . The Guruswami-Sudan algorithm can correct up to  $n(1 - \sqrt{k/n})$  errors. For hard-decision

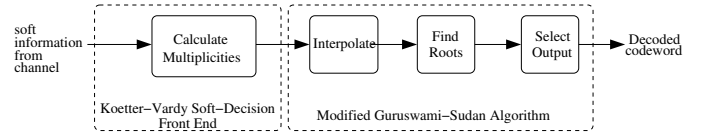


Fig. 1. The Koetter-Vardy algorithm.

decoding, the Guruswami-Sudan algorithm assigns equal multiplicities to the received points.

### C. Soft-Decision Decoding

Guruswami and Sudan hint at a possible soft-decision extension to their algorithm in [6] by allowing each point on the interpolated curve to have its own multiplicity. Koetter and Vardy proposed a method to translate soft-information into multiplicities [2, 3]. The Koetter-Vardy (KV) algorithm performs soft-decision decoding by assigning unequal multiplicities to points according to their relative reliabilities. We note that all possible  $Q \times n$  transmitted/received symbol pairs are considered and not just the ones corresponding to the hard decisions. Once multiplicities have been assigned, the rest of the decoding proceeds according to the Guruswami-Sudan algorithm. A block diagram of the KV algorithm is given in Figure 1

A precise definition of soft information is needed. The symbols are drawn from a finite field  $\text{GF}(Q)$  and transmitted across a memoryless channel. The channel input and output are the random variables  $X$  and  $Y$ . The optimum value of the soft information for symbol  $y_j$  given that symbol  $x_i$  was sent is the *a-posteriori probability* (APP):

$$\pi_{i,j} = P(X = x_i | Y = y_j) \quad (7)$$

where  $1 \leq i \leq Q$  and  $1 \leq j \leq n$ .

A  $(Q \times n)$  *reliability matrix*  $\Pi$  whose columns sum to unity can be constructed from the  $\pi_{i,j}$ . Ultimately, the information in this matrix has to be translated to a set of weighted points.

The weights, or multiplicities can be recorded in a  $(Q \times n)$  *multiplicity matrix*  $M$ . The *score* of a codeword  $c = (c_1, c_2, \dots, c_n)$  over  $\text{GF}(Q)$  with respect to a multiplicity matrix  $M$  is defined as

$$S_M(c) = \sum_{i=1}^Q \sum_{j=1}^n m_{i,j} [c]_{i,j} \quad (8)$$

where the  $[c]_{i,j}$  are elements of the  $(Q \times n)$  matrix  $[c]$  formed by setting  $[c]_{i,j} = 1$  if  $c_j = x_i$  and 0 otherwise. The decoder does not know the codeword, but can only infer information about it from the received soft information. Therefore, the codeword appears as a random vector to the decoder and the score  $S_M(c)$  is a random variable. The goal is to find a multiplicity matrix  $M$  that maximizes the expected value of  $S_M(c)$ . The problem reduces to finding the matrix  $M$  which maximizes the inner product of  $M$  and  $\Pi$  where  $\langle M, \Pi \rangle = \sum_{i=1}^Q \sum_{j=1}^n m_{i,j} \pi_{i,j}$ .

Algorithm 1 [3] is an optimal algorithm for generating a matrix  $M$  which maximizes  $\langle M, \Pi \rangle$  subject to the constraint that  $\sum_{i=1}^Q \sum_{j=1}^n m_{i,j} < s$ . If we let  $s \rightarrow \infty$ ,  $\langle M, \Pi \rangle$  is maximized. The *cost*,  $C_M$  of  $M$  is the number of linear equations that need to be

solved for the interpolation. If an entry in  $M$  is increased from  $m_{i,j}$  to  $m_{i,j} + 1$ , this introduces  $m_{i,j} + 1$  additional linear constraints on the interpolation problem. Therefore, we would like to keep  $s$  as small as possible. We will also see that error-rate performance in general improves with  $C_M$  and hence  $s$ . This gives the KV algorithm a tunable parameter to tradeoff performance with decoding complexity.

---

**Algorithm 1** Algorithm for calculating  $M$  from  $\Pi$  subject to complexity constraint  $s$  (from [3]).

---

```

Choose a desired value for  $s = \sum_{i=1}^Q \sum_{j=1}^n m_{i,j}$ 
 $\Pi^* \leftarrow \Pi$ ;  $M \leftarrow 0$ 
while  $s > 0$  do
  Find the position  $(i, j)$  of the largest entry  $\pi_{i,j}^*$  in  $\Pi^*$ 
   $\pi_{i,j}^* \leftarrow \frac{\pi_{i,j}}{m_{i,j}+2}$ 
   $m_{i,j} \leftarrow m_{i,j} + 1$ 
   $s \leftarrow s - 1$ 
end while

```

---

Algorithm 1 has high complexity as it has to search through a  $(Q \times n)$  matrix  $s$  times. This could require a maximum of  $(Q \times n \times s)$  memory accesses. If we are to have at least the performance of a hard-decision decoder we need  $s \geq n$ . Therefore the complexity of Algorithm 1 is  $O((n+1)(n)(n)) = O(n^3)$ . If  $Q$  is large, say 256, then  $n^3 = 2^{24}$ . We propose a low-complexity algorithm as an alternative to Algorithm 1.  $\langle M, \Pi \rangle$  is maximized if [3]:

$$M = \lfloor \lambda \Pi \rfloor \quad \text{as } \lambda \rightarrow \infty \quad (9)$$

If we instead chose a finite value of  $\lambda \in \mathbb{R}, \lambda > 0$ , then we can obtain a fixed-cost matrix  $M$ . Algorithm 2 is a heuristic that has been experimentally determined to give comparable performance to Algorithm 1. Algorithm 2 only has to make a single pass through  $\Pi$  and therefore has complexity  $O(n^2)$ . We note that in practice,  $\Pi$  is quite sparse with most of its entries  $\approx 0$ . Therefore, storing it explicitly is a waste of memory and computational resources. At the extreme end of the spectrum is the case of high SNR where  $\Pi$  has exactly one non-zero entry per column which is equal to 1.0. This is a hard decision decoding problem and the KV algorithm reduces to the GS algorithm if  $s = n$ . Then the lower bound on the complexity of Algorithm 1 is  $n \times s = O(n^2)$  and Algorithm 2 is  $O(n)$ . The complexity of interpolation is now dependent on the maximum multiplicity in  $M$ ,  $m_{max} = \lfloor \lambda \rfloor$ .

---

**Algorithm 2** for calculating  $M$  from  $\Pi$ . The tunable complexity parameter is  $\lambda$  and the maximum possible entry in  $M$  is  $\lfloor \lambda \rfloor$ .

---

```

for  $i = 1$  to  $n$  do
  for  $j = 1$  to  $Q$  do
     $m_{i,j} \leftarrow \lfloor \lambda \pi_{i,j} \rfloor$ 
  end for
end for

```

---

In practical implementations  $\lambda$  can be a power of two and the multiplication reduces to a shift operation. The floor function is naturally implemented by truncating the bits of the result to the right of the decimal.

### III. SIMULATION RESULTS

#### A. Software Implementation

We have implemented the KV algorithm in software. Soft information is converted to multiplicities by Algorithm 2. The interpolation step finds a Gröbner basis for the ideal of bivariate polynomials which vanish at a set of points with prescribed multiplicities. Fast  $O(n^2)$  algorithms for interpolation are described in [7, 9, 11–14]. We use the algorithm from [7] for the GS algorithm which is easily modified to handle unequal multiplicities. The root-finding algorithm from [9] is used.

The software implementation of the algorithm runs very slowly, especially for large field sizes. Fortunately, an upper-bound on the frame-error rate (FER) can be easily obtained through the following theorem which is proved in [3]:

*Theorem 1*—: Let  $P(x, y)$  be an interpolation polynomial obtained from multiplicity matrix  $M$  corresponding to the transmitted codeword  $c$  with cost  $C_M$ . Then the factorization of  $P(x, y)$  contains a factor  $y - f(x)$  such that  $c = (f(x_1), f(x_2), \dots, f(x_n))$  if:

$$S_M(c) > \min \left\{ \delta \in \mathbb{Z} : \left\lfloor \frac{\delta+1}{k-1} \right\rfloor \left( \delta - \frac{k-1}{2} \left\lfloor \frac{\delta}{k-1} \right\rfloor + 1 \right) > C_M \right\}. \quad (10)$$

If this threshold condition is satisfied then we are guaranteed that the decoding will be successful. Since the decoding could still be successful otherwise, these simulation results might be slightly pessimistic. Simulations for  $n = 15$  and  $k = \{5, 7, 9, 11, 13\}$  and a maximum multiplicity  $m_{max} = \{2, 4\}$  show that the estimated performance matches the actual decoder performance very closely. As an example, see Figure 2. Hybrid simulations are possible where the full decoder is only employed on failure of the threshold condition.

#### B. Coding Gain

Of particular interest is the effect of code rate on the coding gain. The Guruswami-Sudan algorithm can correct up to  $n(1 - \sqrt{k/n})$  errors and therefore improves as the rate  $k/n$  decreases [6]. We would expect that this effect is preserved in soft-decision decoding with the KV algorithm. Simulations for  $n = 15$  and  $k = 3, \dots, 13$  demonstrate this effect. Figure 3 plots the coding gain in dB at a FER of  $10^{-3}$  of the KV algorithm over a conventional hard-decision Reed-Solomon decoder as a function of code rate. The modulation is BPSK and the channel model is additive white Gaussian noise (AWGN). We see that the coding gain ranges from 0.25 dB at high rates and low complexity to 4.25 dB at low rates and high complexity, giving the designer two degrees of freedom to obtain a given coding gain.

Figure 4 shows the performance of the KV algorithm for a very common high-rate (255, 239) Reed-Solomon code. We see that for very high complexities, a maximum gain of 0.47 dB can be achieved. For reasonable complexities, say with  $m_{max} = 4$ , a gain of 0.27 dB is achieved.

We also investigated the performance of the KV algorithm over a Rayleigh fading channel with 16-QAM modulation. Figure 5 shows the performance of a (15, 11) Reed Solomon code. Four-bit symbols from GF(16) are mapped directly to 16-QAM

constellation points. The multiplicative fading factors are independent, simulating the effect of an ideal interleaver. The reliability matrix is calculated directly from the received soft-information assuming perfect channel state information. We see that much larger gains are realized on a fading channel. At a FER of  $10^{-3}$ , the coding gain for a (15, 11) code is 5 dB for  $m_{max} = 4$  and 6.8 dB for  $m_{max} = 100$ .

A simulation setup for a (255, 191) Reed-Solomon code is shown in Figure 6. Each eight-bit symbol of GF(256) is split into two four-bit symbols and two 16-QAM channel uses are needed to transmit the symbol. The simulation results are plotted in Figure 7. The coding gain is 2.1 dB for  $m_{max} = 4$  and 2.9 dB for  $m_{max} = 100$  at FER  $\approx 10^{-3}$ . We note that the coding gain on a Rayleigh channel is not constant but increases as the SNR increases since the two curves diverge. The results given here for a high FER will improve as the SNR increases.

### C. Complexity

Above we saw that the achievable coding gain increases as the rate of the code decreases. Unfortunately, so does the computational complexity of the algorithm, which is dominated by the complexity of the interpolation algorithm. The algorithm maintains a set of  $b$  polynomials of length  $L$  terms, which at the end of each of the  $C_M$  iterations, satisfy one additional linear constraint. When the iterations terminate, the polynomial with minimal  $(1, k-1)$ -weighted degree is chosen as  $P(X, Y)$ . The number of iterations  $C_M$  is [3]:

$$C_M = \frac{1}{2} \sum_{i=1}^Q \sum_{j=1}^n m_{i,j}(m_{i,j} + 1), \quad (11)$$

which is a function of the code length  $n$ , and the maximum multiplicity  $m_{max}$ . The speed of the algorithm is determined by  $C_M$  since iteration  $i$  requires the results of iteration  $i-1$ , creating a dependency loop. The memory requirements for interpolation are  $b \times L$ . The number of polynomials  $b$  is given as [7]

$$b = \left\lfloor \frac{(k-1) + \sqrt{(k-1)^2 + 8C_M(k-1)}}{2(k-1)} \right\rfloor, \text{ which is } \approx \left\lfloor \sqrt{\frac{2C_M}{k-1}} \right\rfloor.$$

Therefore,  $b$  increases with both  $n$  and  $m_{max}$  and also increases as the rate  $k/n$  decreases. For maximum coding gain, we would like the rate to be low and  $m_{max}$  to be high, but this increases the computational complexity and memory requirements. To study this tradeoff, one possible measure is the space-time complexity of interpolation:

$$\begin{aligned} C_{par} &= b \times L \times C_M \\ &\approx b \times C_M^2, \end{aligned} \quad (12)$$

assuming that the  $b$  polynomials can be updated within each loop in parallel in constant time. This expression assumes that the complexity of GS decoding with equal multiplicities of  $m_{max}$  is an upper bound to the complexity of KV decoding with maximum multiplicity  $m_{max}$ . Note that  $L$  can be calculated exactly [3, 7] but is approximately  $C_M$ . For software implementations on a serial machine, each iteration updates  $b$  polynomials of length  $L$  and the complexity is

$$\begin{aligned} C_{ser} &= b^2 \times L^2 \times C_M \\ &\approx b^2 \times C_M^3. \end{aligned} \quad (13)$$

Let us define the *complexity-gain ratios*  $CGR_{par} = C_{par}/\beta_{r,m}$  and  $CGR_{ser} = C_{ser}/\beta_{r,m}$  where  $\beta_{r,m}$  is the coding gain. For the coding gain experiment of Figure 3 (BPSK, AWGN,  $n = 15$ ), we plot  $CGR_{par, norm}$ , the  $CGR_{par}$  normalized by its largest value. From Figure 8 we see that for this particular setup, there are “good” (even values of  $k$ ) and “bad” (odd  $k$ ) choices of the rate. The designer should judiciously choose the parameters of the code with this in mind.

## IV. CONCLUSIONS

The Koetter-Vardy algorithm is a soft-decision decoding algorithm for Reed-Solomon codes that incorporates soft-decisions into an algebraic decoding framework. The soft-decision front end can be implemented with a reasonable complexity using the proposed Algorithm 2. We have presented simulation results to characterize the coding gains possible from the soft-decision Koetter-Vardy algorithm for Reed-Solomon codes. The algorithm can achieve significant gain for soft-decision decoding on an AWGN channel but only for low-rate codes and with very high complexity. For reasonable complexities, the upper limit is 1–1.5 dB. High rate codes only benefit from 0.25 to 0.5 dB of coding gain. Rayleigh fading channels exhibit larger coding gains of about 2 to 6.8 dB. The KV algorithm exhibits a performance-complexity tradeoff which is tunable by the choice of  $m_{max}$ ,  $n$  and  $k$ . The code parameters should be chosen carefully to take advantage of the “sweet spots” in the performance-complexity profile.

## REFERENCES

- [1] A. Brinton Cooper III, “Soft decision decoding of Reed-Solomon codes,” in *Reed-Solomon Codes and Their Applications*, ch. 6, pp. 108–124, New York, New York: IEEE Press, 1994.
- [2] R. Kotter and A. Vardy, “Algebraic soft-decision decoding of Reed-Solomon codes,” in *Proc. of the IEEE Int. Symp. on Information Theory*, p. 61, 2000.
- [3] R. Koetter and A. Vardy, “Algebraic soft-decision decoding of Reed-Solomon codes.” Submitted to IEEE Trans. Inf. Theory, May 31 2000.
- [4] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *SIAM Journal of Applied Math.*, vol. 8, pp. 300–304, 1960.
- [5] M. Sudan, “Decoding of Reed-Solomon codes beyond the error correction bound,” *J. Complexity*, vol. 13, no. 1, pp. 180–193, 1997.
- [6] V. Guruswami and M. Sudan, “Improved decoding of Reed-Solomon and Algebraic-Geometry codes,” *IEEE Trans. Information Theory*, vol. 45, pp. 1757–1767, September 1999.
- [7] R. R. Nielsen, “Decoding AG-codes beyond half the minimum distance,” Master’s thesis, Technical University of Denmark, August 31 1998.
- [8] G. D. Forney, “Reed-Solomon codes.” URL: <http://truth.mit.edu/~eyeh/6.451/L3G.pdf>, 2001.
- [9] R. M. Roth and G. Ruckenstein, “Efficient decoding of Reed-Solomon codes beyond half the minimum distance,” *IEEE Trans. Information Theory*, vol. 46, pp. 246–257, January 2000.
- [10] H. Hasse, “Theorie der höheren differentiale in einem algebraischen funktionenkörper mit vollkommenem konstantenkörper bei beliebiger charakteristik,” in *J. Reine. Ang. Math.*, vol. 175, pp. 50–54, 1936.
- [11] H. M. Möller and B. Buchberger, “The construction of multivariate polynomials with preassigned zeros,” in *EUROCAM ’82, European Computer Algebra Conference* (J. Calmet, ed.), vol. 144 of *Lecture Notes In Computer Science*, (Marseille, France), pp. 24–31, April 1982.
- [12] J. Abbott, A. Bigatti, M. Kreuzer, and L. Robbiano, “Computing ideals of points,” *J. Symbolic Computation*, vol. 30, no. 4, pp. 341–356, 2000.
- [13] G. Feng and K. Tzeng, “A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes,” *IEEE Trans. Inf. Theory*, vol. 37, pp. 1274–1287, September 1991.
- [14] R. Kötter, *On Algebraic Decoding of Algebraic-Geometric and Cyclic Codes*. PhD thesis, Linköping University, 1996.

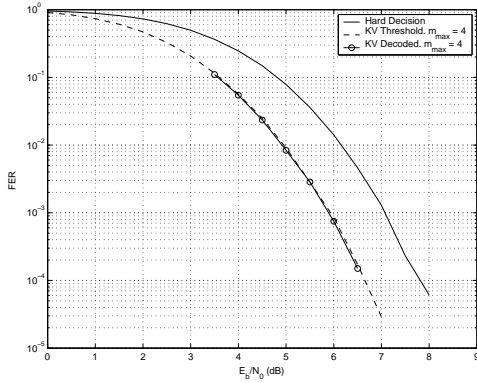


Fig. 2. Simulation of a (15,7) Reed-Solomon code with BPSK modulation over an AWGN channel comparing the actual Koetter-Vardy decoder with  $m_{max} = 4$  and a simulation using the threshold condition in Theorem 1.

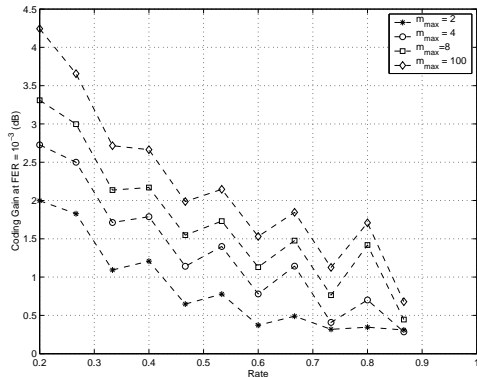


Fig. 3. The effect of code rate on coding gain for a (15, $k$ ) Reed-Solomon code transmitted with BPSK modulation over an AWGN channel. The simulations were performed using the threshold condition in Theorem 1.

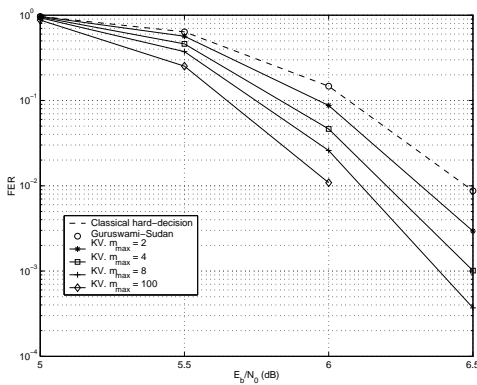


Fig. 4. Simulation of a (255,239) Reed-Solomon code with BPSK modulation over an AWGN channel using the threshold condition from Theorem 1.

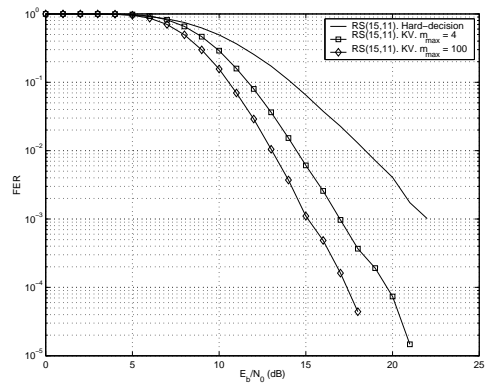


Fig. 5. Simulation of a (15, 11) Reed-Solomon code with 16-QAM modulation over a Rayleigh fading channel using the threshold condition in Theorem 1.

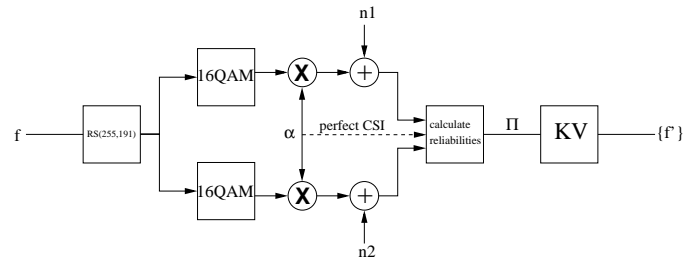


Fig. 6. Simulation setup for decoding the RS(255, 191) code over a Rayleigh fading channel with 16-QAM modulation.

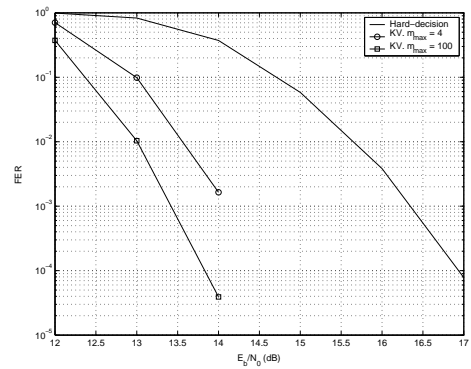


Fig. 7. Simulation of a (255, 191) Reed-Solomon code with 16-QAM modulation over a Rayleigh fading channel using the threshold condition from Theorem 1.

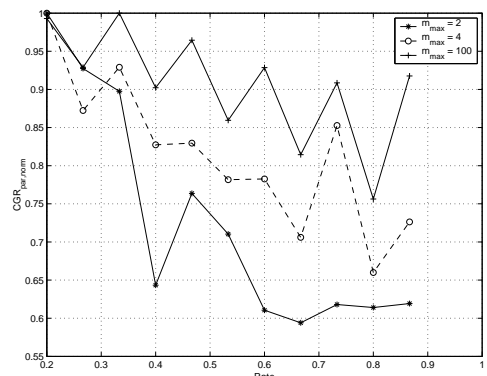


Fig. 8. The normalized complexity-coding gain ratio vs. the code rate.