# Efficient Markov Chain Monte Carlo Algorithms For MIMO and ISI channels

Rong-Hui Peng

Department of Electrical and Computer Engineering
University of Utah

2011/7/16

# Summary of PhD work

- Efficient MCMC algorithms for communication
  - Application to MIMO channel ([Chen-Peng-Ashikhmin-Farhang], *To appear IEEE Trans. Comm.*)
  - Application to noncoherent channel ([Chen-Peng], *to appear IEEE Trans. Comm.'09*)
  - Application to ISI channel and extend to underwater channel ([Peng-Chen-Farhang], *To appear IEEE Trans. Signal Process*.)
  - Achieve near optimal performance with reduced complexity
  - Solve the slow convergence problem
- MIMO-HARQ schemes and combining algorithm ([Peng-Chen], to be *submitted to IEEE Trans. Comm.*)
  - Propose new retransmission scheme
  - Propose new combining algorithm
  - Increase the throughput significantly
  - Applicable to Wimax and LTE

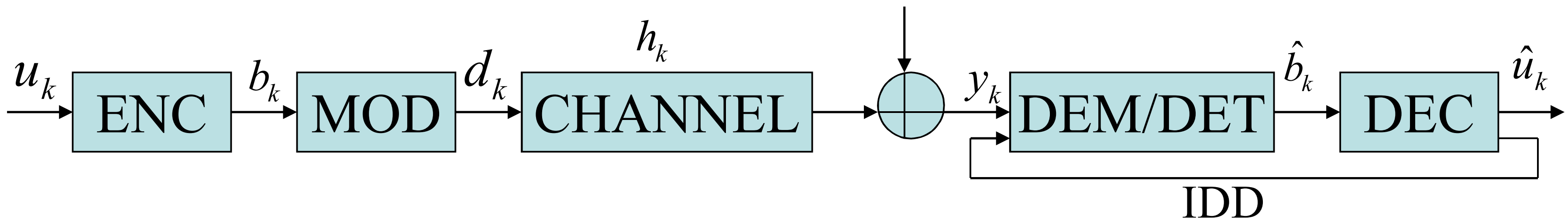
# Summary of PhD work

- Nonbinary LDPC codes ([Peng-Chen], IEEE Trans. Wireless Comm.'08)
  - Nonbinary LDPC coded MIMO system, achieve good performance with reduced complexity
  - Code design based on EXIT chart
  - Hardware-friendly construction
    - Low encoding complexity
    - Parallel architecture
    - Low error floor
- Intern at MERL
  - Low complexity MIMO detection algorithm for MIMO-OFDMA
  - Wimax link-level simulation
  - 1 paper, 1 patent

# Outline

- Background and motivation
- Introduction to MCMC technology
- MCMC MIMO detection
  - Review of MIMO detection
  - QRD-M and MCMC detector
  - Hybrid MIMO detector
  - Complexity analysis
- MCMC ISI equalization
  - Bit-wise MCMC equalizer
  - Group-wise MCMC equalizer
- Conclusion

# Background and motivation




$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{n}$$

MAP detection:

$$\hat{b}_k = \max P(b_k \mid \mathbf{y}, \mathbf{H})$$

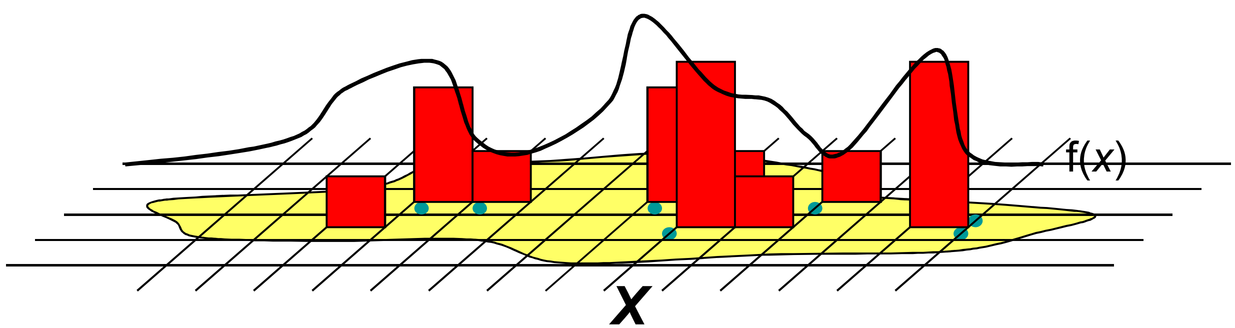
$$P(b_k \mid \mathbf{y}, \mathbf{H}) = \sum_{\mathbf{b}_{-k}} p(\mathbf{Y} \mid \mathbf{H}, \mathbf{d}) P(\mathbf{d})$$

MAP detector has optimal performance but the complexity is exponential with the dimension of $\mathbf{b}_{-k}$

Our objective: reduce the complexity but still obtain near optimal performance

# Markov Chain Monte Carlo

- MCMC is a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its stationary distribution.
- The state of the chain after a large number of steps is then used as a sample from the desired distribution.
- MCMC is suitable for addressing problems involving high-dimensional summations or integrals



# Gibbs sampling

- One kind of the MCMC methods.
- The point of Gibbs sampling is that given a multivariate distribution it is simpler to sample from a conditional distribution rather than integrating over a joint distribution.
- Generate an initial sample as start state
- Using conditional distribution as state transition probability
- Each state jumping allows only one variable change
- Return best state seen across all iterations (may not be the last one)
- Stop after a fixed number of iterations
- Solution is sensitive to the starting state, so we typically run the algorithm several times from different starting points

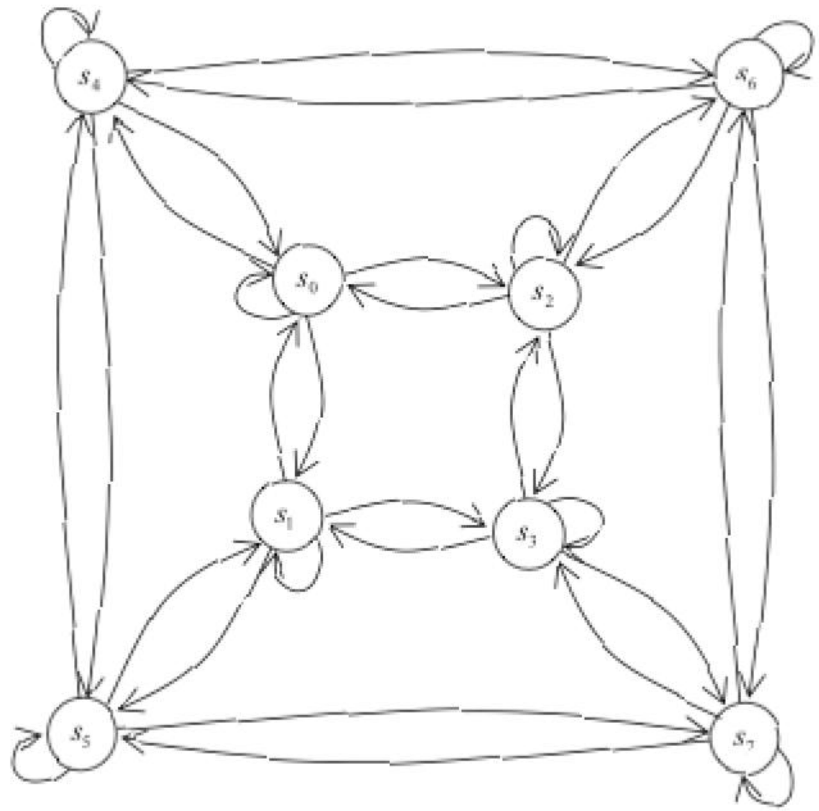
# State transition in Gibbs sampling

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{n} \qquad \mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

Sampling $\mathbf{d}$ according to $p(\mathbf{d} \mid \mathbf{y}, \mathbf{H})$

using conditional PMF $p(d_i \mid \mathbf{y}, \mathbf{H}, \mathbf{d}_{-i})$

| State | $d_3$ | $d_2$ | $d_1$ |
|---|---|---|---|
| $S_0$ | -1 | -1 | -1 |
| $S_1$` | -1 | -1 | +1 |
| $S_2$ | -1 | +1 | -1 |
| $S_3$ | -1 | +1 | +1 |
| $S_4$ | +1 | -1 | -1 |
| $S_5$ | +1 | -1 | +1 |
| $S_6$ | +1 | +1 | -1 |
| $S_7$ | +1 | +1 | +1 |



## Gibbs sampler

Generate initial $\mathbf{d}^{(0)}$ randomly

for n= 1 to $I$

generate $d_0^{(n)}$ from distribution $p(d_0 = b \mid d_1^{(n-1)}, d_2^{(n-1)}, \cdots, d_{N-1}^{(n-1)}, \mathbf{y})$

generate $d_1^{(n)}$ from distribution $p(d_1 = b \mid d_0^{(n)}, d_2^{(n-1)}, \cdots, d_{N-1}^{(n-1)}, \mathbf{y})$

$\vdots$

generate $d_{N-1}^{(n)}$ from distribution $p(d_{N-1} = b \mid d_0^{(n)}, d_1^{(n)}, \cdots, d_{N-2}^{(n)}, \mathbf{y})$

end for

## Why Gibbs sampling works

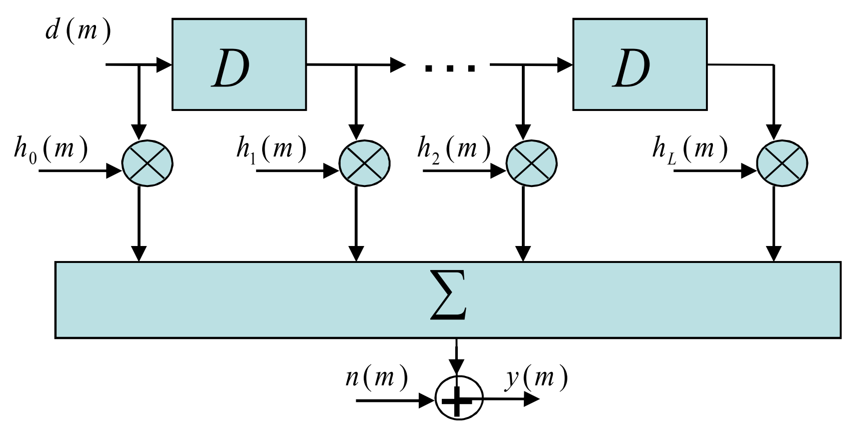
- Retains elements of the greedy approach
  - weighing by conditional PDF makes likely to move towards locally better solutions
- Allows for locally bad moves with a small probability, to escape local maxima (with limitations)

# ISI channel

- Multipath fading channel

$$y(m) = \sum_{l=0}^{L} h_l(m) d(m-l) + n(m),$$

$$\text{for } m = 0, 1, \ldots, N-1+L$$



# ISI channel

- In matrix form

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{n}$$

where

$\mathbf{d} \in \mathbf{C}^{N}$ : transmitting vector

$\mathbf{y} \in \mathbf{C}^{N+L-1}$ : received vector

$\mathbf{H} \in \mathbf{C}^{N+L-1 \times N}$: channel matrix

$\mathbf{n} \in \mathbf{C}^{N+L-1}$: dditive noise vector

$N$ : Equalizer block length

$L$ : Channel memory

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & \cdots & 0 & \cdots & & 0 \\ h_1 & h_0 & 0 & \cdots & & & 0 \\ \vdots & \ddots & & \ddots & & & \vdots \\ 0 & h_L & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & \cdots & 0 & h_L & h_{L-1} & \cdots & h_0 \\ 0 & \cdots & & 0 & \ddots & & \vdots \\ 0 & \cdots & & & 0 & h_L & h_{L-1} \\ 0 & \cdots & & \cdots & & 0 & h_L \end{bmatrix}$$

# Detection for ISI channel

- MAP detection is optimal
  - Efficient implementation of MAP detection is Forward backward algorithm (BCJR)
  - Still exponential complexity with channel memory
- Low complexity algorithm
  - MMSE
  - Decision-feedback
  - …

# Bit-wise MCMC detector

- Transition probability

$$P(b_k = a \mid \overline{\mathbf{b}}_k, \mathbf{y}) \propto p(\mathbf{y} \mid \mathbf{b}^a) P(b_k = a)$$

$$= p(\mathbf{y} \mid \mathbf{d}^a) P(b_k = a)$$

$$= \prod_{j=0}^{N+L-1} p(y_j \mid \mathbf{d}^a_{j-L:j}) P(b_k = a)$$

$$= \boxed{\left\{ \prod_{j=0}^{i-1} p(y_j \mid \mathbf{d}^a_{j-L:j}) \prod_{j=i+L+1}^{N+L-1} p(y_j \mid \mathbf{d}^a_{j-L:j}) \right\}} \prod_{j=i}^{i+L} p(y_j \mid \mathbf{d}^a_{j-L:j}) P(b_k = a)$$

$$= C \cdot \prod_{j=i}^{i+L} p(y_j \mid \mathbf{d}^a_{j-L:j}) P(b_k = a)$$

$$\mathbf{b}^a = \{b_0^{(n)}, \cdots, b_{k-1}^{(n)}, a, b_{k+1}^{(n-1)}, \cdots, b_{M_b N-1}^{(n-1)}\},$$

$\mathbf{d}^a$ denote the symbol vector corresponding to $\mathbf{b}^a$

$b_k$ is mapped to $d_i$

# Bit-wise MCMC detector

- Computing the a posteriori LLR
  - Accurate posteriori LLR involve computations over the whole block (may be very cumbersome since *N* can be very large)
  - Truncated window to approximate

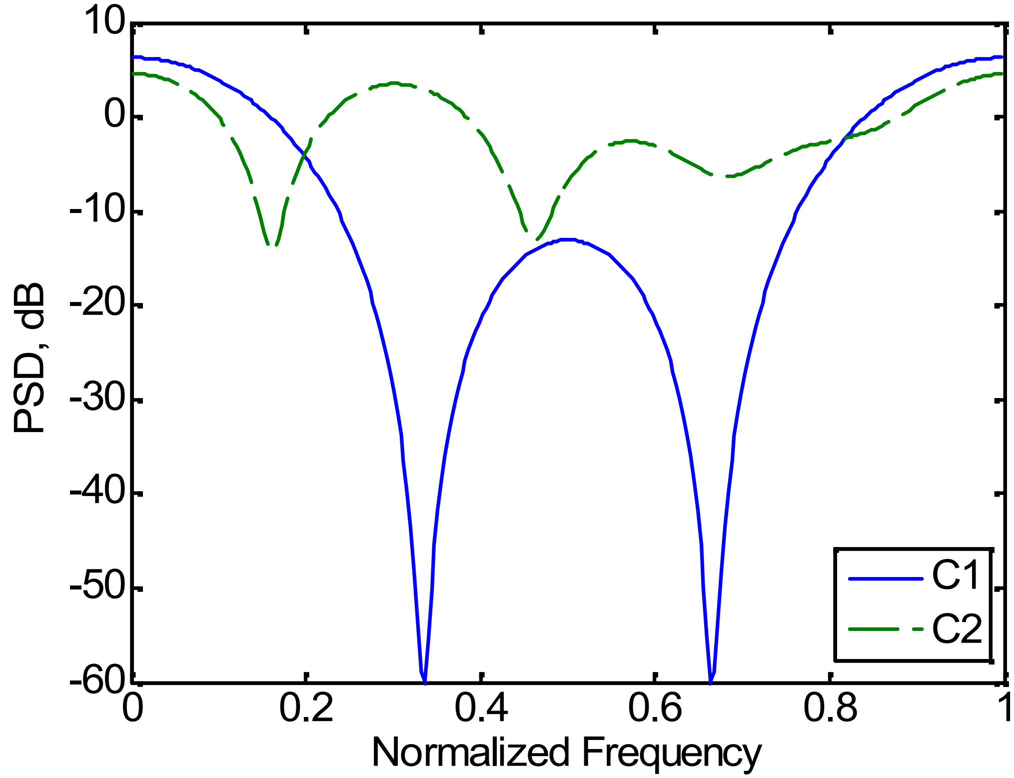
$$\lambda_{1,k}^{e} = \ln \frac{\sum_{\mathbf{b}_{i_1:i_2} \in \mathrm{I}_{i_1:i_2}^{k,0}} p(\mathbf{y}_{i:i+L} \mid \mathbf{b}_{i_1:i_2}) \prod_{l=i_1}^{i_2} P(b_l)}{\sum_{\mathbf{b}_{i_1:i_2} \in \mathrm{I}_{i_1:i_2}^{k,1}} p(\mathbf{y}_{i:i+L} \mid \mathbf{b}_{i_1:i_2}) \prod_{l=i_1}^{i_2} P(b_l)} - \lambda_{2,k}^{e}$$

$\mathrm{I}_{i_1:i_2}$ the set which truncate sequences in I

with bits $\{b_k, i_1 \le k \le i_2\}$

# Severe ISI channel

- For channels with severe ISI, bit-wise MCMC suffers from slow convergence problem
- We found slow convergence problem for ISI channel is mainly caused by high posterior correlation




$$h_1[n] = 0.227\delta[n] + 0.46\delta[n-1] + 0.688\delta[n-2] + 0.46\delta[n-3] + 0.227\delta[n-4]$$

$$h_2[n] = (2+04i)\delta[n] + (15+18i)\delta[n-1] + \delta[n-2] + (12-13i)\delta[n-3] + (08+16)\delta[n-4]$$

# Severe ISI channel

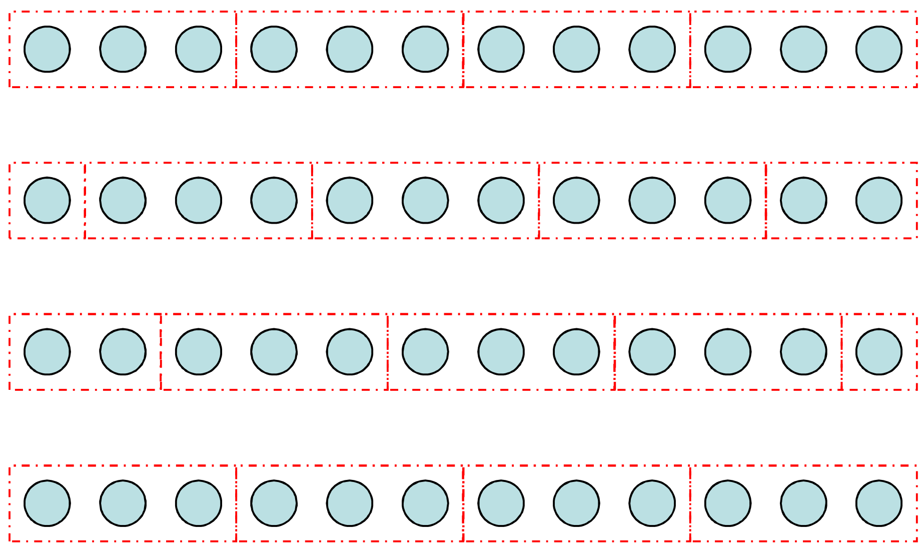
- Our solution
  - Grouping (blocking) the highly correlation variables in Gibbs sampler

1$^{st}$ iteration:

2$^{nd}$ iteration:

3$^{rd}$ iteration:

4$^{th}$ iteration:



Shift grouping different adjacent symbols over iterations to speed up the mixing rate of Gibbs sampler

## Transition probability

- Grouping multiple variables allow a large sample space containing $r = 2^{M_b G}$ values

$$
\begin{bmatrix} y_0 \\ \vdots \\ y_{i-1} \\ y_i \\ \vdots \\ y_{i+P} \\ y_{i+P+1} \\ \vdots \\ y_{N+L-1} \end{bmatrix}
=
\begin{bmatrix}
h_0 & 0 & \cdots & 0 & \cdots & & 0 \\
h_1 & h_0 & 0 & \cdots & & & 0 \\
\vdots & \ddots & & \ddots & & & \vdots \\
0 & h_L & \cdots & h_0 & 0 & \cdots & 0 \\
\vdots & & \ddots & & \ddots & & \vdots \\
0 & \cdots & 0 & h_L & h_{L-1} & \cdots & h_0 \\
0 & \cdots & & 0 & \ddots & & \vdots \\
0 & \cdots & & & 0 & h_L & h_{L-1} \\
0 & \cdots & & \cdots & & 0 & h_L
\end{bmatrix}
\begin{bmatrix} d_0 \\ \vdots \\ d_{i-1} \\ d_i \\ \vdots \\ d_{i+G} \\ d_{i+G+1} \\ \vdots \\ d_{N-1} \end{bmatrix}
+
\begin{bmatrix} n_0 \\ \vdots \\ n_{i-1} \\ n_i \\ \vdots \\ n_{i+P} \\ n_{i+P+1} \\ \vdots \\ n_{N+L-1} \end{bmatrix}
$$

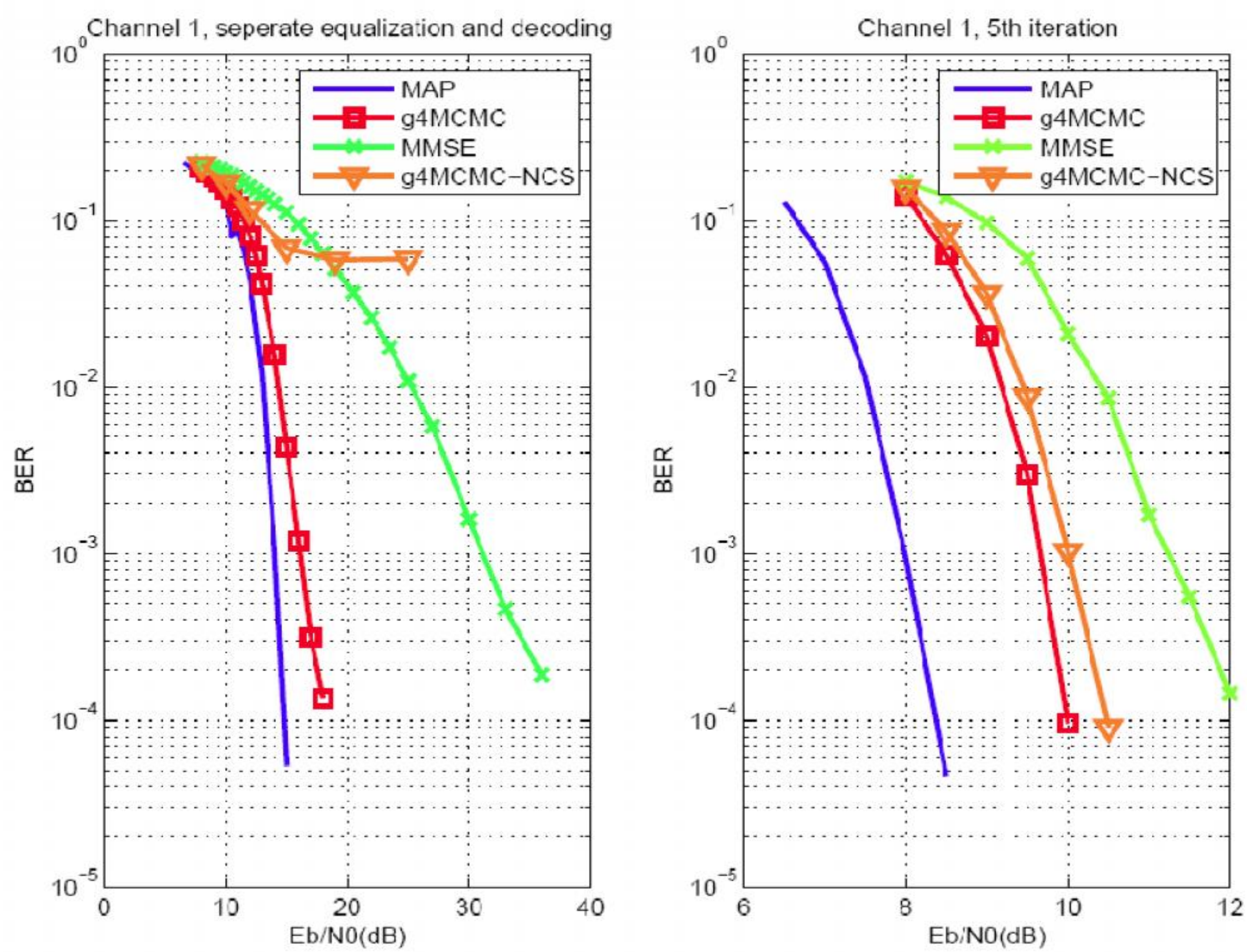
$$P = G + L + 1$$

# Transition probability

$$
\begin{aligned}
\tilde{\mathbf{y}}_{i:i+P} &= \mathbf{y}_{i:i+P} - \mathbf{H}_{i:i+P}^{i-L:i-1}\mathbf{d}_{i-L:i-1} - \mathbf{H}_{i:i+P}^{i+G:i+P}\mathbf{d}_{i+G:i+P} + \mathbf{n}_{i:i+P} \\
&= \mathbf{H}_{i:i+P}^{i+G:i+P}\mathbf{d}_{i:i+G-1} + \mathbf{n}_{i:i+P}
\end{aligned}
$$

- Can be thought as received signals of a MIMO channel with *G* transmit antenna *and P*+1 receive antenna
- Can apply QRD-M to find $r_0 = r$ samples with large APPs
- Gibbs sampler randomly chooses one sample from $r_0$ samples

# Simulation results




Performance of the channel with strong ISI

## Summary of results

- MCMC equalizer for ISI channel is studied
- Near optimal performance can be obtained
- Slow convergence is mainly caused by high posterior correlation
- QRD-M algorithm is applied to reduce the complexity of group-wise MCMC equalizer
- Parallel implementation is proposed

# Conclusions

- Efficient MCMC algorithms for MIMO and ISI channel are studied
  - For MIMO, MCMC works well at low SNR region
  - For ISI, MCMC works well for moderate ISI
- Solutions for slow convergence are proposed
  - For MIMO, use QRD-M as good start point
  - For ISI,
    - Use group-wise Gibbs sampler to group high correlated variables
    - Use QRD-M to reduce the complexity of group-wise Gibbs sampler
- Future work
  - Study the sensitivity of proposed MIMO detector with more practical channel model
  - Study the time-varing ISI channel and adaptive grouping scheme for group-wise MCMC
  - Hardware implementation of MCMC

## Motivation

- Optimal binary code has been designed to approach channel capacity.
  - Long codes
  - Irregular
- Nonbinary LDPC code design has been studied for AWGN and shows better performance than binary codes.
  - Shorter codes
  - More regular

## Contribution

- Apply nonbinary LDPC codes to fading channels and MIMO channels and provide comparison with optimal binary LDPC coded systems
- Propose modified nonbinary LDPC decoding algorithm.
- Extend EXIT chart to nonbinary code design
- Propose parallel sparse encodable nonbinary code with low encoding and decoding complexity

## Introduction of binary LDPC codes

- A subclass of linear block codes
- Specified by a parity check matrix *(n-k)* $\times$ *n*

  *n*: code length *k*: length of information sequence

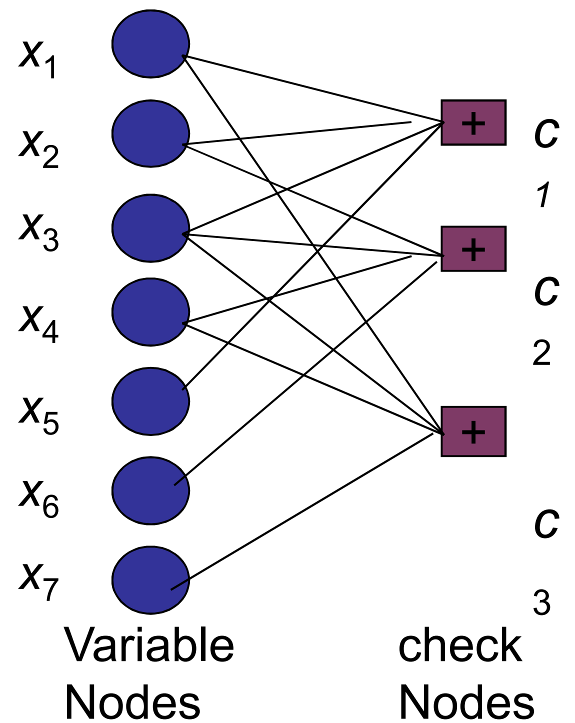
$$
\mathbf{H} = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} x_1 & x_2 & & \Lambda & & & x_7 \end{array} \\
\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}
\end{array}
$$

$$\mathbf{Hx}^T = \mathbf{0}$$

$$c_1 : x_1 + x_2 + x_3 + x_5 = 0$$

$$c_2 : x_2 + x_3 + x_4 + x_6 = 0$$

$$c_3 : x_1 + x_3 + x_4 + x_7 = 0$$



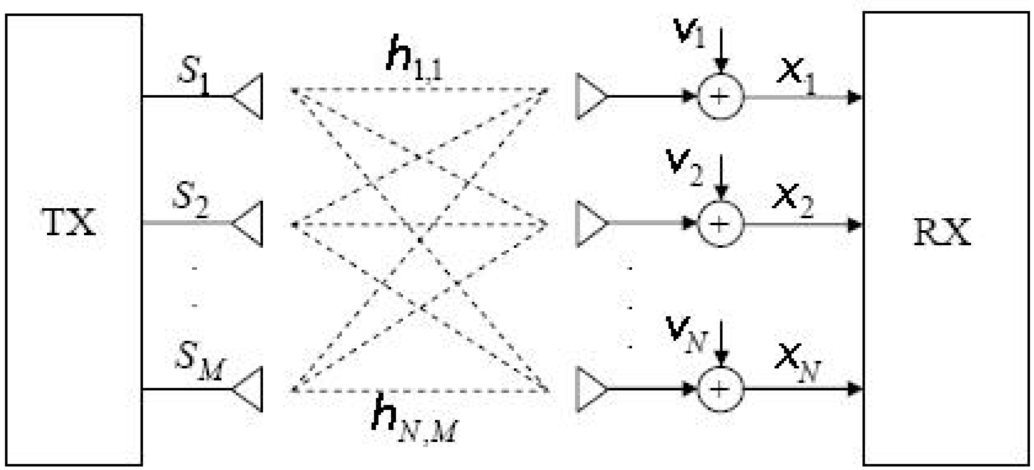
## Definition of nonbinary LDPC codes

- For nonbinary codes, the ones in parity check matrix are replaced by nonzero elements in GF($q$)

$$\mathbf{H}=\begin{bmatrix} 3 & 7 & 1 & 0 & 3 & 0 & 0 \\ 0 & 2 & 5 & 6 & 0 & 3 & 0 \\ 4 & 0 & 2 & 7 & 0 & 0 & 5 \end{bmatrix}$$

## Application to fading channels

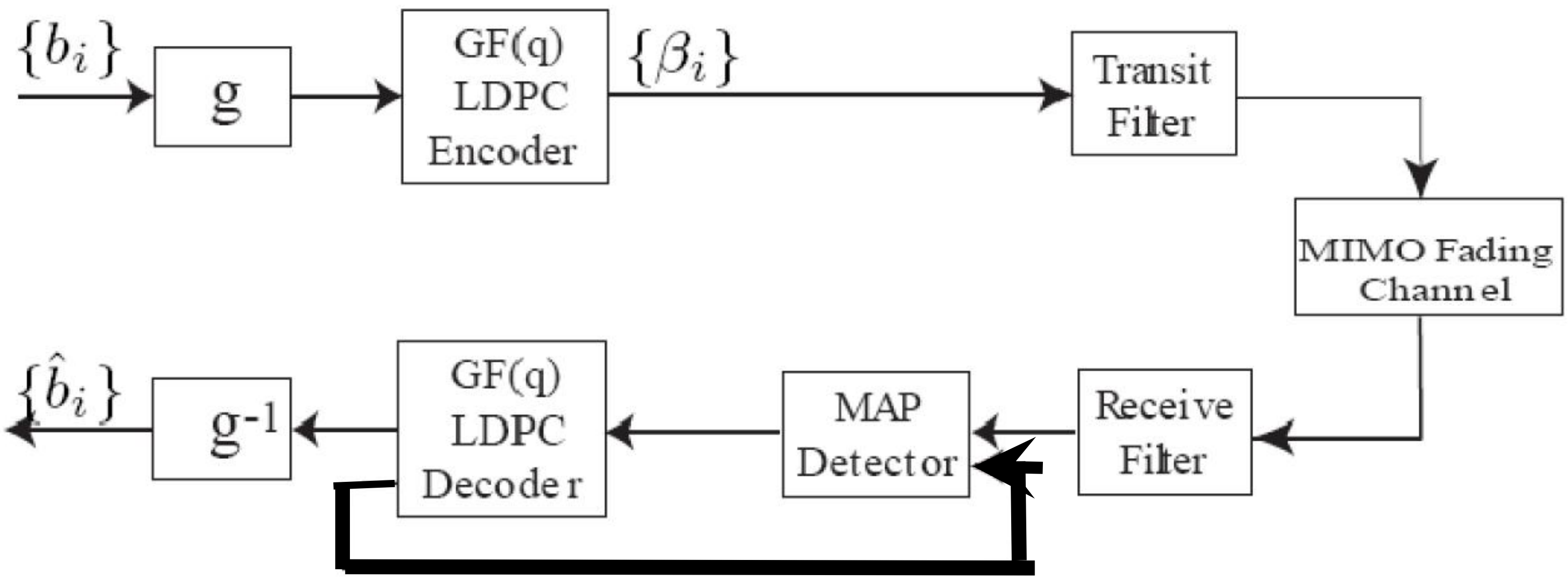
- Channel model




$$\mathbf{X} = \sqrt{\frac{\rho}{M}}\mathbf{HS} + \mathbf{V}$$

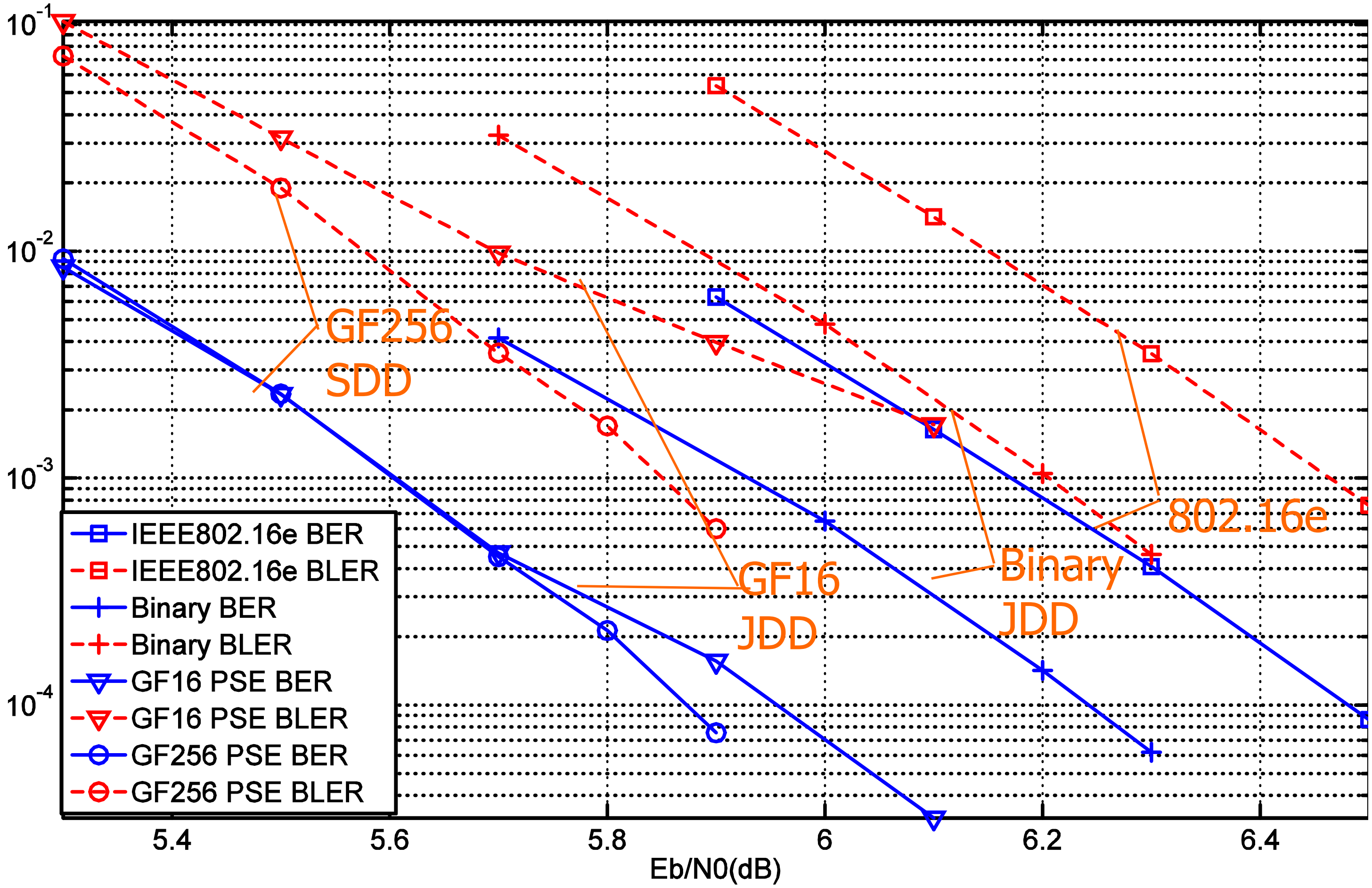
Assume each entry of channel matrix is independent, follows Rayleigh fading, and is known by receiver

## System block diagram




Separate detection and decoding: the detection is performed only once.

## Performance comparison




Performance: GF256>GF16> binary>802.16e

Complexity: GF16<GF256 ≅ 802.16e<binary

Performance comparison of GF(256) SDD, GF(16) JDD and binary JDD system

## Construction of Parallel sparse encodable codes

- Motivation
  - Low encoding complexity
  - Allow parallel implementation
- Parallel sparse encodable (PSE) codes = Qausi cyclic (QC) cycle codes + tree codes
  - QC codes: parallel implementation
  - Cycle code: Low encoding complexity

## Quasi-cyclic construction

- Quasi-cyclic structure

$$\mathbf{H} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,n} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{m,1} & \mathbf{A}_{m,2} & \cdots & \mathbf{A}_{m,n} \end{bmatrix}$$

- $A_{i,j}$ is a circulant: each row is a right cycle-shift of the row above it and the first row is the right cycle-shift of the last row
- The advantage of QC structure
  - Allow linear-time encoding using shift register
  - Allow partially parallel decoding
  - Save memory

## A new QC structure for GF($q$)

- $A_{i,j}$ is a *nonbinary multiplied circulant permutation matrix*

$$\mathbf{A}_{i,j} = \begin{pmatrix} K & 0 & \delta_{i,j} & 0 & \Lambda & \Lambda & 0 \\ \Lambda & & 0 & \delta_{i,j}\beta & 0 & \Lambda & M \\ M & \Lambda & \Lambda & 0 & \delta_{i,j}\beta^{2} & 0 & M \\ M & & & M & 0 & O & 0 \\ M & & & & M & 0 & O \\ O & M & M & & & M & 0 \\ \Lambda & \delta_{i,j}\beta^{q'-2} & 0 & \Lambda & & & 0 \end{pmatrix}$$

$\mathrm{GF}(q') \subset \mathrm{GF}(q)$; $\beta$ is a primitive element of $\mathrm{GF}(q')$

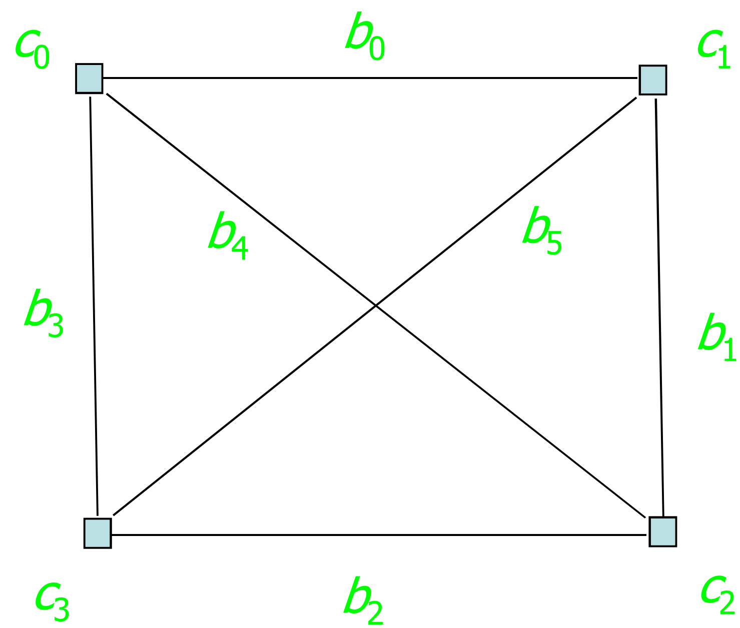
$\delta_{i,j}$ is randomly choosen from $\{\alpha^{0}, \alpha^{1}, \Lambda\ , \alpha^{(q-1/(q'-1)-1}\}$

$\alpha$ is a primitive element of $\mathrm{GF}(q)$

- With degree 2 variable nodes only
- Can be represented by normal graph, every vertex imposes one linear constraint
  - Columns => edges; rows => vertices

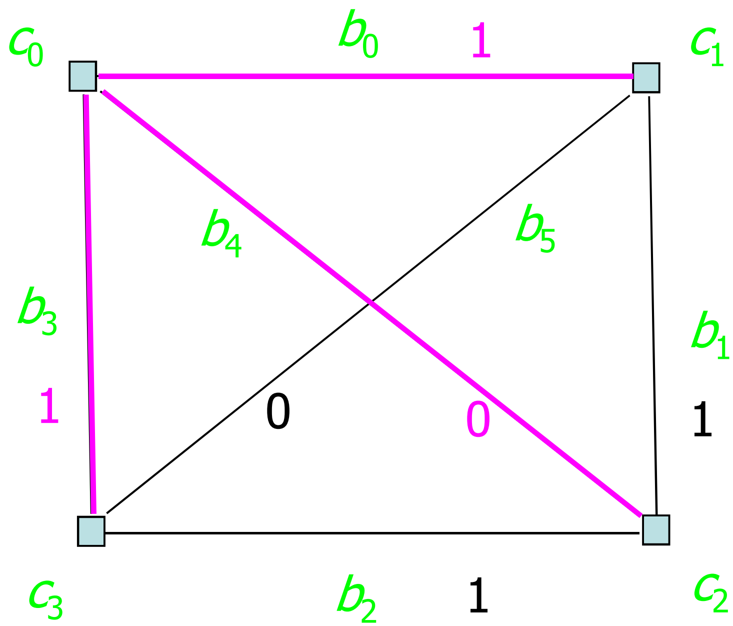
$$\mathbf{H}_m = \begin{array}{c} \\ c_0 \\ \\ \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccccc} b_0 & b_1 & \dots & & & b_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

Parity check matrix




Normal graph representation

## H based encoding of cycle codes




① Find the spanning tree:

$b_0, b_3, b_4$

② Information bits =>

Edges outside SP

$b_1=1, b_2=1, b_5=0$

③ Compute coded bits

$b_3=b_2+b_5=1$

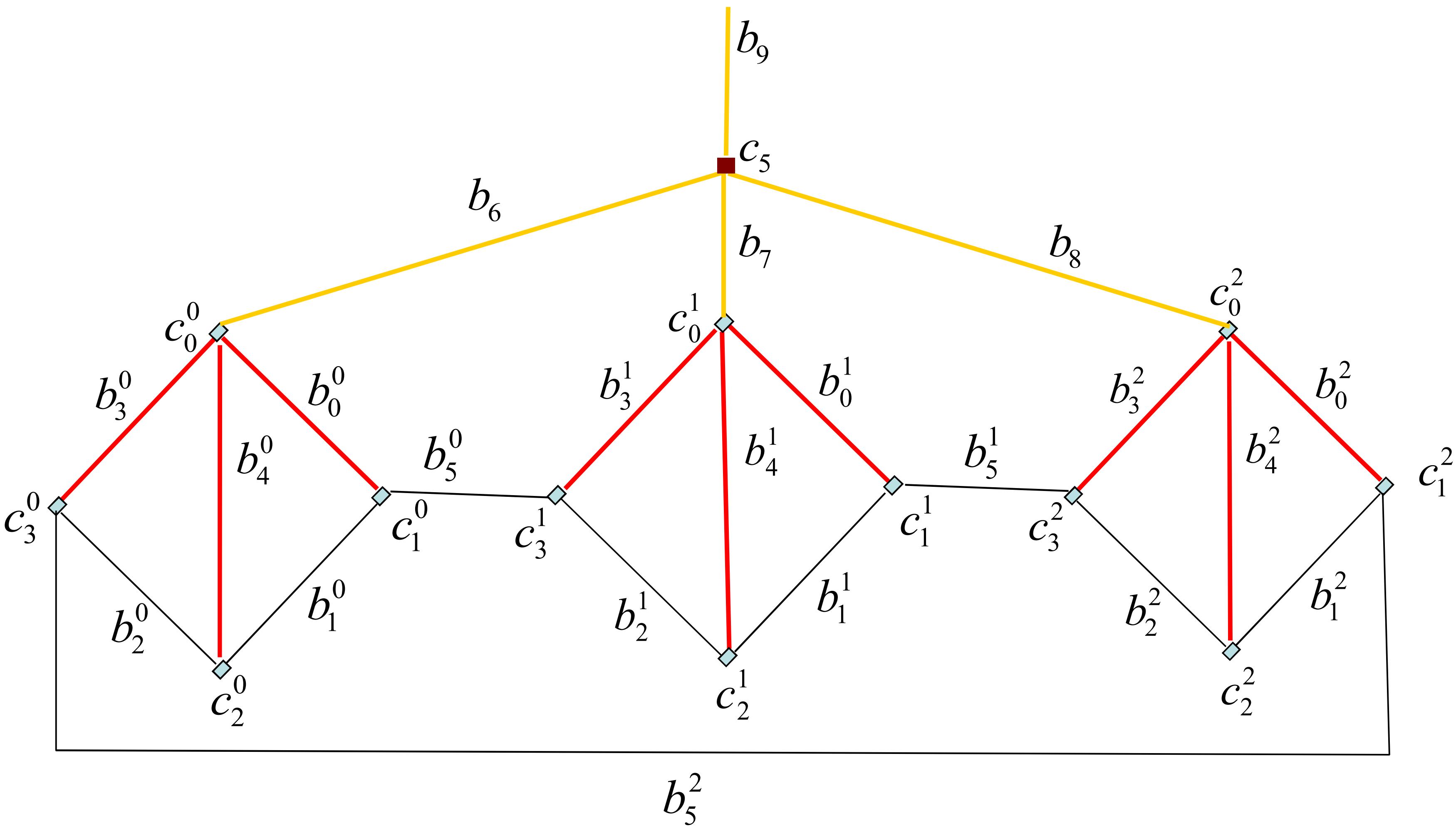
$b_4=b_2+b_1=0$

$b_0=b_1+b_5=1$

For binary code: check $c_0$ is always satisfied.

$b_0+b_3+b_4=0$

Not work for nonbinary cycle codes! Why?

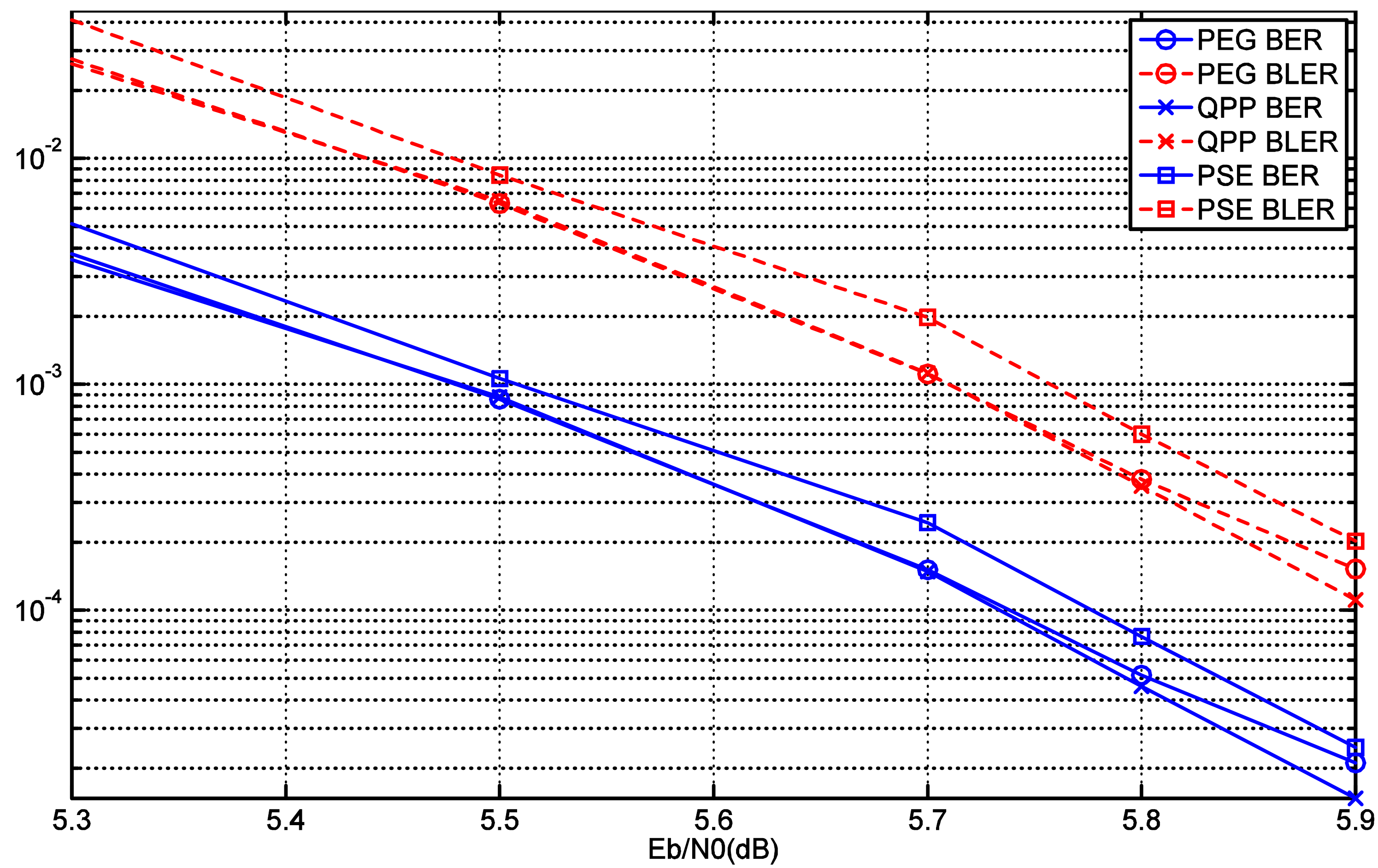



PSE codes = QC cycle codes + tree codes

# Encoding of PSE codes

- Parity check matrix based encoding
- Parallel encoding for QC cycle subcode
- Much lower encoding complexity than normal LDPC codes using generator matrix based encoding because the generator matrix is usually dense

## Simulation results




Performance comparison of PSE codes over GF(256) with QPP (QC) codes and PEG (non-QC) codes

**Thank you !**