

Kite Codes: Design, Analysis and Generalizations

Xiao Ma

maxiao@mail.sysu.edu.cn

Dept. of ECE, Sun Yat-sen University

The Chinese University of Hong Kong, Sept. 2011

- 1 Basics of Kite Codes
- 2 Maximum Likelihood Decoding Analysis of Kite Code
- 3 Design of Kite Code
- 4 Improved Design of Kite Code
- 5 Kite Codes over Groups

Joint work with:

My students [Kai Zhang](#), [Shancheng Zhao](#) and my colleagues [Prof. Baoming Bai](#) from State Key Lab. of ISN, Xidian University and [Prof. Xiaoyi Zhang](#) from National Digital Switching System Engineering and Technological R&D Center, Zhengzhou.

This talk is based on the following works:

“Serial Concatenation of RS Codes with Kite Codes: Performance Analysis, Iterative Decoding and Design”, submitted to IEEE Trans. Inform. Theory, 2009. Available at <http://arxiv.org/abs/1104.4927>.

“Improved Design of Kite Codes and Their Applications”, submitted to IEEE Trans. Commun., 2011

“Kite Codes over Groups”, to appear in 2011 IEEE Information Theory Workshop (ITW2011), Paraty, Brazil, Oct 2011.

- 1 Basics of Kite Codes
- 2 Maximum Likelihood Decoding Analysis of Kite Code
- 3 Design of Kite Code
- 4 Improved Design of Kite Code
- 5 Kite Codes over Groups

Rateless coding

A coding method that can generate potentially infinite parity-check bits for any given fixed-length sequence.

Existing rateless code

- LT-codes;
- Raptor codes.
- ...

Motivations

- Raptor codes are optimized by degree distribution for erasure channels;
- No universal degree distributions exist for AWGN channels.
- How to construct good codes for AWGN channels with arbitrarily designated coding rate?

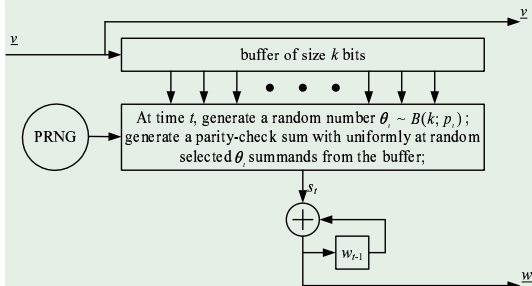
We will propose a class of rateless codes for AWGN channels.

Kite codes

An ensemble of Kite codes is denoted by $\mathcal{K}[\infty, k; \underline{p}]$.

- 1 k is the length of the information sequence;
- 2 \underline{p} is a real sequence, called p -sequence.

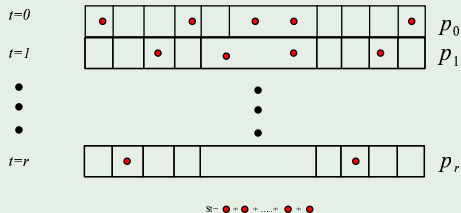
Encoding algorithm of Kite codes



- \underline{v} —systematic sequence of k bits;
- \underline{w} —parity-check sequence of $r = n - k$ bits;
- $B(k; p_t)$ —binomial distribution with k Bernoulli trials and success probability p_t ;
- s_t —parity-check sum at time t ;
- PRNG—pseudo-random number generator.

Encoding algorithm of Kite codes

- Initially, load information sequence of length k into a buffer.
- At time $t \geq 0$, randomly choose, with success probability p_t , several bits from the buffer.
- Calculate the XOR of these chosen bits and use it to drive the accumulator to generate a parity-check bit w_t .

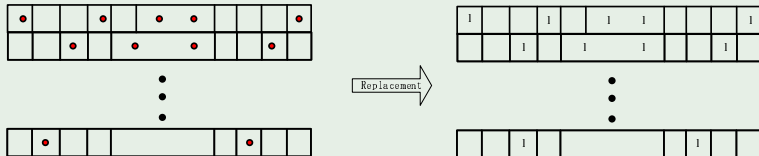


Parity-check matrix of Kite codes

For any given $n \geq k$, the prefix code with length n of a Kite code $\mathcal{K}[\infty, k; \underline{p}]$ is denoted by $\mathcal{K}[n, k]$ and also called Kite code for convenience, whose parity-check matrix can be written as

$$H = (H_v, H_w)$$

where H_v is a matrix of size $(n - k) \times k$ corresponding to the information bits and H_w is a dual diagonal matrix size $r \times r$ corresponding to the parity-check bits.



Obtain H_v by replacing the red dot and the blank in the graph by 1 and 0, respectively.

What is the new aspect of the Kite codes?

As an ensemble of codes, the proposed Kite codes are new.

- 1 The p -sequence instead of degree distributions is used to define the ensemble.
- 2 Kite codes are systematic and can be easily reconfigured for different data lengths and coding rates.
- 3 ...

Outline

- 1 Basics of Kite Codes
- 2 Maximum Likelihood Decoding Analysis of Kite Code**
- 3 Design of Kite Code
- 4 Improved Design of Kite Code
- 5 Kite Codes over Groups

Input-redundancy weight enumerating function

For the prefix codes with dimension k and length n , we define its input-redundancy weight enumerating function (IRWEF) of an ensemble of Kite codes as

$$A(X, Z) = \sum_{i,j} A_{i,j} X^i Z^j$$

where X, Z are two dummy variables and $A_{i,j}$ denotes the ensemble average of the number of codewords $\underline{c} = (\underline{v}, \underline{w})$ consisting of an input information sequence \underline{v} of Hamming weight i and a parity check sequence \underline{w} of Hamming weight j .

Given the information sequence $\underline{v}^{(\ell)}$ with the form $\underline{v}^{(\ell)} \triangleq (\underbrace{1 \cdots 1}_{\ell} \underbrace{0 \cdots 0}_{k-\ell})$. The corresponding parity-check sequence is a random vector $\underline{W}^{(\ell)}$, whose randomness is induced by the ensemble. Let $A^{(\ell)}(Z)$ be the ensemble average weight enumerating function of $\underline{W}^{(\ell)}$.

Maximum Likelihood Decoding Analysis of Kite Code

- $Pr\{S_t = 1\}$ can be calculated recursively.
- $\underline{W}^{(\ell)}$ can be modeled as a Markov process with time-dependent transition probabilities.
- $A^{(\ell)}(Z)$ can be calculated recursively by performing a forward trellis-based algorithm.

Calculation of IRWEF

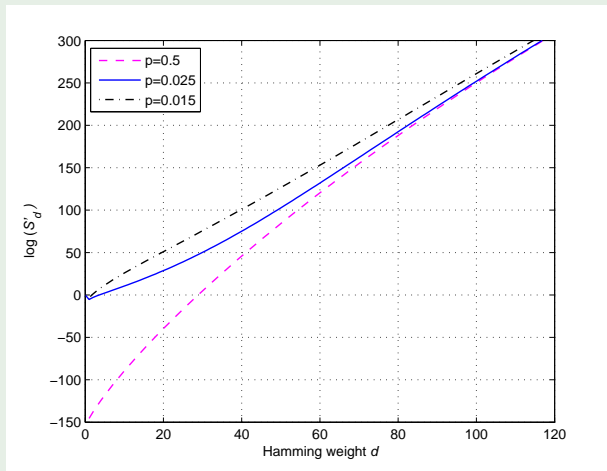
The IRWEF of an ensemble of Kite codes can be calculated as follows.

$$A(X, Z) = \sum_{0 \leq \ell \leq k} \binom{k}{\ell} X^\ell A^{(\ell)}(Z).$$

Maximum Likelihood Decoding Analysis of Kite Code

Numerical results

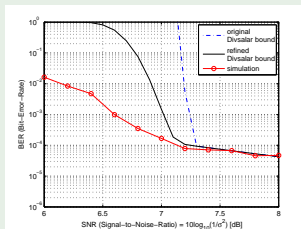
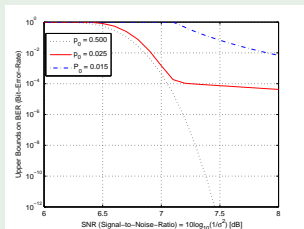
Consider the ensemble of Kite codes $\mathcal{K}[2100, 1890]$ with $p_t = p_0$ for $0 \leq t < 210$. The ensemble weight enumerating function is shown as below.



Maximum Likelihood Decoding Analysis of Kite Code

Numerical results

The performance bound of $\mathcal{K}[2100, 1890]$ is shown as below.



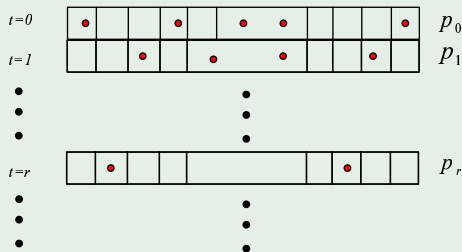
- Under the ML decoding, the performance degrades as the parameter p_0 decreases.
- The refined bound improves the original Divsalar bound especially in the low-SNR region.
- The iterative sum-product decoding algorithm delivers curves that match well with performance bound of the ML decoding.

Outline

- 1 Basics of Kite Codes
- 2 Maximum Likelihood Decoding Analysis of Kite Code
- 3 Design of Kite Code**
- 4 Improved Design of Kite Code
- 5 Kite Codes over Groups

Original problem

Evidently, the performance of Kite codes is determined by the p -sequence.



Then, the whole p -sequence should be optimized jointly such that all the prefix codes of Kite codes are good enough.

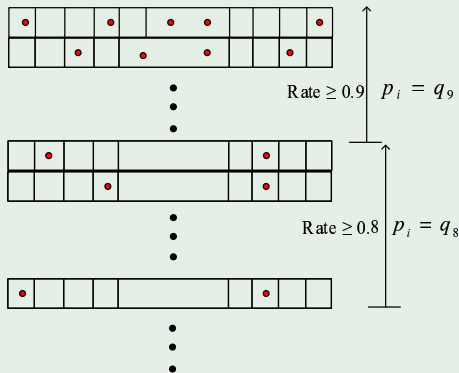
Too complex due to too many (may be infinite) variables involved in.

Design of Kite Code

A simple idea

Partition the p -sequence into groups according to the decoding rate of Kite codes.

$$p_t = \begin{cases} q_9, & 0.9 \leq k/(t+k) < 1.0 \\ q_8, & 0.8 \leq k/(t+k) < 0.9 \\ q_7, & 0.7 \leq k/(t+k) < 0.8 \\ q_6, & 0.6 \leq k/(t+k) < 0.7 \\ q_5, & 0.5 \leq k/(t+k) < 0.6 \\ q_4, & 0.4 \leq k/(t+k) < 0.5 \\ q_3, & 0.3 \leq k/(t+k) < 0.4 \\ q_2, & 0.2 \leq k/(t+k) < 0.3 \\ q_1, & 0.1 \leq k/(t+k) < 0.2 \end{cases} .$$



The task to design a Kite code is to select the parameters $\underline{q} = (q_9, q_8, \dots, q_1)$.

Greedy optimizing algorithm: Main idea

- Firstly, we choose q_9 such that the prefix code $\mathcal{K}[\lfloor k/0.9 \rfloor, k]$ is as good as possible.
- Secondly, we choose q_8 with **fixed** q_9 such that the prefix code $\mathcal{K}[\lfloor k/0.8 \rfloor, k]$ is as good as possible.
- Thirdly, we choose q_7 with fixed (q_9, q_8) such that the prefix code $\mathcal{K}[\lfloor k/0.7 \rfloor, k]$ is as good as possible.
- ...
- we choose q_1 with **fixed** (q_9, q_8, \dots, q_2) such that the prefix code $\mathcal{K}[\lfloor k/0.1 \rfloor, k]$ is as good as possible.

At each step, it is a **one-dimensional optimization problem**. Hence the key is to make a choice between q_ℓ and q'_ℓ .

Greedy optimizing algorithm: Implementation

- Simulation based.
- Density evolution based.

Density evolution based optimization

- Given q , derive the degree distributions for the edges in the normal graph of Kite codes.
- Obtain the threshold with density evolution (Gaussian approximation).
- Choose the parameter q with better threshold.

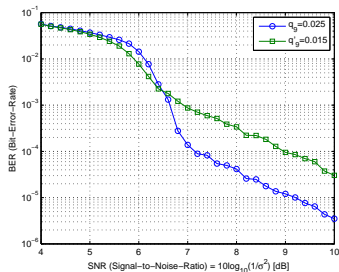
The density evolution is not effective for small k .

Simulation based optimization algorithm

- Set the target BER= 10^{-4} .
- Making the choice between the two parameters by simulations.

This simulation-based optimization algorithm is not time-consuming since

- The optimization target BER is not that low.
- Making choice between two parameters, called **ordinal optimization**, is not equivalent to evaluating performances for each of them.



Numerical results

With data length $k = 1890$ and simulation based greedy optimizing algorithm, we have the following p -sequence.

$$p_t = \begin{cases} q_9 = 0.0249, & 0.9 \leq \frac{k}{k+t} < 1.0 \\ q_8 = 0.0072, & 0.8 \leq \frac{k}{k+t} < 0.9 \\ q_7 = 0.0045, & 0.7 \leq \frac{k}{k+t} < 0.8 \\ q_6 = 0.0034, & 0.6 \leq \frac{k}{k+t} < 0.7 \\ q_5 = 0.0021, & 0.5 \leq \frac{k}{k+t} < 0.6 \\ q_4 = 0.0016, & 0.4 \leq \frac{k}{k+t} < 0.5 \\ q_3 = 0.0010, & 0.3 \leq \frac{k}{k+t} < 0.4 \\ q_2 = 0.0006, & 0.2 \leq \frac{k}{k+t} < 0.3 \\ q_1 = 0.0004, & 0.1 \leq \frac{k}{k+t} < 0.2 \end{cases}$$

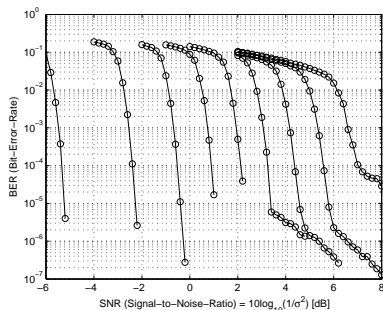


Figure: From left to right, the curves correspond to rates 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9, respectively.

Outline

- 1 Basics of Kite Codes
- 2 Maximum Likelihood Decoding Analysis of Kite Code
- 3 Design of Kite Code
- 4 Improved Design of Kite Code**
- 5 Kite Codes over Groups

Existing issues of Kite Code

- **Issue I:** In the high-rate region, there exist error floors, which is caused by the existence of **all-zero** (or extremely-low-weight) columns in the randomly generated matrix H_v .
- **Issue II:** In the low-rate region, there exists a relatively large gap between the performances of the Kite codes and the Shannon limits.
- **Issue III:** The optimized p -sequence has no closed form, which depends on the data length k .

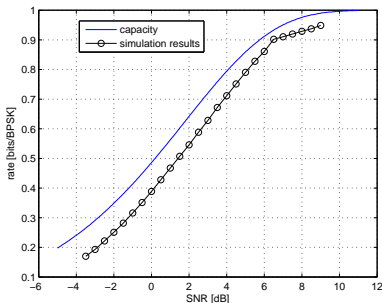


Figure: The average decoding rates of the RS-Kite codes. The data length $k = 50000$.

How to improve the Kite codes?

Given a data length k .

- 1 Partition the decoding rate interval $(0.05, 1]$ into 19 sub-intervals.
- 2 Let $H^{(\ell)} = (H_v^{(\ell)}, H_w^{(\ell)})$ be the parity-check matrix for the prefix code with decoding rate 0.05ℓ . Its size is of $(n_\ell - k) \times n_\ell$, where $n_\ell = \lfloor \frac{k}{0.05\ell} \rfloor$.
- 3 Given $\{q_\ell, 1 \leq \ell \leq 19\}$. We will use a similar incremental method to construct the parity-check matrix $H^{(1)}$, which corresponds to rate 0.05. If this is done, we then have prefix codes with varying rates from 0.05 to 1.0.
- 4 Now assume that $H^{(\ell+1)}$ has been constructed (initially, we set $H^{(20)}$ to be an empty matrix), we construct $H^{(\ell)}$ by padding a submatrix $H^{(\delta)}$ of size $(n_\ell - n_{\ell+1}) \times n_\ell$.

How to lower down the error-floors?

Row-weight concentration algorithm

- 1 Generate a random binary matrix $H_v^{(\delta)}$ of size $(n_\ell - n_{\ell+1}) \times k$ based on a Bernoulli distribution with success probability q_ℓ ; form the matrix corresponding to the information bits as $H_v^{(\ell)} = \begin{pmatrix} H_v^{(\ell+1)} \\ H_v^{(\delta)} \end{pmatrix}$
- 2 From $H_v^{(\delta)}$, find a row with maximum weight (called *maximum-weight row*) and a row with minimum weight (called *minimum-weight row*);
- 3 In $H_v^{(\ell)}$, find a **1** from the *maximum-weight row* satisfying that its column has the maximum weight among all such columns; in the meanwhile, find a **0** from the *minimum-weight row* such that its column has the minimum weight among all such columns.
- 4 Swap the found **1** and the found **0**.
- 5 Repeat Steps 2, 3 and 4 until the row weights of $H_v^{(\delta)}$ are near equal (with differences at most 1).

The process is incremental. The resulting matrix has near equal row weight in each coding interval.

How to narrow the gap from the Shannon limits?

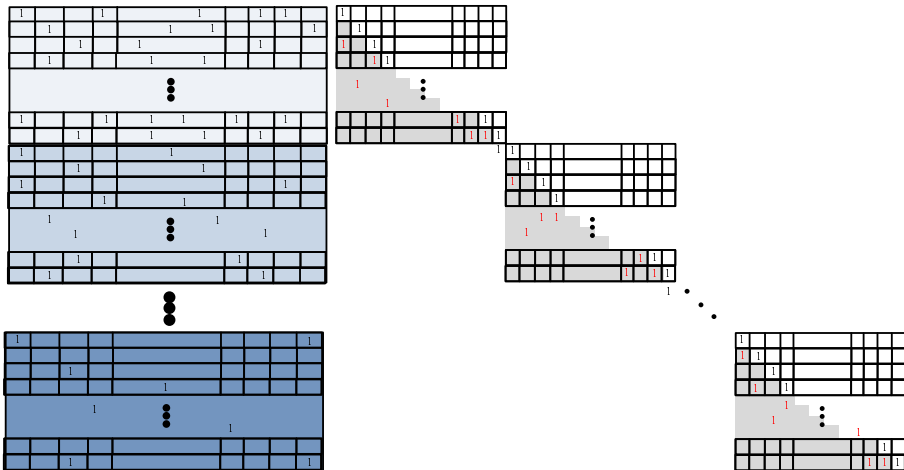
Accumulator randomization algorithm

- 1 Initially, H_w is set to be the identity matrix of size $(n_1 - k) \times (n_1 - k)$.
- 2 For $t = 0, 1, \dots, n_1 - k - 2$, do the following step by step.
 - 1 Find the maximum integer T such that the coding rates $k/(k + T)$ and $k/(k + t + 1)$ falls into the same subinterval, say $(0.05\ell, 0.05(\ell + 1)]$;
 - 2 Choose uniformly at random an integer $i_1 \in [t + 1, T]$;
 - 3 Set the i_1 -th component of the t -th column of H_w to be 1, that is, set $H_w(i_1, t) = 1$.

Remark: The accumulator randomization approach introduces more randomness to the code such that the current parity-check bit depends randomly on previous parity-check bits. It is worth pointing out that both the row-weight concentration algorithm and the accumulator randomization algorithm are executed in an off-line manner. To construct the prefix code of length n_ℓ , both of these two algorithms modify only the incremental $n_\ell - n_{\ell+1}$ rows of the parity-check matrix associated with the original Kite code, which do not affect other rows.

How to narrow the gap from the Shannon limits?

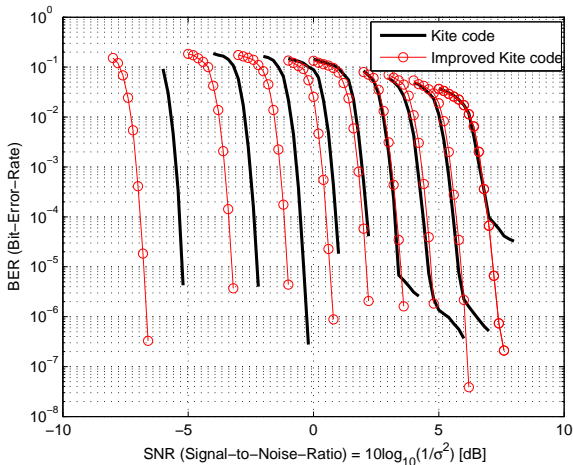
By conducting the *row-weight concentration algorithm* and the *accumulator randomization algorithm*, we can construct a parity-check matrix H given \underline{q} .



How to lower down the error-floors?

Numerical result

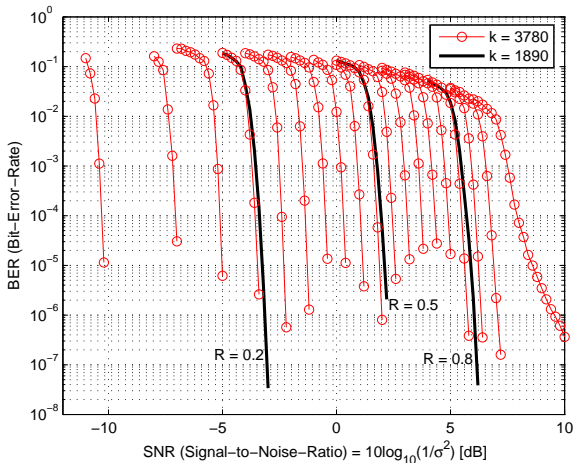
Improved Kite codes with data length $k = 1890$ are constructed and the performances are shown as below.



How to lower down the error-floors?

Numerical result

Improved Kite codes with data length $k = 3780$ are constructed and the performances are shown as below.

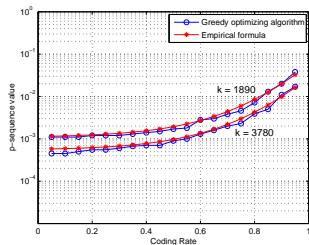


Empirical Formula of p -sequence

To accelerate the design of improved Kite codes, we present the following empirical formula for the p -sequence.

$$q_\ell = \frac{1}{k} \left(\frac{1.65}{(1.5 - 0.05\ell)^6} + 2.0 \right) \quad (1)$$

for $1 \leq \ell \leq 19$. From the following figure, we can see that this formula is well matched to the optimized p -sequence.



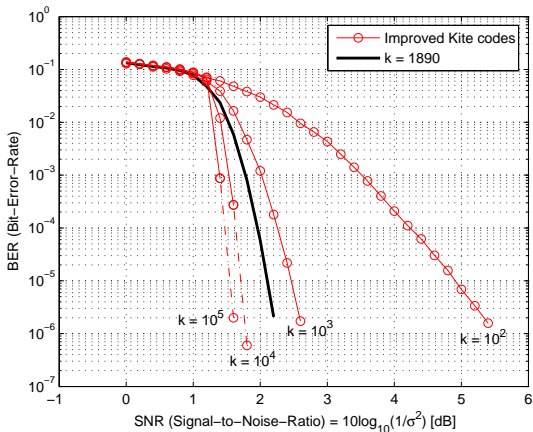
Procedures to construct a code

Given k .

- 1 Compute $\{q_\ell, 1 \leq \ell \leq 19\}$ by the empirical formula for the p -sequence;
- 2 Execute the row-weight concentration algorithm and the accumulator randomization algorithm to construct the parity-check matrix H ;
- 3 To construct a code with rate $R(\geq 0.05)$, we only need take the upper $\lfloor k/R \rfloor - k$ rows of H .

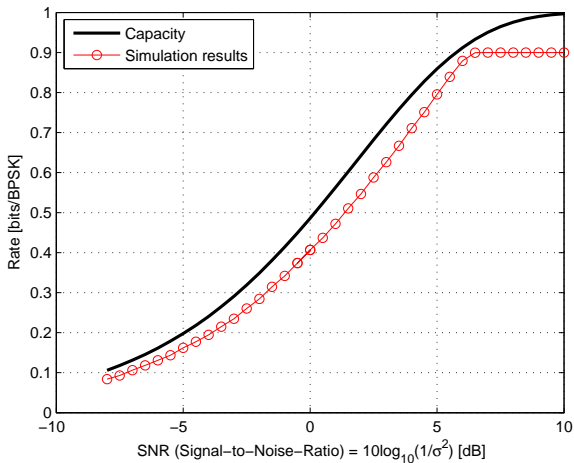
Fixed-rate codes

Kite codes can definitely be used as fixed-rate codes. The nice feature is that we can use the method to construct codes with any given coding rates for arbitrarily given (not too small) data length. Performances of the improved Kite codes of coding rate $1/2$ with $k = 10^2, 10^3, 10^4$ and 10^5 are shown as below.



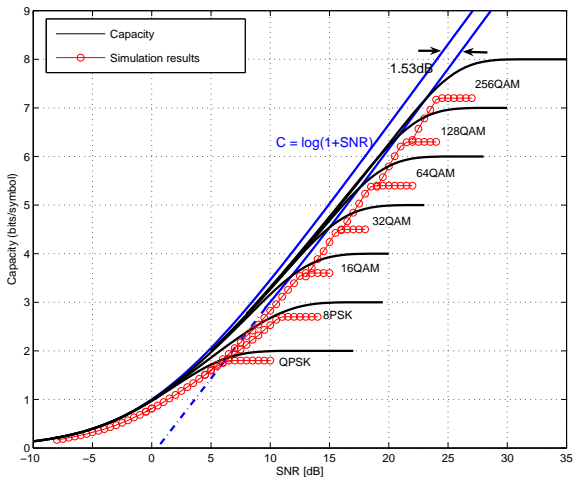
Fountain applications

The average decoding rates (at “zero” error probability) of the improved Kite code with $k = 9450$ over AWGN channels.



Rate-compatible codes and adaptive coded modulation

The average decoding spectral efficiency (at “zero” error probability) of the improved Kite code with data length $k = 9450$ over AWGN channels.



Outline

- 1 Basics of Kite Codes
- 2 Maximum Likelihood Decoding Analysis of Kite Code
- 3 Design of Kite Code
- 4 Improved Design of Kite Code
- 5 Kite Codes over Groups**

Kite codes over groups

Let \mathcal{A}_q be a finite abelian group of size q . Let \mathcal{A}_q^∞ be the set of all infinite sequences over \mathcal{A}_q . A *systematic rateless code* with *degree of freedom* k is defined as

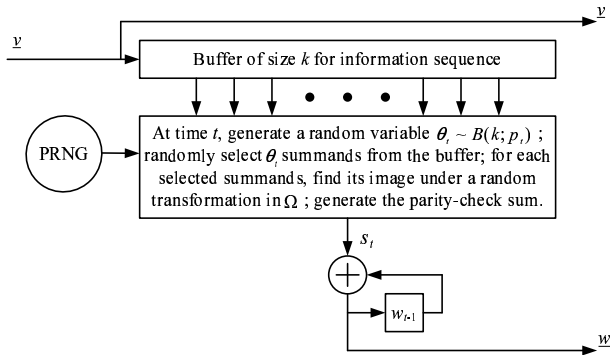
$$C_q[\infty, k] \triangleq \{ \underline{c} = (\underline{v}, \underline{w}) \in \mathcal{A}_q^\infty \mid \underline{v} \in \mathcal{A}_q^k \}, \quad (2)$$

which is a finite subset of \mathcal{A}_q^∞ with size q^k .

An ensemble of Kite codes, denoted by $\mathcal{K}[\infty, k; \underline{p}, \mathcal{A}_q, \Omega]$, is specified by its dimension k , a real sequence $\underline{p} = (p_0, p_1, \dots, p_t, \dots)$ with $0 < p_t < 1$ for $t \geq 0$ and a subset Ω of bijective transformations over \mathcal{A}_q .

Encoder of Kite codes over groups

Let $\underline{v} = (v_0, v_1, \dots, v_{k-1})$ be the information sequence to be encoded. The corresponding codeword is written as $\underline{c} = (\underline{v}, \underline{w})$, where $\underline{w} = (w_0, w_1, \dots, w_t, \dots)$ is the parity-check sequence. The encoding diagram is shown as below.



\underline{v} -- systematic information sequence of length k

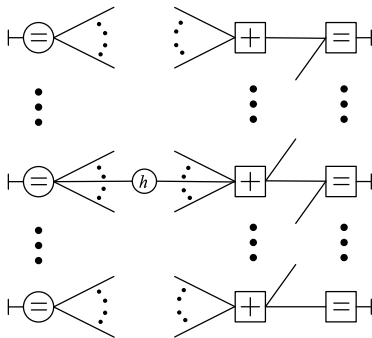
\underline{w} -- parity-check sequence of length $r = n - k$





$B(k; p_t)$ -- binomial distribution with k Bernoulli trials and success probability p_t

PRNG -- pseudo-random number generator

Decoder of Kite codes over groups

The prefix Kite code $\mathcal{K}[n, k; \underline{p}, \mathcal{A}_q, \Omega]$ can be represented by a normal graph shown as below.



-  Information variable node: k in total
-  Parity-check variable node: r in total
-  Check node: r in total
-  Transformation node

Z-Kite codes: Kite codes over one-dimensional lattice

Consider $\mathcal{K}[2100, 1890; \underline{p}, \mathbf{Z}/3\mathbf{Z}, \Omega]$, where $\mathbf{Z}/3\mathbf{Z} = \{-1, 0, +1\}$ and Ω the set of bijective transformations over $\mathbf{Z}/3\mathbf{Z}$. Different from conventional settings for channel coding, we allow the information sequence \underline{v} to have non-uniform *a priori probability*, which may yield shaping gain.

Performances of Z-Kite code

Table: Source Entropies and their Distributions.

$H(V)$ (bits/symbol)	Pr(-1)	Pr(0)	Pr(+1)
$\log_2(3)$	0.3333	0.3334	0.3333
1.3333	0.1886	0.6228	0.1886
1.2222	0.1595	0.6810	0.1595
1.0000	0.1135	0.7730	0.1135
0.8889	0.0946	0.8108	0.0946

Z-Kite codes: Kite codes over one-dimensional lattice

Performances of Z-Kite code

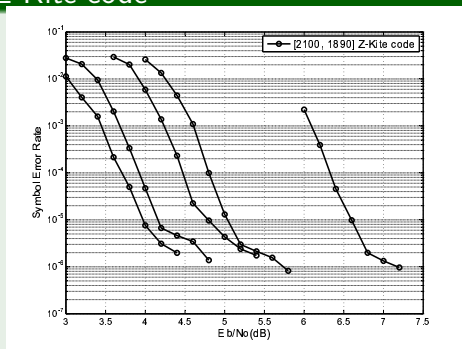


Figure: From left to right, the curves correspond to information rates 0.80, 0.90, 1.10, 1.20 and 1.43 bits/channel symbol, respectively. These curves are obtained by generalizing the original (not the improved) Kite codes.

Lattice-Kite codes: Kite Codes over 4-Dimensional Checkerboard Lattice

Consider the 4-D Lattice-Kite code $\mathcal{K}[2100, 1890; \underline{p}, D_4/4D_4, \Omega_1]$ and 2-D Lattice-Kite code $\mathcal{K}[2100, 1890; \underline{p}, Z^2/4Z^2, \Omega_2]$, where Ω_1 and Ω_2 are the set of bijective transformations of $D_4/4D_4$ and $Z^2/4Z^2$, respectively. Note that $Z^2/4Z^2$ is equivalent to 16QAM signal constellation.

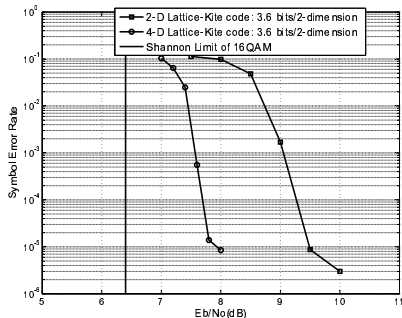
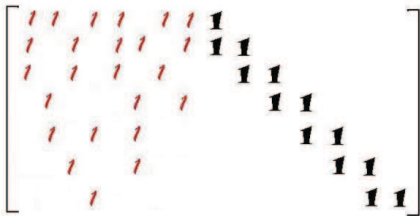


Figure: Performances of Kite (not improved) code [2100, 1890] over $D_4/4D_4$ and $Z^2/4Z^2$.



Codes

How to make a kite that can fly at any designated height?

- Head, tail, and their ratios in size and weight, etc;
- The length and strongness of the thread;

How to design a code that can work at arbitrarily designated rate?

- Denser part, sparser part in the parity-check matrix and their ratios;
- The number of parity-check bits and the strongness of their connections to the information bits.

Conclusions

- We proposed a kind of rateless codes for AWGN channels.
- The ML decoding performances of the proposed codes were analyzed.
- A greedy optimization was presented to optimize Kite codes.
- Three methods were presented either to improve the performance Kite codes, or to accelerate the design of Kite codes.
- Possible applications of Kite codes were investigated.
- Furthermore, Kite codes were generalized into groups and their performances were evaluated.

Thank You for Your Attention!