Reduced Complexity Interpolation for Soft-Decoding of Reed-Solomon Codes

Arshad Ahmed, Ralf Koetter, and Naresh R. Shanbhag
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W. Main Street, Urbana, IL 61801, U.S.A.
{ahmed4,koetter,shanbhag}@uiuc.edu

Abstract — **The re-encoding based interpolation algorithm [1] is modified such that intermediate interpolation results are also useful towards solving the algebraic soft-decoding problem. By factorization of a chosen subset of intermediate results, desired coding gains are obtained at lower interpolation costs.**

I. Introduction

An algorithm for algebraic soft-decision decoding of Reed-Solomon codes based on bivariate polynomial interpolation and factorization [2] is presented in [3]. The Koetter-Vardy (KV) algorithm [3], extends the coding gains obtained in [2] by relaxing the restriction of fixed multiplicity for all interpolation points. In the KV algorithm, interpolation constraints are incrementally assigned to different interpolation points via a greedy algorithm, until the total number of constraints summed over all interpolation points reaches a *preset final cost*. By solving interpolation constraints in the order of generation, a nested sequence of interpolation solutions associated with increasing interpolation costs is obtained. Factorization of intermediate solutions, in addition to the final interpolation solution, can be used to reduce both the average and the final interpolation cost required to obtain a desired coding gain.

In the re-encoding based interpolation algorithm [1], for a $[n, k]$ RS code, *all* interpolation constraints associated with k chosen re-encoded interpolation points are solved at *initialization* by an appropriate choice of *basis polynomials*. Hence in [1], interpolation constraints are solved *out of order*, as the constraints solved at initialization are not the initial constraints generated by the greedy algorithm in [3]. As the number of constraints solved at initialization constitutes a large fraction ($\approx \frac{k}{n}$) of the total constraints, intermediate results in [1] are no longer associated with interpolation solutions of intermediate costs except close to the final cost.

For a desired coding gain, the preset final cost in [1] can be reduced if the interpolation algorithm in [1] is modified such that intermediate results closely match interpolation solutions of intermediate costs, while retaining the initialization of [1]. Also, the number of intermediate results passed to factorization has to be restricted in order to limit factorization complexity. These are the two issues addressed in this paper.

II. Modified re-encoding based interpolation

The re-encoded interpolation algorithm [1], begins with a set of basis polynomials $\tilde{Q}_l(X, Z)$, $0 \leq l \leq t$ which satisfy the constraints associated with k re-encoded interpolation points. The number of basis polynomials $(t + 1)$ is a function of the final interpolation cost. The re-encoded interpolation algorithm determines the minimal $(1, -1)$ weighted bivariate polynomial $Q_M(X, Z) = \sum_{l=0}^{t} \Phi_l(X)\tilde{Q}_l(X, Z)$, that passes through the remaining interpolation points [1]. As the number of basis polynomials is interpolation cost dependent, a first step toward obtaining useful intermediate results is to restrict the number of basis polynomials used at intermediate costs, to that in its corresponding interpolation solution. Further, when the number of basis polynomials in the intermediate result is incremented, a modified polynomial update algorithm is used to solve a fixed number of interpolation constraints without increasing the weighted degree of the intermediate result. In the modified update algorithm, an additional step follows regular polynomial update, in which the weighted degree of the minimal polynomial is decremented by adding appropriately scaled versions of other candidate polynomials. Further details can be found in [4].

From [1], we know that a useful interpolation output contains a factor of the form $(Z - \Omega(X)/\Lambda(X))$, where $\Lambda(X)$ is the error-locater polynomial. Now, $\Lambda(X)$ divides $\Phi_l(X)$, where l denotes the largest Z degree in the intermediate output. The number of roots in $\Phi_l(X)$ can be used to detect the presence of $(Z - \Omega(X)/\Lambda(X))$ in the intermediate result. For each set of intermediate interpolation results which use the same number of basis polynomials, we pass a maximum of two results to factorization; the intermediate result associated with the largest intermediate cost and the first intermediate result that passes a root threshold criterion for $\Phi_l(X)$. Details are given in [4]. The reduction in final cost ranges from 35-55% for a $[255, 239]$ RS code (Figure. 1).

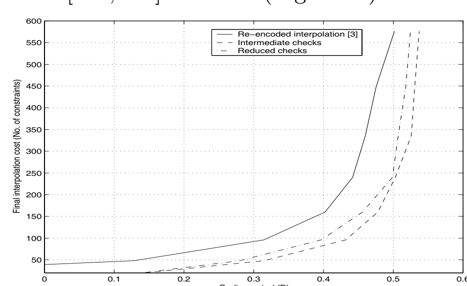

Figure 1: Final cost v/s Coding gain at FER = 10^{-5}.

References

[1] R. Koetter, J. Ma, A. Vardy and A. Ahmed, "Efficient interpolation and factorization in algebraic soft-decision decoding of Reed-Solomon codes," *Proc. IEEE Symp. Info. Theory*, 2003.

[2] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Trans. Info. Theory* , vol. 45, pp. 1757-1767, Sept. 1999.

[3] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Info. Theory*, vol. 49, pp. 2809 - 2825, Nov. 2003.

[4] A. Ahmed, R. Koetter, and N. .R Shanbhag, "Reduced complexity interpolation via intermediate solutions," Manuscript in preparation, Mar. 2004.

[1]This work was funded by NSF grant: CCR 00-73490.