



National Chiao Tung University
Institute of Communication Engineering

Network Technology Laboratory

Design of Low Rate Coding Schemes for Ultra-Reliable Communications

Speaker: Chung-Wei You

Advisor: Po-Ning Chen



Outline

Network Technology Laboratory

- Introduction
- Previous Work and System Model
- Concatenation of Convolutional Code with Single Parity Check Code (CC-SPC)
 - ◆ CC-SPC Encoding Scheme
 - ◆ Decoding of CC-SPC
- Rate-1/9 CC-SPC Coding
- Conclusion and Future Work



Motivation

Network Technology Laboratory

- ◆ For vehicular communication, the transceiving of safety information must be highly reliable. It is thus referred to as the Ultra-Reliable Communications type.
- ◆ The traditional channel coding schemes such as Turbo codes and LDPC codes suffer an error floor effect at high SNR.
- ◆ Low-rate convolutional-coding-based repetition coding scheme is a straightforward design for ultra-reliable communication.
- ◆ Our goal (and contribution) in this thesis is to propose a low-complexity coding scheme that performs better than the low-rate convolutional-coding-based repetition coding scheme.



HARQ with Soft Combining

Network Technology Laboratory

- ◆ Chase Combining
 - ◆ If CRC fails, the transmitter retransmits the same coded bits.
 - ◆ The receiver saves the previous transmission and combines it with the newly received retransmission.

- ◆ Incremental Redundancy (IR)
 - ◆ The IR scheme divides the coded bits into multiple groups. If CRC fails, the transmitter retransmits another set of coded bits.
 - ◆ The receiver combines these coded sets and perform a new decoding round.

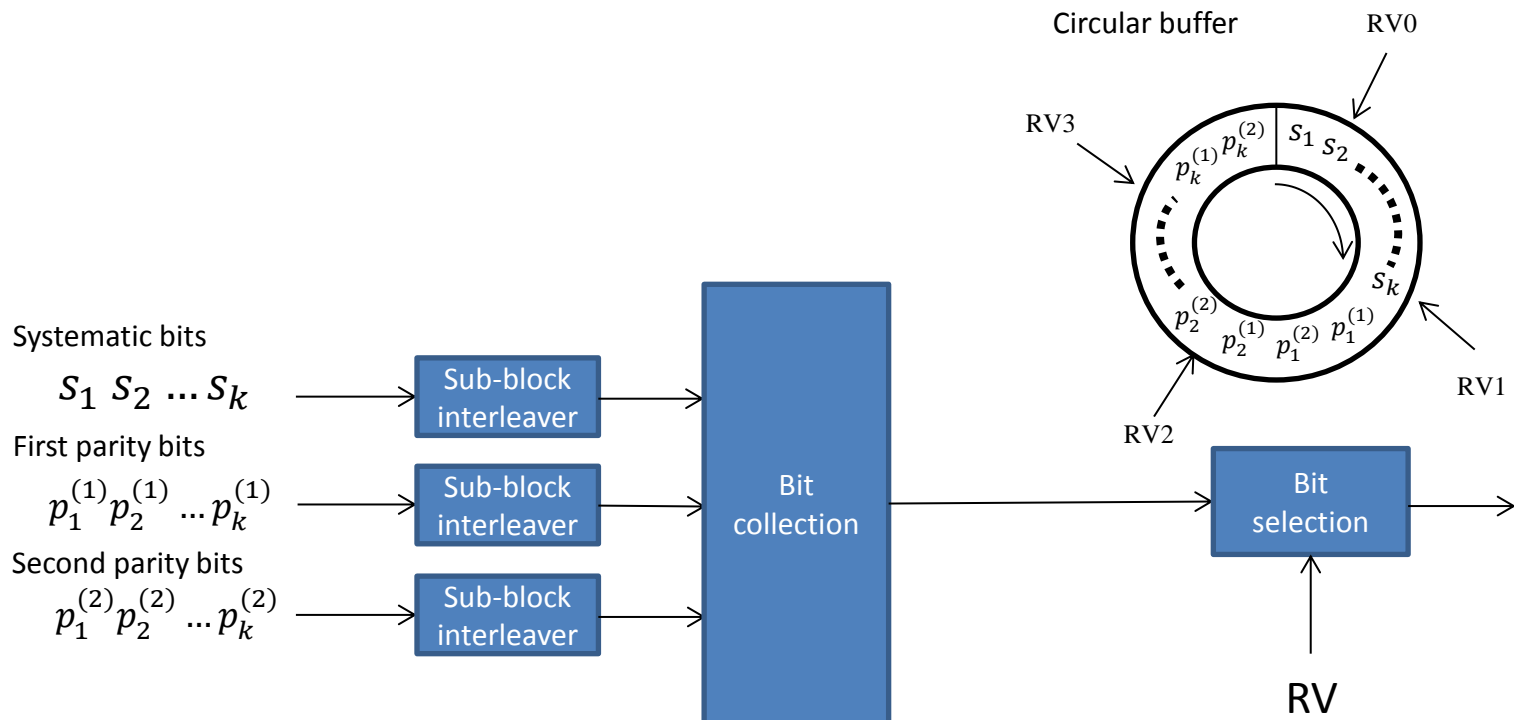


HARQ with Soft Combining

Network Technology Laboratory

◆ Rate matching in LTE

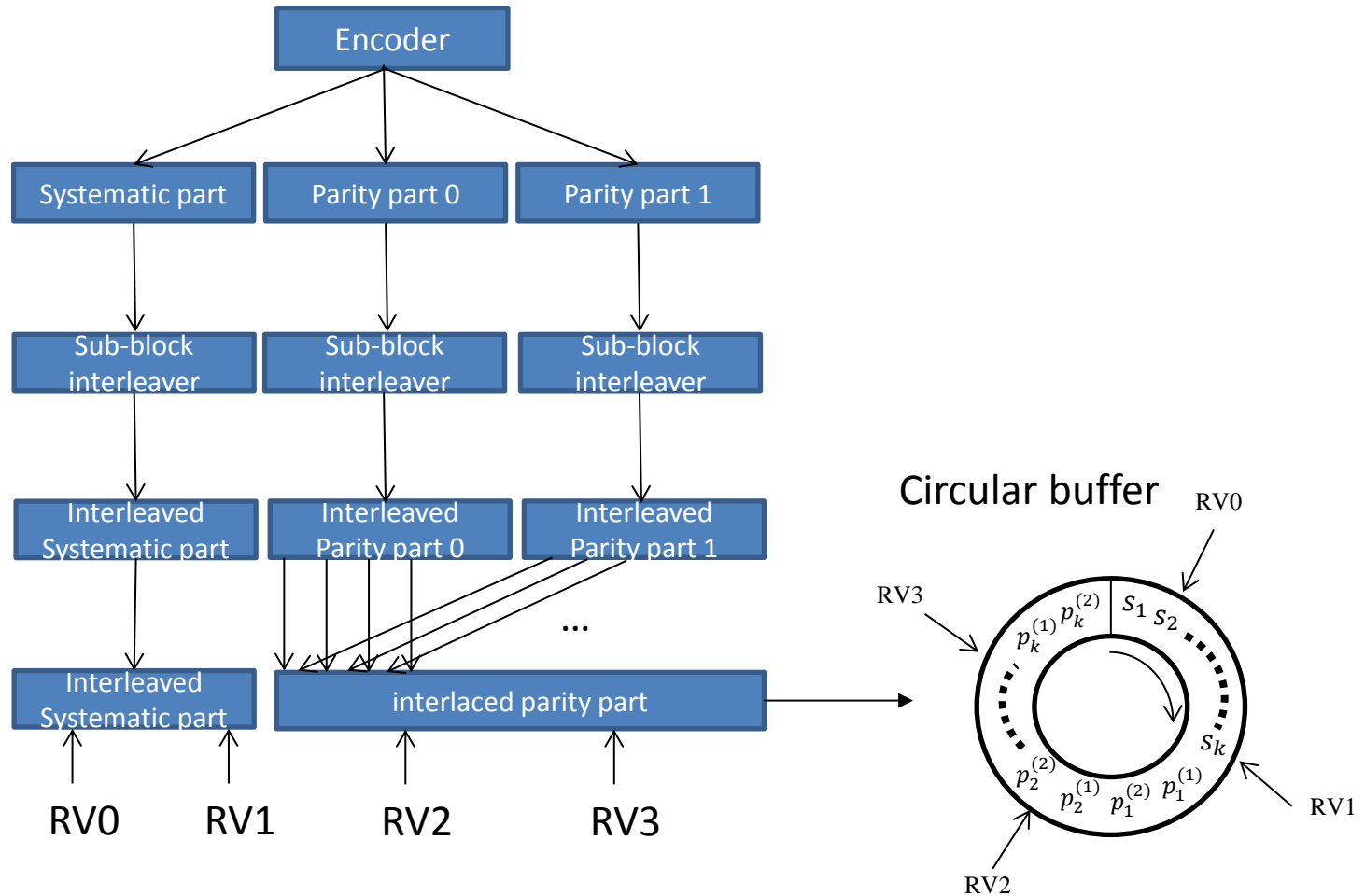
- ◆ Three coded streams are fed into three sub-block interleavers, respectively.
- ◆ The interleaved bits are selective pushed into different redundancy versions (RVs) and then inserted into the circular buffer.
- ◆ The transmitted bits are selected from the circular buffer.





HARQ with Soft Combining

◆ Turbo-coding rate matching in LTE

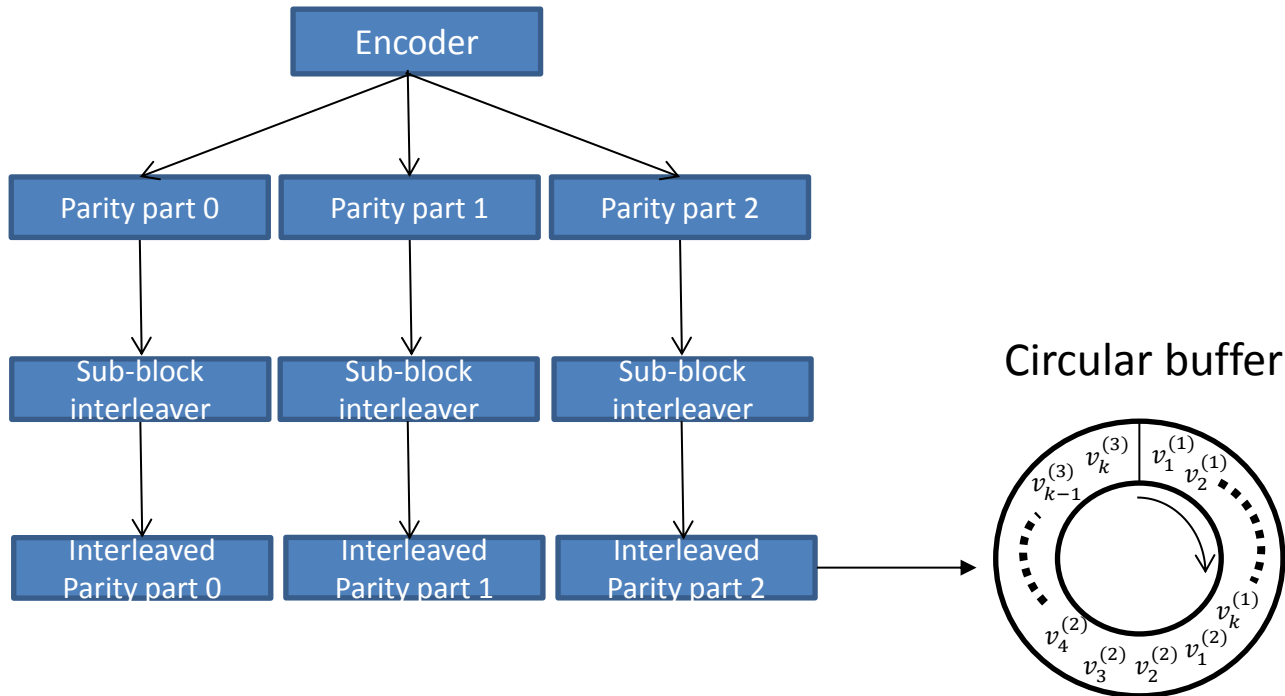




HARQ with Soft Combining

Network Technology Laboratory

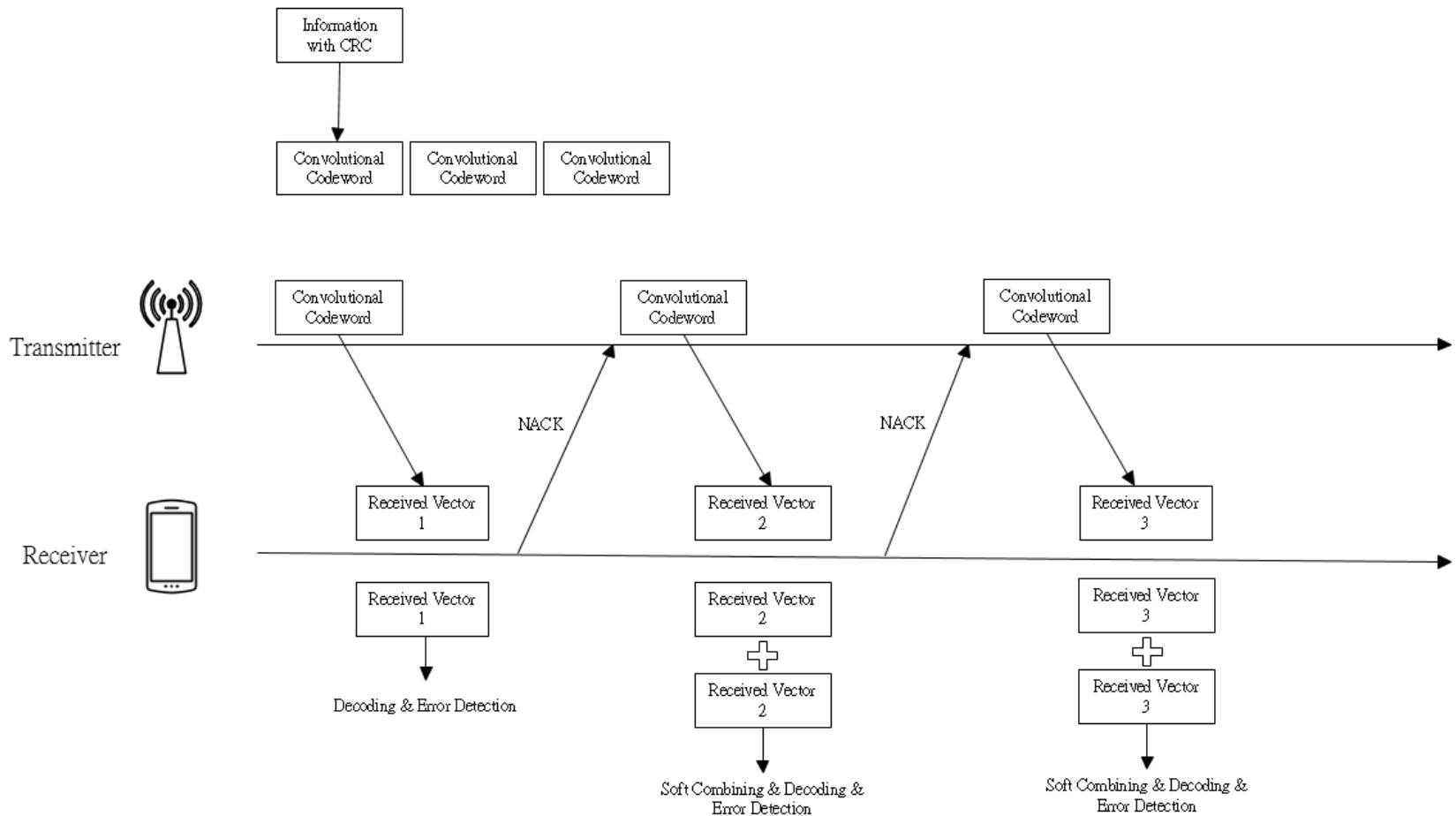
◆ Convolutional-code rate matching in LTE





System Model

◆ HARQ with soft combining for convolutional codes

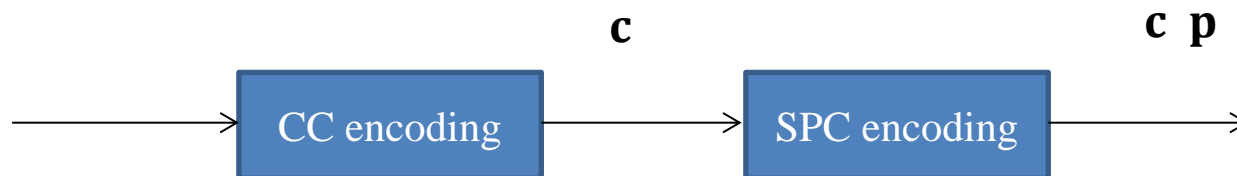




CC-SPC Encoding Scheme

Network Technology Laboratory

- ◆ If CRC of the decoded “initial transmission” fails, the transmitter transmits the parity sequence as the first retransmission. This is different from the legacy retransmission mechanism that retransmits again the “initial transmission.”
- ◆ We propose to SPC-encode the CC codeword to form the parity sequence for retransmission, where SPC code = Single Parity Check code.

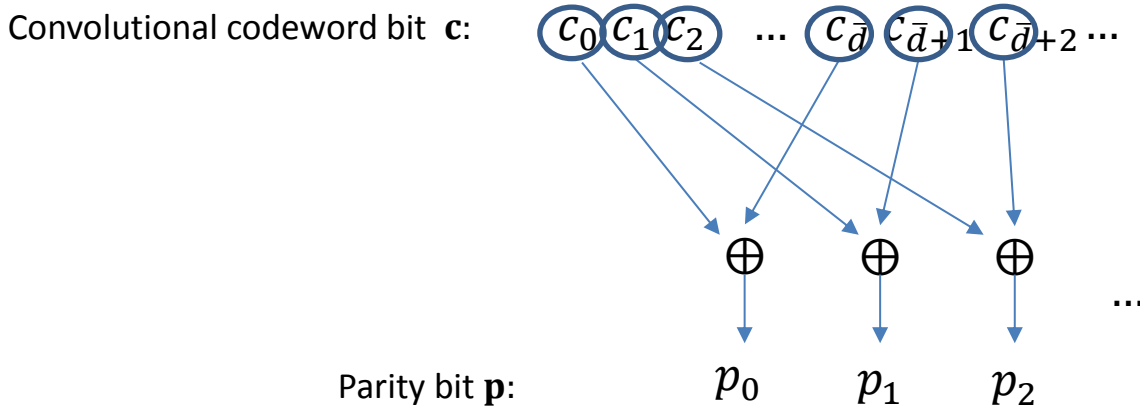




CC-SPC Encoding Scheme

Network Technology Laboratory

◆ Single parity check (SPC) encoding



➤ $p_i = c_{N+i} = c_{i \bmod N} \oplus c_{(i+\bar{d}) \bmod N}$

- N : CC Codeword Length



Matrix Expression of CC-SPC

Network Technology Laboratory

$$\mathbf{p}^T = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 & \dots & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{bmatrix} = \mathbf{A}_{N \times N} \mathbf{c}^T$$

- Matrices used by the SPC under code rate $R = 1/6$

- $\mathbf{A} = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 & \dots & \dots & 0 & 1 \end{bmatrix}_{N \times N}$

- $\mathbf{H}_{\text{SPC } N \times 2N} = [\mathbf{A}_{N \times N} \mid \mathbf{I}_{N \times N}]$

- $\mathbf{G}_{\text{SPC}} = [\mathbf{I}_{N \times N} \mid \mathbf{A}_{N \times N}^T]$

◆ N: CC Codeword Length



Illustration of SPC encoding

Network Technology Laboratory

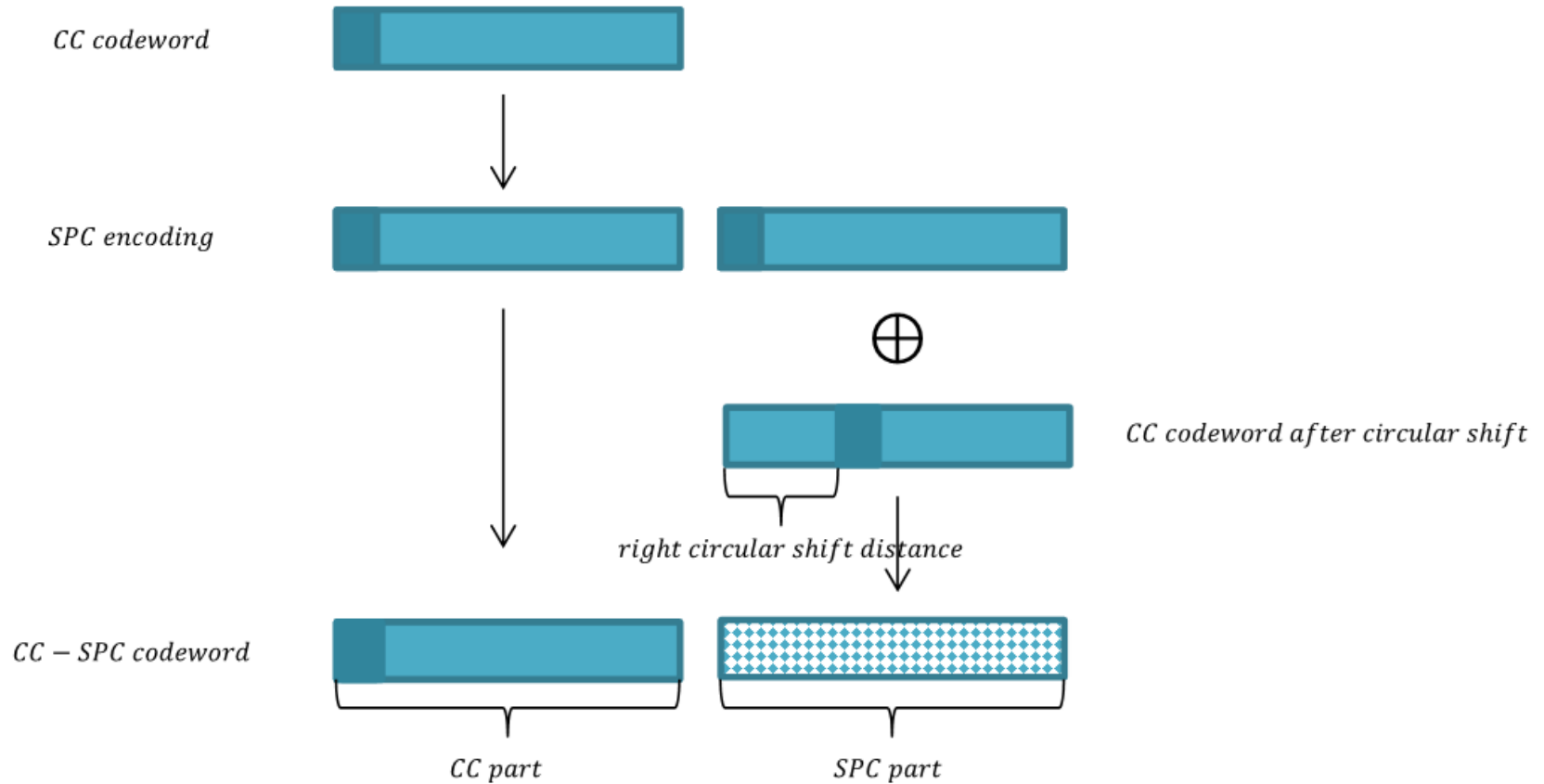


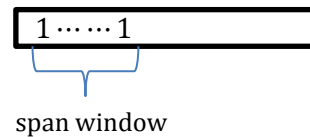


Illustration of d_{min} Tripling

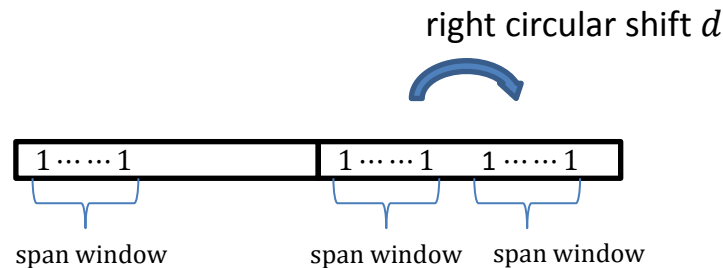
Network Technology Laboratory

- ◆ Consider one of the convolutional codewords with minimum weight

- ◆ CC encoding:



- ◆ SPC encoding:



- No overlapping in 1's positions between the CC codeword with the minimum weight and its right circular shift counterpart.
- Idea Goal: $d_{min}^{(CC-SPC)} = 3 \times d_{min}^{(CC)}$



Illustration of d_{min} Tripling

Network Technology Laboratory

- $\mathbf{G}_{CC}(D) = \left[1, \frac{1+D+D^2+D^3+D^6}{1+D^2+D^3+D^5+D^6}, \frac{1+D+D^2+D^4+D^6}{1+D^2+D^3+D^5+D^6} \right]$

Example of a CC-SPC code	
Information length	24
Convolutional codeword length	$24 \times 3 + 18 = 90$
d_{min} of conv codeword	15
Max size of span window for d_{min} CW	$w = 33$
Suggestive numbers of circular shifts	$w \sim (N - w) = 33 \sim 57$
target d_{min} of CC-SPC	45



d_{min} Results

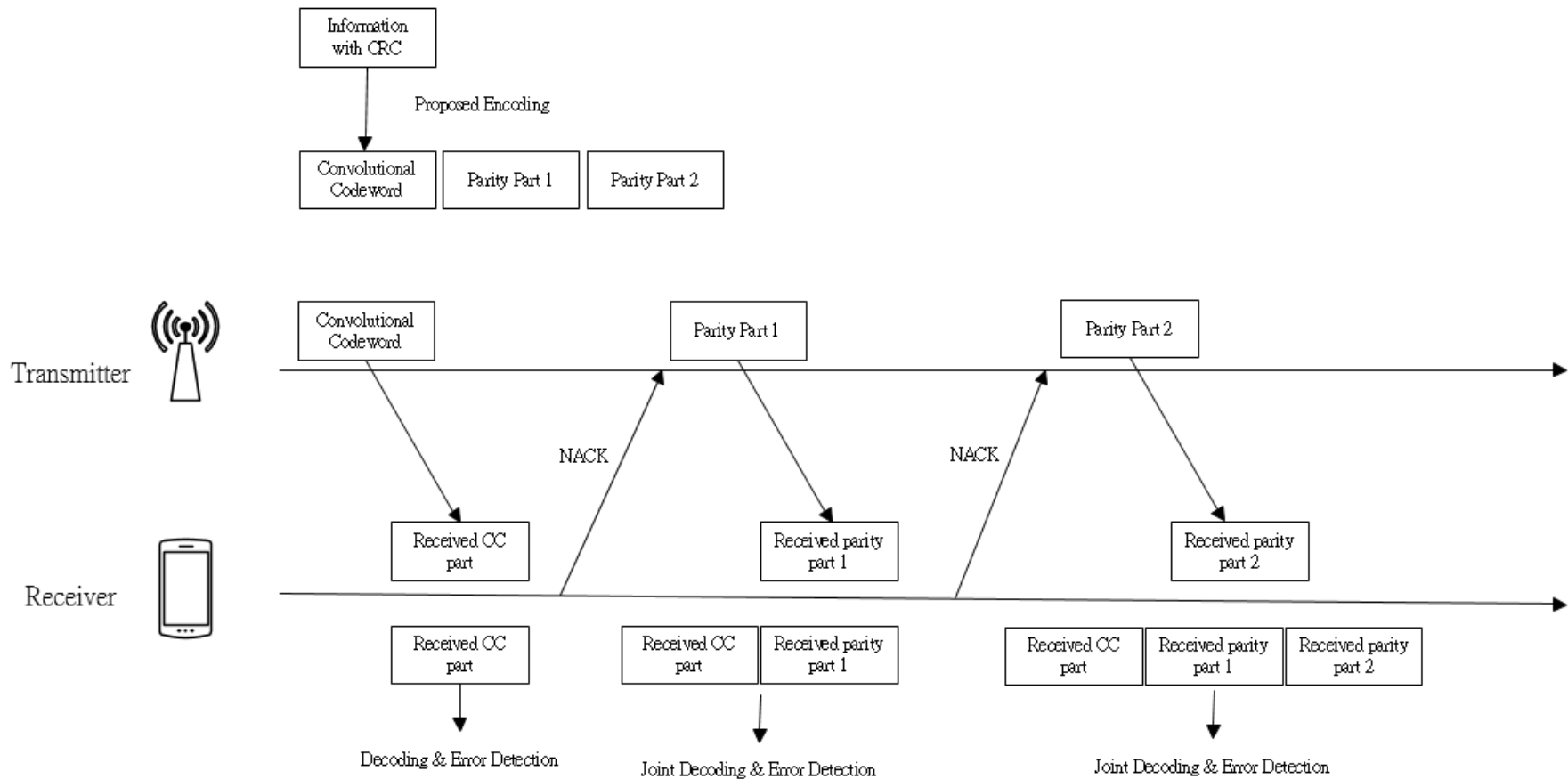
- From the table below, we note that most of the number of right cyclic shifts can triple the minimum pairwise Hamming distance of CC.

circular shifts	d_{min}	circular shifts	d_{min}	circular shifts	d_{min}	circular shifts	d_{min}	circular shifts	d_{min}	circular shifts	d_{min}
1	23	17	37	33	42	49	45	65	39	81	33
2	28	18	39	34	43	50	45	66	39	82	33
3	31	19	38	35	43	51	45	67	37	83	31
4	27	20	39	36	45	52	45	68	37	84	31
5	27	21	39	37	42	53	42	69	39	85	27
6	31	22	37	38	45	54	45	70	39	86	27
7	31	23	37	39	45	55	43	71	38	87	31
8	33	24	39	40	45	56	43	72	39	88	28
9	33	25	39	41	45	57	42	73	37	89	23
10	33	26	39	42	45	58	42	74	33		
11	34	27	40	43	43	59	41	75	33		
12	35	28	41	44	42	60	39	76	35		
13	35	29	39	45	30	61	39	77	35		
14	35	30	39	46	42	62	41	78	35		
15	33	31	41	47	43	63	40	79	34		
16	33	32	42	48	45	64	39	80	33		



Proposed Scheme

◆ Proposed scheme for CC-SPC HARQ

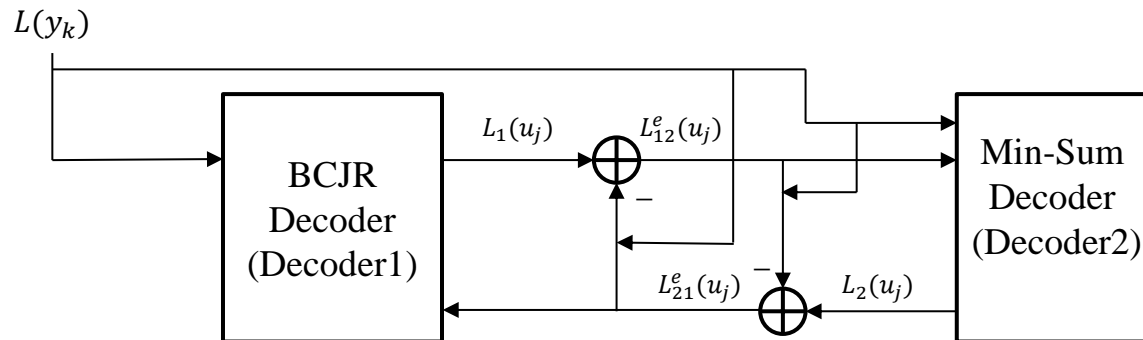




Iterative Decoding for CC-SPC

Network Technology Laboratory

◆ Iterative decoding



➤ Denote

- $N_{\text{CC-SPC}}$ is the codeword length of CC-SPC
- N is the codeword length of CC
- $\forall 0 < k < N_{\text{CC-SPC}} - 1$
- $\forall 0 < j < N - 1$

- $L(y_k)$: LLRs of received signals
- $L_1(u_j)$: a posteriori LLRs from decoder 1
- $L_2(u_j)$: a posteriori LLRs from decoder 2
- $L_{12}^e(u_j)$: a priori LLRs from decoder 1 to decoder 2
- $L_{21}^e(u_j)$: a priori LLRs from decoder 2 to decoder 1



Convolutional Codes for Simulations

Network Technology Laboratory

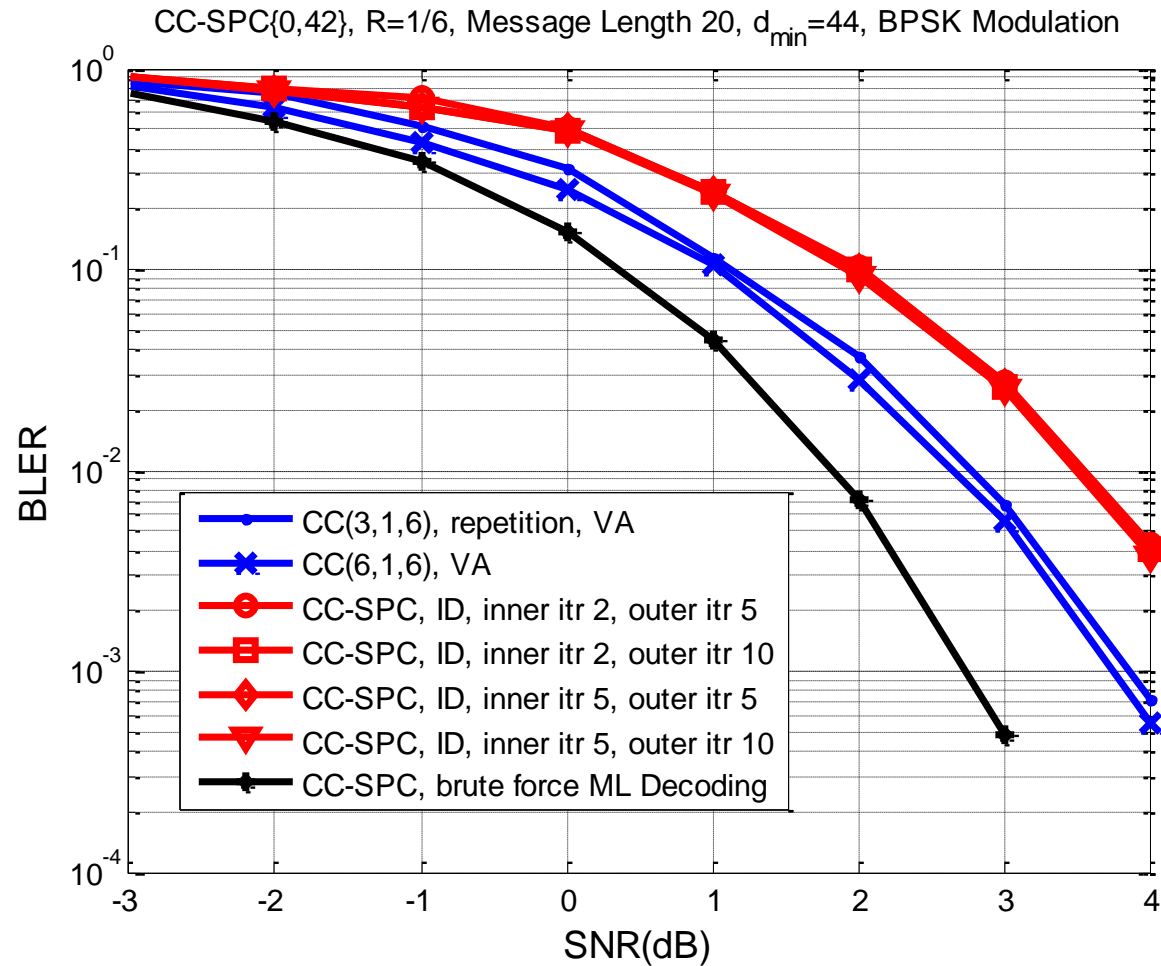
Convolutional codes for simulations	
	Transfer function $G(D)$
ZTCC(3,1,6)	$\left[1, \frac{1 + D + D^2 + D^3 + D^6}{1 + D^2 + D^3 + D^5 + D^6}, \frac{1 + D + D^2 + D^4 + D^6}{1 + D^2 + D^3 + D^5 + D^6}\right]$
ZTCC(6,1,6)	$[1 + D + D^2 + D^3 + D^5 + D^6, 1 + D + D^3 + D^6, 1 + D^2 + D^3 + D^4 + D^6, 1 + D^2 + D^3 + D^4 + D^6, 1 + D + D^2 + D^5 + D^6, 1 + D^2 + D^3 + D^4 + D^5 + D^6]$



Simulation Results $R=1/6$

Iterative Decoding

Network Technology Laboratory



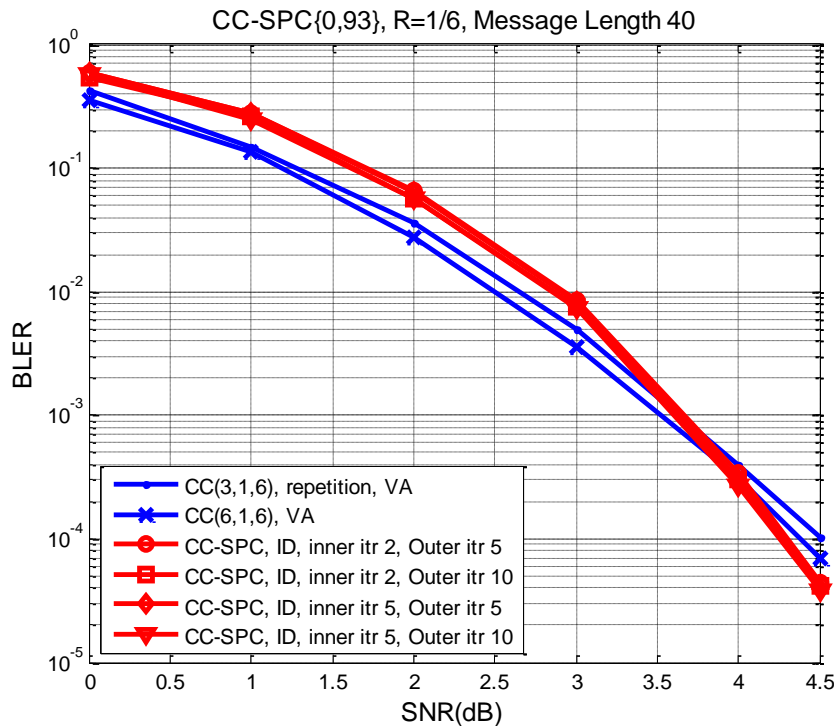
of right circular shifts = $(20+6)*3 - 42 = 36$



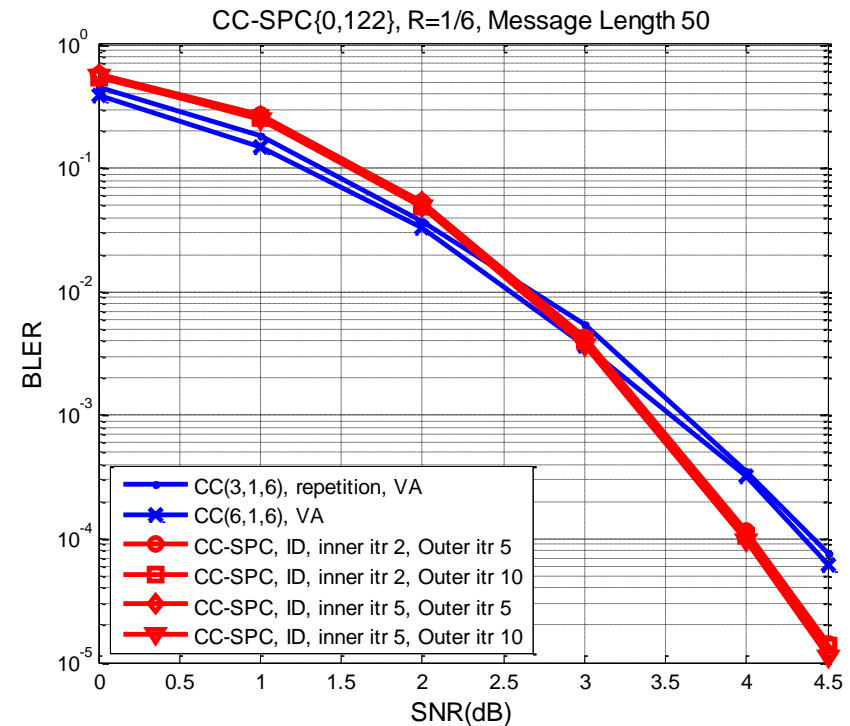
Simulation Results $R=1/6$

Iterative Decoding

Network Technology Laboratory



of right circular shifts = $(40+6)*3 - 93 = 45$



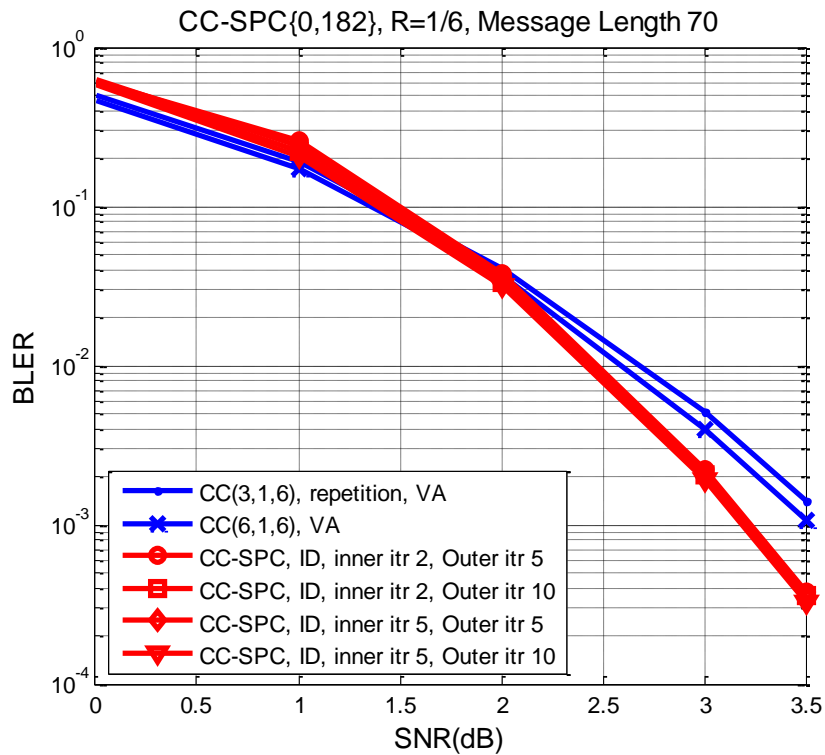
of right circular shifts = $(50+6)*3 - 122 = 46$



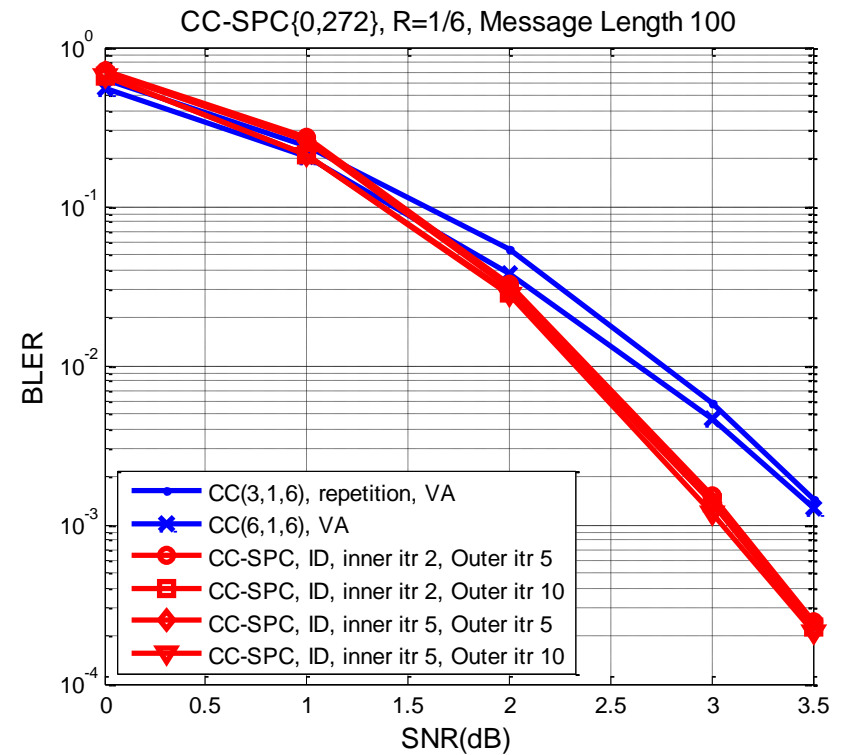
Simulation Results R=1/6

Iterative Decoding

Network Technology Laboratory



of right circular shifts = $(70+6)*3 - 182 = 46$



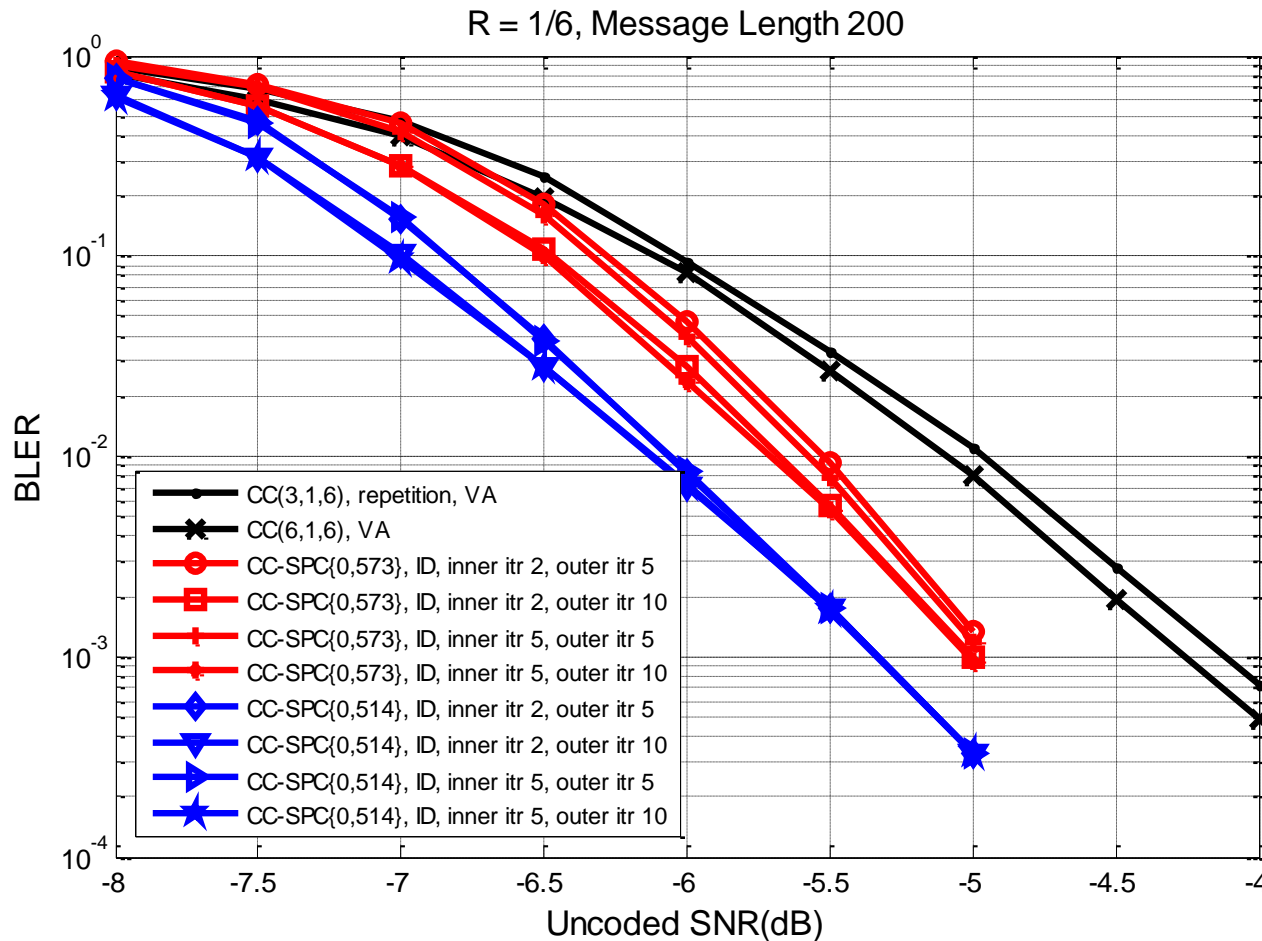
of right circular shifts = $(100+6)*3 - 272 = 46$



Simulation Results $R=1/6$

Iterative Decoding

Network Technology Laboratory



- ◆ CC-SPC{0,573}: # of right circular shifts = 45
- ◆ CC-SPC{0,514}: # of right circular shifts = 104



OSD (Ordered Statistics Decoding)

Network Technology Laboratory

\mathbf{y} : Received vector

\mathbf{G} : Generator matrix

1. Reordering the n bits according to reliabilities

- $\tilde{\mathbf{y}} = \lambda_1 (\mathbf{y}), |\tilde{\mathbf{y}}_1| \geq |\tilde{\mathbf{y}}_2| \geq \dots |\tilde{\mathbf{y}}_n|$
- $\tilde{\mathbf{G}} = \lambda_1 (\mathbf{G})$
- $\tilde{\mathbf{y}}$: Re-ordered received vector
- λ_1 : Permutation operation according to reliabilities

2. Find the first k independent column of $\tilde{\mathbf{G}}$

$$\hat{\mathbf{y}} = \lambda_2 (\tilde{\mathbf{y}}) = \lambda_2 \lambda_1 (\mathbf{y})$$
$$\hat{\mathbf{G}} = \lambda_2 (\tilde{\mathbf{G}}) = \lambda_2 \lambda_1 (\mathbf{G})$$

- λ_2 : Moving these k independent columns to the first k positions

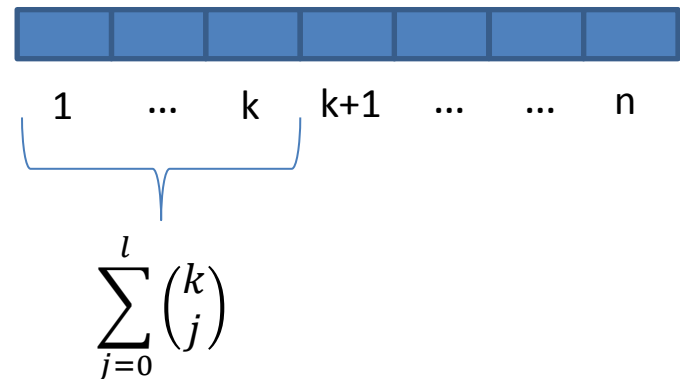


OSD (Ordered Statistics Decoding)

Network Technology Laboratory

3. Transform $\widehat{\mathbf{G}}$ into systematic form $\widehat{\mathbf{G}}_{\text{sys}}$ by row operation
4. Set $\mathbf{a} = [a_1 a_2 \dots a_k a_{k+1} \dots a_n]$ to be the hard-decision vector of $\widehat{\mathbf{y}}$ and let $\mathbf{a}_k = [a_1 a_2 \dots a_k]$. Then set $\widehat{\mathbf{c}} = \mathbf{a}_k \cdot \widehat{\mathbf{G}}_{\text{sys}}$.
For $j = 1$ to $j = l$,

- 1) Generate the test solution set $\mathcal{T}_j = \{\mathbf{v} \in GF(2)^k \mid W_H(\mathbf{v}) = j\}$
- 2) Generate the test codeword set $\mathcal{L}_j = \{(\mathbf{a}_k \oplus \mathbf{v}) \cdot \widehat{\mathbf{G}}_{\text{sys}} \mid \mathbf{v} \in \mathcal{T}_j\}$
- 3) Find \mathbf{c}^* that is closet to \mathbf{y} in Euclidean distance from \mathcal{L}_j .
- 4) If $d_D(\mathbf{c}^*, \mathbf{y}) < d_D(\widehat{\mathbf{c}}, \mathbf{y})$, then $\widehat{\mathbf{c}} = \mathbf{c}^*$.

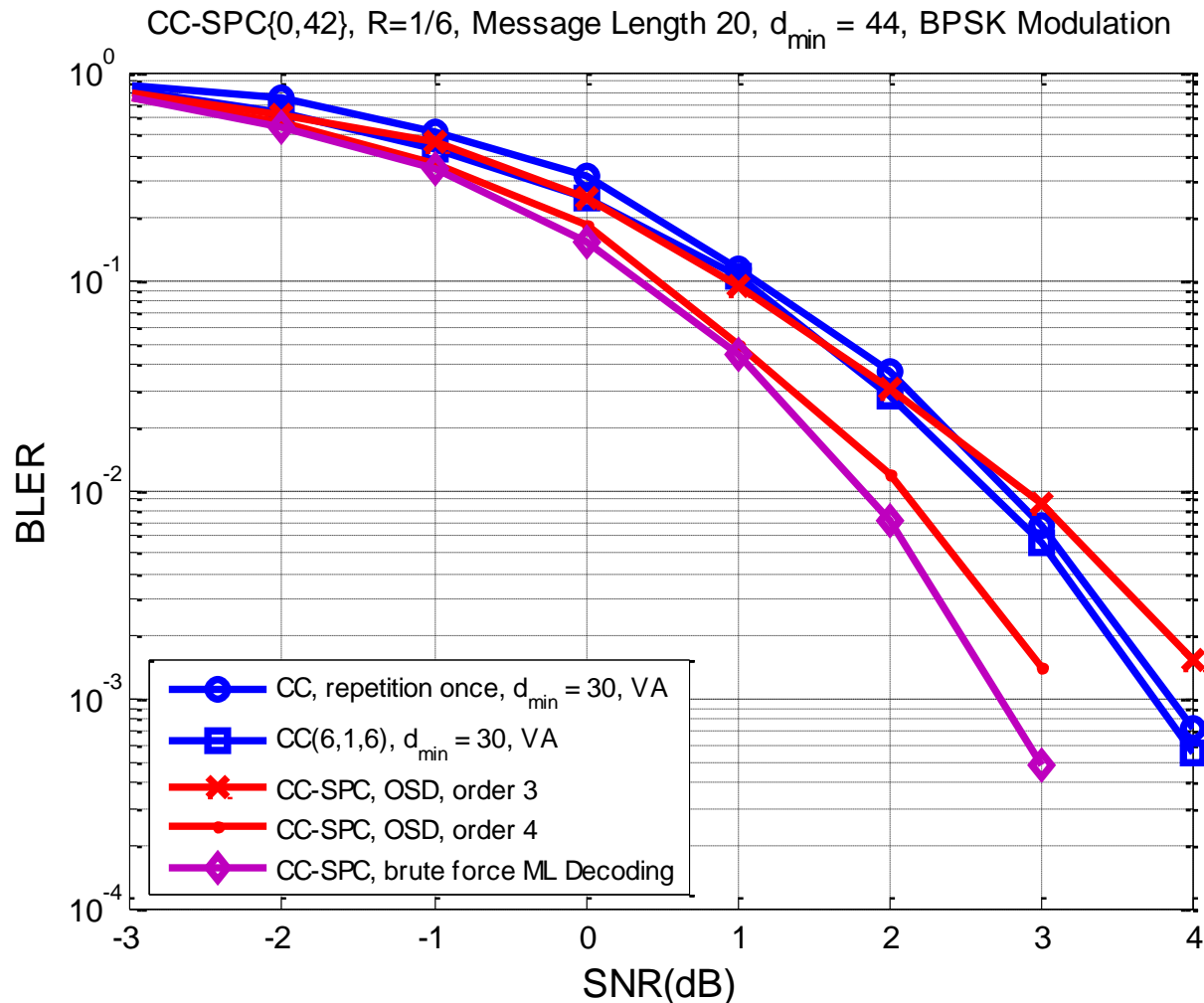


5. Convert $\widehat{\mathbf{c}}$ back to the corresponding codeword of the original code book.



Simulation Results $R=1/6$ Short Code Ordered Statistics Decoding

Network Technology Laboratory



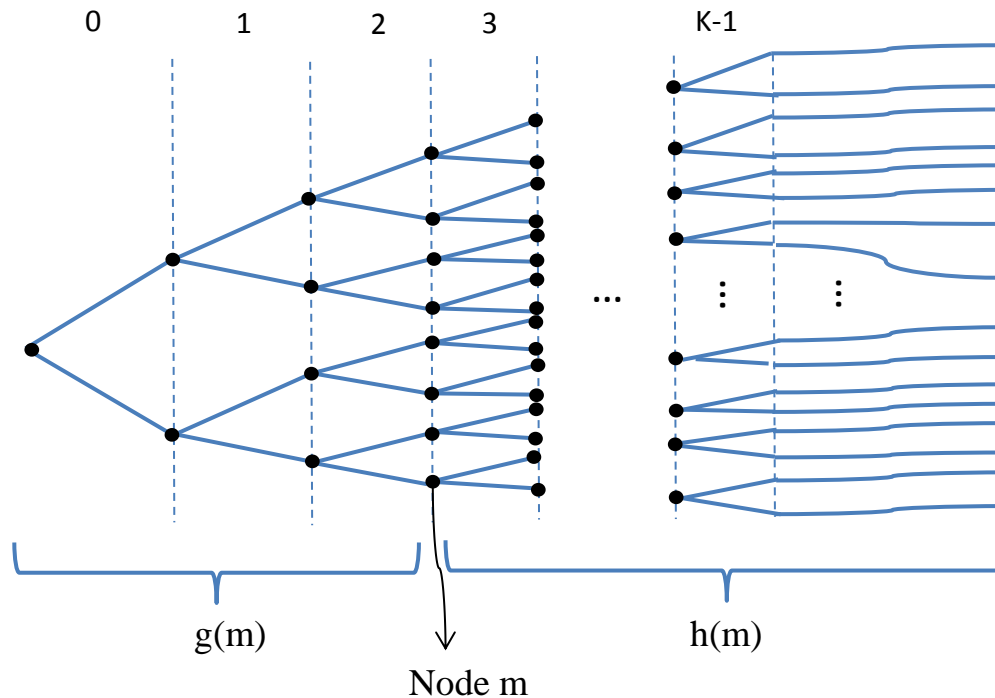


Algorithm A*

□ Priority-First Search Algorithm (a tree search guided by $f(m)$)

- ✓ $g(m)$: path metric
- ✓ $f(m) = g(m) + h(m)$
 - $h(m)$: heuristic function

- $$h(m) = \sum_{j=l+1}^{n-1} (|\phi_j| - 1)^2,$$
$$(|\phi_j| - 1)^2 = \min\{(\phi_j - 1)^2, (\phi_j + 1)^2\}$$





Algorithm A*

Network Technology Laboratory

1. Create an equivalent systematic code (and also its corresponding code tree)
2. Initialize the cost of the starting node as zero and push the starting node into the stack.
3. Push the successor nodes of the top node into the stack and discard the top node from the stack.
4. According to the ascending f -function values, sort the nodes in the stack.
5. If the top node in the stack is a goal node in the code tree, stop the algorithm; else go to Step 3.



Algorithm A*

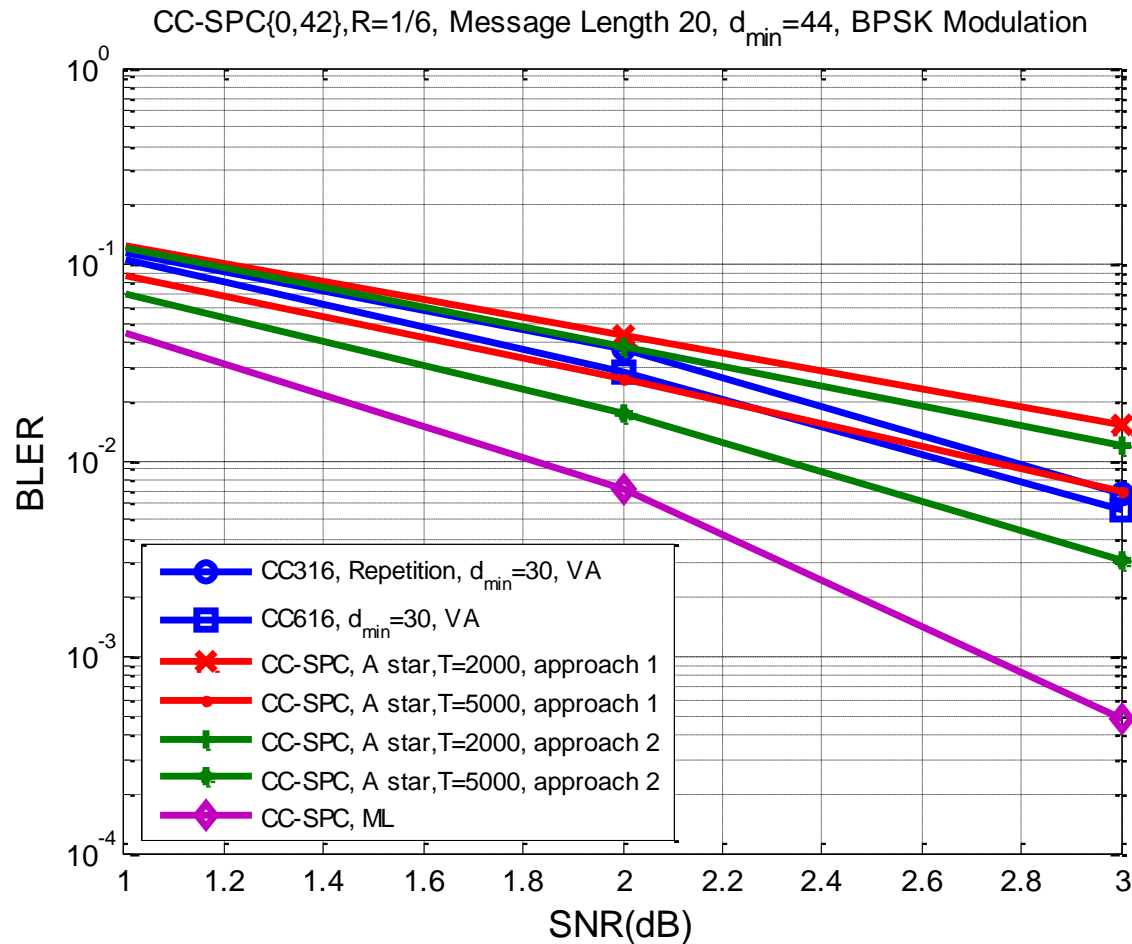
- ◆ If stack overflow occurs,
 - ✓ **Approach 1:** Delete the node with $\max f(m)$
 - ✓ **Approach 2:** Delete the node with min level



Simulation Results $R=1/6$ Short Code

Algorithm A* Decoding

Network Technology Laboratory

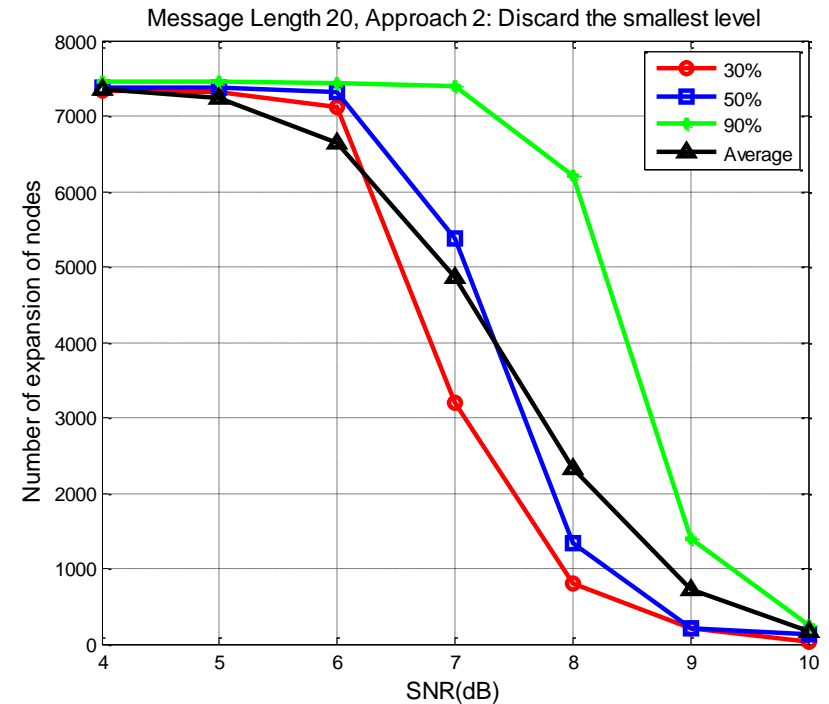
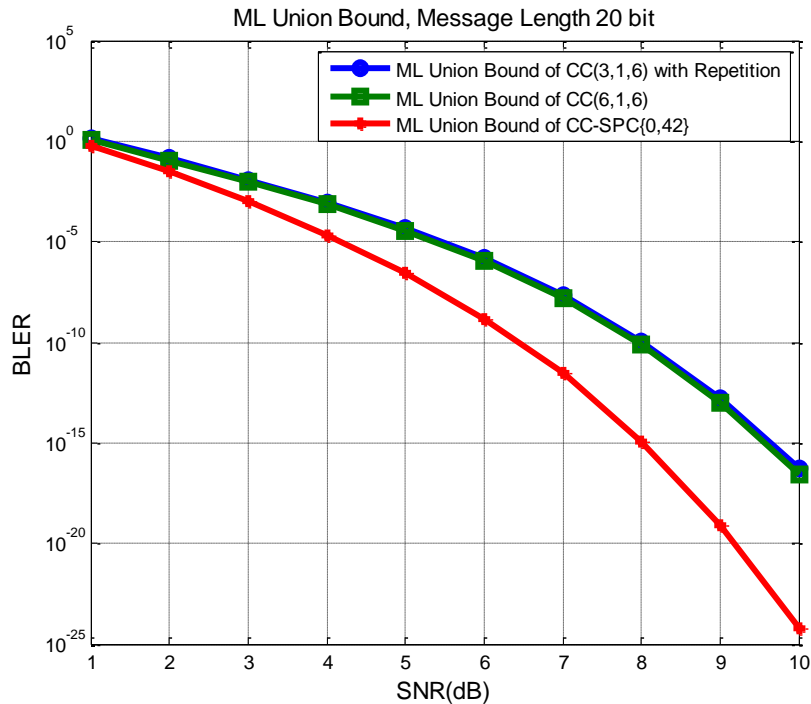




Simulation Results $R=1/6$ Short Code

Algorithm A* Decoding

Network Technology Laboratory

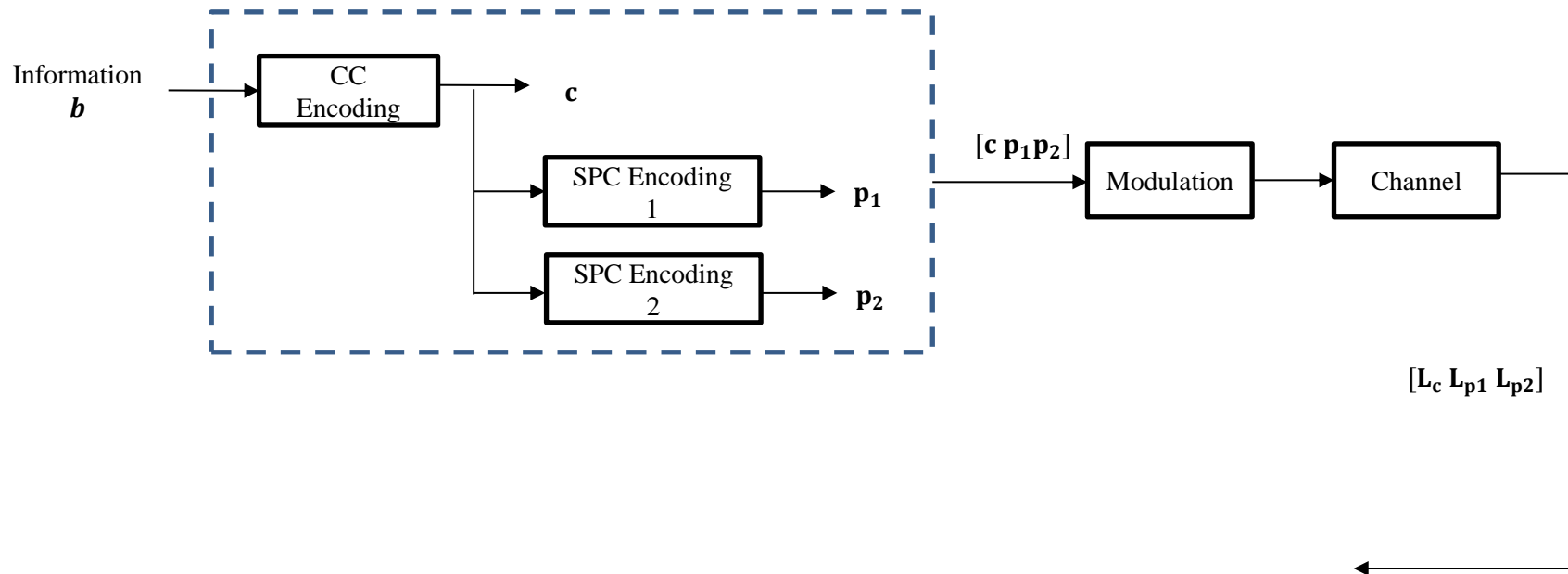




Schemes for Rate 1/9 CC-SPC

Network Technology Laboratory

- Scheme 1: Different SPCs are used for the first and the second retransmissions.
- Encoder

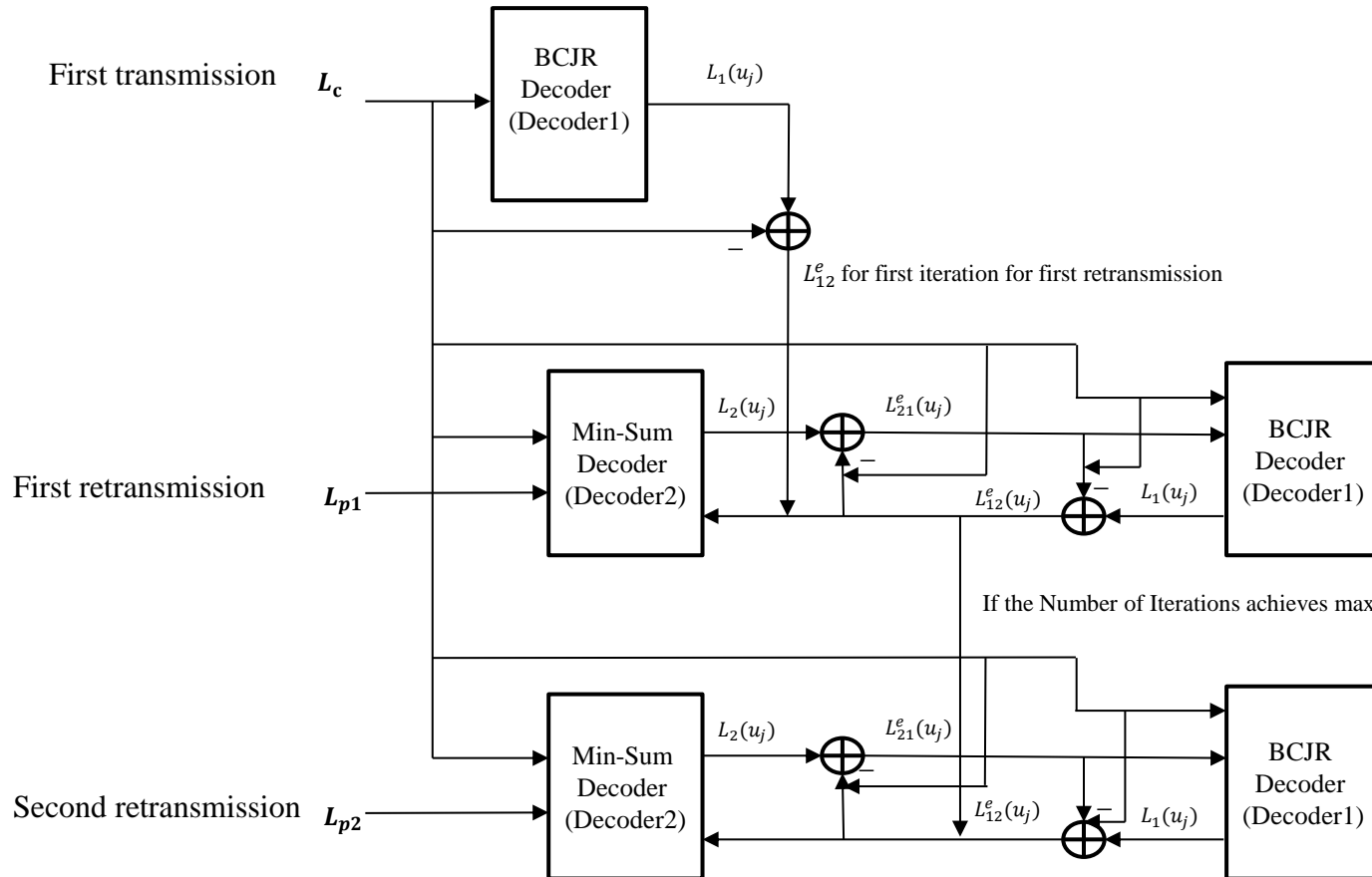




Schemes for Rate 1/9 CC-SPC

Network Technology Laboratory

□ Scheme 1: Decoder

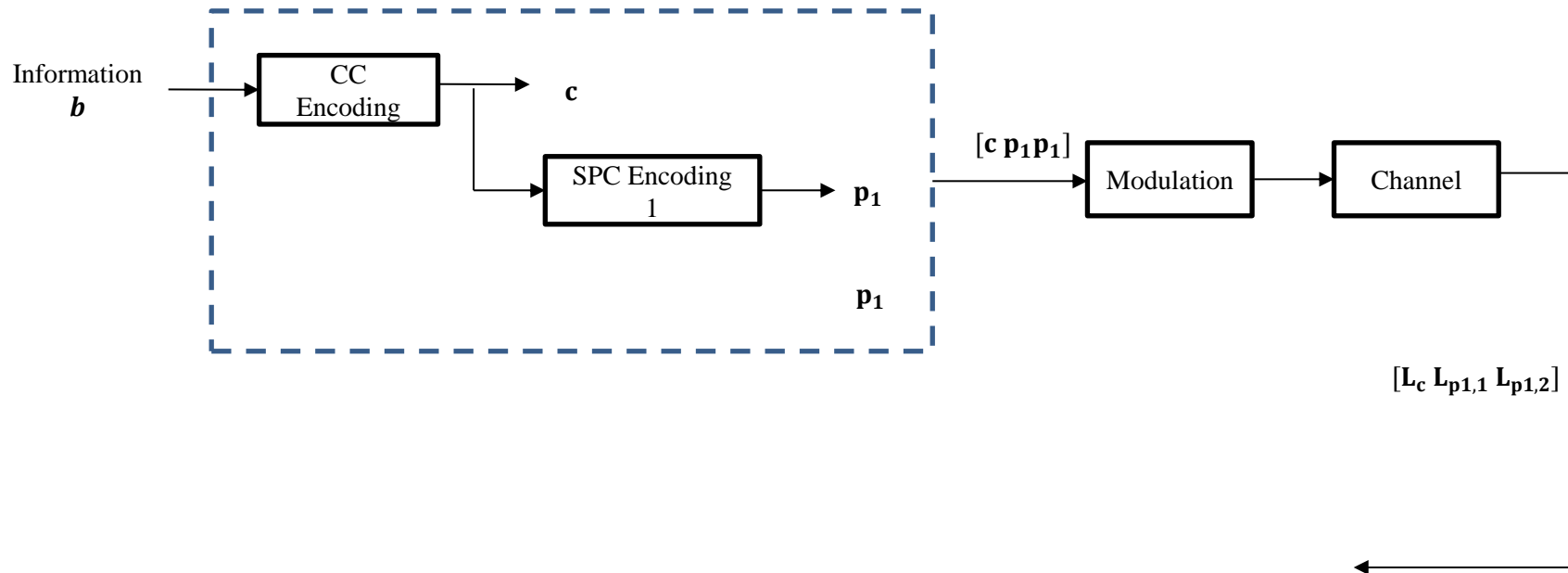




Schemes for Rate 1/9 CC-SPC

Network Technology Laboratory

- Scheme 2: Same SPCs are used for both retransmissions.
- Encoder

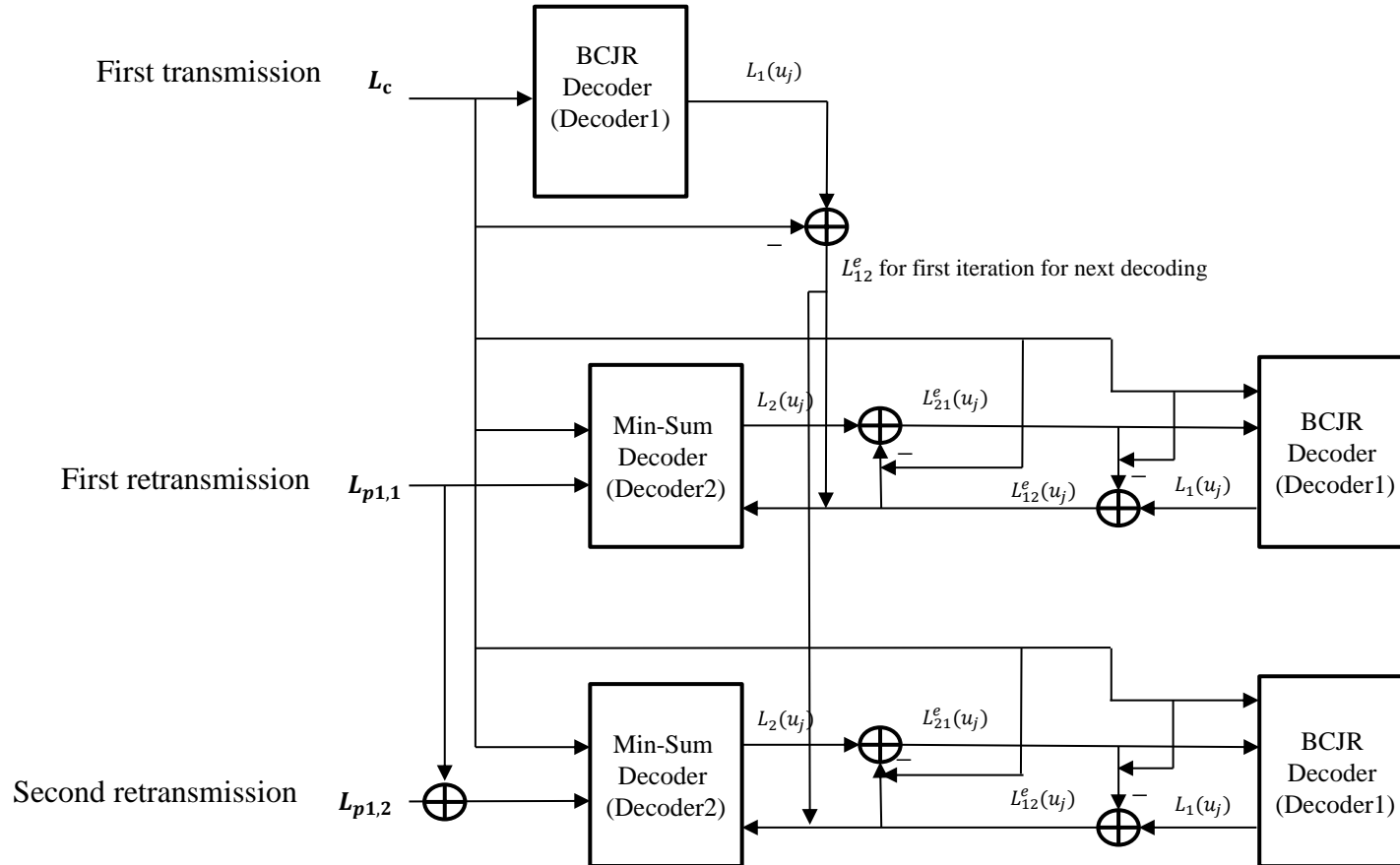




Schemes for Rate 1/9 CC-SPC

Network Technology Laboratory

□ Scheme 2: Decoder

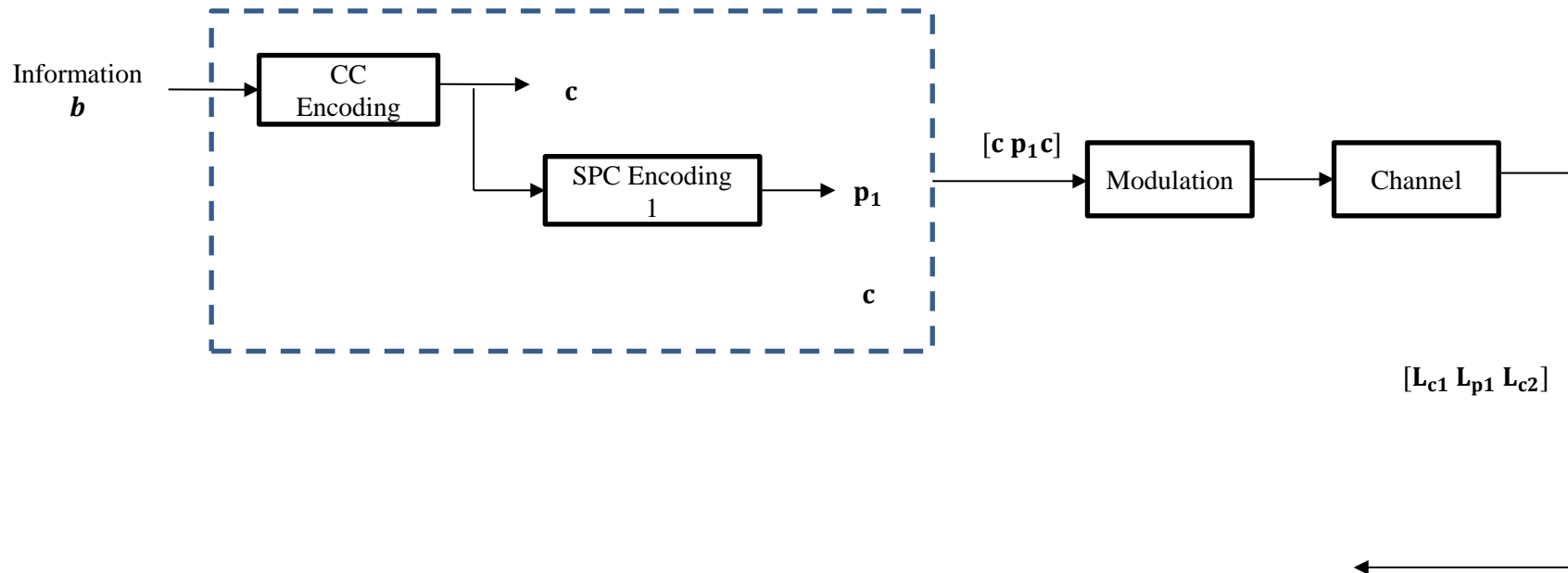




Schemes for Rate 1/9 CC-SPC

Network Technology Laboratory

- Scheme 3: Retransmit CC codeword
 - Encoder

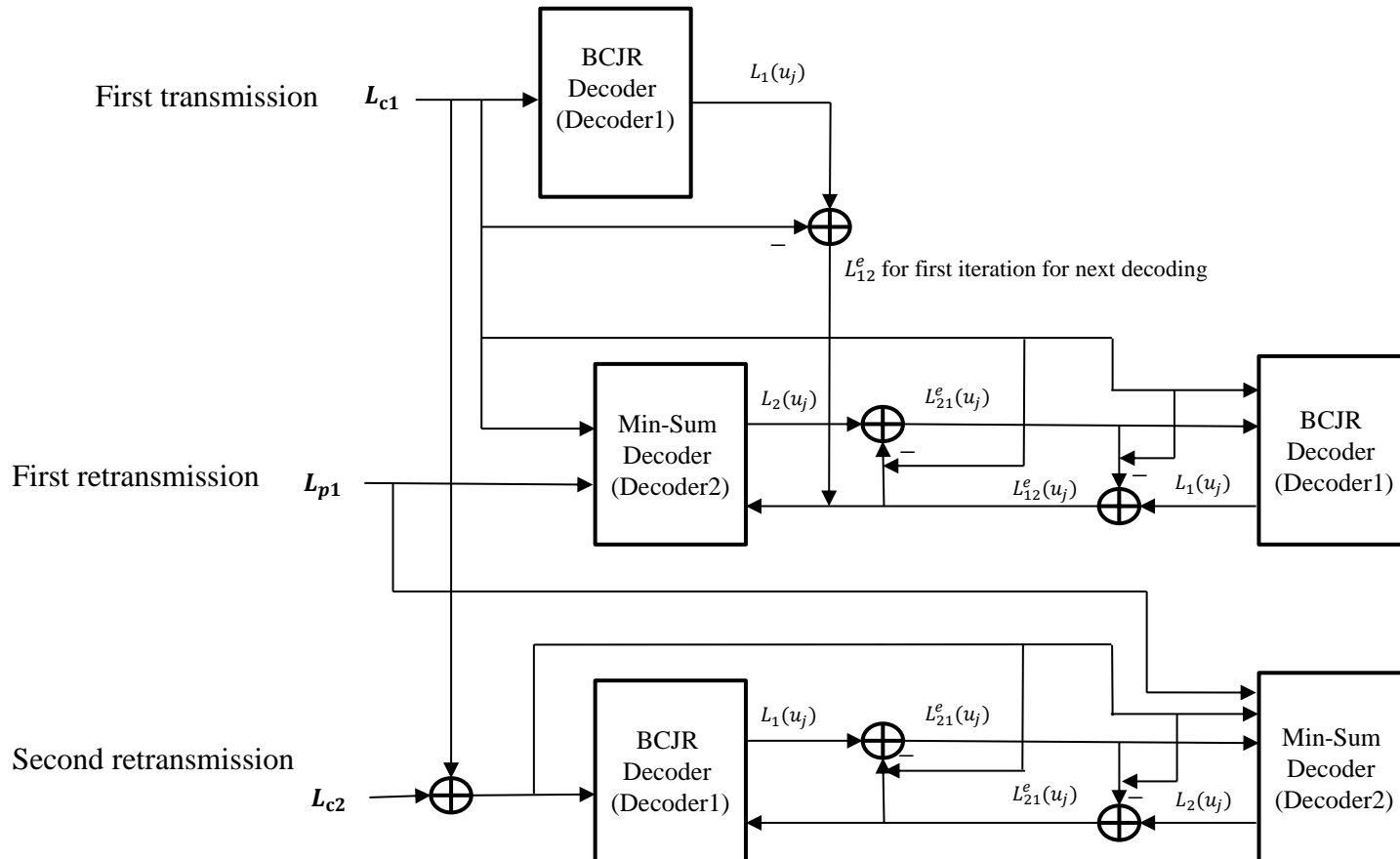




Schemes for Rate 1/9 CC-SPC

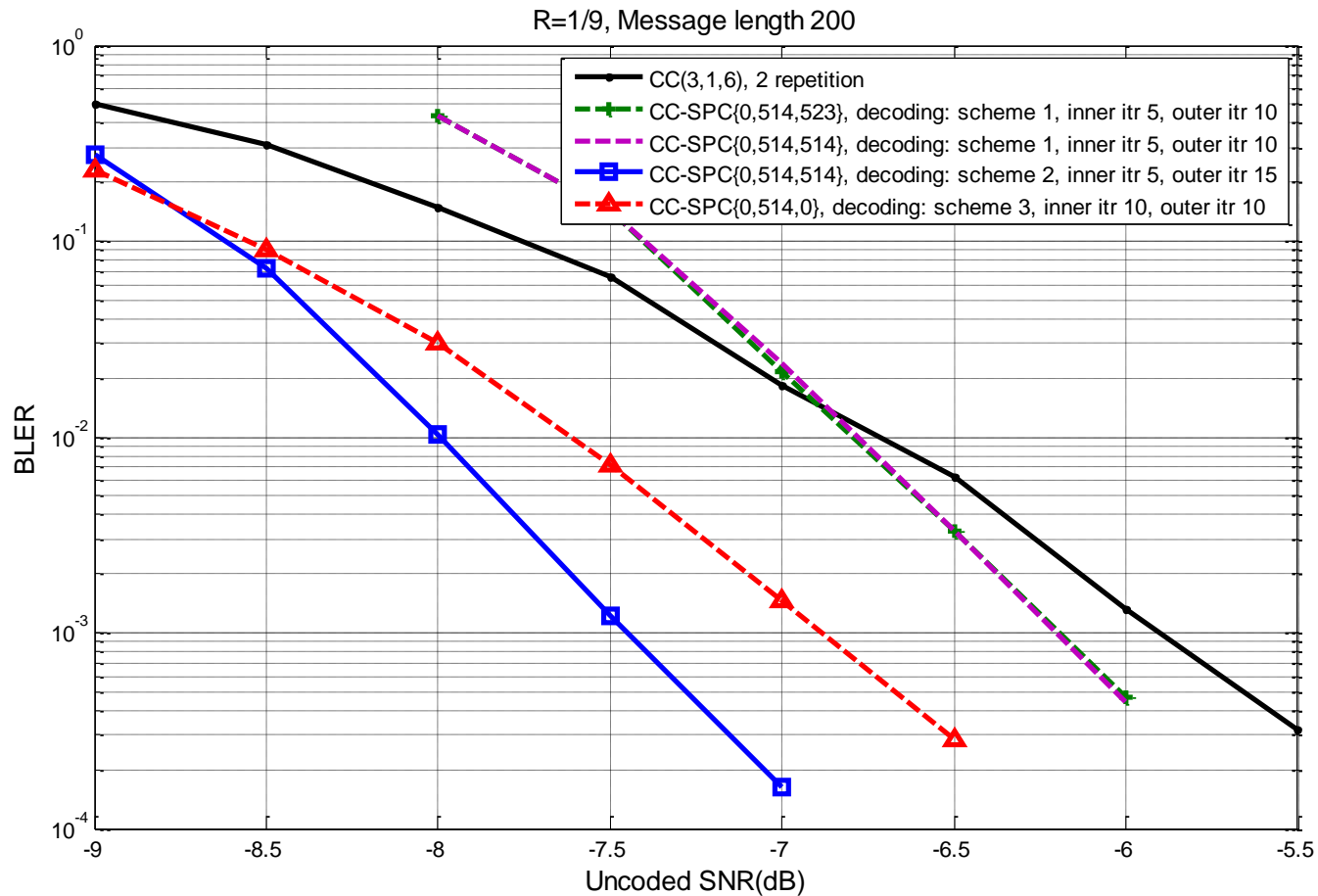
Network Technology Laboratory

□ Scheme 3: Decoder



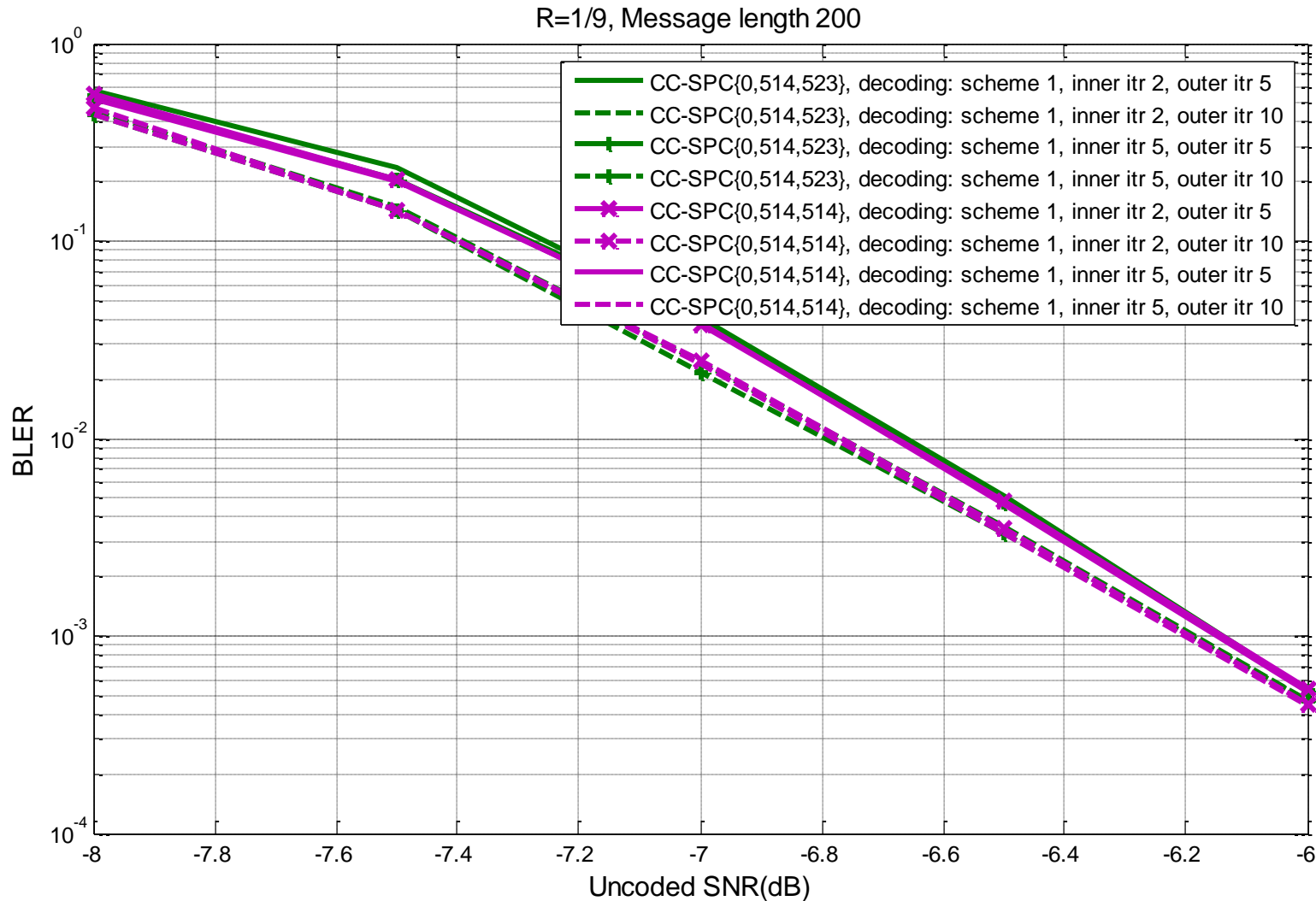


Simulation Results for $R = 1/9$





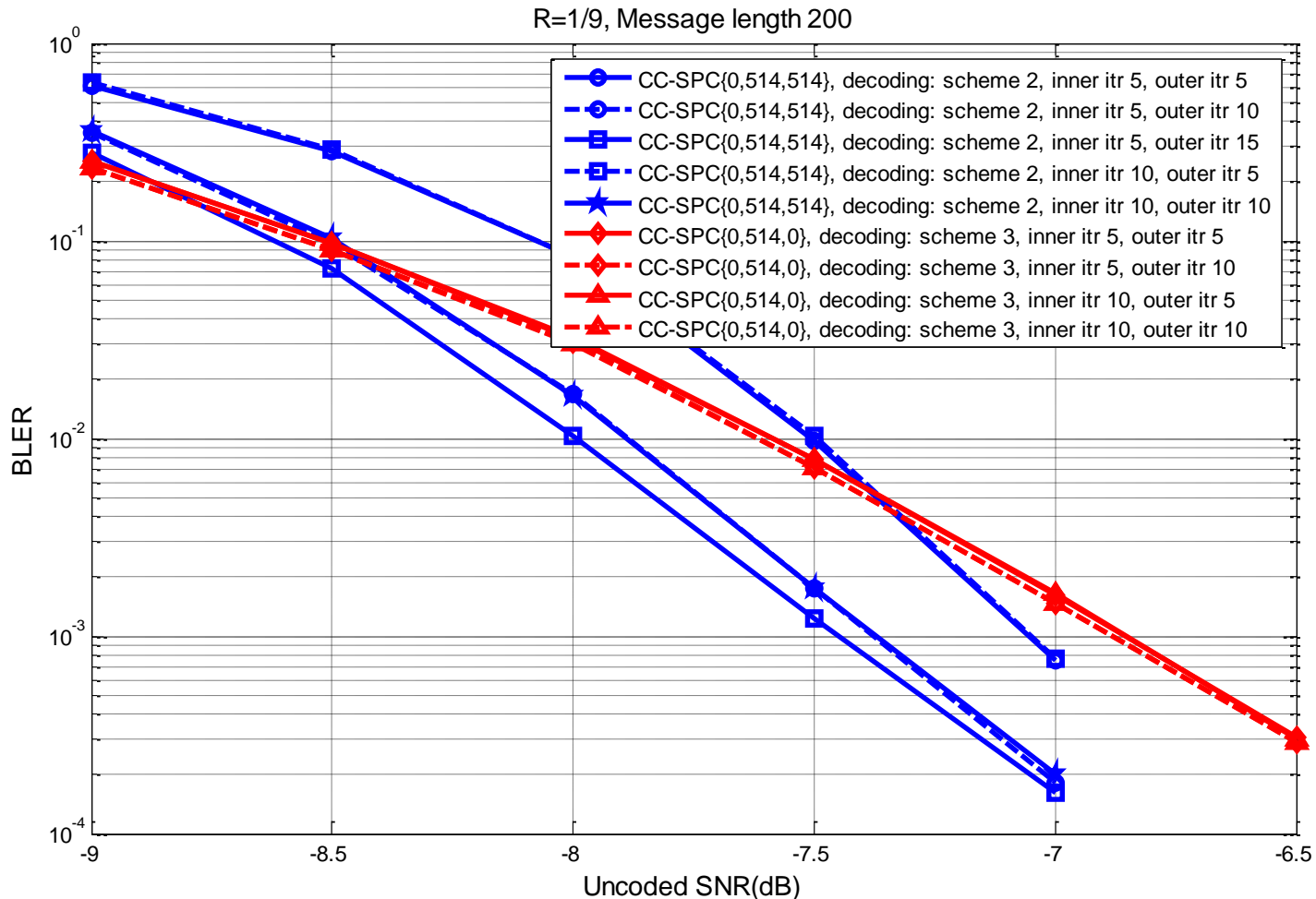
Simulation Results for $R = 1/9$





Simulation Results for $R = 1/9$

- Comparison of iteration numbers required for Scheme 2 and Scheme 3





Conclusions for Rate-1/6

Network Technology Laboratory

◆ Iterative Decoding

- ✓ If codeword length is sufficiently long, simulation results show that the performance of CC-SPC can be better than that of CC under VA decoding.

◆ OSD

- ✓ Simulation results show that CC-SPC/OSD can have around 0.5 dB gain over CC/VA when message length is 20 bits.
- ✓ However, the decoding complexity is large for codes of medium codeword length.

◆ Algorithm A*

- ✓ Low decoding complexity for codes of medium codeword length at high SNR.
- ✓ However, its performance is distant from the ML performance (due to finite-stack implementation).
- ✓ Hence, we suggest the use of Algorithm A* decoding for CC-SPC of medium codeword length at high SNR.



Conclusions for Rate-1/9

Network Technology Laboratory

- ◆ **Scheme 1** has the worst performance because the receiver cannot combine the extrinsic LLRs of the two retransmissions (of different SPC-encoding).
- ◆ **Scheme 2** has a better performance than **Scheme 3** if the number of iterations is sufficiently large.



Future Work

Network Technology Laboratory

- ◆ How to improve the selection of right circular shifts of CC-SPC.
 - Possibly, based on **weight spectrum** and **SPC girth**.
- ◆ XOR-ing Three CC codewords (instead of Two) hopefully to further increase the minimum pairwise Hamming distance.



Comparison of CC-SPCs of $R = 1/6$

Network Technology Laboratory

Message Length 200							
CC-SPC	{0,417}	{0,567}	{0,317}	{0,367}	{0,467}	{0,517}	CC Rate 1/3
Girth	412	412	1236	1236	1236	1236	---
WEF(1)	$x^{60} + 3x^{57} + 2x^{51} + x^{48} + 2x^{45}$	$x^{60} + 3x^{57} + 2x^{51} + x^{48} + 2x^{45}$	$x^{60} + 3x^{57} + 2x^{51} + x^{48} + 2x^{45}$	$x^{60} + 3x^{57} + 2x^{51} + x^{48} + 2x^{45}$	$x^{60} + 3x^{57} + 2x^{51} + x^{48} + 2x^{45}$	$x^{60} + 3x^{57} + 2x^{51} + x^{48} + 2x^{45}$	$x^{20} + 3x^{19} + 2x^{17} + x^{16} + 2x^{15}$
WEF(2)	$8x^{57} + 5x^{54} + 8x^{51} + 3x^{48} + 9x^{45}$	$8x^{57} + 5x^{54} + 8x^{51} + 3x^{48} + 9x^{45}$	$8x^{57} + 5x^{54} + 8x^{51} + 3x^{48} + 9x^{45}$	$8x^{57} + 5x^{54} + 8x^{51} + 3x^{48} + 9x^{45}$	$8x^{57} + 5x^{54} + 8x^{51} + 3x^{48} + 9x^{45}$	$8x^{57} + 5x^{54} + 8x^{51} + 3x^{48} + 9x^{45}$	$8x^{19} + 5x^{18} + 8x^{17} + 3x^{16} + 9x^{15}$
WEF(3)	$200x^{57} + 16x^{54} + 13x^{51} + 9x^{48} + 204x^{45}$	$200x^{57} + 16x^{54} + 13x^{51} + 9x^{48} + 204x^{45}$	$200x^{57} + 16x^{54} + 13x^{51} + 9x^{48} + 204x^{45}$	$200x^{57} + 16x^{54} + 13x^{51} + 9x^{48} + 204x^{45}$	$200x^{57} + 16x^{54} + 13x^{51} + 9x^{48} + 204x^{45}$	$200x^{57} + 16x^{54} + 13x^{51} + 9x^{48} + 204x^{45}$	$200x^{19} + 16x^{18} + 13x^{17} + 9x^{16} + 204x^{15}$
WEF(4)	$206x^{57} + 223x^{54} + 28x^{51} + 14x^{48} + 206x^{45}$	$206x^{57} + 223x^{54} + 28x^{51} + 14x^{48} + 206x^{45}$	$206x^{57} + 223x^{54} + 28x^{51} + 14x^{48} + 206x^{45}$	$206x^{57} + 223x^{54} + 28x^{51} + 14x^{48} + 206x^{45}$	$206x^{57} + 223x^{54} + 28x^{51} + 14x^{48} + 206x^{45}$	$206x^{57} + 223x^{54} + 28x^{51} + 14x^{48} + 206x^{45}$	$206x^{19} + 223x^{18} + 28x^{17} + 14x^{16} + 206x^{15}$
WEF(5)	$397x^{57} + 241x^{54} + 790x^{51} + 18x^{48} + 403x^{45}$	$397x^{57} + 241x^{54} + 790x^{51} + 18x^{48} + 403x^{45}$	$397x^{57} + 241x^{54} + 790x^{51} + 18x^{48} + 403x^{45}$	$397x^{57} + 241x^{54} + 790x^{51} + 18x^{48} + 403x^{45}$	$397x^{57} + 241x^{54} + 790x^{51} + 18x^{48} + 403x^{45}$	$397x^{57} + 241x^{54} + 790x^{51} + 18x^{48} + 403x^{45}$	$397x^{19} + 241x^{18} + 790x^{17} + 18x^{16} + 403x^{15}$
Iterative Decoding: Inner iter 5, Outer iter 10							---
BLER (Uncoded SNR = -7 dB)	1.284522e-001	2.403846e-001	2.557545e-001	9.487666e-002	9.925558e-002	1.141553e-001	---
BLER (Uncoded SNR = -6 dB)	8.950148e-003	1.684211e-002	2.368826e-002	6.801796e-003	6.147982e-003	6.576351e-003	---



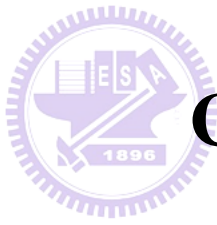
Comparison of CC-SPCs of $R = 1/6$

Network Technology Laboratory

◆ Message Length 200

Inner iter 5, Outer iter 10						
CC-SPC	{0,325}	{0,375}	{0,425}	{0,475}	{0,525}	{0,575}
girth	1236	412	1236	1236	412	1236
BLER (Uncoded SNR = -7 dB)	2.049180e-001	1.017294e-001	1.116695e-001	9.876543e-002	1.140901e-001	3.344482e-001
BLER (Uncoded SNR = -6 dB)	1.205400e-002	7.644091e-003	7.341874e-003	7.486431e-003	8.316008e-003	2.830055e-002

Inner iter 5, Outer iter 10						
CC-SPC	{0,309}	{0,359}	{0,409}	{0,459}	{0,509}	{0,559}
girth	4	1236	1236	412	1236	1236
BLER (Uncoded SNR = -7 dB)	7.575758e-001	1.003009e-001	1.304631e-001	1.199041e-001	9.606148e-002	2.111932e-001
BLER (Uncoded SNR = -6 dB)	3.663004e-001	6.466632e-003	8.237572e-003	7.730066e-003	6.730834e-003	1.317176e-002



Comparison of Iteration Numbers for R = 1/6 Codes

Network Technology Laboratory

Rate 1/6					
BLER, uncoded SNR = -6 dB					
Girth = 1236					
Inner iter	Outer iter	{0,325}	{0,575}	{0,559}	{0,509}
5	5	2.238138e-002	5.011275e-002	2.059944e-002	6.809902e-003
5	10	1.205400e-002	2.830055e-002	1.317176e-002	6.730834e-003
5	15	1.143968e-002	2.597740e-002	1.174398e-002	6.498148e-003
5	20	1.063377e-002	2.426595e-002	1.132182e-002	6.498148e-003
10	5	2.271695e-002	4.206984e-002	1.891253e-002	7.530120e-003
10	10	1.248439e-002	2.792126e-002	1.331647e-002	6.977637e-003
10	15	1.127396e-002	2.528765e-002	1.308301e-002	6.563402e-003
10	20	1.062586e-002	2.485398e-002	1.285430e-002	6.141374e-003
20	5	2.150769e-002	5.028916e-002	1.891253e-002	8.188332e-003
20	10	1.317436e-002	2.799944e-002	1.331647e-002	7.094714e-003
20	15	1.127396e-002	2.426595e-002	1.301321e-002	6.563402e-003
20	20	1.062586e-002	2.318572e-002	1.285430e-002	6.141374e-003



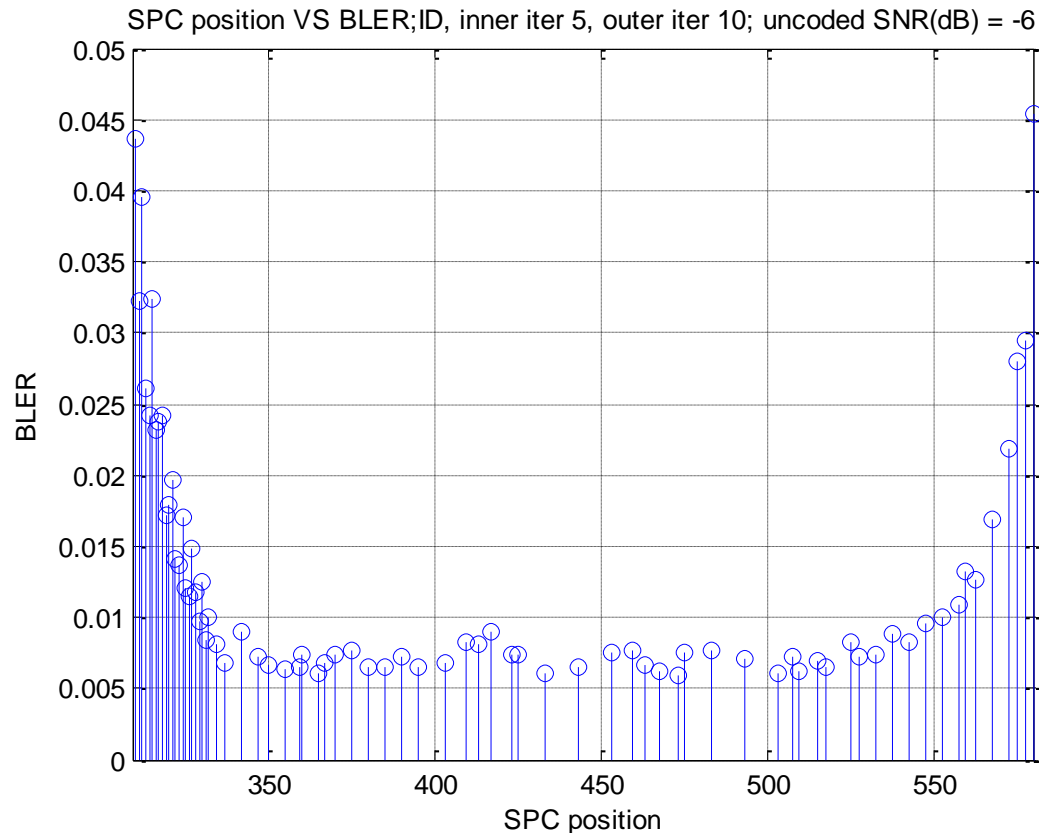
Comparison of Different XOR Positions

Network Technology Laboratory

Message Length 200

Uncoded SNR = -6 dB

Inner itr 5, outer itr 10





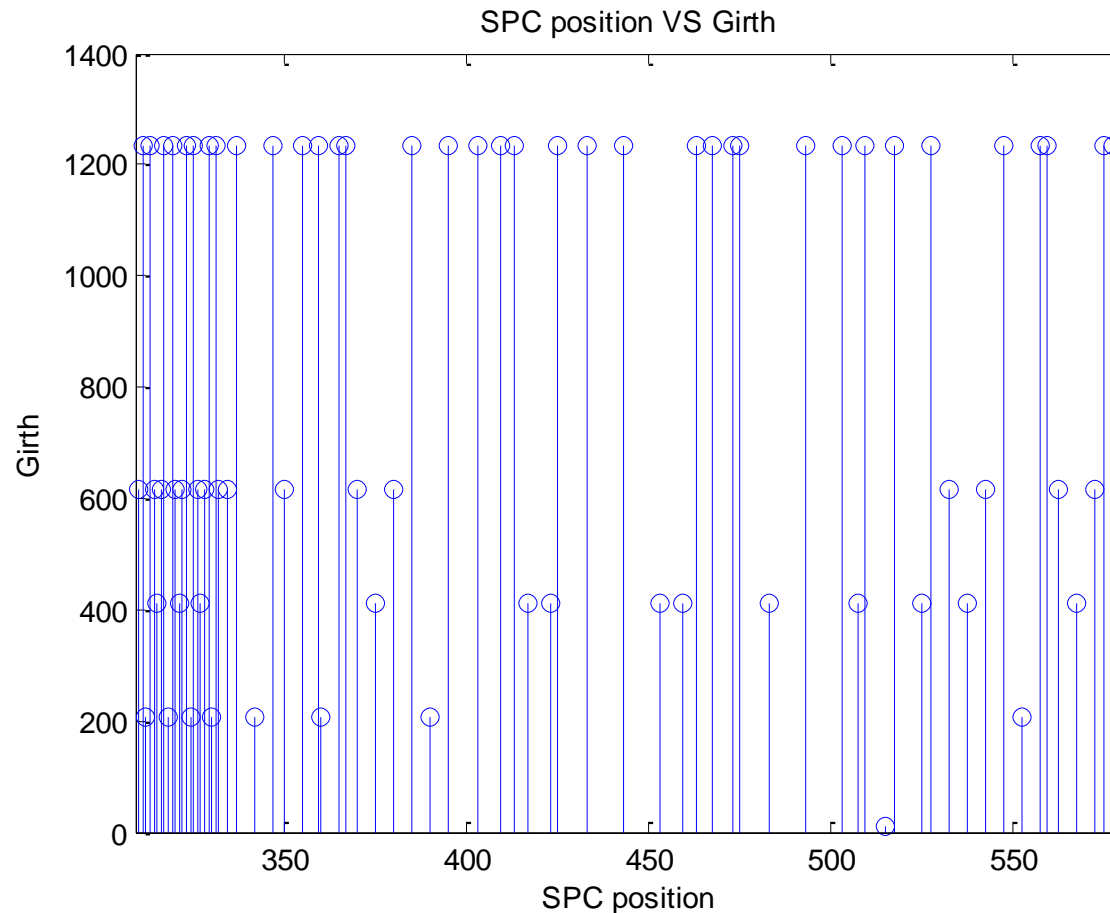
Comparison of Different XOR Positions

Network Technology Laboratory

Message Length 200

Uncoded SNR = -6 dB

Inner itr 5, outer itr 10





Thank you for your attention.



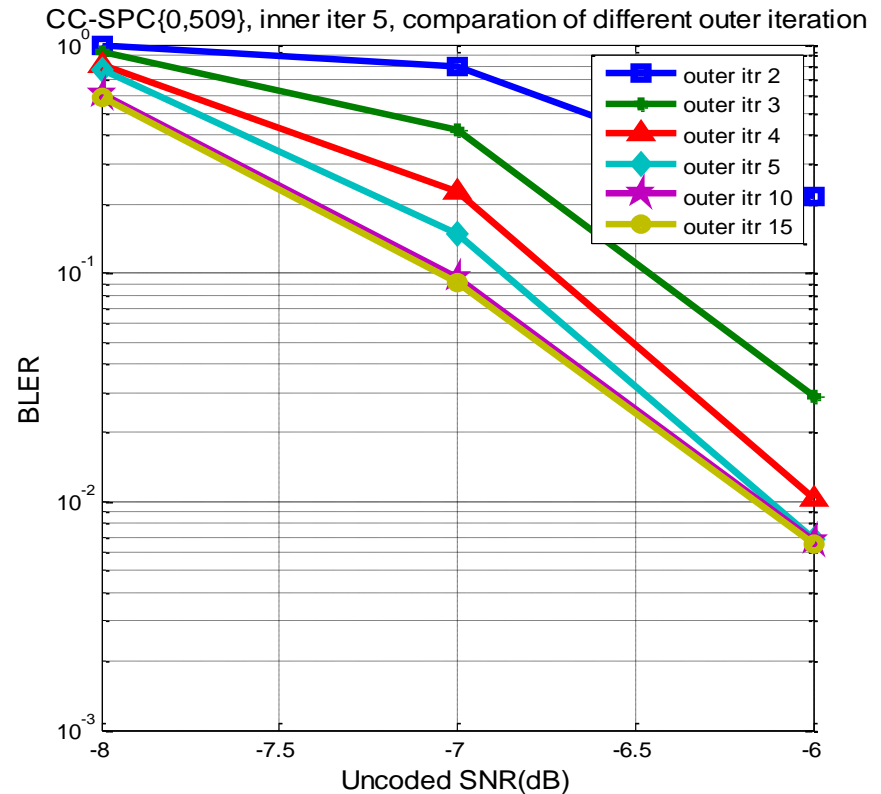
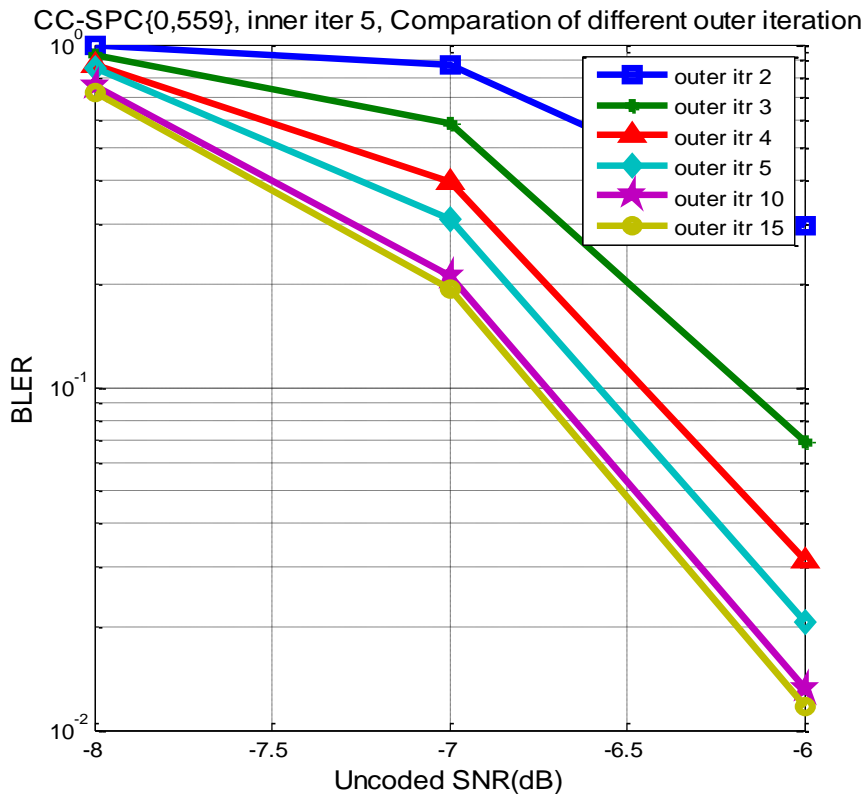
References

- [1] S. A. Abrantes, “From BCJR to turbo decoding: MAP algorithms made easier,” University of Porto, Tech. Rep., 2004.
- [2] NGMN Alliance, *5G White Paper*, Feb. 2015; [https:// www.ngmn.org/uploads/media/ NGMN_5G_White_Paper_V1_0.pdf](https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf)
- [3] E. Dahlman, S. Parkvall, and J. Sköld, *4G: LTE/LTE-Advanced for Mobile Broadband*, Academic Press, ISBN:012385489X, 2011.
- [4] D. G. Daut, J. W. Modestino, and L. D. Wismer, “New short constraint length convolutional code constructions for selected rational rates,” *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 794–800, Sept. 1982.
- [5] ETSI TS 136 212 LTE; *Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing Channel Coding*, ETSI, Route des Lucioles, France, 2016.
- [6] M. Fossorier and S. Lin, “Soft-decision decoding of linear block codes based on ordered statistics,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 1379–1396, Sept. 1995.
- [7] Y. Han, C. Hartmann, and C.-C. Chen, “Efficient priority-first search maximum-likelihood soft-decision decoding of linear block codes,” *IEEE Trans. Inform. Theory*, vol. 39, no. 5, pp. 1514–1523, Sep. 1993.
- [8] M. Fossorier and S. Lin, “Soft-decision decoding of linear block codes based on ordered statistics,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 1379–1396, Sept. 1995.
- [9] T. Levanen and M. Renfors, “Efficient parity bit estimation for turbo codes,” *SPAWC2009*, Perugia, Italy, pp.663-667, June 2009.
- [10] P. Popovski, “Ultra-reliable communication in 5G wireless systems,” *Proc. 5GU Conf.*, 6 pp., Nov. 2014.
- [11] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge University Press, 2009.
- [12] S. L. Shieh, P.-N. Chen, and Y. S. Han, “Reduction of computational complexity and sufficient stack size of the MLSDA by early elimination,” *IEEE International Symposium on Information Theory*, Nice, France, pp. 1671-1675, Jun. 2007.



Comparison of Different Number of Iterations

- ◆ Comparison of max numbers of Outer iterations
- ◆ Message Length 200, $R=1/6$

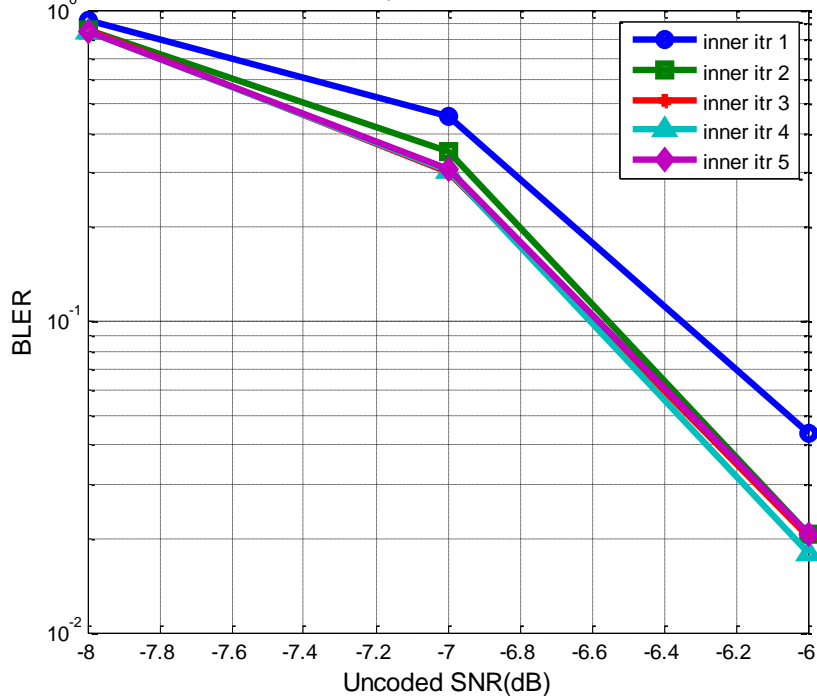




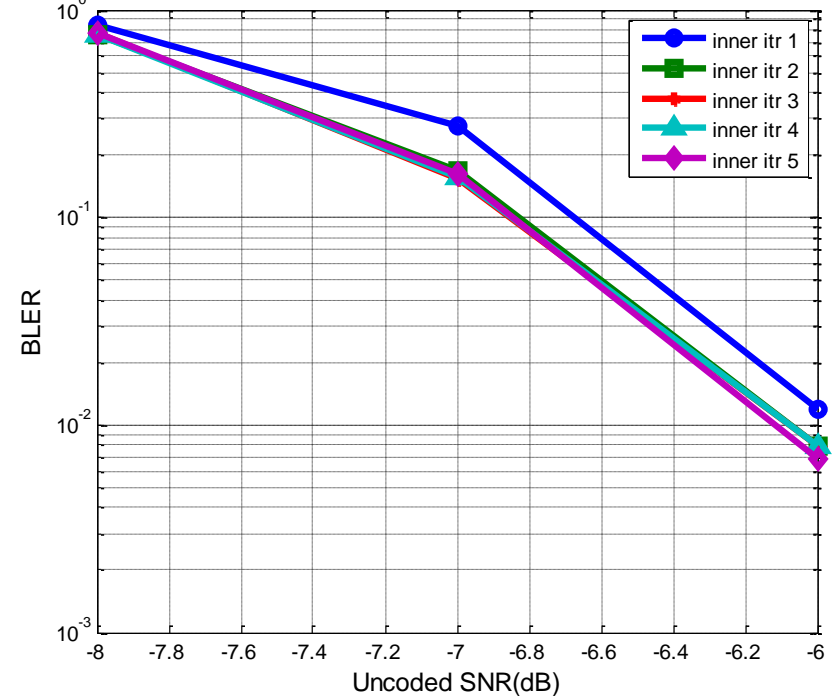
Comparison of Different Numbers of Iterations

- ◆ Comparison of max numbers of Inner iterations
- ◆ Message Length 200, $R=1/6$

CC-SPC{0,559}, outer itr 5, comparison of different number of inner iteration



CC-SPC{0,509}, outer itr 5, comparison of different number of inner iteration





Appendix

◆ Sum-Product Algorithm

□ Denote

- Variable node v_j
- Check node c_i
- Received signal y_j
- Noise power σ^2

1. Initialization

$$L_{i \rightarrow j} = \frac{2y_j}{\sigma^2}$$

2. Check node update

$$L_{i \rightarrow j} = 2 \tanh^{-1} \left(\prod_{j' \in N(i), j' \neq j} \tanh\left(\frac{1}{2} L_{j' \rightarrow i}\right) \right)$$

3. Variable node update

$$L_{j \rightarrow i} = L_j + \sum_{i' \in N(j), i' \neq i} L_{i' \rightarrow j}$$

4. LLR Total

$$L_j^{total} = L_j + \sum_{i \in N(j)} L_{i \rightarrow j}$$

5. Stop Criteria

$\hat{\mathbf{v}}\mathbf{H}^T = 0$, or iteration number achieves max limit; else, go to 2.



Appendix

◆ Simplification of Sum-Product Algorithm

Set

$$\begin{aligned}L_{j \rightarrow i} &= \alpha_{j,i} \beta_{j,i} \\ \alpha_{j,i} &= \text{sign}(L_{j \rightarrow i}) \\ \beta_{j,i} &= |L_{j \rightarrow i}|\end{aligned}$$

Then,

$$\prod_{j' \in N(i), j' \neq j} \tanh\left(\frac{1}{2} L_{i \rightarrow j}\right) = \prod_{j' \in N(i), j' \neq j} \alpha_{j',i} \prod_{j' \in N(i), j' \neq j} \tanh\left(\frac{1}{2} \beta_{j',i}\right)$$

$$\begin{aligned}L_{i \rightarrow j} &= \prod_{j' \in N(i), j' \neq j} \alpha_{j',i} \cdot 2 \tanh^{-1}\left(\prod_{j' \in N(i), j' \neq j} \tanh\left(\frac{1}{2} \beta_{j',i}\right)\right) \\ &= \prod_{j' \in N(i), j' \neq j} \alpha_{j',i} \cdot 2 \tanh^{-1} \log^{-1} \log\left(\prod_{j' \in N(i), j' \neq j} \tanh\left(\frac{1}{2} \beta_{j',i}\right)\right) \\ &= \prod_{j' \in N(i), j' \neq j} \alpha_{j',i} \cdot 2 \tanh^{-1} \log^{-1}\left(\sum_{j' \in N(i), j' \neq j} \log(\tanh\left(\frac{1}{2} \beta_{j',i}\right))\right)\end{aligned}$$

□ Check Node Update

$$\begin{aligned}L_{i \rightarrow j} &= \prod_{j' \in N(i), j' \neq j} \alpha_{j',i} \cdot \phi\left(\sum_{j' \in N(i), j' \neq j} \phi(\beta_{j',i})\right), \\ \text{where } \phi(x) &= -\log\left[\tanh\left(\frac{x}{2}\right)\right] = \log\left(\frac{e^x + 1}{e^x - 1}\right)\end{aligned}$$



Appendix

◆ Min-Sum Algorithm

$$\phi \left(\sum_{j' \in N(i), j' \neq j} \phi(\beta_{j',i}) \right) \approx \phi \left(\phi \left(\min_{j' \in N(i), j' \neq j} \beta_{j',i} \right) \right) = \min_{j' \in N(i), j' \neq j} \beta_{j',i}$$

Check Node Update: $L_{i \rightarrow j} = \prod_{j' \in N(i), j' \neq j} \alpha_{j',i} \cdot \min_{j' \in N(i), j' \neq j} \beta_{j',i}$



Appendix

◆ Bahl-Cocke-Jelinek-Raviv (BCJR) Algorithm

➤ Assume

□ \mathbf{y} : received sequence

□ $L_a(u_k) = \ln \left(\frac{p(u_k=+1)}{p(u_k=-1)} \right)$: Log-likelihood ratio for prior probability

□ $L(u_k) = \ln \left(\frac{p(u_k=+1|\mathbf{y})}{p(u_k=-1|\mathbf{y})} \right)$: Log-likelihood ratio for a posteriori probability



Appendix

- ◆ $p(u_k = +1|\mathbf{y}) = \frac{p(u_k=+1,\mathbf{y})}{p(\mathbf{y})} = \sum_{u^+} p(s_{k-1} = s', s_k = s, \mathbf{y})$
- ◆ $L(u_k) = \ln \frac{\sum_{u^+} p(s', s, \mathbf{y})}{\sum_{u^-} p(s', s, \mathbf{y})}$
, $p(s', s, \mathbf{y}) = p(s', s, \mathbf{y}_{t < k}, \mathbf{y}_k, \mathbf{y}_{t > k}) = p(\mathbf{y}_{t < k}, s') p(s, \mathbf{y}_k | s') p(\mathbf{y}_{t > k} | s)$

- ◆ Forward Recursive $\alpha_{k-1}(s') \equiv p(\mathbf{y}_{t < k}, s')$
- ◆ $\gamma_k(s', s) \equiv p(s, \mathbf{y}_k | s')$
- ◆ Backward Recursive $\beta_k(s) \equiv p(\mathbf{y}_{t > k} | s)$

Then $p(s', s, \mathbf{y}) = \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s)$



Appendix

◆ Calculations of α

$$\text{◆ } \alpha_k(s) = p(s', \mathbf{y}_{t < k+1}) = \sum_{s'} p(s', s, \mathbf{r}_{t < k+1})$$

$$= \sum_{s'} p(s, \mathbf{y}_k | s', \mathbf{y}_{t < k}) p(s', \mathbf{y}_{t < k}) = \sum_{s'} p(s, \mathbf{y}_k | s') p(s', \mathbf{y}_{t < k}) = \sum_{s'} \gamma_k(s', s) \alpha_{k-1}(s')$$

$$\text{◆ } \alpha_0(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases}$$

$$\text{◆ } \beta_{k-1}(s') = \sum_s \gamma_k(s', s) \beta_k(s)$$

$$\text{◆ } \beta_N(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases}$$

$$\text{◆ } \gamma_k(s', s) = p(\mathbf{y}_k | x_k) p(u_k) = C_k e^{u_k L(u_k)/2} \exp\left(\frac{L_c}{2} \sum_{l=1}^n x_{kl} y_{kl}\right)$$

$$\text{◆ } L_c = 4a \frac{E_c}{N_0} = 4a R_c \frac{E_b}{N_0}$$

- ◆ a : fading amplitude
- ◆ R_c : code rate
- ◆ $N_0/2$: Noise power
- ◆ E_b : bit energy



Appendix

◆ Simplified MAP algorithm

- ◆ $\Gamma_k(s', s) = \ln \gamma_k(s', s) = \ln C_k + \frac{u_k L(u_k)}{2} + \frac{L_c}{2} \sum_{l=1}^n x_{kl} y_{kl}$
- ◆ $A_k(s) = \ln \alpha_k(s) = \max_{s'}^* [A_{k-1}(s') + \Gamma_k(s', s)]$
- ◆ $B_{k-1}(s') = \ln \beta_{k-1}(s') = \max_s^* [B_k(s) + \Gamma_k(s', s)]$

where $\max^*(a, b) = \begin{cases} \max(a, b) + \ln(1 + e^{-|a-b|}), & \text{log - MAP} \\ \max(a, b), & \text{max - log - MAP} \end{cases}$