On transforming Generator Matrices of GRS_k Codes for systematic Encoding

A Literature Review

Roman Wetenkamp*

January 29, 2023

Abstract

Linear codes like GRS_k are frequently used for error-correction and even cryptographic purposes. Since the encoding is often required to be *systematic*, the goal of this paper is to find algorithms that transform *non-systematic* generator matrices into *systematic* ones. The conducted literature review shows that most frequently the inverse of the $k \times k$-submatrix is computed and multiplied to the generator matrix to achieve the *standard form*.

1 Introduction

The encoding of information tupels with linear codes is done by matrix multiplication. Since several generator matrices for the same linear code exist, approaches like *systematic encoding* make use of this in order to produce codewords with certain properties. This article deals with the question how existing generator matrices of a linear code can be transformed into *systematic* ones algorithmically.

2 Problem

Definition 1 ([1])**.** *A code C with length n over a finite field $\mathbb{F}_q$ with $q \in \mathbb{P}$ or $q = p^m, p \in \mathbb{P} \wedge m \in \mathbb{N}$ is said to be **linear** if C forms a linear subspace of $\mathbb{F}_q^n$.*

Linear Codes can be represented by a generator matrix and a parity-check matrix which is the generator matrix of the dual code.

An example for linear codes is the class of *Generalized Reed Solomon* codes:

Definition 2 ([2])**.** *Let $a = \langle a_0, a_1, \ldots, a_{n-1}\rangle$ be a tupel of pairwise distinct components from $\mathbb{F}_q$ and $v = \langle v_0, v_1, \ldots, v_{n-1}\rangle$ be a tupel of components from $\mathbb{F}_q \setminus \{0\}$. The **generalized Reed-Solomon code** $GRS_k(a, v)$ is now defined by all codewords*

$$c = \langle v_0 f(a_0), v_1 f(a_1), \ldots, v_{n-1} f(a_{n-1})\rangle$$

for all polynomials $f \in \mathbb{F}_q[x]/(x^n - 1)$ with $\deg f < k$.

The generator matrix for this class of codes as in [2] is given by

$$G_{GRS} = \begin{pmatrix} v_0 & v_1 & \cdots & v_{n-1} \\ v_0 a_0 & v_1 a_1 & \cdots & v_{n-1} a_{n-1} \\ \vdots & \vdots & & \vdots \\ v_0 a_0^{k-1} & v_1 a_1^{k-1} & \cdots & v_{n-1} a_{n-1}^{k-1} \end{pmatrix}$$

Definition 3 ([1])**.** *A generator matrix G is called to be **systematic** if it has the form $G = [I_k \mid P_{n-k}]$ with I_k being the $k \times k$-identity matrix and P_{n-k} being the $k \times (n-k)$-matrix for generating parity-check bits.*

Since the matrix G_{GRS} is in *non-systematic* form, an algorithm for transforming this matrix into a *systematic* form is required.

*✉ s200376@student.dhbw-mannheim.de

3 Method

A literature review is conducted.

- *Question of Research:* Which algorithms generate *systematic* generator matrices for linear codes like Generalized Reed Solomon?
- *Derived search terms:* "Generalized Reed Solomon" ∧ ("Systematic Encoding" ∨ "Generator Matrix"); "Generator Matrix" ∧ ("Standard Form" ∨ "Systematic Form");
- *Databases:* IEEE Xplore (1); Google Scholar (2); JSTOR (3); KIT KVK (4);
- *Scope:* 1+3: Metadata; 2+4: Full-texts
- *Criteria for Selection:* Implementability; Efficiency; Applicability;

Since a variety of different shaped and constructed generator matrices is found among linear codes in general, it seems ineffective to search for terms like "Linear codes *systematic* encoding" containing no information on specific code classes. To find general algorithms that work for several types of linear codes, the second search term was introduced.

4 Results

Using the first search term "Generalized Reed Solomon *systematic* encoding", the following results were gathered:

- BRAUCHLE points out that matrices of Generalized Reed Solomon codes are *Vandermonde* matrices. He proposes a new algorithm for the recovery of erased symbols [3]. The search term "Vandermonde matrix *systematic* conversion" will be added to this papers scope. Since [3] references [4], it will be added to the review.
- In [4] an algorithm for the *systematic* encoding of Reed-Solomon codes with arbitrary parity positions (neither pre- nor postponed to the information bits) is proposed. BRAUCHLE AND KOETTER show that for a given parity-check *Vandermonde* matrix $H =$

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^0 & \alpha^1 & \cdots & \alpha^{n-1} \\ \vdots & \vdots & & \vdots \\ \alpha^{0\cdot(2t-1)} & \alpha^{1\cdot(2t-1)} & \cdots & \alpha^{(n-1)\cdot(2t-1)} \end{pmatrix}$$

  a *systematic* form can be derived by multiplying the original matrix with the inverse of the $2t \times 2t$-submatrix B_{2t}:

$$H = [A \mid B_{2t}] \Rightarrow H_{sys} = B_{2t}^{-1} H = [P \mid I_{2t}]$$

- MATTOUSSI, ROCA, AND SAYADI compare in [5] the construction of *systematic* generator matrices using *Vandermonde* and *Hankel* matrices. Their *Hankel*-based approach has no need for matrix inversions and is therefore considered to be faster and simpler [5].

The second search term "Generalized Reed Solomon Generator Matrix" delivers one additional result:

- ROTH AND SEROUSSI proved in [6] that a $GRS_k(\alpha, v)$ code with length n has a *systematic* generator matrix $G = [I \mid A]$ where A is a $k \times (n-k)-$ Generalized Cauchy Matrix. Their proof is constructive, hence a *systematic* generator matrix can be derived from a given *non-systematic* generator matrix using the formulas in [6].

The term "Generator Matrix Standard Form" gives related procedures:

- NAKKIRAN, RASHMI, AND RAMCHANDRAN describe in [7] the "systematic remapping" operation in order to generate *systematic* codewords from *non-systematic* generator matrices: Let G be a $k \times n$-generator matrix. G_k denotes the $k \times k$-submatrix consisting of the first k columns of G. Since encoding is done by $c = G \cdot m$ for an information tuple m of length k, the *remapping step* and *systematic* encoding is defined by

$$\overline{m} = G_k^{-1} \cdot m \Rightarrow c_{sys} = G \cdot \overline{m}$$

Several papers described here reference [8]. HILL proposes an algorithm for transforming a generator matrix G to *standard form* using elementar operations like row/column permutation, row multiplications with scalars or adding multiples of a row to another.

Algorithm 1 Algorithm for generating a standard form matrix after [8]

Require: $G = (g_{ij})_{1 \le i \le k, 1 \le j \le n}$
  for $j \in \{1, \ldots, k\}$ **do**
    if $g_{jj} = 0$ **then**
      if $i \in \{j+1, \ldots, k\}\colon g_{ji} \neq 0$ **then**
        $\mathrm{row}(g_{jj}) \leftarrow \mathrm{row}(g_{ij}))$
      else
        $\mathrm{col}(g_{jj}) \leftarrow \mathrm{col}(g_{jh} \neq 0)$
      end if
    end if
    $\mathrm{row}(g_{jj}) \leftarrow \mathrm{row}(g_{jj}) \cdot g_{jj}^{-1}$
    for $i \in \{1, 2, \ldots, k\}\colon i \neq j$ **do**
      $\mathrm{row}(g_{i0}) \leftarrow \mathrm{row}(g_{i1}) - g_{ij} \cdot \mathrm{row}(g_{1j})$
    end for
  end for

5 Conclusion

The research has shown that the default approach for *systematic* encoding of linear codes is given by *Vandermonde* matrices. For a given *non-systematic* generator matrix, the $k \times k$-submatrix is inverted and multiplied with G to achieve the *systematic* form $G_{sys} = [I_k \mid P]$. Since matrix inversion can be an exhaustive task for large matrices, alternative approaches using *Hankel* or *Cauchy* matrices could be taken into consideration. Even row and column operations could be used algorithmically to achieve a *systematic* generator matrix, but it should be pointed out that this algorithm does not seem to be more efficient than matrix inversion since matrix inversion is performance engineered in many programming libraries.

Specific solutions for GRS_k codes were not discovered during the conducted literature review. Since this literature review was not done systematically and a comparison of results did not take place, further investigation is required to find and verify the most efficient algorithm for the underlying question.

References

[1] W. C. Huffman and V. Pless, "Basic concepts of linear codes," in *Fundamentals of Error-Correcting Codes*, Cambridge, UK: Cambridge University Press, 2010, ch. 1, pp. 1–52, ISBN: 9780521131704. [Online]. Available: `https://archive.org/details/fundamentalsofer0000huff/` (visited on 01/25/2023).

[2] F. MacWilliams and N. Sloane, "10 reed-solomon and justesen codes," in *The Theory of Error-Correcting Codes*, ser. North-Holland Mathematical Library, vol. 16, Elsevier, 1977, pp. 294–316. DOI: `https://doi.org/10.1016/S0924-6509(08)70535-X`.

[3] J. Brauchle, "On efficient recovery of erased symbols in generalized reed-solomon codes," in *2011 IEEE International Conference on Communications (ICC)*, 2011, pp. 1–5. DOI: `10.1109/icc.2011.5962475`.

[4] J. Brauchle and R. Koetter, "A systematic reed-solomon encoder with arbitrary parity positions," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, 2009, pp. 1–4. DOI: `10.1109/GLOCOM.2009.5426304`.

[5] F. Mattoussi, V. Roca, and B. Sayadi, "Complexity comparison of the use of vandermonde versus hankel matrices to build systematic mds reed-solomon codes," in *2012 IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2012, pp. 344–348. DOI: `10.1109/SPAWC.2012.6292924`.

[6] R. M. Roth and G. Seroussi, "On generator matrices of mds codes (corresp.)," *IEEE Transactions on Information Theory*, vol. 31, no. 6, pp. 826–830, 1985. DOI: `10.1109/TIT.1985.1057113`.

[7] P. Nakkiran, K. V. Rashmi, and K. Ramchandran, "Optimal systematic distributed storage codes with fast encoding," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 430–434. DOI: `10.1109/ISIT.2016.7541335`.

[8] R. Hill, "Introduction to linear codes," in *A First Course in Coding Theory*, ser. Oxford Applied Mathematics and Computing Science Series, Oxford University Press, 1986, ISBN: 0-19-853804-9.