Robust decoding of DFT-based error-control codes for impulsive and additive white Gaussian noise channels

P. Azmi and F. Marvasti

Abstract: A novel decoding technique for DFT-based error control codes that is robust against quantisation noise and additive white Gaussian noise is proposed. The algorithm simultaneously determines the number and the positions of the corrupted samples. In contrast to the conventional decoding technique, the proposed decoding method is stable in the presence of both quantisation noise and additive white Gaussian noise.

1 Introduction

The discrete Fourier transform (DFT) is commonly used for real field error-control codes. In this coding scheme code vectors are produced by padding zeros to the information vectors in DFT domain [1–4]. It has been shown that the decoding of this scheme reduces to solving a set of t Toeplitz equations, where t is the number of errors or missing samples [3, 4]. Theoretically, the number of corrupted samples by impulsive noise t can be estimated using the well-known recursive methods such as the Levinson–Durbin, Berlekamp–Massey and Euclidian algorithms [5]. But in practice, because of the presence of additive noise and quantisation noise, these algorithms become unstable and fail to estimate the number of errors. By impulsive noise we mean that a finite number of samples are either erased or drastically changed, while the remaining samples are almost similar to the original samples and are only different slightly due to quantisation noise and additive white Gaussian noise. Impulsive noise is a realistic model for a non-binary channel with non-binary modulations such as QAM with hard decisions.

This deficiency has encouraged us to consider a novel decoding technique for real field error-control codes under the impulsive channel model. It is shown that the proposed method can properly estimate the number and positions of corrupted samples in the presence of both quantisation noise and additive white Gaussian noise.

2 Proposed decoding method

In a (N,K) DFT-based error-control code, each information vector K-tuple $\boldsymbol{u}$ is encoded into an N-tuple $\boldsymbol{v}$, called a codevector (codeword), where $N > K$. The encoding procedure is as follows:

Step 1: Take the DFT of $\boldsymbol{u}$ to get a K-tuple $\boldsymbol{U}$.

Step 2: Insert N-K consecutive zeros to get an N-tuple $\boldsymbol{V}$.

Step 3: Take the inverse DFT to get an N-tuple codeword $\boldsymbol{v}$.

In the decoding procedure the following steps are usually taken:

Step 1: An error locator polynomial is generated.

Step 2: The errors due to lost samples are corrected.

We propose a novel decoding technique that has the same decoding steps as the conventional technique. However, in contrast to the conventional technique, a new error locator polynomial is generated and several new error-correction procedures are applied.

2.1 New error locator polynomial

Let $\boldsymbol{r}$ be the received vector and suppose that an unknown error vector $\boldsymbol{e}$ is introduced as follows:

$$\boldsymbol{r} = \boldsymbol{v} + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{v}$ is the transmitted codeword. We assume that the error vector $\boldsymbol{e}$ is due to lost samples in an impulsive channel. In this case the ith component of the error vector, $e(i)$ is zero where i denotes the set of integers related to the positions of good samples.

Let $\boldsymbol{E}$ be the DFT of the error vector $\boldsymbol{e}$ that coincides with the DFT of the received vector $\boldsymbol{R}$ in the positions that zeros are added in the encoding procedure. For (N,K) DFT-based code in the proposed decoding method we produce an error locator polynomial as follows:

$$S(z) = \sum_{r=0}^{\lfloor \frac{N-K}{2} \rfloor} h_r z^r = 1 + \sum_{r=1}^{\lfloor \frac{N-K}{2} \rfloor} h_r z^r \tag{2}$$

where each $h_i\, i = 0, 1, 2, \ldots, \lfloor (N-K)/2 \rfloor$ satisfies the following equation:

$$S(e^{j\frac{2\pi}{N} i_m}) = \sum_{r=0}^{\lfloor \frac{N-K}{2} \rfloor} h_r e^{j\frac{2\pi}{N} i_m r} = 0 \tag{3}$$

IEE Proceedings online no. 20045160

doi:10.1049/ip-com:20045160

Paper first received 14th August and in revised form 20th October 2004

P. Azmi is with the Electrical and Computer Engineering Department, Tarbait Modarres University, P.O. Box 14115-143, Tehran, Iran and the Signal Processing and Multimedia Lab., Iran Telecommunication Research Centre, Tehran, Iran

F. Marvasti is with the Electrical Engineering Department, Sharif University of Technology, Tehran, Iran and the Advanced Communications Research Institute (ACRI), Sharif University of Technology

E-mail: pazmi@modares.ac.ir

and $\{i_m : m = \{1, 2, \ldots, t\} \quad t \le \lfloor (N-K)/2 \rfloor\}$ denotes the positions of the lost samples in the impulsive channel and t denotes the number of lost samples. In this case $S(z)$ can be written as

$$S(z) = \left\{ \prod_{m=1}^{t} \left(1 - z e^{-j\frac{2\pi i_m}{N}}\right) \right\} P(z) \tag{4}$$

where $P(z)$ is a polynomial such that its order is

$$\left\lfloor \frac{N-K}{2} \right\rfloor - t \tag{5}$$

and from (4) and (2), because $S(0) = 1$,

$$P(0) = 1 \tag{6}$$

In the conventional decoding method the upper limits of summations (2) and (3) are t. Therefore the proposed method is a generalisation of the conventional technique and if $\lfloor (N-K)/2 \rfloor = t$ the proposed method reduces to the conventional method discussed in [1–4].

As with the conventional technique, multiplying (3) by $e(i_m)\exp(-j2\pi i_m p/N)$ and summing over i_m gives the following recursive equations:

$$\begin{aligned} &\sum_{\{i_m: m=1,2,\ldots,t \quad t \le \lfloor \frac{N-K}{2} \rfloor\}} e(i_m) e^{-j\frac{2\pi}{N} p i_m} \sum_{r=0}^{\lfloor \frac{N-K}{2} \rfloor} h_r e^{j\frac{2\pi}{N} i_m r} \\ &= \sum_{r=0}^{\lfloor \frac{N-K}{2} \rfloor} h_r \sum_{\{i_m: m=1,2,\ldots,t \quad t \le \lfloor \frac{N-K}{2} \rfloor\}} e(i_m) e^{-j\frac{2\pi}{N} p i_m} e^{j\frac{2\pi}{N} i_m r} \\ &= \sum_{r=0}^{\lfloor \frac{N-K}{2} \rfloor} h_r \sum_{i=0}^{N-1} e(i) e^{-j\frac{2\pi}{N}(p-r)i} \\ &= \sum_{r=0}^{\lfloor \frac{N-K}{2} \rfloor} h_r E(p-r) = 0 \quad p = 0, 1, \ldots, N-1 \end{aligned} \tag{7}$$

where $E(k)$ is the kth component of $\boldsymbol{E}$ that is the DFT of the error vector $\boldsymbol{e}$. In deriving the recursive equations (7) note that $e(i)$ is zero in the positions where

$$i \notin \{i_m : m = \{1, 2, \ldots, t\} \quad t \le \lfloor \tfrac{N-K}{2} \rfloor\}$$

In the proposed decoding method, for (N, K) DFT-based code, the idea is to use recursive equations

$$\sum_{r=0}^{\lfloor \frac{N-K}{2} \rfloor} h_r E(p-r) = 0 \quad \text{for } p = 0, 1, \ldots, N-1$$

to determine $\lfloor (N-K)/2 \rfloor$ unknown coefficients h_r. To solve these equations we need to know only N–K samples of $\boldsymbol{E}$ in the positions where zeros are added in the encoding procedure. As one can see, the proposed decoding technique is a generalisation of the conventional decoding technique discussed in [1–4]. In other words, in the conventional decoding method, the decoder uses $\sum_{r=0}^{t} h'_r E(p-r) = 0$ for $p = 0, 1, \ldots, N-1$ to determine t unknown coefficients h'_r. Therefore, in principle, it is necessary to know the number of lost samples. But in contrast to the conventional method, in the proposed method the upper bound of the summation of the recursive equations (that is equal to the number of unknown coefficients h_r) is $\lfloor (N-K)/2 \rfloor$. Therefore it is not necessary to know the number of lost samples beforehand.

Because $\boldsymbol{e}$ is real, the equations are Toeplitz and hermitian and therefore (assuming N is even)

$$\begin{aligned} &E(N/2 - r - 1) = E^*(N/2 + r) \quad r = 0, 1, \ldots, N/2 - 1 \\ &E(0) = E^*(0) \end{aligned} \tag{8}$$

Thus the linear equations shown in (7) are Yule–Walker equations and can be rewritten as

$$\boldsymbol{R} * \boldsymbol{H} = -\boldsymbol{E} \tag{9}$$

where

$$\boldsymbol{E} = \left[E(N/2) \quad E(N/2+1) \quad \ldots E\left(N/2 + \left\lfloor \frac{N-K}{2} \right\rfloor + 1\right) \right]^T \tag{10}$$

$$\begin{aligned} \boldsymbol{R} &= [R_{ij}] \\ R_{ij} &= E(N/2 + i - j - 1) \quad 1 \le i, j \le \left\lfloor \frac{N-K}{2} \right\rfloor \end{aligned} \tag{11}$$

and

$$\boldsymbol{H} = \left[h_1 \quad h_2 \quad \ldots \quad h_{\lfloor \frac{N-K}{2} \rfloor} \right]^T \tag{12}$$

where $h_i \; i = 1, 2, \ldots, \lfloor (N-K)/2 \rfloor$ are unknown coefficients of (2) and (7). In the absence of additive and quantisation noise, $\boldsymbol{E}$ will be an autoregressive (AR) process with order t. Therefore, if $t < \lfloor (N-K)/2 \rfloor$, the matrix $\boldsymbol{R}$ will be singular [4]. In this case there are infinite solutions for $\boldsymbol{H}$ that can be used as an acceptable solution. One way to find a proper solution is to put an additional constraint on $\boldsymbol{H}$. It has been shown that if the constraint is to require the vector $\boldsymbol{H}$ to have the smallest energy

$$\sum_{i=1}^{\lfloor \frac{N-K}{2} \rfloor} |h_i|^2$$

the unique solution is given by the Moore–Penrose pseudoinverse [6]

$$\boldsymbol{H} = -\mathrm{Pinv}(\boldsymbol{R}) * \boldsymbol{E} \tag{13}$$

where $\mathrm{Pinv}(\boldsymbol{R})$ is a matrix of the same dimensions as $\boldsymbol{R}$ so that

$$\begin{aligned} &\boldsymbol{R} * \mathrm{Pinv}(\boldsymbol{R}) * \boldsymbol{R} = \boldsymbol{R} \\ &\mathrm{Pinv}(\boldsymbol{R}) * \boldsymbol{R} * \mathrm{Pinv}(\boldsymbol{R}) = \mathrm{Pinv}(\boldsymbol{R}) \end{aligned} \tag{14}$$

To calculate the locations of errors from (3) it can be seen that i_m denotes the location of an error if $\exp(j2\pi i_m/N)$ is a zero of the polynomial $S(z)$. Furthermore, if an N-tuple vector $\boldsymbol{H}^e$ is constructed as follows:

$$\boldsymbol{H}^e = \left[1 \quad h_1 \quad h_2 \quad \ldots \quad h_{\lfloor \frac{N-K}{2} \rfloor} \quad 0 \quad \ldots \quad 0 \right]^T \tag{15}$$

where h_i are components of the solution of (9); the zeros of the DFT of $\boldsymbol{H}^e$ are in the locations of errors.

2.2 New error-correction procedures

In a decoding procedure for DFT-Based error-control codes, after producing an error locator polynomial, an error correction technique is applied. Three techniques for error correction, or equivalently, for calculating the error vector are proposed.

In method 1, using the identity of $E(i) = R(i)$ for i denoting the positions of N–K consecutive zeros in the DFT domain, the remaining values of $\boldsymbol{E}$ can be found by (7).

In the second method, after finding the positions of errors, the conventional erasure recovery technique [1] is applied. In other words, after calculating the locations of errors the steps of the error-correction procedure are as follows:

(i) We produce the actual error locator polynomial $S'(z)$ such that

$$S'(z) = \prod_{m=1}^{t'} \left(1 - ze^{-j\frac{2\pi i'_m}{N}}\right) = \sum_{r=0}^{t'} h'_r z^r \qquad (16)$$

where $\{i'_m : m = \{1, 2, \ldots, t'\}\}$ and t', respectively, denote the locations and the number of errors that are extracted by the technique presented in Section 2.1.

(ii) By using the identity of $E(i) = R(i)$, for i denoting the positions of $N-K$ consecutive zeros in the DFT domain, the remaining values of E can be found by the following recursive equations:

$$\sum_{r=0}^{t'} h'_r E(p-r) = 0 \quad p = 0, 1, \ldots, N-1 \qquad (17)$$

where h'_r are the coefficients of the polynomial (16).

In the third technique, after finding the positions of errors, using the identity of $E(i) = R(i)$ for i denoting the positions of N–K consecutive zeros in the DFT domain we try to solve the following linear equations (assuming N is even):

$$\sum_{i'_m \in \{\text{Locations of errors}\}} e(i'_m) e^{-j\frac{2\pi}{N} i'_m k} = E(k),$$

$$\text{for } k = N/2,\ N/2 \pm 1,\ N/2 \pm 2, \ldots, N/2 \pm \left\lfloor \frac{N-K}{2} \right\rfloor \qquad (18)$$

where $\{i'_m : m = \{1, 2, \ldots, t'\}\}$ and t', respectively, denote the locations and the number of errors that are extracted by the technique of Section 2.1. In this case we have $2 \times \lfloor (N-K)/2 \rfloor + 1$ equations and $t' < N-K$ unknowns. Therefore they form an overdetermined system of equations. This set of equations may not have an exact solution. In this case we try to find a least-square solution.

We rewrite the set of equations of (18) as follows:

$$\boldsymbol{R}_s * \boldsymbol{e}_s = \boldsymbol{E}_s \qquad (19)$$

where

$$\boldsymbol{R}_s = \begin{bmatrix} e^{-j\frac{2\pi}{N} i_1 \left(N/2 - \left\lfloor \frac{N-K}{2} \right\rfloor\right)} & e^{-j\frac{2\pi}{N} i_2 \left(N/2 - \left\lfloor \frac{N-K}{2} \right\rfloor\right)} & \cdots & e^{-j\frac{2\pi}{N} i_{t'} \left(N/2 - \left\lfloor \frac{N-K}{2} \right\rfloor\right)} \\ e^{-j\frac{2\pi}{N} i_1 \left(N/2 - \left\lfloor \frac{N-K}{2} \right\rfloor + 1\right)} & e^{-j\frac{2\pi}{N} i_2 \left(N/2 - \left\lfloor \frac{N-K}{2} \right\rfloor + 1\right)} & \cdots & e^{-j\frac{2\pi}{N} i_{t'} \left(N/2 - \left\lfloor \frac{N-K}{2} \right\rfloor + 1\right)} \\ \vdots & \vdots & \vdots & \vdots \\ e^{-j\frac{2\pi}{N} i_1 \left(N/2 + \left\lfloor \frac{N-K}{2} \right\rfloor\right)} & e^{-j\frac{2\pi}{N} i_2 \left(N/2 + \left\lfloor \frac{N-K}{2} \right\rfloor\right)} & \cdots & e^{-j\frac{2\pi}{N} i_{t'} \left(N/2 + \left\lfloor \frac{N-K}{2} \right\rfloor\right)} \end{bmatrix} \qquad (20)$$

$$\boldsymbol{e}_s = [\, e_{i_1} \quad e_{i_2} \quad \ldots \quad e_{i_{t'}} \,]^T \qquad (21)$$

$$\boldsymbol{E}_s = \left[E\left(N/2 - \left\lfloor \frac{N-K}{2} \right\rfloor\right) \; E\left(N/2 - \left\lfloor \frac{N-K}{2} \right\rfloor + 1\right) \ldots E\left(N/2 + \left\lfloor \frac{N-K}{2} \right\rfloor\right) \right]^T \qquad (22)$$

and the dimensions of $\boldsymbol{R}_s$, $\boldsymbol{e}_s$, and $\boldsymbol{E}_s$ are $(2\lfloor (N-K)/2 \rfloor + 1) \times t'$, $t' \times 1$, and $(2\lfloor (N-K)/2 \rfloor + 1) \times 1$, respectively. From the properties of Vandermonde matrices, it can be easily shown that $\boldsymbol{R}_s$ is a full-rank matrix. Therefore the least-square solution of (19) can be found as follows [7–10]:

$$\boldsymbol{e}_s = (\boldsymbol{R}_s^H * \boldsymbol{R}_s)^{-1} * \boldsymbol{R}_s^H * \boldsymbol{E}_s = \text{Pinv}(\boldsymbol{R}_s) * \boldsymbol{E}_s \qquad (23)$$

We expect that method 3 outperforms the others. This is because in this technique we use the information conveyed by all the padded zeros to compensate for the effects of additive noise and quantisation error. Our simulation results in Sections 3 and 4 confirm this prediction. They show that the sensitivity of method 3 to additive noise for bursty losses and erasures is considerably lower than the sensitivity of methods 1 and 2.

3 Numerical results

In the absence of additive noise and quantisation noise, the vector $\boldsymbol{E}$ is an AR process with order t, which denotes the number of lost samples. In the conventional decoding method, the Levinson–Durbin, Berlekamp, and Euclidian algorithms can be used to find the number of errors. The idea of these algorithms is to recursively compute the solutions of the Yule–Walker equations for the top principal submatrices in (9). In the conventional technique, to properly estimate the number of errors t, these algorithms require that the top $t \times t$ principal matrix to be nonsingular and the top $(t+1) \times (t+1)$ principal matrix to be singular. In other words, the curve of determinants of the top principal submatrices against their dimensions should rapidly decrease at the point of t.

To show the sensitivity of the conventional decoding technique to additive noise, which may be generated by quantisation process of the known samples of $\boldsymbol{E}$, for a system with $N = 64$, N–$K = 31$ and $t = 12$ in Fig. 1, the curves of determinants against dimensions of submatrices are plotted for the no quantisation and 8-bit quantisation cases. It can be seen that in the case of no quantisation noise, because of the rapid fall-off in the values of determinants at dimensions greater than 12, the conventional algorithm can estimate the number of errors, i.e. 12. Nevertheless, in the presence of the quantisation noise the conventional algorithm cannot exactly predict the number of errors.

To evaluate the performance of the proposed decoding method, the transmitted, received and estimated (reconstructed) signals using the proposed decoding technique in the absence of quantisation noise and additive noise are plotted in Fig. 2. Our results show that all the proposed error-correction procedures have the same performances and they can exactly estimate the transmitted signal. For the signals shown in Fig. 2, Fig. 3 shows the positions of zeros of the polynomial $S(z)$. The positions of zeros with the form of $\exp(j2\pi m/N)$ exactly show the location of errors. In Fig. 4 the amplitude of the DFT of $\boldsymbol{H}^e$, see (15), is shown under the same conditions of Figs. 2 and 3. The minima of this curve are at the locations of errors.

In Fig. 5 the estimated (reconstructed) signal is compared with the received and transmitted signals with 8-bit quantisation. Our results show that in this case all the

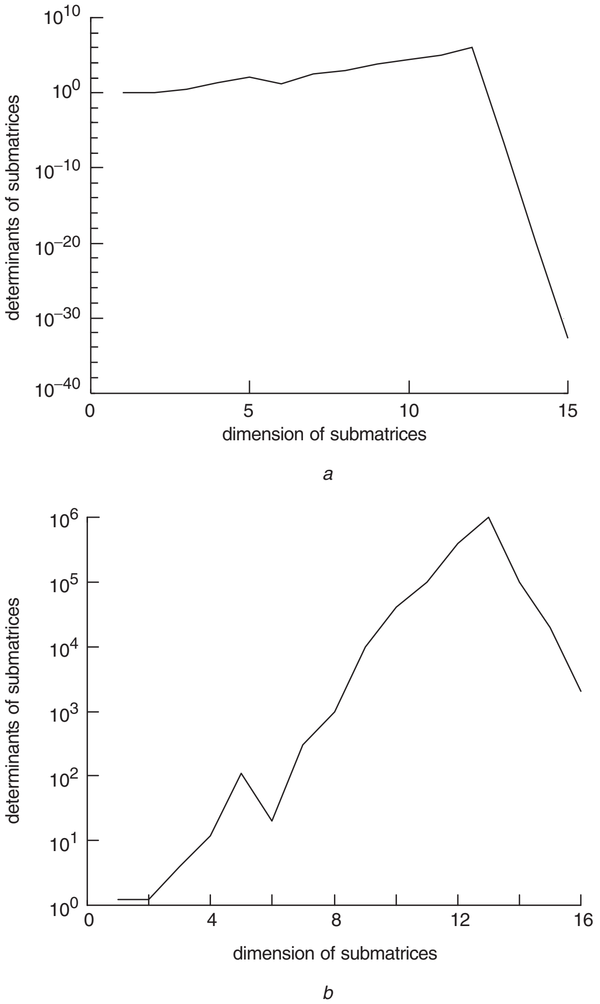

Fig. 1 *Determinants against dimensions of submatrices*
a No quantisation
b 8-bit quantisation

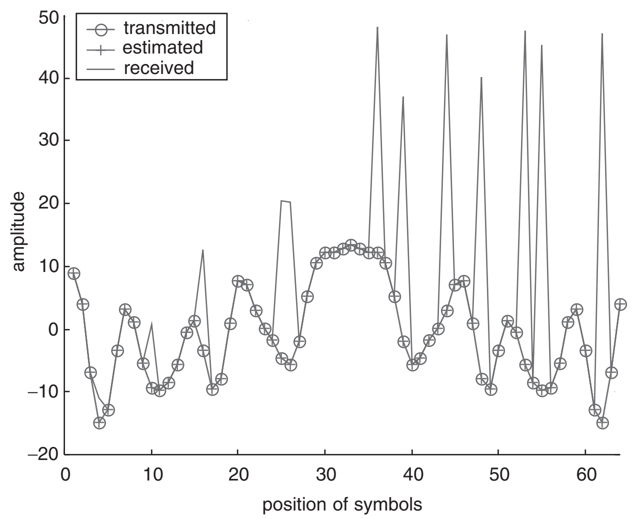

Fig. 2 *Transmitted, received and estimated signals in absence of quantisation noise and additive noise*

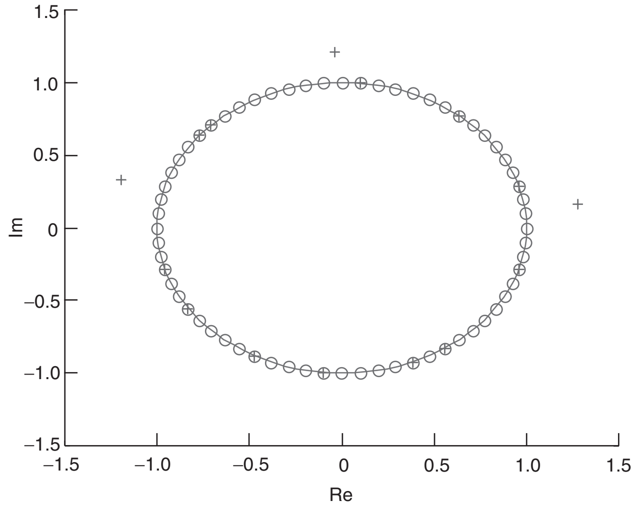

Fig. 3 *Positions of zeros of polynomial $S(z)$ in absence of quantisation noise and additive noise*

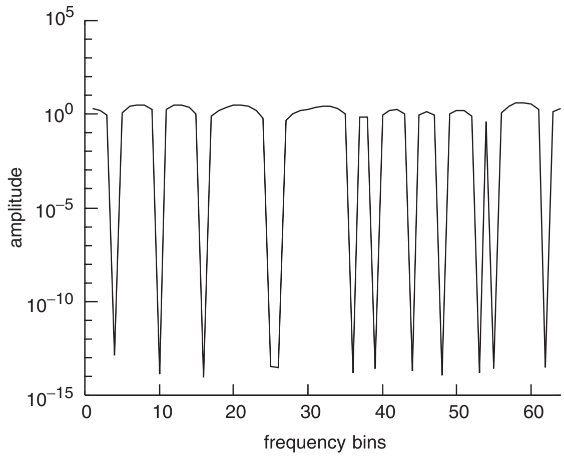

Fig. 4 *Amplitude of DFT of $\boldsymbol{H}^e$ in absence of quantisation noise and additive noise*

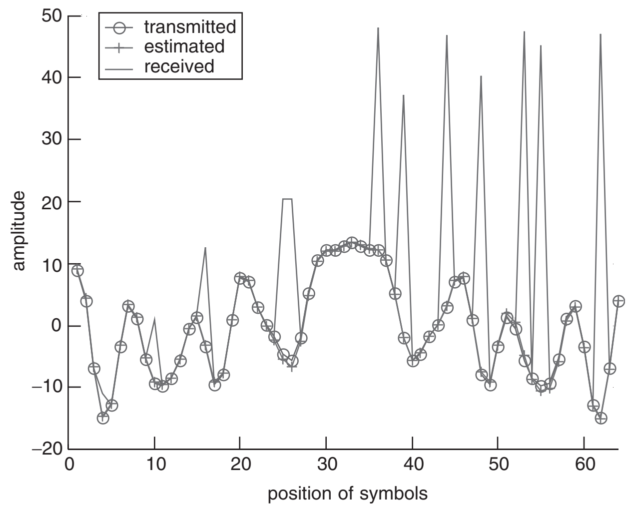

Fig. 5 *Transmitted, received and estimated signals with 8-bit quantisation*

proposed error-correction procedures have the same performance. It can also be seen that the proposed decoding method works well and can reconstruct the transmitted signal; but the conventional decoding method fails to estimate the transmitted signal. Under the same conditions as Fig. 5, Fig. 6 shows the positions of zeros of $S(z)$ and also shows that the positions of zeros whose locations denote the positions of errors are insensitive to the quantisation noise. Figure 7 shows the amplitude of the DFT of $\boldsymbol{H}^e$. It can be seen that the presence of quantisation noise does not change the locations of the curve minima.

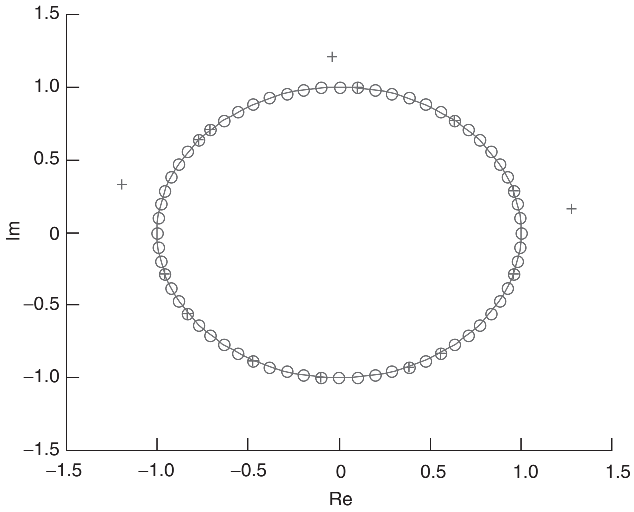

Fig. 6 *Positions of zeros of polynomial $S(z)$ with 8-bit quantisation*

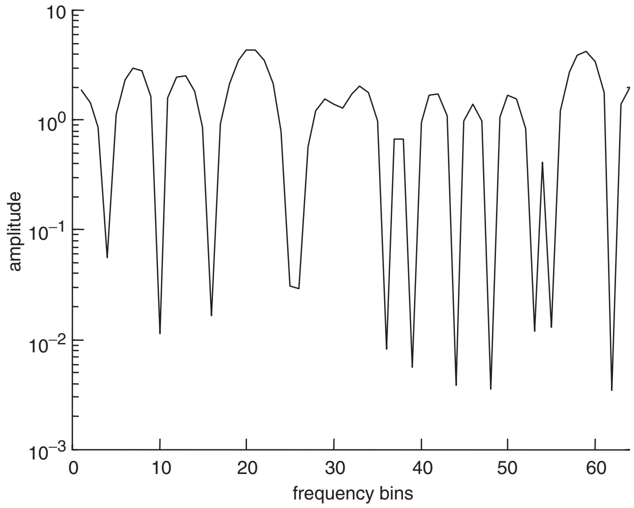

Fig. 7 *Amplitude of DFT of $\boldsymbol{H}^e$ with 8-bit quantisation*

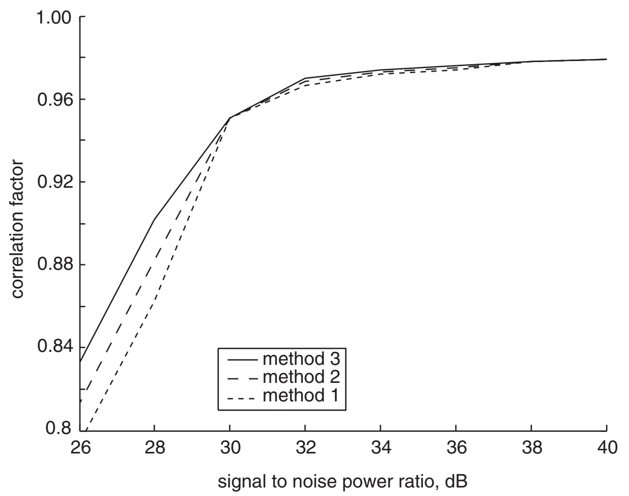

Fig. 8 *Correlation factor between transmitted and estimated signal against the ratio of signal to additive white Gaussian noise power*

4 Stability analysis

The curves of Figs. 5–7 have been obtained for quantised symbols in the absence of additive noise. In Fig. 8, to quantitatively analyse the stability of the proposed method in the presence of both quantisation noise and additive noise, the curves of the correlation factor between the transmitted and estimated signals are plotted against the ratio of signal to additive white Gaussian noise power (SNR) for a system with $N = 64$, $N - K = 31$ and $t = 21$. To mitigate the dependency of the results on the original signal, the sample mean of the correlation factor is evaluated by using 1000 different independent signals with the same lost sample positions. It can be seen that with minimum acceptable correlation factor of 0.95 the proposed decoding technique can tolerate the additive white Gaussian noise up to SNR = 30 dB. Furthermore, it can be seen that method 3 slightly outperforms the other error-correction procedures. With 8-bit quantisation the conventional decoding technique, even in the absence of additive noise, fails to reconstruct the transmitted signal and the correlation factor between the reconstructed and transmitted signal will be lower than 0.8.

Now we present the sensitivity analysis of the proposed decoding method for bursty losses. As with the conventional technique [1–4], our simulation results show that the proposed decoding technique is sensitive to additive noise for bursty losses, and its sensitivity can be significantly reduced by using the sorted DFT (SDFT) transformation [Note 1]. The kernel of SDFT is

$$t_{kl} = \exp\left(-j\frac{kl}{N}q2\pi\right) \tag{24}$$

where $\boldsymbol{T} = [t_{kl}]_{kl}$ is the SDFT transformation matrix and q is an integer relatively prime to N. Table 1 presents the optimum values of q for different values of block size N enabling the SDFT to correct consecutive error samples up to its maximum capability ($\lfloor N - K/2 \rfloor$) with 8 or 16 bit quantisation. It is observed that the optimum values of q have symmetry. In other words, the sorted transformations with q and $n - q$ show the same performance.

Table 1: Optimum values of *q* for SDFT kernel with 8 or 16 bit quantisation

Block size	*Q*
16	3,5,11,13
32	5,7,9,23,25,27
64	17,19,23,25,27,37,39,41,45,47

To calculate the positions of errors, matrix equation (9) should be solved. In the absence of quantisation noise and additive noise, if $t < \lfloor (N - K)/2 \rfloor$, the matrix $\boldsymbol{R}$ will be singular. In this case, we have proposed to use the unique solution that is given by Moore–Penrose pseudoinverse. In the presence of additive or quantisation noise, $\boldsymbol{R}$ may become a full-rank matrix, so we can find a solution by inverting the matrix $\boldsymbol{R}$. Our simulation results show that using pseudoinverse always leads to a perfect solution for (9) but in some cases the inverse matrix may lead to a false solution due to a poor condition number. As an example, Fig. 9 demonstrates the results of error-signal reconstruction for 12 consecutive errors in a (64, 9) code with $q = 27$. As seen, the pseudoinverse method outperforms the simple matrix inverse.

The least-square based method (method 3) uses the information conveyed by all padded zeros to compensate

Note 1: Marvasti, F.: 'Error concealment or correction of speech, image and video signals'. European patent (no. 1 006 665), 28 May 2003; and US patent (no. 6 601206), 29 July 2003

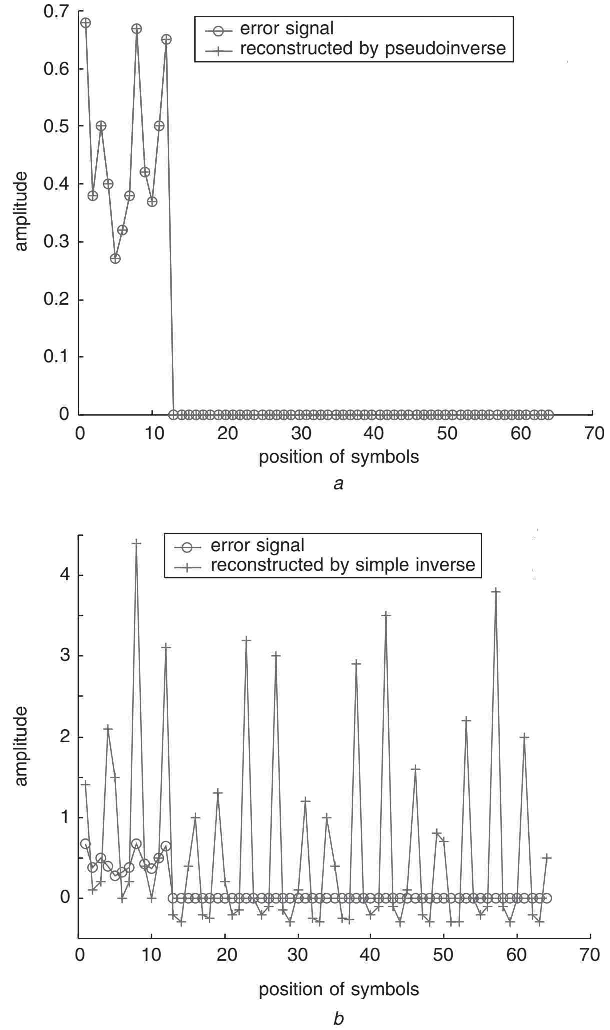

Fig. 9 *Actual and reconstructed error signals in presence of burst of 12 consecutive errors*
a Reconstructed by pseudoinverse
b Reconstructed by simple inverse

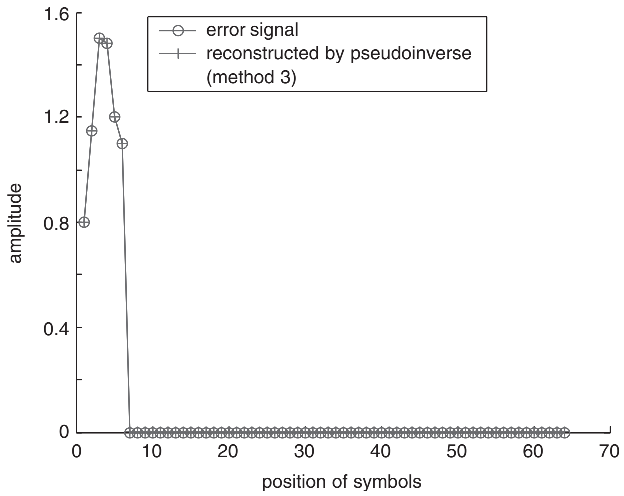

Fig. 10 *Actual and reconstructed by pseudoinverse-based error-correction method (method 3) error signals for $N=64$, $K=32$ in presence of six consecutive bursty losses with 8-bit quantisation*
For this example the other proposed error-correction techniques fail to work

for the effect of additive and quantisation noises; therefore this error-correction procedure becomes more stable. Figure 10 shows an example for bursty losses in which both

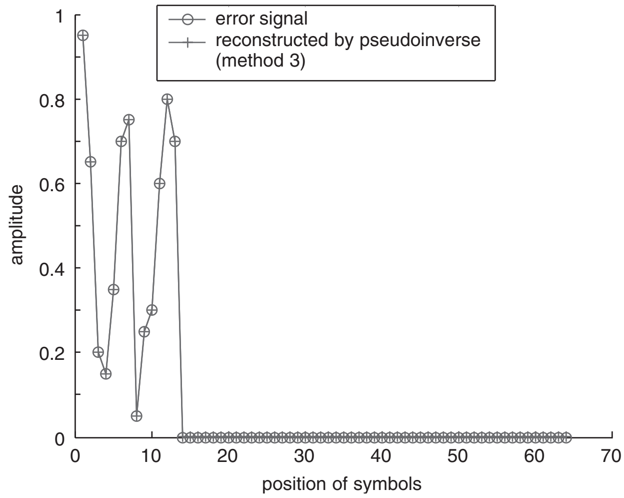

Fig. 11 *Actual and reconstructed by pseudoinverse-based error-correction method (method 3) error signals for $N=64$, $K=32$ in presence of 13 consecutive erasure errors with 8-bit quantisation*
For this example the other proposed error-correction techniques fail to work

methods 1 and 2 fail but method 3 is able to estimate the error signal. In this Figure the reconstructed error signal by method 3 is compared with the actual error signal in a system with $N=64$, $K=31$ and $t=6$ with 8-bit quantisation and in the absence of additive white Gaussian noise. It can be seen that method 3 is able to reconstruct the error signal whereas the first and the second techniques fail to work in bursty losses.

In the presence of consecutive erasure errors, when the positions of error samples are known, the proposed error-correction procedures can be applied to recover the values of erasure samples. As with the impulsive channel case, our simulation results show methods 1 and 2 have almost the same performance whereas method 3 outperforms the others. Figure 11 shows the capability of method 3 for a system with $N=32$, $K=15$ in the presence of $t=13$ consecutive erasure errors with 8-bit quantisation and in the absence of additive noise. In this case, both the first and second error-correction techniques fail but method 3 performs well. In the absence of quantisation noise and additive noise all three techniques perform equally well whenever $t<\lfloor N-K \rfloor$.

5 Computational complexity

This Section compares the complexity of the proposed decoding technique with the complexity of the conventional method, choosing to use the order of the number of flops required for each technique as the measure of complexity.

For the proposed decoding technique, as the first step, error locator polynomial coefficients must be found from (9). To provide vector $\boldsymbol{E}$ and matrix $\boldsymbol{R}$ from the received vector we need a DFT operation and therefore the required number of flops has an order of $O(N\log_2 N)$. To solve (9) by using Moore–Penrose pseudoinverse, the required number of computations has an order of approximately $O\left((N-K)^3\right)$ [8]. Therefore the order of total flops required to produce the error locator polynomial is $O(N\log_2 N)+O\left((N-K)^3\right)$.

The second step of the proposed decoding technique is the error-correction procedure. In this paper three methods have been proposed for error correction. In method 1 the

Table 2: Computational complexities of decoding methods

Decoding technique	Computational complexity
Method 1	$O(K(N-K)) + O(N\log_2 N) + O\left((N-K)^3\right)$
Method 2	$O(N^2 - K^2) + O(N\log_2 N) + O\left((N-K)^3\right)$
Method 3	$O(N\log_2 N) + O\left((N-K)^3\right)$
Conventional method	$O(K(N-K)) + O\left((N-K)^2\right) + O(N\log_2 N)$

recursive equations (7) are applied to find the unknown values of $\boldsymbol{E}$. This operation needs to $O(K(N-K))$ flops. To produce error vector $\boldsymbol{e}$ from $\boldsymbol{E}$ and then to subtract it from the received vector $\boldsymbol{r}$ requires $O(N\log_2 N)$ operations. Therefore the total flops required for method 1 has an order of $O(K(N-K)) + O(N\log_2 N)$.

In method 2, to produce the actual error locator polynomial requires $O(N(N-K))$ flops [2, 11]. Then, as with method 1, finding the unknown values of $\boldsymbol{E}$ needs $O(K(N-K))$ flops. To correct the errors, we need to find the IFFT of $\boldsymbol{E}$ and then subtract the result from the received vector. Therefore $O(N\log_2 N)$ additional flops are required. So the total computational complexity of the proposed decoding procedure with method 2 has an order of $O(N^2 - K^2) + O(N\log_2 N)$. In method 3, we need to calculate Moore–Penrose pseudoinverse of $\boldsymbol{R_s}$ which needs $O\left((N-K)^3\right)$ flops. It can be easily shown that the required number of flops for method 3 has an order of $O(N\log_2 N) + O\left((N-K)^3\right)$.

In Table 2 the computational complexities of the proposed decoding method with different error-correction procedures are compared with the complexity of the conventional method. It can be seen that for the examples presented in Sections 3 and 4, and generally speaking for low-rate DFT-based codes where $N-K \geq K$, calculating the Moore–Penrose pseudoinverse is the most complex part of the proposed decoding algorithm. Therefore the proposed technique is considerably more complex than the conventional method. But as it has been shown in Sections 3 and 4 that in the presence of quantisation and additive noise, the proposed technique can work but the conventional method fails. Therefore in such cases one may need to accept the computational price of the proposed method.

The main burden of computation of the proposed method is due to finding pseudoinverses of the Toeplitz and Vandermonde matrices. The computation of the pseudoinverses of structured matrices such as the Toeplitz and Vandermonde has become an important line of research in numerical linear algebra in recent years [12]. Therefore we expect that the proposed method will have a comparable complexity to the conventional method within a short time.

6 Conclusions

A novel decoding technique for real field error control codes has been proposed. The proposed algorithm determines the number and the positions of the error samples simultaneously. It has been shown that in contrast to the conventional decoding technique, the proposed method can work in the presence of additive and quantisation noises. Furthermore, it has been shown that if sorted DFT is used in real field error codes, the sensitivity of the proposed decoding method to additive noise for bursty losses and erasures significantly reduces. Among the three proposed error-correction procedures the simulation results show that the sensitivity of the pseudoinverse-based method to additive noise is lower than the sensitivity of the other error-correction techniques.

7 Acknowledgment

The authors would like to thank Ms Madani for help in providing some of the numerical results.

8 References

1 Marvasti, F.: 'Nonuniform sampling theory and practice' (Kluwer Academic/Plenum, New York, 2001)
2 Marvasti, F., Hassan, M., Echhart, M., and Talebi, S.: 'Efficient algorithms for burst error recovery using FFT and other transform kernels', *IEEE Trans. Signal Process.*, 1999, **47**, (4), pp. 1065–1075
3 Wolf, J.K.: 'Redundancy, the discrete Fourier transform, and impulse noise cancellation', *IEEE Trans. Commun.*, 1983, **COM-31**, (3), pp. 458–461
4 Ferreira, P.J.S.G., and Vieira, J.M.N.: 'Locating and correcting errors in images'. Proc. IEEE Conf. on Image Processing, ICIP-97, Oct. 1997, pp. 691–694
5 Zhang, H., and Duhamel, P.: 'On the methods for solving Yule–Walker equations', *IEEE Trans. Signal Process.*, 1992, **40**, (12)
6 Rao, C.R., and Mitra, S.K.: 'Generalized inverse of matrices and its applications' (Wiley, New York, 1971)
7 Heining, G., and Rost, K.: 'Split algorithms for symmetric Toeplitz matrices with any rank profile', *Num. Linear Algebr. Appl.*, 2000, **6**, pp. 1–7
8 Golub, G., and van Loan, C.: 'Matrix computations' (Johns Hopkins University Press, 1996, 3rd edn.)
9 Rath, G., and Guillemot, C.: 'Performance analysis and recursive syndrome decoding of DFT codes for bursty erasure recovery', *IEEE Trans Signal Process.*, 2003, **51**, (5), pp. 1335–1350
10 Azmi, P., and Marvasti, F.: 'A novel decoding procedure for real field error control codes in the presence of quantisation noise'. Proc. Int. Conf. on ASSP 2003, Hong Kong, May 2003, pp. IV-265–268
11 Azmi, P., Meleas, D., and Marvasti, F.: 'An efficient method for demodulating PPM signals based on Reed–Solomon decoding algorithm', *Signal Process.*, 2004, **84**, (10)
12 Martinez, J.J.: 'A new source of structured SVD problems'. Presented at Int. Workshop on Numerical Linear Algebra, Madrid, Spain, June 2003