Decoding Interleaved Reed Solomon Codes over Noisy Channels

Daniel Bleichenbacher¹, Aggelos Kiayias², and Moti Yung³

¹ Bell Laboratories, Murray Hill, NJ, USA
bleichen@bell-labs.com
² Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA
aggelos@cse.uconn.edu
³ RSA Laboratories Bedford, MA, USA, and
Department of Computer Science, Columbia University, New York, NY, USA
moti@cs.columbia.edu

Abstract. We consider error-correction over the Non-Binary Symmetric Channel (NBSC) which is a natural probabilistic extension of the Binary Symmetric Channel (BSC). We propose a new decoding algorithm for interleaved Reed-Solomon Codes that attempts to correct all “interleaved” codewords simultaneously. In particular, interleaved encoding gives rise to multi-dimensional curves and more specifically to a variation of the Polynomial Reconstruction Problem, which we call Simultaneous Polynomial Reconstruction. We present and analyze a novel probabilistic algorithm that solves this problem. Our construction yields a decoding algorithm for interleaved RS-codes that allows efficient transmission arbitrarily close to the channel capacity in the NBSC model.

1 Introduction

Random noise assumptions have been considered extensively in the coding theory literature with substantial results. One prominent example is Forney Codes [5] that were designed over the binary symmetric channel (BSC). The BSC suggests that when transmitting binary digits, errors are independent and every bit transmitted has a fixed probability of error. The BSC provides a form of a random noise assumption, which allows probabilistic decoding for message rates that approach the capacity of the channel.

Worst-case non-ambiguous decoding (i.e., when only a bound on the number of faults is assumed and a unique solution is required) has a natural limitation of correcting a number of errors that is up to half the distance of the code. Going beyond this natural bound, either requires re-stating the decoding problem (e.g. consider list-decoding: output all possible decodings for a corrupted codeword), or assuming some “noise assumption” that will restrict probabilistically
the combinatorial possibilities for a multitude of possible solutions. Typically, such assumptions are associated with physical properties of given channels (e.g., bursty noise, etc.). Recent breakthrough results by Guruswami and Sudan, and later by Parvaresh and Vardy in list-decoding ([14, 6, 11]) demonstrated that decoding beyond the natural error-correction bound is possible in the worst-case, by outputting all possible decodings. Naturally, there are still limitations in the case of worst-case decoding that prohibit the efficient decoding of very high error-rates (e.g., the size of the list is the trivial lower bound).

In this work, motivated by the above, we investigate a traditional channel model that is native to the non-binary setting. The channel is called “Non-Binary Symmetric Channel” (NBSC), presented in figure 1.

As a channel model for bit-level transmission, the Non-Binary Symmetric Channel model usually applies to settings where aggregates of bits are sent and errors are assumed to be bursty. Thus, in contrast with the Binary Symmetric Channel, errors in consecutive bits are assumed from a Coding Theoretic perspective to be correlated. There are additional situations that have been considered in a number of settings where the NBSC describes the transmission model. For example, consider the case of Information Dispersal Algorithms (IDA) introduced by Rabin in [12] for omission errors, and extended by Krawczyk [8] to deal with general errors. In this setting, a word is encoded into a codeword and various portions of the codeword are sent over different radio network channels, some of which may introduce errors. In the case where the channels are operating in different frequencies, errors may be introduced by jammed channels which emit white noise (i.e., they randomize the transmitted symbol). As a result the communication model in this case approximates the NBSC. Another setting which approximates the NBSC is the transmission of encrypted data where each sub codeword is sent encrypted with what is called “error propagation encryp-

![Fig. 1. A non-binary symmetric channel over an alphabet of \( q \) symbols. The probability of successful transmission is \( 1 - \epsilon + \epsilon/q \). We will refer to \( \epsilon \) as the error-rate of the NBSC.](image)
tion mode." These popular modes (e.g. the CBC mode), over noisy channels, will produce a transmission that also approximates the NBSC model ([10], page 230). Moreover the NBSC model has been used in the cryptographic setting as a way to hide information in schemes that employ intractability assumptions related to the hardness of decoding, see e.g. [7].

In this work we concentrate on Reed-Solomon Codes. The decoding problem of Reed-Solomon Codes (also known as the Polynomial Reconstruction problem — PR) has been studied extensively, see e.g. [1, 14, 6]. Here, we present a variation of the PR, which we call “Simultaneous Polynomial Reconstruction” and we present a novel probabilistic algorithm that solves it for settings of the parameters that are beyond the currently known solvability bounds for PR (without any effect on the solvability of the latter problem). Our algorithm is probabilistic and is employed in settings where errors are assumed to be random.

Next we concentrate on the “code interleaving” encoding schema, see e.g. section 7.5, [15], which is a technique used to increase the robustness of a code in the setting of burst errors. We consider the problem of decoding interleaved Reed-Solomon Codes and we discover the relationship of this problem to the problem of Simultaneous Polynomial Reconstruction. In particular we show that the two problems are equivalent when interleaved Reed-Solomon Codes are applied over a channel that satisfies the NBSC model.

Subsequently using our algorithm for Simultaneous Polynomial Reconstruction we present a novel decoding algorithm for interleaved Reed-Solomon Codes in the NBSC model that is capable of correcting (probabilistically) any error-rate up to $\frac{r}{r+1}(1 - \kappa)$ where $r$ is the “amount of interleaving” and $\kappa$ is the message rate.

We note that traditional decoding of interleaved RS-Codes does not improve the error-rate that can be corrected. In fact, error-rates only up to $\frac{1 - \kappa}{2}$ can be corrected (uniquely) in the worst-case, but also list-decoding algorithms [6] can be also employed thus correcting error-rates up to $1 - \sqrt{\kappa}$ (but producing more than one solutions would be possible in this case).

If the channel on the other hand follows the NBSC model, improving the solvability of (non-interleaved) PR is a open with the only result known the fact that list-decoding will produce a single polynomial as output and thus error-rates of up to $1 - \sqrt{\kappa}$ can be assumed uniquely decodable. The recent results of [11] suggest further improvement for unique decodability are possible.

Considering interleaved RS-codes on the other hand, the situation changes dramatically: using our algorithm for Simultaneous Polynomial Reconstruction we can correct error-rates up to $\frac{r}{r+1}(1 - \kappa)$. An immediate corollary is that we can correct any error-rate bounded away from $(1 - \kappa)$ provided that the alphabet-size is selected to be large enough. In other words, interleaved RS-Codes reach the channel’s capacity assuming the interleaving $r \to \infty$.

Concurrently with the present work, that was originally publicised in [3], Coppersmith and Sudan [4] also presented an algorithm that essentially solves the Simultaneous Polynomial Reconstruction Problem and following our approach, it can also be applied to the interleaved Reed-Solomon codes setting. Their anal-
ysis states that their algorithm works when the error-rate is below $1 - \kappa - \kappa \epsilon r$. Note that this algorithm is especially geared towards vanishing message-rates (i.e., when the limit of the message-rate as a function of the code-length is 0).

The results of the present work as well as its comparison to the related work in the context of decoding probabilistic interleaved Reed-Solomon Codes and deterministic worst-case decoding of such codes are presented in figure 2.

Fig. 2. Reed-Solomon Codes decoding algorithms. BW = [2], GS = [6], CS = [4]. Note that BW, GS are deterministic, worst-case, they require no interleaving and are merely included for the sake of comparison. This paper and CS operate in the interleaved case and employ an (essentially) equivalent distribution assumption on the instance space. The parameters of the table are $\kappa$ the message-rate and $\epsilon$ the error-rate. The plots are based on the probabilistic bounds that were proved in the respective works rather than experimental results (which may potentially exhibit improved performance).

Organization. In section 2 we present our variation of the Polynomial Reconstruction problem and we describe and analyze a probabilistic algorithm that solves this problem. Subsequently in section 3 we describe the relation of this problem to the decoding of Interleaved Reed-Solomon codes and we show how our algorithm is employed in this domain.
Notation. We will use standard notation throughout. $\mathbb{F}$ will denote a finite field. $\mathbb{N}$ will denote the set of natural numbers; we use $[n]$ to denote the set $\{1, \ldots, n\}$ for any $n \in \mathbb{N}$. If $A$ is a finite set we denote by $|A|$ its cardinality and by $a \leftarrow_R A$ the process of sampling an element of $A$ following the uniform distribution; i.e., if $a \leftarrow_R A$ it holds that $a$ will assume any value from $A$ with probability $1/|A|$.

2 Simultaneous Polynomial Reconstruction

In this section we present a probabilistic algorithm that solves efficiently the following problem, which we call the Simultaneous Polynomial Reconstruction:

Definition 1. (Simultaneous Polynomial Reconstruction — SPR) for parameters $n, k, t, r \in \mathbb{N}$ and $z_1, \ldots, z_n \in \mathbb{F}$ with $i \neq j \rightarrow z_i \neq z_j$; an instance of SPR is a set of tuples $(y_{i,1}, \ldots, y_{i,r})_{i=1}^n$ over a finite field $\mathbb{F}$ that satisfies the following: There exists an $I \subseteq [n]$ with $|I| = t$, and polynomials $p_1, \ldots, p_r \in \mathbb{F}[x]$ of degree less than $k$, such that $p_\ell(z_i) = y_{i,\ell}$ for all $i \in I$ and $\ell \in [r]$.

The solution of an SPR instance as above would be the tuple $\langle p_1, \ldots, p_r \rangle$.

We will consider the SPR problem not in the worst-case but under a distributional assumption that will be based on the following instance generator. First let $D_{\text{msg}}$ be any samplable probability distribution over $(\mathbb{F}^k[x])^r$. Given $D_{\text{msg}}$ we define the distribution $D$ over $(\mathbb{F}^r)^n$ with parameters $n, k, t, r$ and $z_1, \ldots, z_n \in \mathbb{F}$, using the following sampling procedure:

1. Select $p_1, \ldots, p_r$ distributed according to $D_{\text{msg}}$.
2. Select $I \subseteq \{1, \ldots, n\}$ with $|I| = t$ at random.
3. Select $y_{i,\ell}$ for $i \not\in I$ and $\ell \in [r]$ uniformly at random from $\mathbb{F}$.
4. Set $y_{i,\ell} = p(zi)$ for $i \in I$ and $\ell \in [r]$.
5. Output $\langle y_{1,1}, \ldots, y_{1,r}, \ldots, y_{n,1}, \ldots, y_{n,r} \rangle_{i=1}^n$.

We remark that the goal of Simultaneous Polynomial Reconstruction, assuming a large underlying finite-field $\mathbb{F}$, over the distribution $D$, is well-defined: indeed, one can show that the probability that there exists a second tuple of $r$ polynomials $p'_1, \ldots, p'_r$ that would fit the data in the same way as $p_1, \ldots, p_r$ do, is very small. Taking this into account, the SPR problem with parameters $n, k, t, r$ reduces easily to the Polynomial Reconstruction Problem with parameters $n, k, t$, (by simply reducing the $n$ tuples by discarding $r - 1$ coordinates — it follows easily that the recovery of $p_1$ would reveal the remaining polynomials). Naturally, we would be interested in algorithmic solutions for the SPR problem when the parameters $n, k, t$ are selected to be beyond the state-of-the-art solvability of the PR problem.

2.1 Description of the Algorithm

The algorithmic construction that we present amends the prototypical decoding paradigm (fitting the data through an error-locator polynomial, see e.g. [2, 1]) to the setting of Simultaneous Polynomial Reconstruction. More specifically our
algorithm can be seen as a generalization of the Berlekamp-Welch algorithm for Reed-Solomon Decoding, [2]. The parameter settings where our algorithm works is

\[ t \geq \frac{n + rk}{r + 1} \]  

Observe that for \( r = 1 \) the above bound on \( t \) coincides with the bound of the algorithm of [2], whereas when \( r > 1 \) less agreement is required (\( t \) is allowed to be smaller).

Let \( \langle y_{i,1}, \ldots, y_{i,r} \rangle_{i=1}^{n} \) be an instance of the SPR problem with parameters \( n, k, t, r \). Further observe that the condition on \( t \) above implies that

\[ r \geq \frac{n - t}{t - k} \]  

Define the following system of \( rn \) equations:

\[
\begin{align*}
[m_1(z_i) = y_{i,1}E(z_i)]_{i=1}^{n} \ldots [m_r(z_i) = y_{i,r}E(z_i)]_{i=1}^{n} \tag{\ast}
\end{align*}
\]

where the unknowns are the coefficients of the polynomials \( m_1, \ldots, m_r, E \). Each \( m_\ell \) is a polynomial of degree less than \( n - t + k \) and \( E \) is a polynomial of degree at most \( n - t \) with constant term equal to 1. It follows that the system has \( r(n - t + k) + n - t \) unknowns and thus it is not underspecified (i.e., the number of equations is at least as large as the number of unknowns); this follows from the condition on \( r \) in equation 2.

The algorithm for the SPR problem is then specified as follows:

**Input:** \( \langle y_{i,1}, \ldots, y_{i,r} \rangle_{i=1}^{n} \). Parameters \( n, k, t, r \in \mathbb{N} \), \( z_1, \ldots, z_n \in \mathbb{F} \).

**Step 0:** Randomize input: select \( q_1, \ldots, q_r \in \mathbb{F}^k[x] \) random polynomials and compute \( y_{i,\ell} := y_{i,\ell} + q_\ell(z_i) \) for all \( i \in [n], \ell \in [r] \).

**Step 1:** Form the linear system (\( \ast \)). Let the matrix of the system be \( A \).

**Step 2:** Eliminate a number of rows to obtain a square subsystem of (\( \ast \)) with corresponding square matrix \( \hat{A} \).

**Step 3:** If \( \hat{A} \) is singular then fail, otherwise compute solution of system.

**Step 4:** Parse solution of system as polynomials \( m_1, m_2, \ldots, m_r, E \) and return as solution the polynomials \( p_1 = m_1/E - q_1, p_2 = m_2/E - q_2, \ldots, p_r = m_r/E - q_r \). If this is not possible fail.

This completes the description of our algorithm. We argue about its correctness in the following two sections where we prove the feasibility of the system (\( \ast \)) and the uniqueness of solution. The exact choice of the square submatrix \( A \) in step 2 above will be given in section 2.3.

### 2.2 Feasibility

In this section we argue that for a given SPR instance \( \langle z_i, y_{i,1}, \ldots, y_{i,r} \rangle_{i=1}^{n} \), one of the possible outputs of the algorithm of section 2.1 is the solution of the SPR instance. Observe that due to definition 1, there exist \( I \subseteq [n] \) with \( |I| = t \) and \( p_1, \ldots, p_r \in \mathbb{F}[x] \) such that \( p_\ell(z_i) = y_{i,\ell} \) for \( i \in I \) and all \( \ell \in [r] \).
Given the existence of \( p_1, \ldots, p_r, I \) we will construct a solution of system (\ref{eq:system}), by defining explicitly the solution polynomials \( \tilde{E}, \tilde{m}_1, \ldots, \tilde{m}_r \).

Let \( \tilde{E}(x) := (-1)^{n-|I|} \prod_{i \notin I} (x/z_i - 1) \). Observe that \( \tilde{E} \) has constant term 1 and degree \( n-t \). Additionally it is easy to see that \( \tilde{E}(z_i) = 0 \) if and only if \( i \notin I \), i.e., \( \tilde{E} \) is an error-locator polynomial for the given instance.

Further, if \( \tilde{m}_\ell(x) := y_\ell(x) \tilde{E}(x) \) it holds that \( \tilde{m}_\ell(z_i) = p_\ell(z_i) \tilde{E}(z_i) = y_\ell(z_i) \tilde{E}(z_i) \), for all \( i = 1, \ldots, n \). The degree of \( \tilde{m}_\ell \) is less than \( n-t+k \). Observe that the polynomials \( \tilde{E}, \tilde{m}_1, \ldots, \tilde{m}_r \) constitute a possible solution of the system (\ref{eq:system}). Moreover (by construction) \( \tilde{m}_\ell(x)/\tilde{E}(x) = p_\ell(x) \) for \( \ell = 1, \ldots, r \) and as a result one of the possible outputs of the algorithm of section 2.1 is indeed the solution of the given SPR instance.

### 2.3 Uniqueness

In the previous section we have established that one of the possible outputs of our algorithm is the solution of the given SPR instance.

In this section we will show that the matrix constructed by the algorithm is with very high probability of full rank assuming that the SPR input to the algorithm is distributed according to the instance distribution \( D \). In a nutshell, we will present a technique for constructing a minor for the matrix of system (\ref{eq:system}) that is non-singular with high probability (this will be the square matrix employed in step 2 of the algorithm).

Below let \( \langle y_{1,1}, \ldots, y_{r,r} \rangle_{i=1}^n \) be an instance to the SPR problem and let \( A \) denote the matrix of the system of linear equations (\ref{eq:system}).

**Structure of \( A \).** We will start by investigating the structure of the matrix \( A \). Consider the following matrices, for \( \ell = 1, \ldots, r \):

\[
M = \begin{pmatrix}
1 & z_1 & z_1^2 & \cdots & z_1^{n-t+k-1} \\
1 & z_2 & z_2^2 & \cdots & z_2^{n-t+k-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & z_n & z_n^2 & \cdots & z_n^{n-t+k-1}
\end{pmatrix}
\]

\[
M_\ell = \begin{pmatrix}
y_{1,\ell} z_1 & y_{1,\ell} z_1^2 & \cdots & y_{1,\ell} z_1^{n-t} \\
y_{2,\ell} z_2 & y_{2,\ell} z_2^2 & \cdots & y_{2,\ell} z_2^{n-t} \\
\vdots & \vdots & \ddots & \vdots \\
y_{n,\ell} z_n & y_{n,\ell} z_n^2 & \cdots & y_{n,\ell} z_n^{n-t}
\end{pmatrix}
\]

Given these definitions, it follows that the matrix of the system (\ref{eq:system}) can be written as follows (where \( \mathbf{0} \) stands for a \( n \times (n-t+k) \)-matrix with \( \mathbf{0} \)'s everywhere):

\[
A = \begin{pmatrix}
M & \mathbf{0} & \cdots & \mathbf{0} & -M_1 \\
\mathbf{0} & M & \cdots & \mathbf{0} & -M_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & M & -M_r
\end{pmatrix}
\]

We index each row of \( A \) by the pair \( \langle i, \ell \rangle \) with \( i \in \{1, \ldots, n\} \) and \( \ell \in \{1, \ldots, r\} \). The \( \ell \)-th block row of \( A \) contains the rows \( \langle 1, \ell \rangle, \ldots, \langle n, \ell \rangle \).

**Choosing a Square submatrix of \( A \).** We will define a square sub-matrix \( \hat{A} \) of \( A \) by removing \( r(t-k)-(n-t) \) rows (recall that the condition \( t \geq (n+r)/(r+1) \) implies that \( r(t-k)-(n-t) \geq 0 \)) with the following criterion: we remove from
the \( \ell \)-th block row \( x_\ell \) rows indexed by \( (n-x_\ell+1, \ell), \ldots, (n, \ell) \). Let \( c = \lceil (r(t-k) - (n-t))/(t-k) \rceil \); observe that \( c \in \{0, 1, \ldots, r\} \). The sequence \( x_1, \ldots, x_r \) is defined as follows: \( x_r = \ldots = x_{r-c+1} = t-k, \ x_{r-c} = (r(t-k) - (n-t)) \mod (t-k) \), and \( x_{r-c-1} = \ldots = x_1 = 0 \).

Now observe that \( \hat{A} \) is a square matrix: first it has the same number of columns as \( A \), that is \( r(n-t+k)+n-\ell \); moreover, by construction, the number of rows of \( \hat{A} \) is equal to \( rn-r(t-k)+(n-t) = r(n-t+k)+n-t \).

Clearly if \( \hat{A} \) is non-singular, one may proceed to solve the linear system \((*)\) by solving the subsystem that corresponds to \( \hat{A} \). The square matrix \( \hat{A} \) will be used at step 2 of the algorithm in section 2.1.

The fact that \( \hat{A} \) is non-singular will be argued in the probabilistic sense based on the randomization that is performed by our algorithm prior to the construction of the system \((*)\) as well as the distributional properties of the given SPR instance that is assumed to follow the probability distribution \( \mathcal{D} \).

First we will define a family of rearrangements of the rows of \( \hat{A} \) according to an index-set \( I \) with \(|I|=t\). Note that what follows in the remaining of the section is for the sake of the presentation of the probabilistic argument - it does not affect the operation of the algorithm in any way (i.e., no rearrangement of the rows of \( \hat{A} \) is necessary from the algorithmic viewpoint; however rearranging the rows of \( \hat{A} \) allows the probabilistic argument to be presented more directly).

**Rearrangements the rows of the matrix \( \hat{A} \) according to \( I \).** Let \( I \subsetneq \{1, \ldots, n\} \) and \(|I|=t\). Observe that the number of rows in \( \ell \)-th block row of \( \hat{A} \) equals \( n-x_\ell \). The number \( n-x_\ell-(n-t+k)=t-k-x_\ell \) will be called the surplus of the \( \ell \)-th block row and will be denoted by \( s_\ell \). Note that the sum of all the surpluses satisfies

\[
\sum_{\ell \in [c]} s_\ell = r(t-k) - \sum_{\ell \in [c]} x_\ell = n-t
\]

The first \( r-c-1 \) block rows have surplus \( s_\ell = t-k \) while the block-rows \( r-c+1, \ldots, r \) have surplus \( s_\ell = 0 \).

Two rows \((i, \ell)\) and \((i', \ell')\) of the matrix \( A \) will be called *unrelated* if \( i \neq i' \). Now we are ready to define the reordered matrix \( \hat{A}^* \); informally, we will move a number of pairwise unrelated rows that is equal to the surplus to the lower part of the matrix so that exactly \( s_\ell \) rows are selected from the \( \ell \)-block row and moreover if the row \((i, \ell)\) is selected it holds that \( i \notin I \). From this point on, for the sake of clarity, in order to study the structure of the rearranged matrix \( \hat{A}^* \), we will make the assumption that \( I = \{n-t+1, \ldots, n\} \); the formal definition of the rearrangement as given below can immediately generalize to any \( I \). The rearrangement is also depicted in figure 3.

First recall that an \( h \)-Vandermonde matrix over \( a_1, \ldots, a_h \) is a matrix defined as follows:

\[
\begin{pmatrix}
1 & a_1 & \ldots & a_1^{h-1} \\
1 & a_2 & \ldots & a_2^{h-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & a_h & \ldots & a_h^{h-1}
\end{pmatrix}
\]
Let us denote by $N_{\ell}$ a $(n - t + k)$-Vandermonde matrix over the elements \( \{ z_1, \ldots, z_n \} \setminus \{ z_1 + (r - c - 1)(t - k), \ldots, z_{r-c-c} + (r - c - 1)(t - k) \} \) for $\ell \in [r - c - 1]$. Also let $N_{r-c}$ be the $(n - t + k)$-Vandermonde matrix over the elements,

\[
\{ z_1, \ldots, z_{n-x_{r-c}} \} \setminus \{ z_1 + (r - c - 1)(t - k), \ldots, z_{s_{r-c-c}} + (r - c - 1)(t - k) \}
\]

Moreover, for $\ell \in [r - c]$, let $N_{\ell}$ be the $(n - t + k)$-Vandermonde matrix over $\{ z_1, \ldots, z_{n-t} \}$.

Similarly, for $\ell \in [r - c]$, we define $M'_{\ell}$ to be the sub-matrix of $M_{\ell}$ with the rows $\langle x + (\ell - 1)(t - k), \ell \rangle$ removed for $x = 1, \ldots, t - k$. $M'_{r-c}$ is the sub-matrix of $M_{r-c}$ with the rows $\langle i, r - c \rangle$ removed, where

\[
i \in \{ 1 + (r - c - 1)(t - k), \ldots, s_{r-c-c} + (r - c - 1)(t - k) \} \cup \{ n - x_{r-c} + 1, \ldots, n \}
\]

Finally $M'_{\ell}$ for $\ell \in [r - c + 1, \ldots, r]$ is simply the submatrix of $M_{\ell}$ with the rows $\langle i, \ell \rangle$ removed where $i \in \{ n - t + k + 1, \ldots, n \}$.

Using the above notations the structure of the matrix $\hat{A}^*$ is presented below:

\[
\hat{A}^* = \begin{pmatrix}
N_1 & 0 & \ldots & 0 & -M'_{1} \\
0 & N_2 & \ldots & 0 & -M'_{2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & N_{r} & -M'_{r} \\
V_1 & V_2 & \ldots & V_{r} & -\hat{M}
\end{pmatrix}
\]
We will argue that $\hat{A}^{*}$ is non-singular (and drop the subscript $I$).

This completes the description of $\hat{A}^{*}$, a rearrangement of the rows of the square matrix $\hat{A}$ based on the index set $I = \{n-t+1, \ldots, n\}$. The rearrangement for an arbitrary index set $I$ with $|I| = t$ would work in the same way but it would rearrange the rows so that the indices appearing in the matrix $\hat{M}$ would be the set $\{1, \ldots, n\} - I$. In general we will denote the rearrangement of the rows of $\hat{A}$ according to $I$ by $\hat{A}^{*}_I$.

The following lemma is the basis of our probabilistic analysis. The main idea behind the proof is considering the determinant of a random matrix as a multivariate polynomial defined over random variables uniformly drawn from the underlying finite field and then applying Schwartz’s lemma to argue that it is unlikely that it will be 0 assuming the size of the field is large enough. With such an approach the only nuisance then becomes to prove that there exists a non-zero value of such a multivariate polynomial something that in the proof we demonstrate constructively.

**Lemma 1.** Fix $I \subseteq \{1, \ldots, n\}$ with $|I| = t$, and consider the random sequence of tuples $(y_{i,1}, \ldots, y_{i,r})_{i=1}^n$ that is distributed according to the following distribution over $(\mathbb{F}^{r+1})^n$: for $i \notin I$, each $y_{i,\ell}$ is uniformly distributed over $\mathbb{F}$ and for $i \in I$, each $y_{i,\ell} = p_{\ell}(z_i)$ where $p_1, \ldots, p_r$ are polynomials defined as $p_{\ell}(x) = a_{0,\ell} + a_{1,\ell}x + \ldots + a_{k-1,\ell}x^{k-1}$ with each $a_{j,\ell}$ uniformly distributed over $\mathbb{F}$. Constructed over such a random sequence of tuples, the matrix $\hat{A}$ defined as above is non-singular with probability at least $1 - (n-t)/|\mathbb{F}|$.

**Proof.** Given the matrix $\hat{A}$ we first rearrange its rows to obtain the matrix $\hat{A}^{*}_I$. Clearly, if $\hat{A}^{*}_I$ is non-singular then $\hat{A}$ will be non-singular. In the remaining of the proof we will consider without loss of generality that $I = \{n-t+1, \ldots, n\}$. We will argue that $\hat{A}^{*}$ is non-singular (and drop the subscript $I$).
Recall based on the distributional conditions of the theorem’s statement it holds that \( y_{1, \ell}, \ldots, y_{n-t, \ell} \) for \( \ell \in [r] \), are uniformly distributed over \( \mathbb{F} \) and furthermore \( y_{i, \ell} = a_0 + a_1 z_1^i + \cdots + a_{k-1, \ell} z_{k-1}^i \) for \( i \in I, \ell \in [r] \), where \( a_{0, \ell}, \ldots, a_{k-1, \ell} \) for \( \ell \in [r] \), are uniformly distributed over \( \mathbb{F} \).

Now, observe that the determinant of \( \hat{A}^* \) can be seen as a multivariate polynomial \( M_{\hat{A}} \), over the variables \( y_{i, \ell}, a_{j, \ell} \) where \( i \in [n-t], j \in [k-1] \cup \{0\} \) and \( \ell \in [r] \). In order to argue about the non-singularity of \( \hat{A}^* \), we will first show that \( M_{\hat{A}}, \neq 0 \) where \( 0 \) stands for the zero polynomial. First note that if we perform any number of linear row operations in the matrix \( \hat{A}^* \) that result in a matrix \( A' \), then the multivariate polynomial \( M' \) that corresponds to the resulting matrix \( A' \) has the property that \( M' \neq 0 \iff M_{\hat{A}} \neq 0 \).

Furthermore, in order to show that \( M_{\hat{A}}, \neq 0 \) it suffices to find an assignment to the variables of \( M_{\hat{A}} \), for which it holds that the polynomial evaluates to a non-zero value. We will use the following assignment: for all \( \ell \in [r] \) such that \( s_{\ell} \neq 0 \) and \( u = 1 + (\ell - 1)(t - k), \ldots, s_{\ell} + (\ell - 1)(t - k) \) we will set \( y_{u, \ell} = 0 \); observe that the only \( u \)'s that satisfy the given condition have the property that \( u \leq n - t \), therefore we are allowed to do any assignment to \( y_{u, \ell} \) (since it is a free-variable). The remaining variables \( y_{i, \ell} \) with \( i \leq n - t \) and \( \ell \in [r] \) will be set to 1. Finally we set \( a_{0, \ell} = 1 \) and \( a_{j, \ell} = 0 \) for all \( j \in [k-1] \cup \{0\} \) and \( \ell \in [r] \) (observe that this assignment forces \( y_{i, \ell} = 1 \) for \( i \in I \) and \( \ell \in [r] \)). Given this assignment observe that the sub-matrix \( \hat{M} \) is equal to the zero \((n-t) \times (n-t)\) matrix. Furthermore, the matrix \( M_{1}' \) will be equal to

\[
\begin{pmatrix}
y_{t-k+1,1} z_{t-k+1} & \cdots & y_{t-k+1,1} z_{t-k+1}^{n-t} \\
y_{t-k+2,1} z_{t-k+2} & \cdots & y_{t-k+2,1} z_{t-k+2}^{n-t} \\
y_{n,1} z_{n} & \cdots & y_{n,1} z_{n}^{n-t}
\end{pmatrix}
\begin{pmatrix}
y_{t-k+1,1} z_{t-k+1}^{n-t} \\
y_{t-k+2,1} z_{t-k+2}^{n-t} \\
y_{n,1} z_{n}^{n-t}
\end{pmatrix}
\]

and similarly for the other matrices \( M_{2}', \ldots, M_{r}' \).

Let us now consider a linear row operation inside \( \hat{A}^* \) that will eliminate the first row of \( V_1 \) which is equal to \( (1, z_1, \ldots, z_{n-t+k-1}) \). To accomplish this, we find \( \lambda_{t-k+1}, \ldots, \lambda_{n} \) such that \( \sum_{j=t-k+1}^{n} \lambda_{j} z_{j}^m = -z_{1}^{m} \) for each \( m \in [n-t+k-1] \cup \{0\} \). It follows easily (given the structure of \( M_{1}' \)) that after the elimination of the first row of \( V_1 \) the first row of \( \hat{M} \) becomes equal to \( (z_{1}, \ldots, z_{n-t}) \). By applying the same elimination method to the remaining non-zero rows of \( V_1, V_2, \ldots, V_r \), it is easy to observe that given the structure of the \( M_{1}', \ldots, M_{r}' \) matrices, after we complete the elimination, the matrix \( \hat{M} \) will be transformed to the non-singular matrix

\[
\begin{pmatrix}
z_1 & z_1^2 & \cdots & z_1^{n-t} \\
z_2 & z_2^2 & \cdots & z_2^{n-t} \\
\vdots & \vdots & \ddots & \vdots \\
z_{n-t} & z_{n-t}^2 & \cdots & z_{n-t}^{n-t}
\end{pmatrix}
\]

The above linear transformations on the matrix \( \hat{A}^* \) for the given assignment to its variables define a matrix \( A' \) that is non-singular: indeed it is block-
triangular and its block diagonal is comprised of non-singular square matrices. From this we deduce that the multivariate polynomial that corresponds to $A'$ is non-zero and as a result the polynomial $M_\hat{A}$ is non-zero.

Now observe that the polynomial $M_\hat{A}$ has combined degree $n - t$; this is so, by expanding the determinant that defines $M_\hat{A}$. By Schwartz’s Lemma [13], it follows that $M_\hat{A}$ it cannot be 0 in more than a $(n - t)/|F|$-fraction of its domain. As a result det($\hat{A}^*$) will be 0 with probability at most $(n - t)/|F|$. ■

**Theorem 1.** Assuming that the given SPR instance is distributed according to $\mathcal{D}$, the system $(*)$ constructed by our algorithm accepts at most one solution with probability at least $1 - (n - t)/|F|$.

**Proof.** The matrix $A$ of the system $(*)$ is of dimensions $rn \times r(n - t + k) - (n - t)$; let $\hat{A}$ its square submatrix as selected above. It holds that for any $I$, lemma 1 suggests that $\hat{A}$ is singular with probability at most $(n - t)/|F|$. It follows that $A$ is of full rank with probability at least $1 - (n - t)/|F|$ and as a result it can accept at most one solution. Observe that step 0 of the algorithm totally randomizes the solution polynomials thus essentially transforming $D_{msg}$ to a uniform distribution as required in the proof of lemma 1. ■

**2.4 Correctness**

The non-singularity of $\hat{A}$ as suggested by theorem 1 is not sufficient to ensure the existence of a solution of the system $(*)$. Nevertheless we know that $(*)$ accepts at least one solution (as constructed explicitly in section 2.2). Based on these results, it will follow that system $(*)$ has a unique solution (that coincides with the solution constructed in section 2.2) and this solution can be found by solving the linear system with $\hat{A}$ as its square matrix.

**Theorem 2.** The probabilistic algorithm of section 2.1 when given input distributed according to $\mathcal{D}$ returns the solution of the given SPR instance with probability at least $1 - (n - t)/|F|$.

**Proof.** It follows from the construction of the feasibility section 2.2 and theorem 1. ■

We remark that the efficiency of our algorithm can be further improved. Indeed, it is not necessary to solve the linear-system with matrix $\hat{A}$ directly; instead, we can derive easily a system of $n - t$ equations that completely determines the polynomial $E$; it is obvious that the recovery of $E$ will reveal all solutions of the given SPR instance. This is so, since finding all roots of $E$ will reveal the error-locations of the given SPR-instance and then the recovery of $p_1, \ldots, p_r$ can be done by interpolation. A system of $n - t$ equations that determines $E$ completely can be found by eliminating all variables that correspond to the polynomials $m_\ell$ from at most $t - k$ rows of the $\ell$-th block row of matrix $\hat{A}$, for $\ell = 1, \ldots, r$. Such elimination will be possible for exactly $n - t$ rows.
3 Decoding Interleaved RS-Codes in the NBSC model

In this section we present a coding theoretic application of our algorithm of section 2 in the setting of interleaved Reed-Solomon Decoding. First we recall the notion of interleaved codes.

3.1 Interleaved codes

Interleaved codes are not an explicit family of codes, but rather an encoding mode that can be instantiated over any concrete family of codes. The mode can be applied to any family of codes; in this section we give a code independent description.

Let $\Sigma, \Sigma'$ be two alphabets with $|\Sigma'| = \sqrt{|\Sigma|}$. Let $\phi : \Sigma \to (\Sigma')^r$ be some 1-1 mapping. We use the notation $\phi(x) := x^0[1]x^0[2] \ldots x^0[r]$, where $x^0[\ell] \in \Sigma'$, for $\ell = 1, \ldots, r$, for any $x \in \Sigma$.

Now let $\text{enc} : (\Sigma')^k \to (\Sigma')^n$ be an encoding function. An interleaved code w.r.t. $\phi$ for $\text{enc}$ is a function $\text{enc}_\phi : \Sigma^k \to (\Sigma)^n$ that is defined as follows: Let $m_0m_1 \ldots m_{k-1} \in (\Sigma)^k$. First the following strings of $(\Sigma')^n$ are computed:

$$c_{1,1} \ldots c_{n,1} = \text{enc}(m_0^0[1] \ldots m_{k-1}^0[1])$$

$$\vdots$$

$$c_{1,r} \ldots c_{n,r} = \text{enc}(m_0^0[r] \ldots m_{k-1}^0[r])$$

The interleaved encoding is defined as follows:

$$\text{enc}_\phi(m_0m_1 \ldots m_{k-1}) = \phi^{-1}(c_{1,1} \ldots c_{1,r}) \ldots \phi^{-1}(c_{n,1} \ldots c_{n,r})$$

A graphical representation of code interleaving is presented in figure 4.

![Fig. 4. Encoding schema for an interleaved code. Single subscript symbols ($m_i, c_i$) belong to the “outer” alphabet $\Sigma$; double subscript symbols ($m_{i,j}, c_{i,j}$) belong to the “inner” alphabet $\Sigma'$.](image-url)
Such interleaved encodings will be said to be of degree $r$ over the alphabet $\Sigma'$ (we will also call it “amount of interleaving”).

The common way to use an interleaved code, is simply decode each of the code words $(c_{1,i}, \ldots, c_{n,i})$ separately. Such a decoding does not increase the error correction rate. The advantage is the fact that burst errors are distributed over several code words, and therefore employing interleaving over bursty channels increases the chances of error-correction.

We emphasize here that under reasonable channel assumptions it might be possible to take advantage of interleaving and attempt to correct all code words simultaneously. Indeed, in contrast to the standard approach of decoding each one of the codewords individually, we will present a decoding technique that attempts to correct all codewords simultaneously assuming that the NBSC model describes the transmission channel in the setting of Reed-Solomon Codes. This methodology will increase the possible error-rates that the interleaved code can withstand.

An extended version of the above schema, known as cross-interleaving is to replace the 1-1 mapping $\phi^{-1}$ with a second error correcting code. Cross-interleaving increases the code size, but allows to correct a larger class of errors. By testing the second (outer) code potential error locations can be found. Then decoding the inner code can be done by treating those potential error locations as erasures. This increases the error correcting capabilities of the scheme, because generally a code allows more erasures than errors. The present work focuses on simple interleaving (without an outer code).

### 3.2 Interleaved Reed-Solomon Codes

In this section we focus on explaining interleaving in the context of Reed-Solomon Codes. Let $\Sigma = GF(2^B)$ be the alphabet for the encoding function (without loss of generality we will focus only on binary extension fields — all our results hold also for general finite fields). The parameters are $n, k \in \mathbb{N}$ where $\kappa := k/n$ is the message rate. We assume additionally a parameter $r \in \mathbb{N}$ with the property that it splits $B$ as follows $r = B/b$ (in fact this is not a necessary requirement but it simplifies the presentation so we will adopt it). Let $z_1, \ldots, z_n \in GF(2^b)$ be fixed distinct constants.

We now describe the case of interleaved Reed-Solomon Codes. First, recall that there exists a straightforward bijection mapping $\phi : GF(2^B) \rightarrow (GF(2^b))^r$. If $m \in GF(2^B)$ we define by $m_0[\ell]$ the element of $GF(2^b)$ which is the $\ell$-th coordinate of $m$ under the bijection $\phi$. Given $m_0 \ldots m_{k-1} \in GF(2^B)$ we define the following polynomials over $GF(2^b)$, for $\ell = 1, \ldots, r$:

$$p_\ell(x) := m_0[\ell] + m_1[\ell]x + \ldots + m_{k-1}[\ell]x^{k-1}$$

The encoding of $m_0 \ldots m_{k-1}$ is set to be the string over $GF(2^B)^n$,

$$\phi^{-1}(p_1(z_1) \ldots p_r(z_1)) \ldots \phi^{-1}(p_1(z_n) \ldots p_r(z_n))$$
The straightforward way to decode RS-interleaved-codes is to concentrate in each of the \( r \) coordinates individually and employ the decoding algorithm of the underlying RS-Code over \( \text{GF}(2^b) \). This can be done as follows: given a (partially corrupted) codeword \( c_1 \ldots c_n \in (\text{GF}(2^B))^n \) we treat the string \( c_{\phi[1]}^0[1] \ldots c_{\phi[n]}^0[1] \in (\text{GF}(2^B))^n \) as a partially corrupted RS-codeword over \( \text{GF}(2^b) \) and we employ the RS-Decoding of Berlekamp-Welch to recover \( p_1 \). Observe that the recovery of \( p_1 \) will imply the recovery of \( p_2, \ldots, p_r \) immediately, provided that the error-rate is at most \((1 - \kappa)/2\) (recall that the errors are induced over the channel that transmits \( \text{GF}(2^B) \) symbols; based on this, it is easy to see that all codewords \( c_{\phi[1]}^0[\ell] \ldots c_{\phi[n]}^0[\ell] \) for each \( \ell = 1, \ldots, r \) will have identical error-pattern; it follows that the recovery of \( p_1 \) reveals the error-pattern of the first coordinate and thus by interpolation it is possible to recover all remaining \( p_2, \ldots, p_r \).

Moreover, if unique solvability is somehow assured with high probability, e.g., assuming the NBSC model, one can further employ the Guruswami-Sudan list-decoding algorithm that will produce a unique solution with high probability for error-rates up to \( 1 - \sqrt{\kappa} \). The main focus of the next section is to go beyond this bound.

### 3.3 The Decoding Algorithm

In this section we reduce the problem of decoding interleaved Reed-Solomon Codes in the NBSC model to the problem of Simultaneous Polynomial Reconstruction. In the light of this reduction, our algorithm for the latter problem will provide a decoder for interleaved RS-codes.

Consider interleaved RS-Codes with parameters \( r, n, k \in \mathbb{N} \), where \( r \) is the amount of interleaving. Also let \( \phi : \text{GF}(2^B) \rightarrow \text{GF}(2^b)^r \) be the bijection mapping employed for the interleaving. We will suppose that the interleaved RS encoded codewords will be transmitted over a channel with symbol alphabet \( \text{GF}(2^B) \) that follows the NBSC model with error-rate \( \epsilon \) (refer to figure 1). Recall that, in the NBSC model each symbol that is placed on the channel will be delivered without any alteration with probability \( 1 - \epsilon \). We call this event a direct transmission. On the other hand with probability \( \epsilon \) the symbol will be dropped by the channel and substituted by a uniformly random element from the channel alphabet (note that there is still a small probability of correct transmission).

**Lemma 2.** Let \( c_1 \ldots c_n \in \text{GF}(2^B)^n \) be the encoding of an arbitrary message from \( \text{GF}(2^B)^k \) using the interleaved RS encoding schema with parameters \( n, k, r \) and \( z_1, \ldots, z_n \in \text{GF}(2^b) \). Let \( c_1^* \ldots c_n^* \in \text{GF}(2^B)^n \) be a random variable that is drawn from the conditional probability space induced by the NBSC transmission of \( c_1 \ldots c_n \) with error probability \( \epsilon \in \mathbb{Q} \) conditioned on the number of direct transmissions being \( t \). Then, it holds that the random variable \( \langle \phi(c_1^*)[1], \ldots, \phi(c_n^*)[r] \rangle_{i=1}^n \) is distributed identically to the probability distribution of SPR instances \( \mathcal{D} \) over \( (\text{GF}(2^b)^r)^n \) with parameters \( n, k, t, r \) and \( z_1, \ldots, z_n \in \text{GF}(2^b) \).

**Proof.** Suppose that the message to be transmitted using the interleaved RS code is \( \langle p_1, \ldots, p_r \rangle \in (\mathbb{F}[x])^r \). Based on the interleaved schema the message will be encoded as a \( n \)-symbol long sequence over \( \text{GF}(2^b) \) as follows:
The NBSC transmission of the above codeword with error rate $\epsilon$ can be simulated by the following probabilistic procedure (assuming that $\epsilon \in \mathbb{Q}$ and $u \in \mathbb{Z}$ is such that $u \cdot \epsilon^{-1} \in \mathbb{Z}$):

**Input.** Codeword $c_1 \ldots c_n \in GF(2^B)^n$.
1. For $i = 1, \ldots, n$ choose $r_i \leftarrow R \{u \epsilon^{-1} 2^B\}$.
2. For $i = 1, \ldots, n$, if $r_i \in [(\lambda - 1)u, \ldots, \lambda \cdot u]$ for some $\lambda \in [2^B]$ set $c_i^* := a_\lambda$ otherwise set $c_i^* := c_i$. Note that $GF(2^B) = \{a_1, \ldots, a_{2^B}\}$.
3. Output $c_1^* \ldots c_n^*$.

It is easy to see that the above procedure simulates the NBSC transmission with error probability $\epsilon$ when a codeword of length $n$ is transmitted. A direct transmission of a symbol $c_i$ in the above sampling procedure corresponds to selecting the randomness $r_i$ to belong to $\{2^Bu + 1, \ldots, u \epsilon^{-1} 2^B\}$ an event that has probability $(2^B \epsilon^{-1} u - 2^Bu)/(2^B \epsilon^{-1} u) = (\epsilon^{-1} - 1)/(\epsilon^{-1}) = 1 - \epsilon$. The conditional probability space assuming that exactly $t$ direct transmissions have occurred means that we restrict the above probability space to those tuples $[r_1, \ldots, r_n]$ for which it holds that exactly $n - t$ among $r_i$ belong to $[2^Bu]$ and the remaining $t$ belong to $[\epsilon^{-1} u 2^B] - [2^Bu]$. Based on the above, it is easy to verify that the sequence of tuples

$$\langle \phi(c_1^*)[1], \ldots, \phi(c_n^*)[r]\rangle_{i=1}^n$$

distributed according to the conditional NBSC distribution with $t$ direct transmissions is identical to the distribution $D$ for the SPR problem with parameters $n, k, t, r$ and $z_1, \ldots, z_n \in GF(2^B)$.

The above lemma indicates that we can employ any channel over symbols of $GF(2^B)$ that simulates the NBSC model to transmit arbitrary message distributions provided that we are given a guarantee that exactly $t$ direct transmissions will occur when sending $n$ symbols. Indeed, as suggested by the lemma, it holds that we can employ an interleaved RS code with parameters $n, k, t, r$ over $GF(2^B)$ and expect to correct the transmission with probability at least $1 - \frac{n}{2^t}$ provided that the parameters satisfy $t \geq \frac{n+rk}{r+1}$. Of course the specification of the NBSC model does not offer such a guarantee regarding the number of direct transmissions. Nevertheless it suggests that the expected number of direct transmissions is $(1 - \epsilon)n$.

This gives rise to the following decoder algorithm for interleaved RS codes with parameters $n, k, r, \epsilon$ over the NBSC model for a channel with alphabet $GF(2^B)$ and error-rate $\epsilon \in \mathbb{Q}$.

**Interleaved RS decoder.** Parameters $n, k, r, \epsilon, z_1, \ldots, z_n \in GF(2^B)$.

**Input:** string $c_1^* \ldots c_n^* \in GF(2^B)^n$ that is the output of a $n$-symbol NBSC transmission of a codeword $c_1 \ldots c_n$ that encodes a message drawn from
an arbitrary distribution $D_{msg}$ over $GF(2^B)^k$ according to the interleaved RS encoding schema with parameters $n, k, r$ and $z_1, \ldots, z_n \in GF(2^B)$.

**Algorithm.** Apply the SPR algorithm of section 2.1 for parameters $n, k, t, r$ by setting $t$ to its expected value $(1-\epsilon)n$ to the input $\langle \phi(c_i^*)[1], \ldots, \phi(c_i^*)[r]\rangle_{i=1}^n$. If the algorithm fails repeat with the same input and increase/decrease $t$ doing a “left-right” search of the set $\{(n+rk)/(r+1), \ldots, n\}$.

Note that if $v$ is a variable initialized to a value $v_0$, a left-right search of the set $\{v_L, \ldots, v_R\}$ such that $v_0 \in \{v_L, \ldots, v_R\}$ involves the assignment of the following values to $v$: $(v_0, v_0 + 1, v_0 - 1, v_0 + 2, v_0 - 2, \ldots, v_L, \ldots, v_R)$.

**Theorem 3.** The interleaved RS decoder presented above with parameters $n, k, r \in \mathbb{N}$ and $z_1, \ldots, z_n \in GF(2^B)$, employed over an NBSCh channel with symbol alphabet $GF(2^B)$ can correct any error-rate $\epsilon$ up to

$$\epsilon < \frac{r}{r + 1}(1 - \kappa)$$

with probability at least $1 - (1 - \kappa) \cdot n/2^b - \delta(n, r, \kappa, \epsilon)$, where $n$ is the block length and $\delta(n, r, \kappa, \epsilon)$ is a function that expresses the probability that a codeword of length $n$ transmitted in the NBSC channel with error-rate $\epsilon$ is corrupted by a number of errors $e$ beyond the solvability bound of our SPR algorithm which is $e \leq n - (n + kr)/(r + 1)$. The function $\delta$ drops exponentially and it holds that $\delta(n, r, \kappa, \epsilon) \leq \exp(-e\beta^2 n)$ where $\beta = r(1 - \kappa)/(r + 1)(1 - \epsilon) - \epsilon/(1 - \epsilon)$.

**Proof.** First let us fix a certain block length $n \in \mathbb{N}$. Suppose a codeword of length $n$ is transmitted over the NBSC. Let us condition on the event that the channel does any number $t$ of direct transmissions that belongs to $\{(n+kr)/(r+1), \ldots, n\}$.

Since (i) our interleaved decoder will effectively try every possible $t$ (starting from the most likely) and (ii) the decoder never returns false output (all decoders are “Las-Vegas” algorithms) it is easy to see that the SPR algorithm will eventually be simulated on the correct $t$ and as a result it will hold that the probability of success of the interleaved decoder is exactly the one that the SPR algorithm has for the conditioned probability space where exactly $t$ direct transmissions occur. Based on theorem 2 and lemma 2 we obtain that the interleaved decoder above will have success probability $1 - (n-t)/2^b$. Since we conditioned on the fact that $t \geq (n+kr)/(r+1)$ it holds that $1 - (n-t)/2^b \geq 1 - nr(1 - \kappa)/((r+1)2^b)$ which is greater or equal to $1 - (1 - \kappa)n/2^b$. It follows that the success probability of the algorithm will be equal to $(1 - (1 - \kappa)n/2^b)/(1 - \delta(n, r, k, \epsilon))$ from which the statement of the theorem follows easily.

Regarding bounding the function $\delta$, observe that $t$ in the NBSC channel follows a binomial probability distribution with success probability $(1 - \epsilon)$ and the event $e > n - (n + kr)/(r + 1)$ is equivalent to the event $t < (n + kr)/(r + 1)$. It follows that the probability of the event $t < (n + kr)/(r + 1)$ will be less than $\exp(-e\beta^2 n)$ by a direct application of the Chernoff bound.
Observe that \( \lim_{r \to \infty} \frac{r}{r+1} (1 - \kappa) = 1 - \kappa \). It follows from the above theorem that our decoding algorithm will be capable decoding any error rate arbitrarily close to the information-theoretic bound \( 1 - \kappa \) as follows: first select an amount of interleaving \( r \) so that the bound of theorem 3 is satisfied. Then select \( n \) and \( b \) depending on the choice of \( r \) so that the success probability of theorem 3 becomes sufficiently large.

**Example:** Suppose that the message-rate is \( 1/4 \) and the error-rate is \( 11/16 \). We employ the interleaved RS-schema for \( r = 11 \) with alphabets \( \Sigma = GF(2^B) = GF(2^{40}) \) and \( \Sigma' = GF(2^b) = GF(2^{40}) \). Observe that such error-rates are not correctable by considering the interleaved codewords individually (indeed, even list-decoding algorithms, e.g. the list-decoder of [6] would work only for error-rates up to 1/2). Suppose now that the block-size is \( n = 64 \). Our probabilistic decoding algorithm for such interleaved RS-codes corresponds to solving the SPR problem on parameters \( n = 64, k = 16, t = 20, r = 11 \) over the finite-field \( GF(2^{40}) \) and thus we will succeed in decoding with probability least \( 1 - 2^{-34} \) conditioning on the fact that the number of errors is always within the bound of the solvability of our SPR algorithm \( e \leq n - \frac{2 + kr}{r+1} \).

**Remark:** We note that employing our methodology, setting and analysis techniques in other cases (i.e. simultaneous decoding of all interleaved codewords for other families of interleaved codes in the NBSC model) is an interesting research direction.

**Remark:** As noted in the introduction, an independent solution of the Simultaneous Polynomial Reconstruction Problem was presented recently by Coppersmith and Sudan in [4]. The probabilistic analysis described in [4] for the stronger of the two algorithms they present assumes a similar distributional assumption as the present work and requires \( t > r + \sqrt{nk^r} + k + 1 \). It follows that they can correct any error-rate \( \epsilon < 1 - \kappa - \kappa^{1+r} \). Observe that \( \lim_{r \to \infty} (1 - \kappa - \kappa^{1+r}) = 1 - 2\kappa \) and thus the information theoretic bound cannot be reached by the analysis of [4]. It should be noted though that, on the one hand the analysis may potentially be improved (this is an open question) and on the other hand, the algorithm of [4] is not geared towards asymptotically reaching the channel capacity \( 1 - \kappa \) but rather it is designed for decoding instances with vanishing message-rates, high noise and small interleaving (cf. figure 2).

**Acknowledgement**

The authors wish to thank Alexander Barg and Jorn Justesen for helpful discussions.

**References**


5. G. David Forney, Concatenated Codes, MIT Press, Cambridge, MA, 1966


