Distributed Lossy Source Coding Using Real-Number Codes

Mojtaba Vaezi and Fabrice Labeau

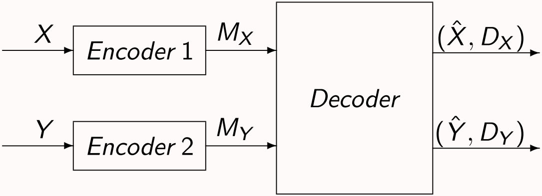
McGill University

76th Vehicular Technology Conference
Québec City, Canada

September 5, 2012

Problem setup

- Distributed *lossy* source coding



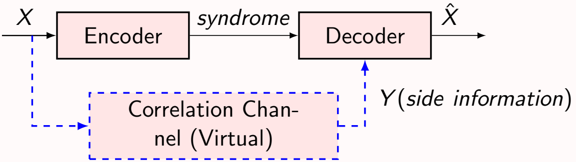
A communication system with

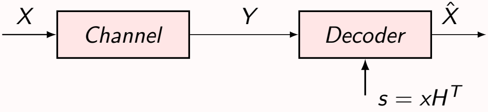
- Two separate, correlated signals (X and Y)
- The sources cannot communicate with each other; thus, encoding is done independently or in a distributed manner.
- The receiver, however, can perform joint decoding.

Practical code construction

Correlation channel model

- DSC is essentially a channel coding problem (view Y as corrupted version of X)





Practical code construction

Linear channel codes

- Lossless DSC

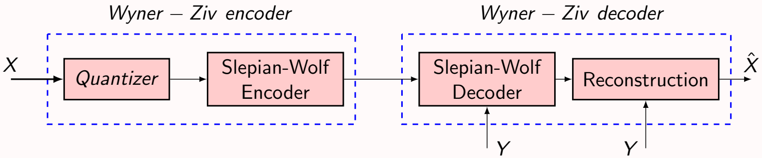
- The channel is usually modeled as a binary symmetric channel (BSC) with a crossover probability p
- Capacity-approaching channel codes (for the correlation channel) result in good DSC
  Examples: Turbo and LDPC codes
- The linear channel code design and its rate depends on the correlation model

Practical code construction

Wyner-Ziv coding

What if the source is a continuous-valued sequence? (many practical applications)

- Current approach

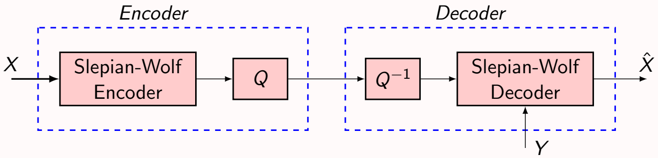


- There are source coding loss (or quantization loss) and channel coding loss (or binning loss)

Practical code construction

Wyner-Ziv coding in the real field

- Alternative approach



- Similarities and differences
  - There are still coding loss and quantization loss
  - Coding is before quantization $\Rightarrow$ error correction in the real field (soft redundancy)
- Advantages
  1. Correlation channel model is more realistic
  2. Quantization error can be reduced by a factor of coderate (it vanishes if X and Y are completely correlated)
  3. Better performance w.r.t. delay and complexity

Real BCH-DFT Codes

Encoding

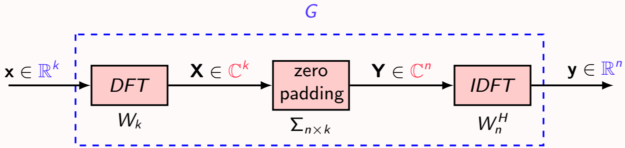


Figure: Real BCH-DFT encoding scheme

$$G = \sqrt{\frac{n}{k}} W_n^H \Sigma W_k$$

- $\Sigma_{n\times k}$ inserts n-k consecutive zeros in the transform domain $\Longrightarrow$ BCH code
- DFT is used to convert vector $\mathbf{x} \in \mathbb{R}^k$ to a circularly symmetric $\mathbf{X} \in \mathbb{C}^k$, guaranteeing a real $\mathbf{y}$
- Removing the DFT block, we obtain complex BCH-DFT codes

Real BCH-DFT codes

Encoding

- H takes N-K columns of W_N^H corresponding to zeros of Σ
- For every codeword, $s = Hy = HGx \equiv 0$

-Example: The (6,3) DFT code

$$G = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & \frac{-1}{3} \\ 0 & 1 & 0 \\ \frac{-1}{3} & \frac{2}{3} & \frac{2}{3} \\ 0 & 0 & 1 \\ \frac{2}{3} & \frac{-1}{3} & \frac{2}{3} \end{pmatrix}$$

- Reconstruction:

$$x = G^{\dagger}y = (G^TG)^{-1}G^Ty = \frac{K}{N}G^Ty$$

Real BCH-DFT Codes

Channel coding

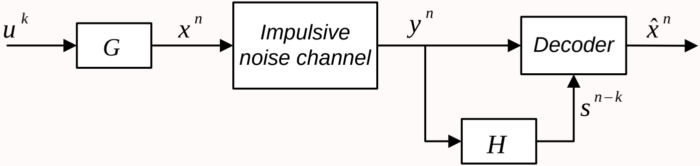


Figure: Channel coding using real-valued BCH codes

- H takes N-K columns of W_N^H corresponding to zeros of Σ
- For every codeword, $s = Hy = HGx \equiv 0$

Real BCH-DFT Codes

Channel coding

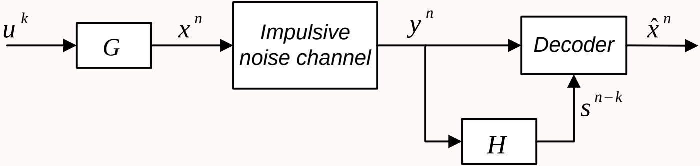


Figure: Channel coding using real-valued BCH codes

- H takes N-K columns of W_N^H corresponding to zeros of Σ
- For every codeword, $s = Hy = HGx \equiv 0$

Without quantization:

$y^n = x^n + e^n \Rightarrow s_y = s_e$

Real BCH-DFT codes

Decoding

- How can we decode?

1. Without quantization error
   - $y^n = x^n + e^n \Rightarrow s_e = s_y$
   - Decoding algorithms (e.g., the Peterson-Gorenstein-Zierler) for a BCH code, in general, has the following major steps
     1. Detection (to determine the *number* of errors)
     2. Localization (to find the *location* of errors)
     3. Calculation (to calculate the *magnitude* of errors)
2. With quantization error
   - $y^n = x^n + q^n + e^n \Rightarrow s_e = s_y - s_q$
   - Modify the above algorithm
   - Each step becomes an estimation problem
   - Least squares solution largely improves the decoding accuracy

Practical code construction
Wyner-Ziv coding using DFT codes

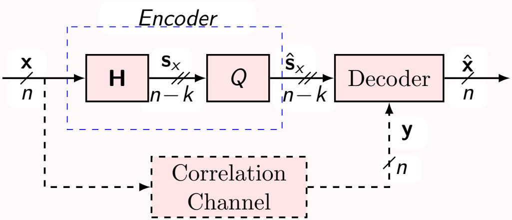


Figure: Wyner-Ziv coding using DFT codes: Syndrome approach.

- $y = x + e \Rightarrow s_e = s_y - s_x$
- Syndrome samples are complex numbers ($s_x = Hx$)

Practical code construction

Wyner-Ziv coding using DFT codes

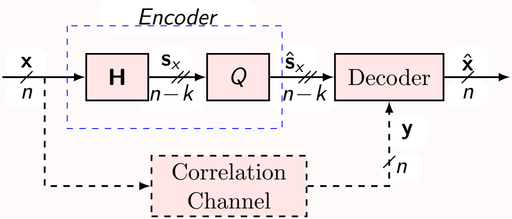


Figure: Wyner-Ziv coding using DFT codes: Syndrome approach.

- $y = x + e \Rightarrow s_e = s_y - s_x$
- Syndrome samples are complex numbers ($s_x = Hx$)

Q:

Can we do better?

Practical code construction

Wyner-Ziv coding using DFT codes

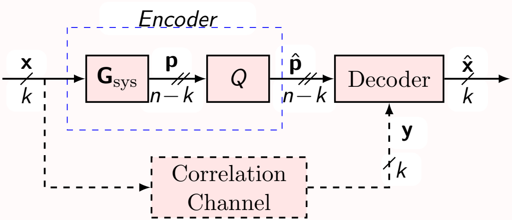


Figure: Wyner-Ziv coding using DFT codes: Parity approach.

$$G_{sys} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} G_1^{-1} = GG_1^{-1} \Longrightarrow HG_{sys} = 0$$

Thus, $W_n G_{sys}$ also has $n-k$ consecutive zeroes at the same positions; i.e., G_{sys} is the generator matrix of the same BCH code.

Practical code construction

Comparison

Compression ratio given an (n, k) code

- Syndrome approach: $\eta_s = \frac{n}{2(n-k)}$
- Parity approach: $\eta_p = \frac{k}{n-k}$

$\eta_p/\eta_s = 2k/n = 2R_c > 1 \Rightarrow$Parity approach is more efficient

Practical code construction

Comparison

Compression ratio given an (n, k) code

- Syndrome approach: $\eta_s = \frac{n}{2(n-k)}$
- Parity approach: $\eta_p = \frac{k}{n-k}$

$\eta_p/\eta_s = 2k/n = 2R_c > 1 \Rightarrow$Parity approach is more efficient

Given the same compression)

Which codes result in $\eta = \frac{n}{n-k}$?

- Syndrome approach: $(n, \frac{n+k}{2})$
- Parity approach: $(2n-k, n)$

Thus, for a given compression ratio the parity approach implies a code with smaller rate

Wyner-Ziv using real-number codes
MSE for reconstructed signal

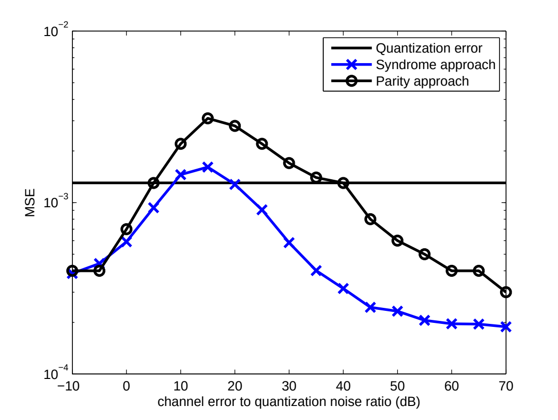


Figure: Reconstruction error for Wyner-Ziv coding using a $(7, 5)$ DFT code: Syndrome and parity approaches.

Wyner-Ziv using real-number codes

Probability of error localization

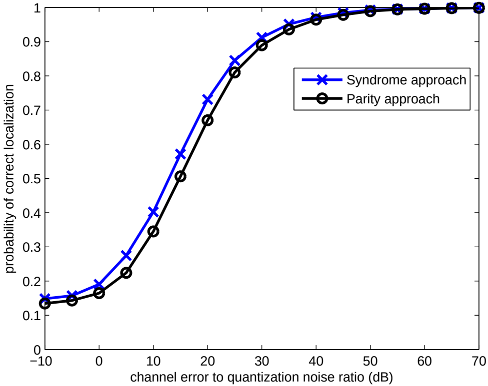


Figure: Relative frequency of correct localization of correlation channel error in the syndrome and parity approaches, using a $(7,5)$ DFT code.

Wyner-Ziv using DFT codes

Summary

- DFT codes could be better than binary codes for lossy DSC
- Delay and complexity is much less compared to Turbo and LDPC codes
- Parity approach is more efficient than syndrome approach
- Error localization is crucial in the performance of compression

Thank you for your attention

Real BCH-DFT codes

Decoding

The Peterson-Gorenstein-Zierler (PGZ) algorithm

1. Compute vector of syndrome samples
2. Determine the number of errors ν by constructing a syndrome matrix and finding its rank
3. Find coefficients $\Lambda_1, \ldots, \Lambda_\nu$ of error-locating polynomial $\Lambda(x) = \prod_{i=1}^{\nu}(1 - xX_i)$ whose roots are the inverse of error locations
4. Find the zeros $X_1^{-1}, \ldots, X_\nu^{-1}$ of $\Lambda(x)$; the errors are then in locations $i_1, \ldots, i_\nu$ where $X_1 = \alpha^{i_1}, \ldots, X_\nu = \alpha^{i_\nu}$ and $\alpha = e^{-j\frac{2\pi}{N}}$
5. Finally, determine error magnitudes by solving a set of linear equations whose constants coefficients are powers of X_i.

Error correction for BCH code

The Peterson-Gorenstein-Zierler (PGZ) algorithm

- Suppose there are $\nu \leq t$ errors in locations $i_1, \ldots, i_\nu$ with magnitudes $e_{i_1}, \ldots, e_{i_\nu}$
- Then $r(x) = c(x) + e(x)$ where $e(x) = e_{i_1}x^{i_1} + \ldots + e_{i_\nu}x^{i_\nu}$ is the error polynomial
- The partial syndromes are defined as $s_j = r(\alpha^j) = c(\alpha^j) + e(\alpha^j) = e(\alpha^j) = e_{i_1}\alpha^{i_1 j} + \ldots + e_{i_\nu}\alpha^{i_\nu j}$.
  -change of variables
  1. error locators: $X_1 = \alpha^{i_1}, \ldots, X_\nu = \alpha^{i_\nu}$
  2. error magnitudes: $Y_1 = e_{i_1}, \ldots, Y_\nu = e_{i_\nu}$
- Syndrome equations ($2t$ equations with 2ν unknowns)

$$
\begin{aligned}
s_1 &= Y_1X_1 + \ldots + Y_\nu X_\nu \\
s_2 &= Y_1X_1^2 + \ldots + Y_\nu X_\nu^2 \\
&\vdots \\
s_{2t} &= Y_1X_1^{2t} + \ldots + Y_\nu X_\nu^{2t}
\end{aligned}
$$

Error correction techniques in real-field

BCH-DFT decoding

Then the PGZ algorithm has the following steps

1. Detection $(\nu = ?)$

$$\mathbf{S}_t = \begin{bmatrix} s_1 & s_2 & \dots & s_t \\ s_2 & s_3 & \dots & s_{t+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_t & s_{t+1} & \dots & s_{2t-1} \end{bmatrix}$$

$\nu = \mu$ iff $\mathbf{S}_\nu$ is nonsingular for $\nu = \mu$ but is singular for $\nu > \mu$
$\mathbf{S}_\mu = V_\mu D V_\mu^T$

$$V_\mu = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ X_1^{\mu-1} & \dots & X_\mu^{\mu-1} \end{bmatrix}, D = \begin{bmatrix} Y_1 X_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & Y_\mu X_\mu \end{bmatrix}$$

Error correction techniques in real-field

BCH-DFT decoding

1. Localization ($X_i = ?$)
   - Define error-locator polynomial as

$$\Lambda(x) = \prod_{i=1}^{\nu}(1 - xX_i) = \Lambda_0 + \Lambda_1 x + \ldots + \Lambda_\nu x^\nu$$

     The roots of $\Lambda(x)$, i.e. $X_1^{-1}, \ldots, X_\nu^{-1}$, give the reciprocals of of error locators.
   - Find $\Lambda_1, \ldots, \Lambda_\nu$ by solving $S_\nu \left[\Lambda_\nu \ \Lambda_{\nu-1} \ldots \Lambda_1\right]^T = -\left[s_{\nu+1} \ s_{\nu+2} \ldots s_{2\nu}\right]^T$
   - Determine the roots of $\Lambda(x)$ evaluating $\Lambda(\alpha^i), i = 1 \ldots N$, where $\alpha = e^{-j\frac{2\pi}{N}}$ for BCH-DFT codes

Error correction techniques in real-field

BCH-DFT decoding

1. Estimation ($Y_i = ?$)

   Finally, determine error magnitudes by solving a set of linear equations whose constants coefficients are powers of X_i

$$\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{2t} \end{bmatrix} = \begin{bmatrix} X_1 & \dots & X_\nu \\ X_1^2 & \dots & X_\nu^2 \\ \vdots & \ddots & \vdots \\ X_1^{2t} & \dots & X_\nu^{2t} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{2\nu} \end{bmatrix} \quad (1)$$

DSC using real error-correcting codes

Syndrome approach,6-bit quantization

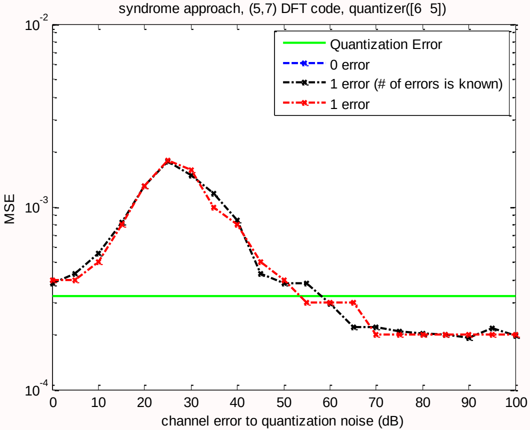


Figure: Wyner-Ziv coding using a (7,5) DFT code