Further study of Advanced MIMO receiver

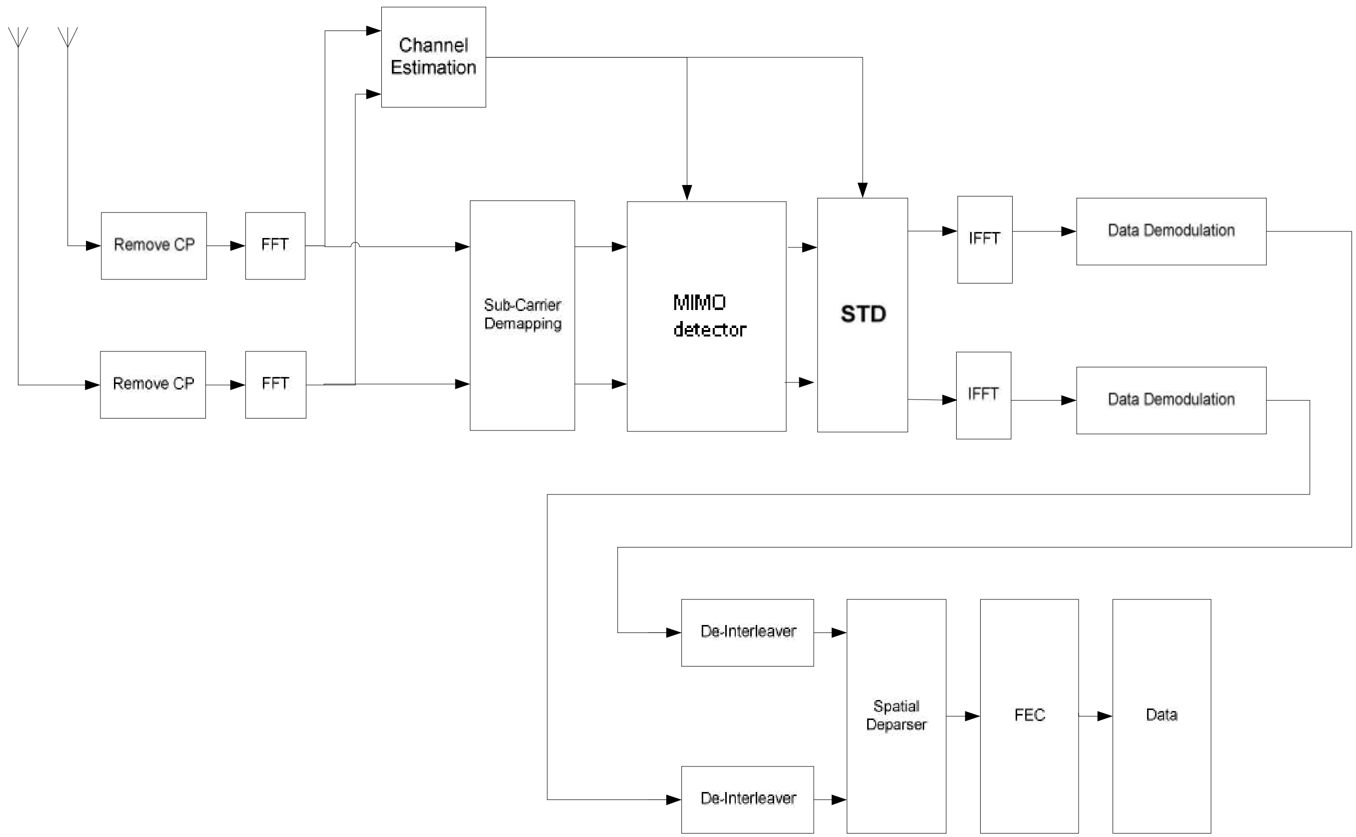
Ronghui Peng

10/5/2008

Outline

- MIMO detection
- Channel estimation

Receiver block diagram



2011/7/16

Receiver design

- Receiver design is challenging
  - Synchronization
  - Channel estimation
  - MIMO detection
  - Channel decoding
- Balance between performance and complexity

Detection in MIMO

- Consider frequency selective channel, OFDM is used to convert the frequency selective channel to a number of parallel flat fading channels. Accordingly, each subcarrier channel has the following model:

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

where

$\mathbf{d} \in \mathbf{C}^{N_t}$ is a vector of transmit symbols

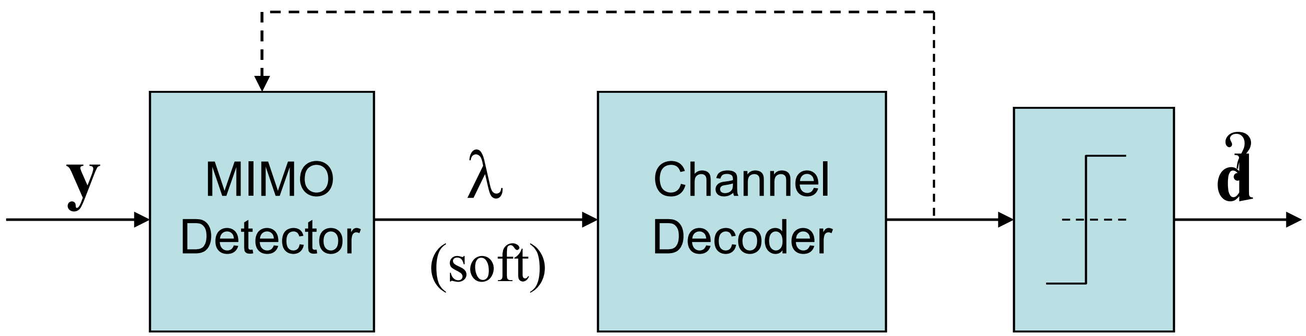
$\mathbf{y} \in \mathbf{C}^{N_r}$ is a vector of received signal

$\mathbf{H} \in \mathbf{C}^{N_r \times N_t}$ is the channel gain matrix

$\mathbf{n} \in \mathbf{C}^{N_r}$ is an additive noise vector

N_t, N_r is the number of transmit antenna and receive antenna

Receiver structure



• Separate detection and decoding (SDD) : no feedback from channel decoder

• Joint detection and decoding (JDD) : exchange soft information between detector and decoder

MIMO detection

- Maximal likelihood (ML) detection or Maximum *a posteriori* (MAP) is optimal
- Optimal detection usually has exponential complexity and is computation infeasible for practical system
- Low complexity sub-optimal detectors
  - MMSE, SIC, VBLAST …
- Approximate optimal detectors
  - Sphere decoding, MCMC …

Hard output VS. Soft output

- Hard output detector make hard decision after detection
  - ML: $$\max_{\mathbf{d}} P(\mathbf{y} \mid \mathbf{d})$$
- Soft output detector generate soft message, usually log likelihood ratio (LLR). It will be used by soft channel decoder
  - MAP:

$$\lambda_k = \ln \frac{P(b_k = +1 \mid \mathbf{y})}{P(b_k = -1 \mid \mathbf{y})} = \ln \frac{\sum_{\mathbf{b}_{-k}} P(b_k = +1, \mathbf{b}_{-k} \mid \mathbf{y})}{\sum_{\mathbf{b}_{-k}} P(b_k = -1, \mathbf{b}_{-k} \mid \mathbf{y})} \approx \ln \frac{\max_{\mathbf{b}_{-k}} P(b_k = +1, \mathbf{b}_{-k} \mid \mathbf{y})}{\max_{\mathbf{b}_{-k}} P(b_k = -1, \mathbf{b}_{-k} \mid \mathbf{y})}$$

where

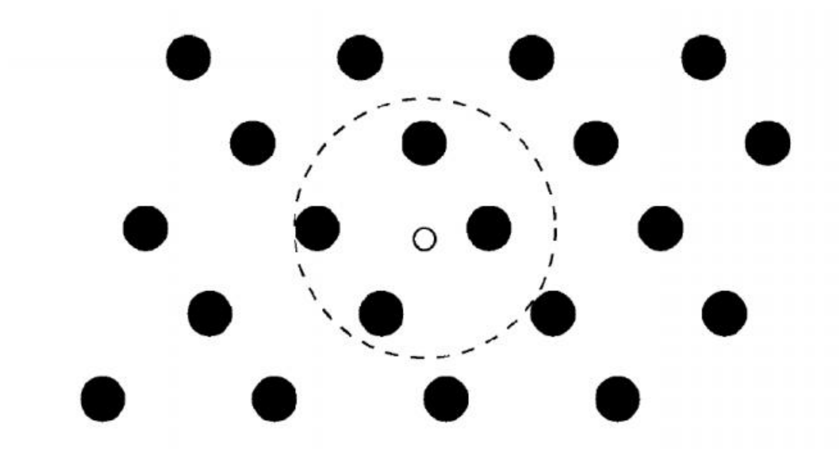
$$\mathbf{d} = (b_1, b_2, \Lambda\ , b_{N_t M_c}); \quad \mathbf{b}_{-k} = (b_1, \Lambda\ b_{k-1}, b_{k+1}, \Lambda\ , b_{N_t M_c}); \quad b_i \in \{-1, 1\}$$

ML detection

- The ML detection solves optimally the closest lattice point problem, i.e., finds **d** which minimizes

$$\hat{\mathbf{d}}_{ML} = \arg\min_{\mathbf{d}} \left\| \mathbf{y} - \mathbf{H}\mathbf{d} \right\|^2$$

- The problem can be solved with exhaustive search, i.e., checking all possible symbol vectors and selecting the closest point
  - Computationally very heavy and often not practical



Low complexity detection

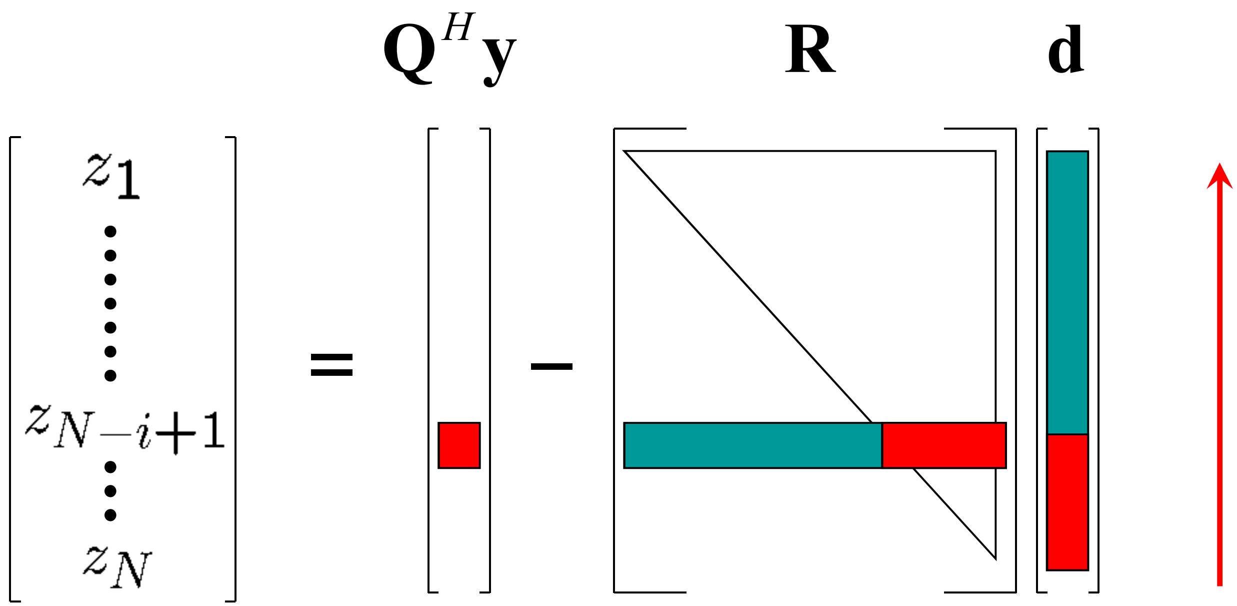
- Zero-forcing detector:
  - Estimate of $\mathbf{d} = \mathbf{Q}\{(\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*\mathbf{y}\}$
- MMSE detector:
  - Estimate of $\mathbf{d} = Q\{(\mathbf{H}^*\mathbf{H}+\sigma^2\mathbf{I})^{-1}\mathbf{H}^*\mathbf{y}\}$
- VBLAST/Successive Interference Canceller (SIC) detector:
  - Detects the streams with highest SNR, subtract the detected streams, and continue with the successive detection and cancellation of the rest of the streams.
- Sphere decoding
  - Limit the search to a sphere around the most likely symbols
  - The radius of the sphere can be adjusted to achieve a tradeoff of complexity and performance

Low complexity detection

- ZF has lowest complexity but worst performance
- VBLAST is ordered version of ZF or MMSE
- Sphere decoding and Markov chain Monte Carlo (MCMC) is a class of algorithms that have potential to achieve ML performance at relatively lower complexity
- Sphere decoding is more popular and extensively studied in the literature
  - Easy to understand and more flexibility to improve

Sphere decoding

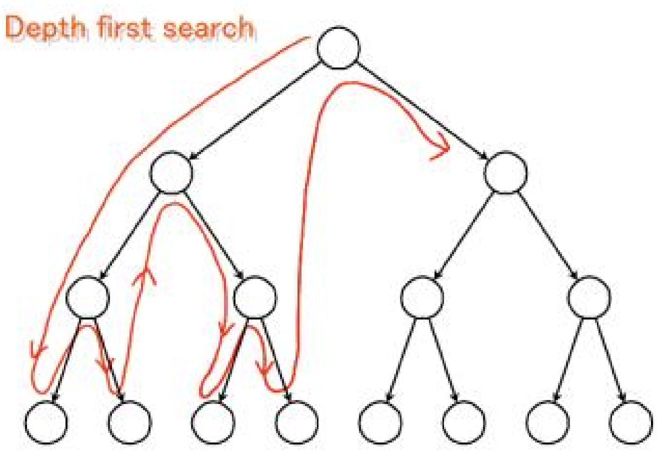
$$r^2 = \|\mathbf{y} - \mathbf{Hd}\|^2 = \|\mathbf{y} - \mathbf{QRd}\|^2 = \|\mathbf{Q}^H\mathbf{y} - \mathbf{Rd}\|^2 \quad (N_t \le N_r)$$

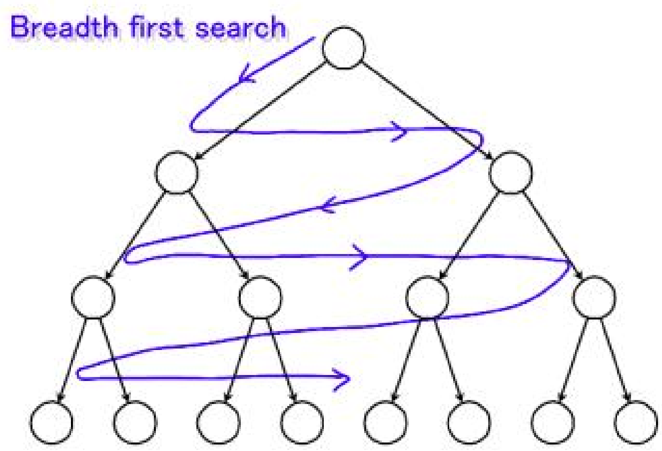


$$\lambda_i = |z_{N-i+1}|^2$$

Tree search

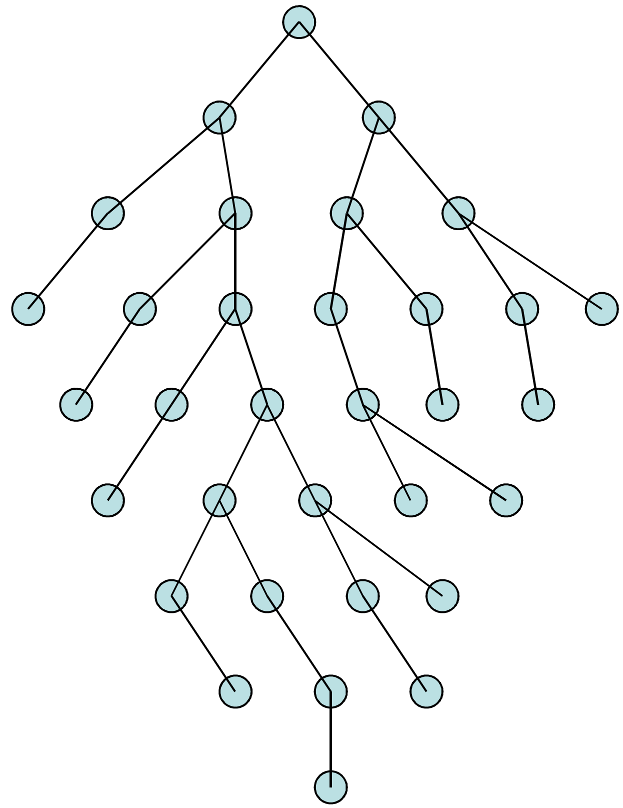
- Sequential sphere decoding –Depth-First Search (DFS)
  - Variable throughput with average polynomial complexity but exponential at low SNR
  - Can be optimum
- K-best sphere decoding (QRM-MLD) – Breadth-First Search (BFS)
  - Fixed number of visited nodes, constant throughput
  - Sub-optimal





Sequential sphere decoding

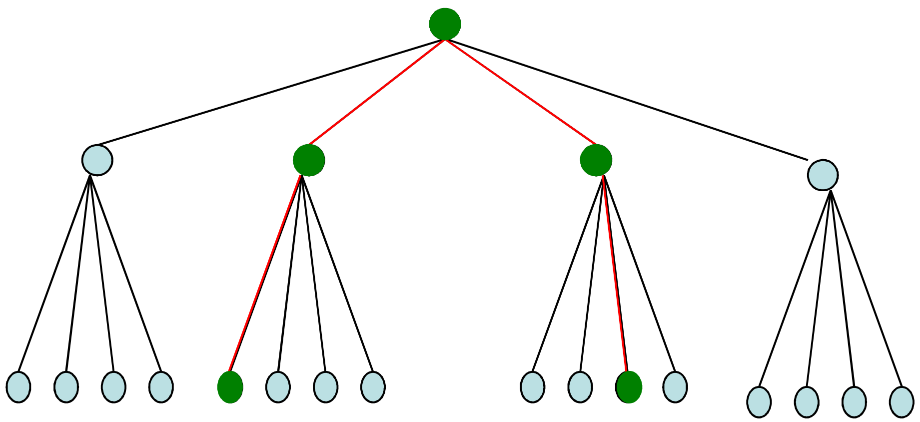
$$
\begin{aligned}
\lambda_1 &< r^2 \\
\lambda_1 + \lambda_2 &< r^2 \\
&\vdots \\
\lambda_1 + \cdots + \lambda_i &< r^2 \\
&\vdots \\
\lambda_1 + \lambda_2 + \cdots + \lambda_N &< r^2
\end{aligned}
$$



The parameter *r* decide the complexity and accuracy of Algorithm, different SNR need to decide *r*.

QRM-MLD

- At level 1, select M path as survival path with minimal λ_1 discard other path
- At level 2, From the survival nodes of level 2, select M path with minimal $\lambda_1 + \lambda_2$ and discard other path
- ……



$\mathrm{M} = 2$

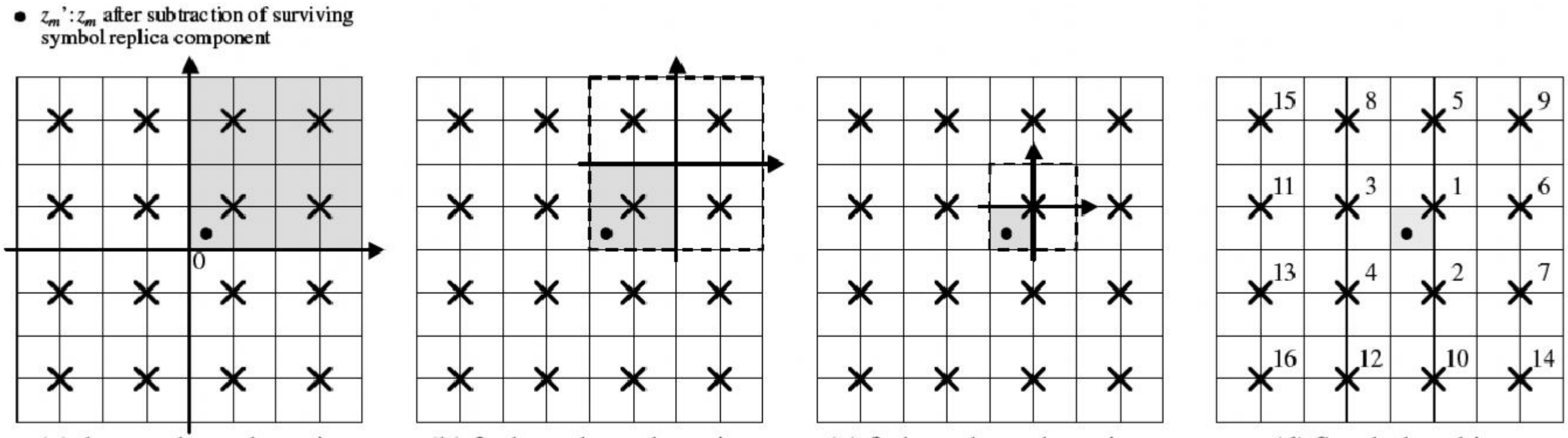
Complexity of QRM-MLD

- Need preprocessing QR decomposition, $O(N_t^3)$
- At each layer (except root layer), MM_c square euclidian distance calculations are needed to find M minimal distance from MM_c path where M_c is the number of symbols in a constellation
- The complexity of QRM-MLD depends on the parameter M, $O(MM_c)$
- If M is too large, the complexity is very high; if M is too small, good candidates have been discarded before the process proceeds to the lowest layer, note that M is also increase with the dimension of problem to preserve performance.

QRM-MLD with ASESS

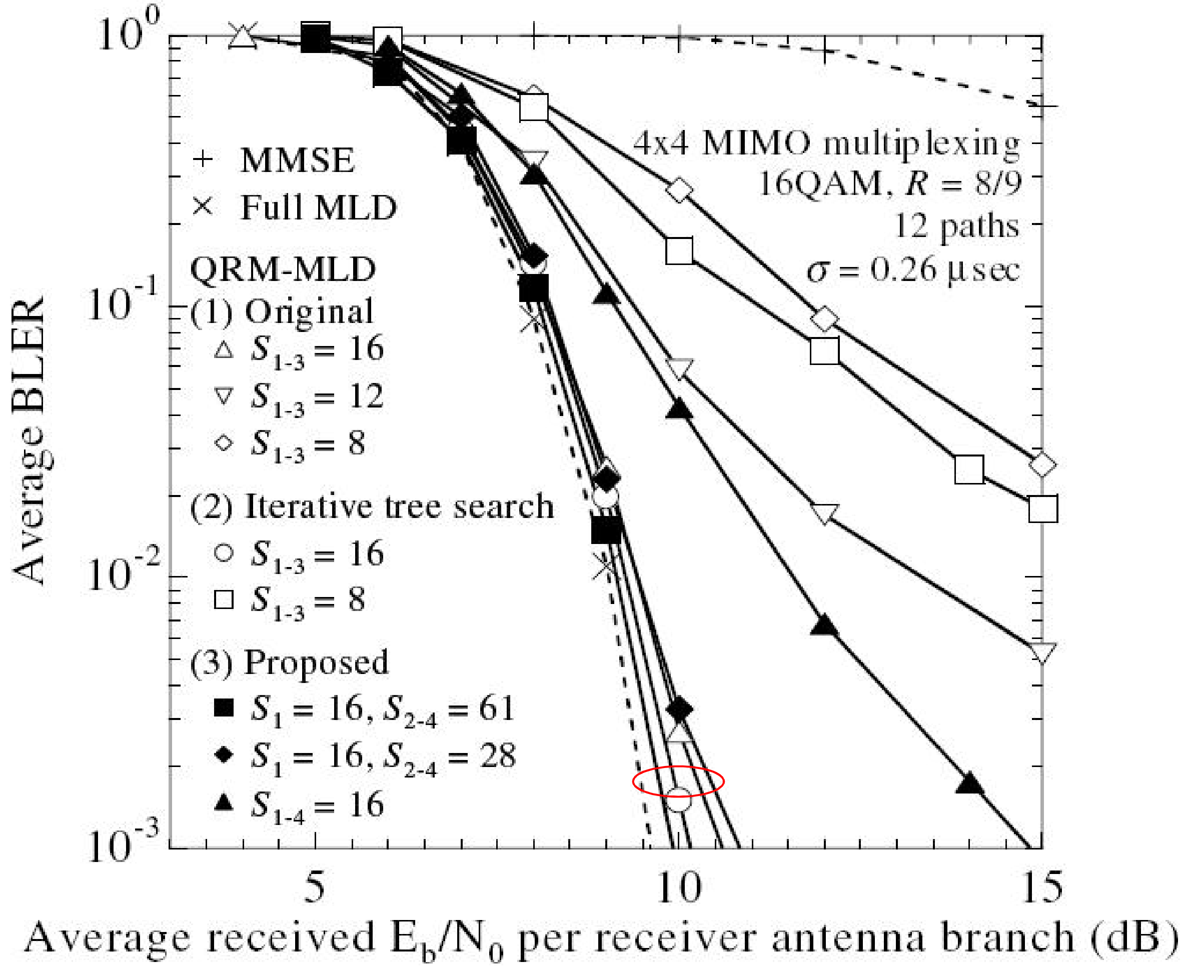
- A modified QRM-MLD with lower complexity proposed by NTT DoCoMo
- Reduce the number of multiplications by replacing the Euclidian distance calculation with quadrant detection for selecting candidates
- After candidates are selected, the real Euclidian distance is calculated for the selected candidates only

QRM-MLD with ASESS



(a) 1st quadrant detection (b) 2nd quadrant detection (c) 3rd quadrant detection (d) Symbol ranking

Performance



4x4 16QAM 8/9 Turbo code

Complexity

NUMBER OF REAL MULTIPLICATIONS PER FRAME REQUIRED FOR SIGNAL SEPARATION

Signal separation method	Operation	Required number of multiplications per frame	Example of total required number for 1-Gbps data rate ($N_{ant} = 4$, $C = 16$ (16QAM), $N_{sub} = 768$, $N_d = 48$)
Full MLD	Generation of symbol replica candidates	$4\, N_{ant}^{\;2}\, C\, N_{sub}$	➢1.9 x 10^{+10}
	Calculation of squared Euclidian distances	$2\, N_{ant}\, C^{N_{ant}}\, N_{sub}\, N_d$	
Original QRM-MLD	QR decomposition of channel matrix **H**	$4\, N_{ant}^{\;3}\, N_{sub}$	➢6.1 x 10^{+7} when $S_{1\text{-}3} = 16$
	Multiplication of $\mathbf{Q}^H$ to received signal vector	$4\, N_{ant}^{\;2}\, N_{sub}\, N_d$	➢4.7 x 10^{+7} when $S_{1\text{-}3} = 12$
	Generation of symbol replica candidates	$4\, (N_{ant}\, (N_{ant} + 1) / 2)\, C\, N_{sub}$	
	Calculation of squared Euclidian distances	$2\,(1+\sum_{m=1}^{N_{ant}-1} S_m)\, C\, N_{sub}\, N_d$	➢3.3 x 10^{+7} when $S_{1\text{-}3} = 8$
Proposed QRM-MLD	QR decomposition of channel matrix **H**	$4\, N_{ant}^{\;3}\, N_{sub}$	➢1.8 x 10^{+7} when $S_1 = 16$, $S_{2\text{-}4} = 61$
	Multiplication of $\mathbf{Q}^H$ to received signal vector	$4\, N_{ant}^{\;2}\, N_{sub}\, N_d$	➢1.0 x 10^{+7} when $S_1 = 16$, $S_{2\text{-}4} = 28$
	Generation of symbol replica candidates	$4\, (N_{ant}\, (N_{ant} + 1) / 2)\, C\, N_{sub}$	➢7.8 x 10^{+6} when $S_{1\text{-}4} = 16$
	Calculation of squared Euclidian distances	$2\,(\sum_{m=1}^{N_{ant}} S_m)\, N_{sub}\, N_d$	
MMSE	MMSE weight generation	$4\,(2\, N_{ant}^{\;3} + N_{ant}\,(N_{ant} + 1)^2)\, N_{sub}$	➢7.8 x 10^{+6}
	Signal separation using MMSE weight	$4\, N_{ant}^{\;2}\, N_{sub}\, N_d$	
	Calculation of squared Euclidian distances	$2\, N_{ant}\, C\, N_{sub}\, N_d$	

From: "Adaptive selection of surviving symbol replica candidates based on maximum reliability in QRM-MLD for OFCDM MIMO multiplexing" Globecom04

Other tree based detectors

- Following the same framework
  - Modifications of the algorithm to marginally reduce the complexity but requiring additional operations or the calculation of limiting thresholds
  - Simplifications of the algorithm for specific constellation types
  - Adapting dynamically select M
  - A combination of the sequential SD and the QRM-MLD
  - Utilize the information of discarded paths
  - ……

MCMC

- Markov chain Monte Carlo (MCMC) detector [1]
  - Proposed by wireless communication lab of University of Utah
- The idea is to find those terms with significant contributions (important vectors) efficiently
  - Similar to sphere decoding, but completely different approach
- MCMC detector finds important vectors using Markov chain Monte Carlo
  - Create Markov chain with state space: **d** and stationary distribution P(**d**|**Y**)
  - Run Markov chain using Gibbs sampler
  - After Markov chain converge, the samples are generated according to P(**d**|**Y**). Those samples with large P(**d**|**Y**) generated with high probabilities

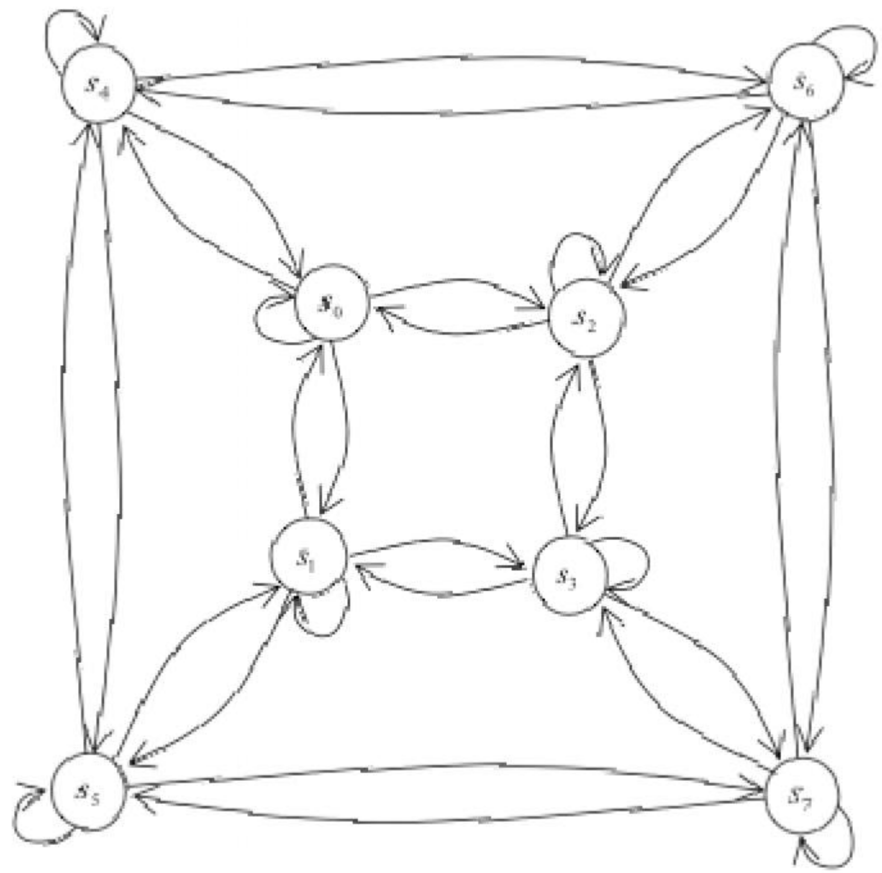
[1] B. Farhang-Boroujeny, H. Zhu, and Z. Shi, "Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems," IEEE Trans. Signal Processing

Gibbs sampler

- Run full Markov chain is impossible because of huge number of states

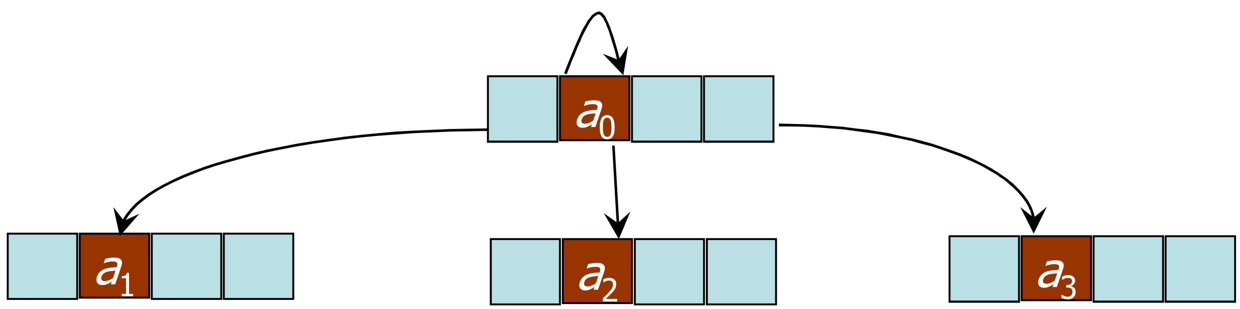
$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

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

- Gibbs sampler limit the states jumping with only one variable change for each state jumping



Gibbs sampler

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

for $\mathrm{n} = 1$ to I

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

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

M

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

end for

where $p(d_i = b \mid d_0^{(n)}, \Lambda\ d_{i-1}^{(n)}, d_{i+1}^{(n-1)}, \Lambda\ , d_{NM_c-1}^{(n-1)}, \mathbf{y})$

$$\propto p(\mathbf{y} \mid d_0^{(n)}, \Lambda\ d_{i-1}^{(n)}, d_i = b, d_{i+1}^{(n-1)}, \Lambda\ , d_{NM_c-1}^{(n-1)},) p(d_i = b)$$

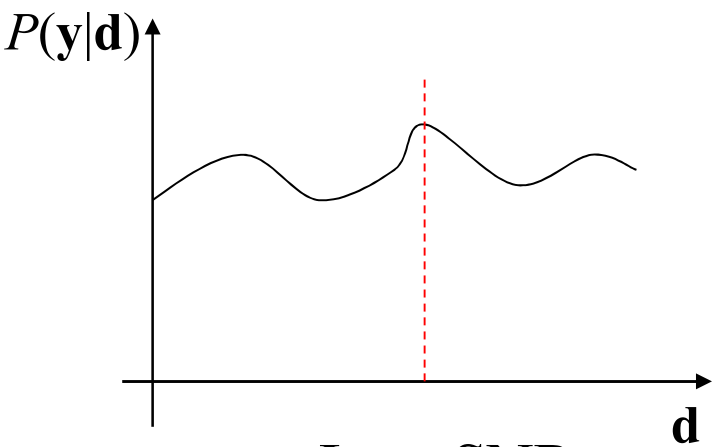
M_c is the # of bits per constellation symbol

MCMC

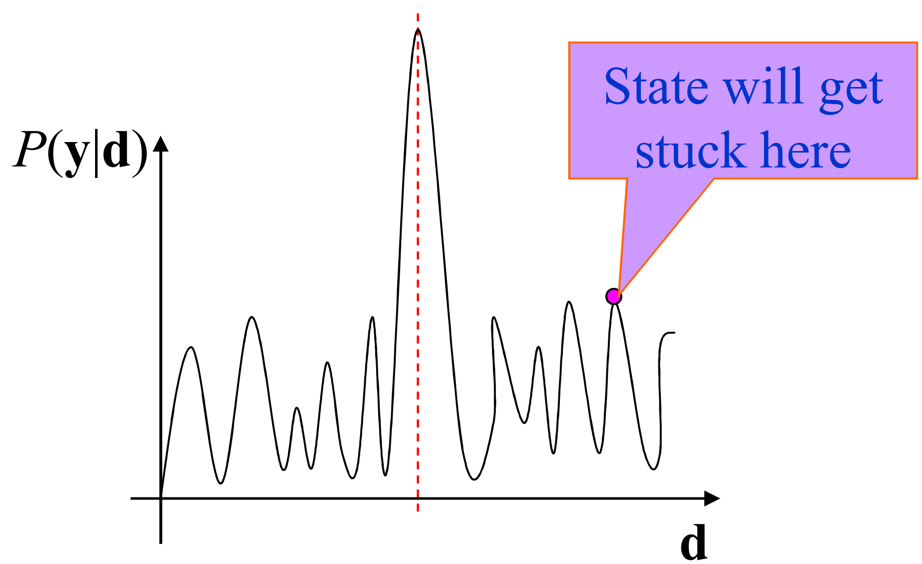
- The samples generated by Gibbs sampler are used to compute the soft message for soft decoder
- To accelerate Markov train converge, *L* independent parallel Gibbs samplers are runned and each Gibbs sampler run *I* iterations
- Complexity : $2LI\log(M_c)N_tN_r$ per subcarrier per symbol

High SNR problem

- At high SNR, MCMC takes long time to converge, leads performance degradation, this is because of the multimodal property of channel PDF at high SNR



Low SNR

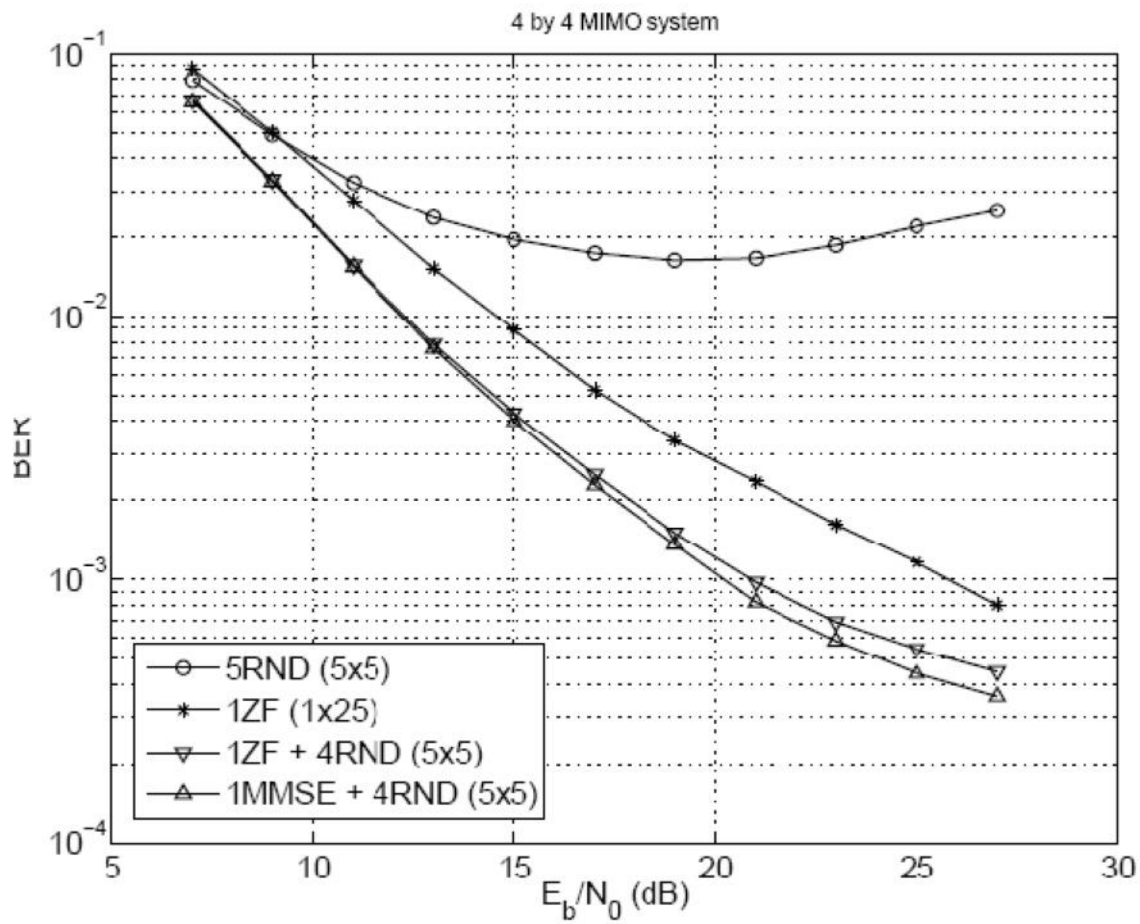


High SNR

Solutions

- Multimodal problem exists in many MCMC algorithm.
- No general method to overcome it
- For MIMO detection, one solution is to generate initial candidates using other low complexity detector (warm start)
- Forced state jump
- Joint detection and decoding
- All these solutions are experimental, no theoretical results available yet
- Still need more study to improve

Performance



4x4 16QAM with ½ convolutional code

Joint detection and decoding

- JDD system achieve better performance than SDD system with higher complexity (More than 2dB)
- MAP detection

$$\lambda_k = \ln \frac{P(b_k = +1 \mid \mathbf{y})}{P(b_k = -1 \mid \mathbf{y})} = \ln \frac{\sum_{\mathbf{b}_{-k}} P(b_k = +1 \mid \mathbf{y}, \mathbf{b}_{-k}) P(\mathbf{b}_{k+} \mid \mathbf{y})}{\sum_{\mathbf{b}_{-k}} P(b_k = -1 \mid \mathbf{y}, \mathbf{b}_{-k}) P(\mathbf{b}_{k-} \mid \mathbf{y})}$$

where $\mathbf{b}_{-k} = [b_1 \quad b_2 \quad \cdots \quad b_{k-1}\, b_{k+1} \quad \cdots \quad b_N], \mathbf{b}_{k+} = [b_1 \quad b_2 \quad \cdots \quad b_{k-1}\ 1\ b_{k+1} \quad \cdots \quad b_N]$

and $\mathbf{b}_{k-} = [b_1 \quad b_2 \quad \cdots \quad b_{k-1}\ -1\ b_{k+1} \quad \cdots \quad b_N]$

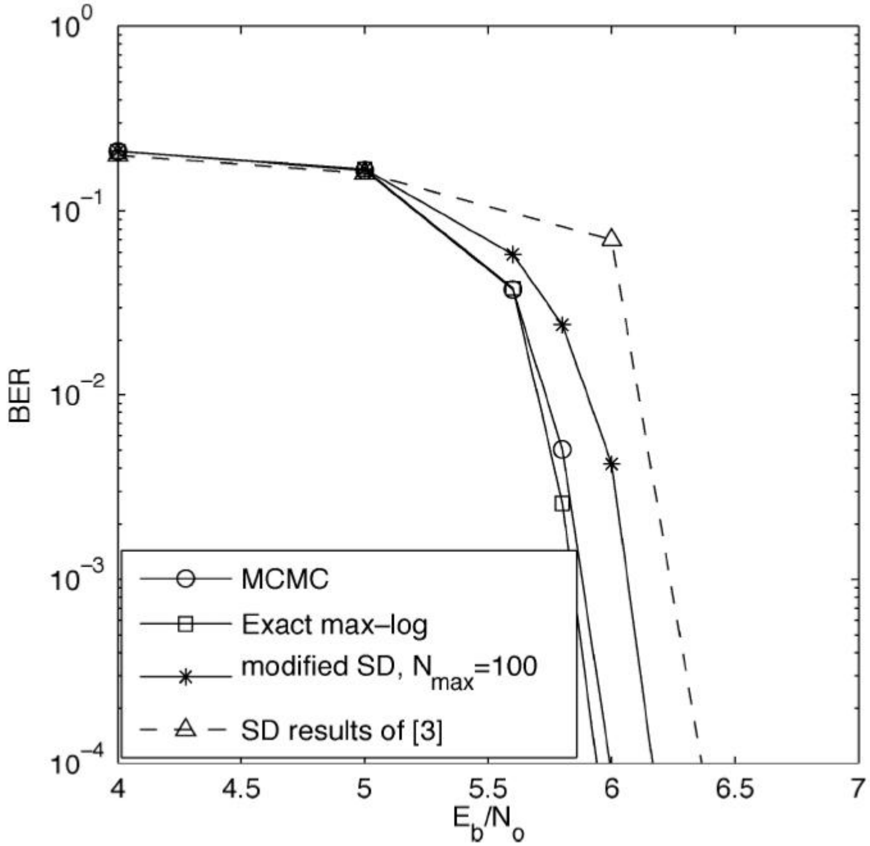
Problem: the number of combinations that $\mathbf{b}_{-k}$ takes is 2^{N-1}!

Important observation: Most of the terms in the numerator and denominator are insignificant. We may just need to sum those terms with significant contributions

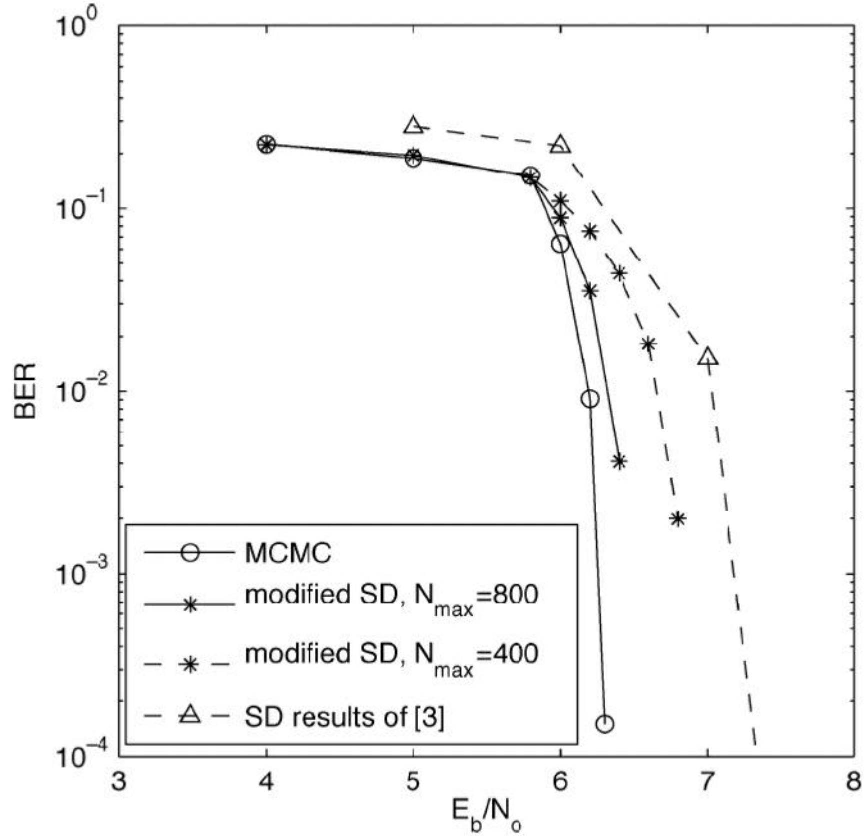
Low complexity JDD

- Most of previous detector can be applied to JDD system with minor modification
- List sphere decoding and MCMC can achieve near optimal MAP performance
- JDD with strong channel codes usually can achieve near capacity performance but at the cost of higher delay, lower throughput and higher complexity

Sphere decoding VS. MCMC (JDD)



4x4 16QAM ½ turbo code [2]



8x8 16QAM ½ turbo code [2]

[2] H. Zhu, B. Farhang-Boroujeny, B. and R-R. Chen, "On performance of sphere decoding and Markov chain Monte Carlo detection methods,"

Sphere decoding VS. MCMC (JDD)

TABLE I
COMPLEXITY STUDY OF SD AND MCMC DETECTORS

	SNR (dB)	Channel size	L, N_s or N_{max}, N_{min}	No. of Operations
MCMC	5.6	4×4	10, 10	1.32×10^8
SD	5.6	4×4	100, 10	1.16×10^8
MCMC	6.5	8×8	20, 20	9.72×10^8
SD	6.5	8×8	400, 10	1.40×10^{10}
SD	6.5	8×8	800, 10	1.69×10^{10}

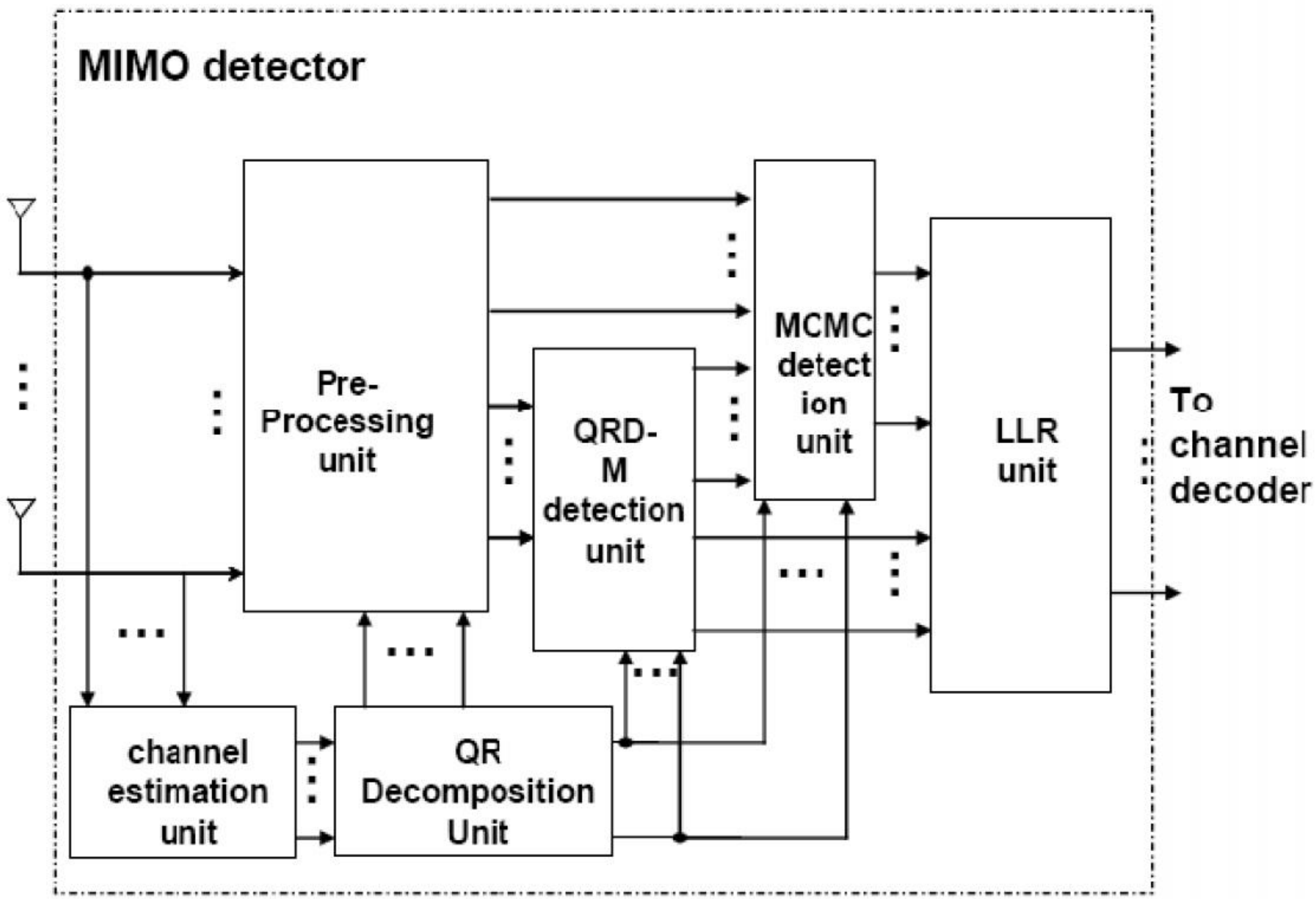
Sphere decoding VS. MCMC (JDD)

- Sphere decoding finds important vectors by limiting the search to sphere.
  - Exponential complexity at low SNR
  - Complexity is increased quickly with the dimension of problem
- MCMC finds important vectors using the P(**d**|**Y**)
  - Works very well at low SNR
  - Complexity is independent of SNR
  - Complexity is increase not too much with the dimension of problem
  - At high SNR, need the help of ZF or MMSE

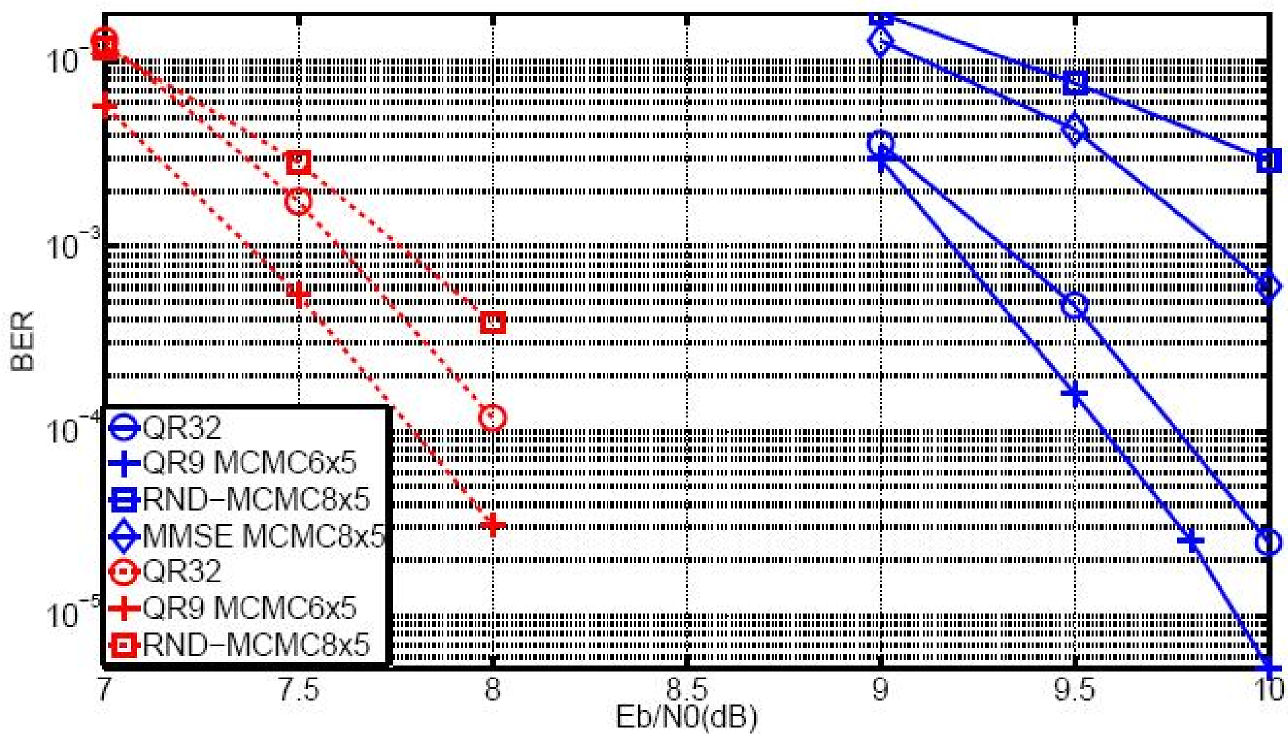
Some thoughts

- Combine sphere decoding and MCMC
  - Combine QRM-MLD and MCMC
    - Use QRM-MLD with a small M to generate initial candidates
    - Not necessary increase the complexity of MCMC because upper-triangular matrix can reduce half of Euclidian distance calculation which is most computation extensive part

Hybrid QRD-MCMC

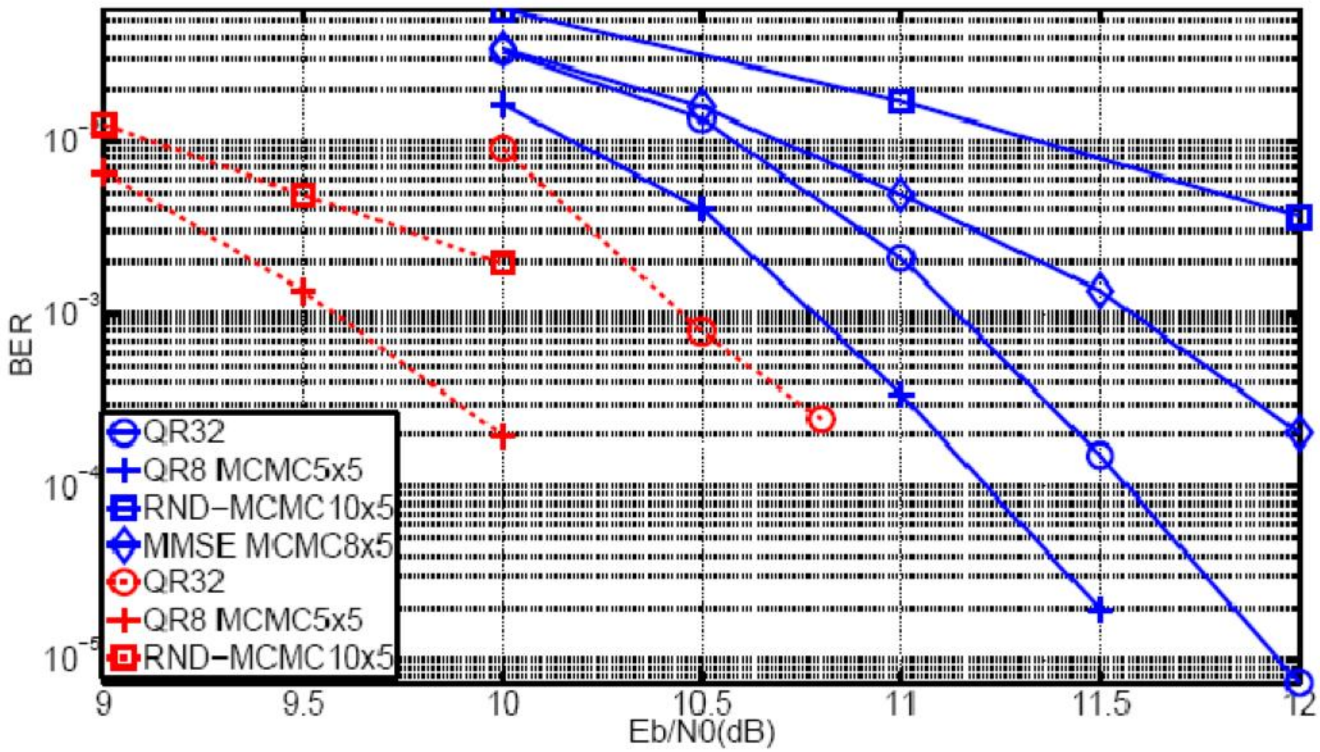


Results



Performance comparison of 4x4 16QAM SDD and JDD systems

Results



Performance comparison of 8x8 16QAM SDD and JDD systems

Complexity

QRD-M	4x4 M=32	50256
	8x8 M=32	127344
RND-MCMC	4x4 $D = 8$ $L = 5$	73760
	8x8 $D = 10$ $L = 5$	331920
QRD-MCMC	4x4 $M = 9$ $D = 6$ $L = 5$	41498
	8x8 $M = 8$ $D = 5$ $L = 5$	114284

Imperfect channel estimation

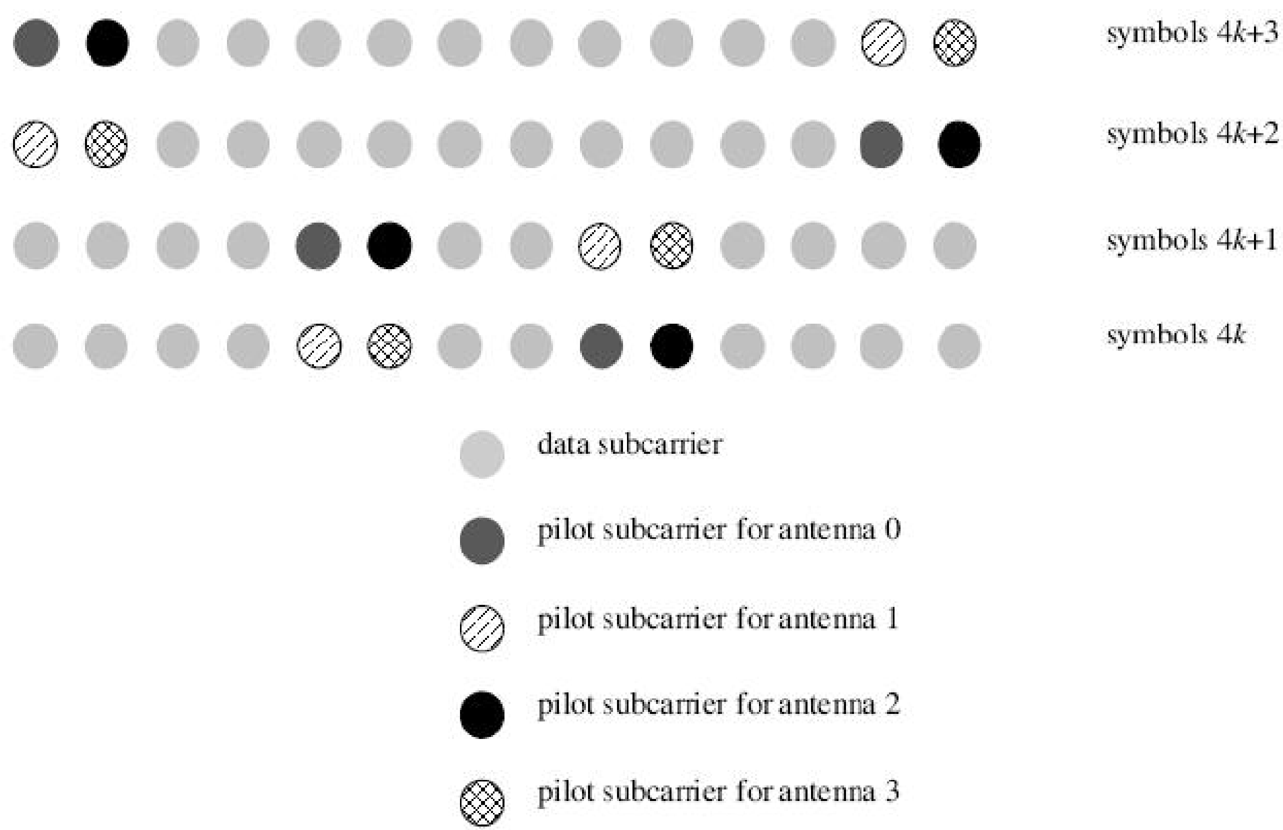
- In practice, channel is unknown, the accuracy of channel estimation influence performance of the detection greatly
  - Channel estimation for MIMO-OFDM is more challenging
    - Doppler frequency caused time selective
    - Multipath caused frequency selective
    - Need to track the variance at 2-D
    - For high speed UE, the channel can change during even one symbol which will cause ICI
  - Optimal pilot structure for MIMO-OFDM need to be studied
  - Low complexity channel estimation algorithm should be studied
  - Joint channel estimation and detection
    - Combined with iterative coding can achievement very good performance but with higher delay, lower throughput and higher complexity

MIMO-OFDMA system in 802.16e

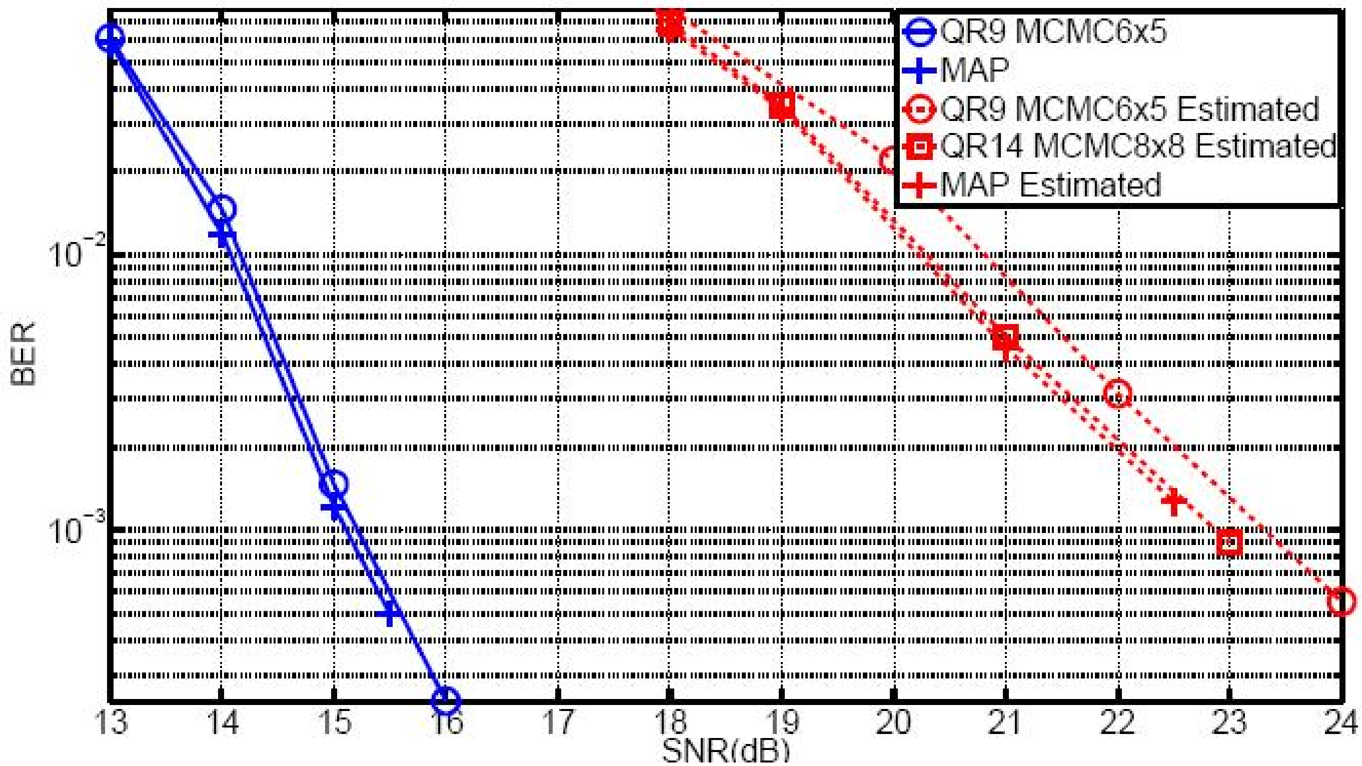
SYSTEM PARAMETERS

Parameter	Value
Channel bandwidth	10 MHz
Number of subcarriers	1024
Subcarrier permutation	PUSC
Cyclic prefix	1/8
Channel coding	Convolutional turbo codes
Carrier frequency	2500 MHz
Sampling frequency	11.2 MHz
Multipath channel	ITU VehA
MS speed	120 km/hr

Pilot structure



Results



Performance comparison of 4x4 16QAM MIMO-OFDMA system using R = 1/2 IEEE 802.16e convolutional turbo codes with perfect and 2D MMSE channel estimation.

Imperfect channel estimation

- In practice, channel is unknown, the accuracy of channel estimation influence performance of the detection greatly
  - Channel estimation for MIMO-OFDM is more challenging
    - Doppler frequency caused time selective
    - Multipath caused frequency selective
    - Need to track the variance at 2-D
    - For high speed UE, the channel can change during even one symbol which will cause ICI
  - Optimal pilot structure for MIMO-OFDM need to be studied
  - Low complexity channel estimation algorithm should be studied
  - Joint channel estimation and detection
    - Combined with iterative coding can achievement very good performance but with higher delay, lower throughput and higher complexity

Interference

- Internal interference
  - ICI caused by frequency offset
    - Loss of synchronization
    - Phase noise if the oscillators
    - Relatively less challenging since frequency offset parameter is constant over all the subcarriers and can be tracked iteratively
  - ICI caused by fast fading channel
    - Channel change too fast and is not constant over one OFDM symbol
    - Need more complexity algorithm to compensate
- External interference
  - Impulse noise
  - Narrowband interference can be modeled as Gaussian noise
  - Synchronous and asynchronous interferences

Challenges

- Good detector should achieve good performance and complexity tradeoff
- Imperfect channel estimation and/or synchronization may have bad effect on the performance of detector
- More simulations should be performed under more practical channel models

Thank you !