On the Universality of the Block Markov Superposition Transmission

Speaker: Xiao Ma
maxiao@mail.sysu.edu.cn

Dept. Electronics and Comm. Eng.
Sun Yat-sen University,

Shenzhen, March, 2014

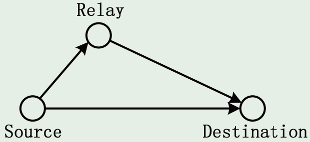
Outline

Outline

1. Superposition Block Markov Encoding in the Relay Channel
2. Block Markov Superposition Transmission
3. General Behavior of BMST
4. BMST with BPSK
5. BMST with High-order Modulations
6. BMST with Continuous phase modulation (CPM)
7. BMST with Spacial Modulation (SM)
8. Conclusions

Superposition Block Markov Encoding in the Relay Channel

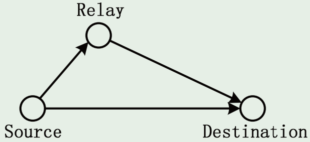
Gaussian Relay Channel Model



- one sender and one receiver with a number of relays;
- relays help the communication from the sender to the receiver.

Superposition Block Markov Encoding in the Relay Channel

Gaussian Relay Channel Model



- one sender and one receiver with a number of relays;
- relays help the communication from the sender to the receiver.

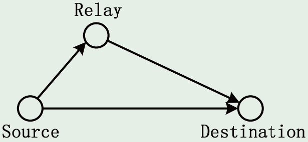
- A sender X and an ultimate intended receiver Y;
- The Gaussian relay channel is given by

$$\begin{aligned} Y_1 &= X + Z_1 \\ Y &= X + X_1 + Z_2, \end{aligned}$$

where Z_1 and Z_2 are independent zero-mean Gaussian random variables with variance N_1 and N_2, respectively;

Superposition Block Markov Encoding in the Relay Channel

Gaussian Relay Channel Model



- one sender and one receiver with a number of relays;
- relays help the communication from the sender to the receiver.

- A sender X and an ultimate intended receiver Y;
- The Gaussian relay channel is given by

$$\begin{aligned} Y_1 &= X + Z_1 \\ Y &= X + X_1 + Z_2, \end{aligned}$$

where Z_1 and Z_2 are independent zero-mean Gaussian random variables with variance N_1 and N_2, respectively;

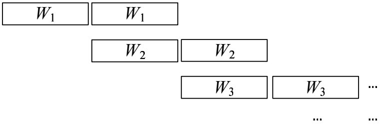
Achievable rate of the Gaussian Relay Channel

The decode-forward achievable rate is

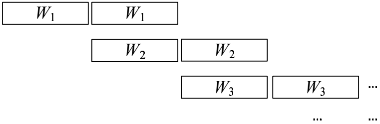
$$C = \max_{0 \leq \alpha \leq 1} \min\left\{C\left(\frac{P + P_1 + 2\sqrt{\overline{\alpha}PP_1}}{N_1 + N_2}\right), C\left(\frac{\alpha P}{N_1}\right)\right\},$$

where $\overline{\alpha} = 1 - \alpha$.

Superposition Block Markov Encoding (SBME) - Decode-Forward (DF) Strategy

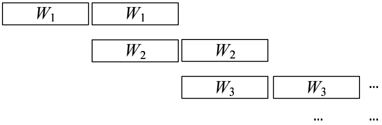


Superposition Block Markov Encoding (SBME) - Decode-Forward (DF) Strategy

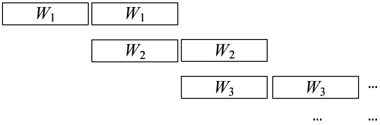


- The data are equally grouped into B blocks;

Superposition Block Markov Encoding (SBME) - Decode-Forward (DF) Strategy

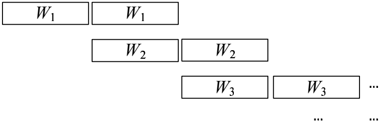


- The data are equally grouped into B blocks;
- Initially, the source (S) broadcasts a codeword that corresponds to the first data block to the relay (R) and the destination (D). Since the code rate is greater than the capacity of the link $\mathrm{S} \to \mathrm{D}$ (otherwise, no relay is required), D is not able to recover reliably this data block;



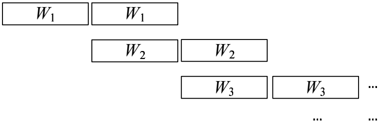
- The data are equally grouped into B blocks;
- Initially, the source (S) broadcasts a codeword that corresponds to the first data block to the relay (R) and the destination (D). Since the code rate is greater than the capacity of the link $\mathrm{S} \rightarrow \mathrm{D}$ (otherwise, no relay is required), D is not able to recover reliably this data block;
- Then the source and the relay cooperatively transmit more information about the first data block;

Superposition Block Markov Encoding (SBME) - Decode-Forward (DF) Strategy



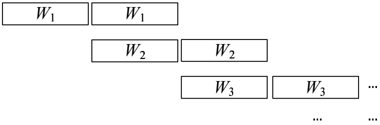
- The data are equally grouped into B blocks;
- Initially, the source (S) broadcasts a codeword that corresponds to the first data block to the relay (R) and the destination (D). Since the code rate is greater than the capacity of the link $\mathrm{S} \rightarrow \mathrm{D}$ (otherwise, no relay is required), D is not able to recover reliably this data block;
- Then the source and the relay cooperatively transmit more information about the first data block;
- In the meanwhile, the source "superimposes" a codeword that corresponds to the second data block;

Superposition Block Markov Encoding (SBME) - Decode-Forward (DF) Strategy

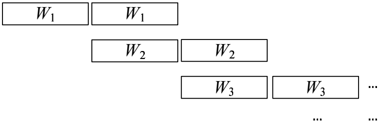


- The data are equally grouped into B blocks;
- Initially, the source (S) broadcasts a codeword that corresponds to the first data block to the relay (R) and the destination (D). Since the code rate is greater than the capacity of the link $\mathrm{S} \rightarrow \mathrm{D}$ (otherwise, no relay is required), D is not able to recover reliably this data block;
- Then the source and the relay cooperatively transmit more information about the first data block;
- In the meanwhile, the source "superimposes" a codeword that corresponds to the second data block;
- Finally, the destination recovers (reliably) the first data block from the two successive received blocks;

Superposition Block Markov Encoding (SBME) - Decode-Forward (DF) Strategy



- The data are equally grouped into B blocks;
- Initially, the source (S) broadcasts a codeword that corresponds to the first data block to the relay (R) and the destination (D). Since the code rate is greater than the capacity of the link $\mathrm{S} \to \mathrm{D}$ (otherwise, no relay is required), D is not able to recover reliably this data block;
- Then the source and the relay cooperatively transmit more information about the first data block;
- In the meanwhile, the source "superimposes" a codeword that corresponds to the second data block;
- Finally, the destination recovers (reliably) the first data block from the two successive received blocks;
- After removing the effect of the first data block, the system returns to the initial state;



- The data are equally grouped into B blocks;
- Initially, the source (S) broadcasts a codeword that corresponds to the first data block to the relay (R) and the destination (D). Since the code rate is greater than the capacity of the link $\mathrm{S} \rightarrow \mathrm{D}$ (otherwise, no relay is required), D is not able to recover reliably this data block;
- Then the source and the relay cooperatively transmit more information about the first data block;
- In the meanwhile, the source "superimposes" a codeword that corresponds to the second data block;
- Finally, the destination recovers (reliably) the first data block from the two successive received blocks;
- After removing the effect of the first data block, the system returns to the initial state;
- This process iterates $B+1$ times until all B blocks of data are sent successfully.

Superposition Block Markov Encoding (SBME) in the Relay Channel

- The SBME is a powerful technique in the multiuser information-theoretic field;

Superposition Block Markov Encoding (SBME) in the Relay Channel

- The SBME is a powerful technique in the multiuser information-theoretic field;
- Can we apply a multiuser technique to single-user systems?

Superposition Block Markov Encoding (SBME) in the Relay Channel

- The SBME is a powerful technique in the multiuser information-theoretic field;
- Can we apply a multiuser technique to single-user systems?
- It is possible. Actually, we have ever shown how to design bandwidth-efficient coded modulation by the use of "multiple-access signalling" together with the successive decoding [See, for example, Xiao Ma and Li Ping 2004: Coded Modulation Using Superimposed Binary Codes];

Superposition Block Markov Encoding (SBME) in the Relay Channel

- The SBME is a powerful technique in the multiuser information-theoretic field;
- Can we apply a multiuser technique to single-user systems?
- It is possible. Actually, we have ever shown how to design bandwidth-efficient coded modulation by the use of "multiple-access signalling" together with the successive decoding [See, for example, Xiao Ma and Li Ping 2004: Coded Modulation Using Superimposed Binary Codes];
- We apply a similar strategy (SBME) to the single-user communication system, resulting in the block Markov superposition transmission (BMST) scheme.

Outline

Short Codes

Short Convolutional Codes

Convolutional codes with short constraint lengths: e.g.,

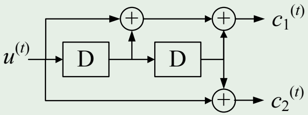


Figure: A $(2, 1, 2)$ convolutional code encoder.

Short Codes

Short Convolutional Codes

Convolutional codes with short constraint lengths: e.g.,

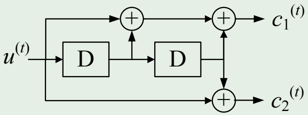


Figure: A $(2, 1, 2)$ convolutional code encoder.

Short Block Codes

Block codes with short length: repetition codes, single parity-check codes, Hamming codes, etc. We are actually interested in Cartesian product of short block codes. For example $[2, 1, 2]^{5000}$, $[6, 5, 2]^{2000}$, $[7, 4, 3]^{2500}$;

$[7, 4, 3]^{2500}$: Suppose that we intend to transmit 10000 bits using Hamming $[7, 4, 3]$ code. We first group the bits into sub-blocks of length 4 and then encode (independently) each sub-block into a sub-block of length 7.

Short Codes

Short Convolutional Codes

Convolutional codes with short constraint lengths: e.g.,

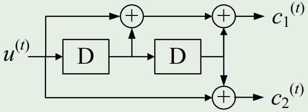


Figure: A $(2, 1, 2)$ convolutional code encoder.

Short Block Codes

Block codes with short length: repetition codes, single parity-check codes, Hamming codes, etc. We are actually interested in Cartesian product of short block codes. For example $[2, 1, 2]^{5000}$, $[6, 5, 2]^{2000}$, $[7, 4, 3]^{2500}$;

$[7, 4, 3]^{2500}$: Suppose that we intend to transmit 10000 bits using Hamming $[7, 4, 3]$ code. We first group the bits into sub-blocks of length 4 and then encode (independently) each sub-block into a sub-block of length 7.

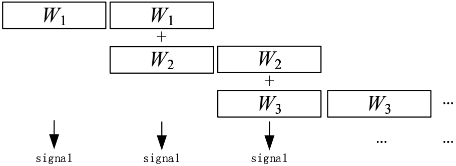
Actually, short codes can be any code that has fast encoding algorithm and soft-in soft-out (SISO) decoding algorithm.

Block Markov Superposition Transmission

Let $\mathscr{C}$ be the short code (called *basic code*) in the transmission scheme.

Block Markov Superposition Transmission

Let $\mathscr{C}$ be the short code (called *basic code*) in the transmission scheme.

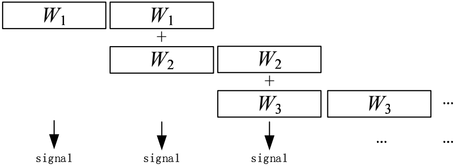


BMST Scheme

1. The data are equally grouped into B blocks;

Block Markov Superposition Transmission

Let $\mathscr{C}$ be the short code (called *basic code*) in the transmission scheme.

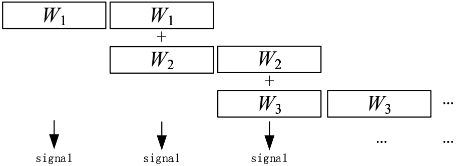


BMST Scheme

1. The data are equally grouped into B blocks;
2. Initially, the transmitter sends a codeword from $\mathscr{C}$ that corresponds to the first data block;

Block Markov Superposition Transmission

Let $\mathscr{C}$ be the short code (called *basic code*) in the transmission scheme.

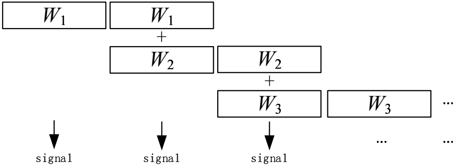


BMST Scheme

1. The data are equally grouped into B blocks;
2. Initially, the transmitter sends a codeword from $\mathscr{C}$ that corresponds to the first data block;
3. Since the short code is weak, the receiver is unable to recover reliably the data from the current received block. Hence the transmitter transmits the codeword (possibly in its interleaved version) one more time.

Block Markov Superposition Transmission

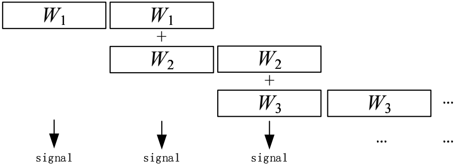
Let $\mathscr{C}$ be the short code (called *basic code*) in the transmission scheme.



BMST Scheme

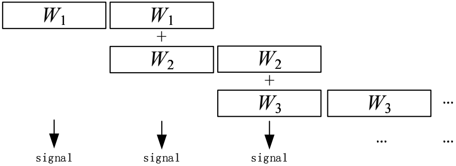
1. The data are equally grouped into B blocks;
2. Initially, the transmitter sends a codeword from $\mathscr{C}$ that corresponds to the first data block;
3. Since the short code is weak, the receiver is unable to recover reliably the data from the current received block. Hence the transmitter transmits the codeword (possibly in its interleaved version) one more time.
4. In the meanwhile, a fresh codeword from $\mathscr{C}$ that corresponds to the second data block is superimposed on the second block transmission.

Block Markov Superposition Transmission



BMST Scheme (Continued)

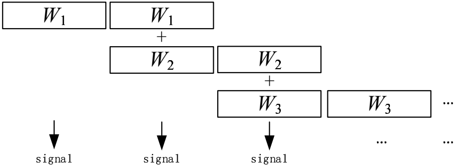
Block Markov Superposition Transmission



BMST Scheme (Continued)

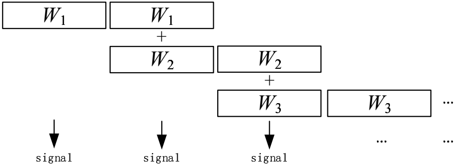
5. Finally, the receiver recovers the first data block from the two successive received blocks.

Block Markov Superposition Transmission



BMST Scheme (Continued)

5. Finally, the receiver recovers the first data block from the two successive received blocks.
6. After removing the effect of the first data block, the system returns to the initial state;



BMST Scheme (Continued)

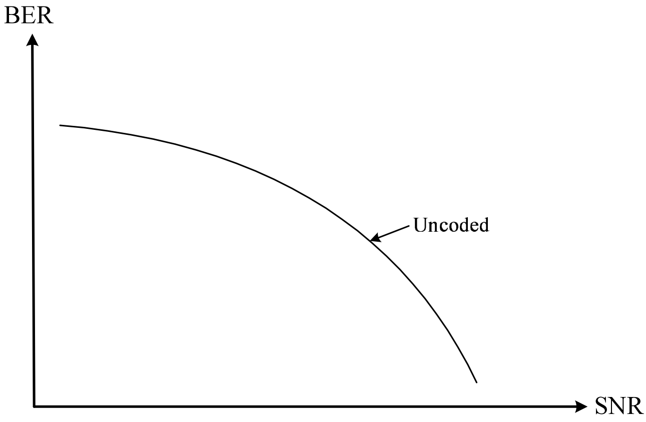
5. Finally, the receiver recovers the first data block from the two successive received blocks.
6. After removing the effect of the first data block, the system returns to the initial state;
7. This process iterates $B + 1$ times until all B blocks of data are sent successfully.

- Repetition increases reliability.
- Superposition keeps rate unchanged.

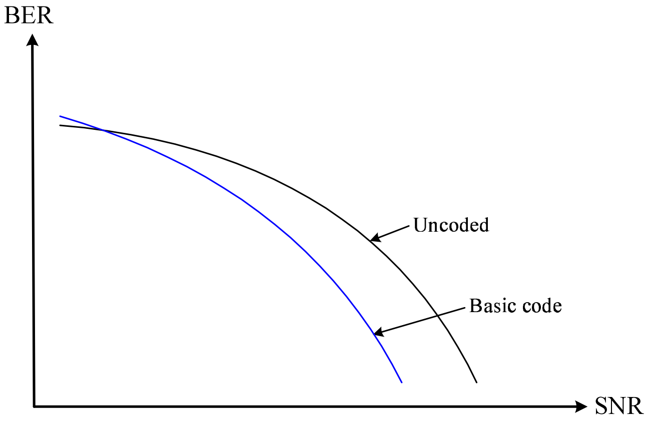
Outline

What does the performance curve look like?

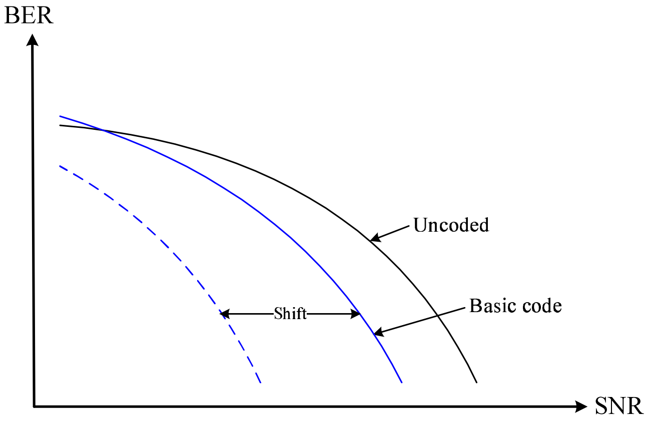
What does the performance curve look like?



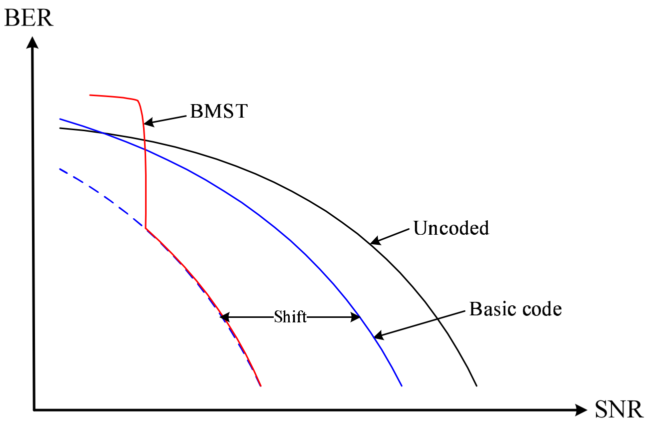
What does the performance curve look like?



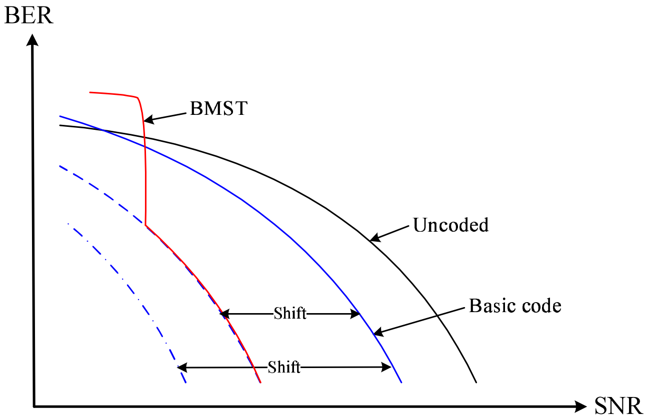
What does the performance curve look like?



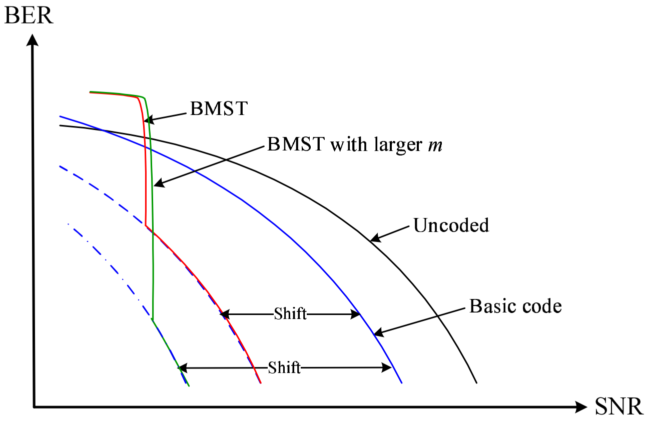
What does the performance curve look like?



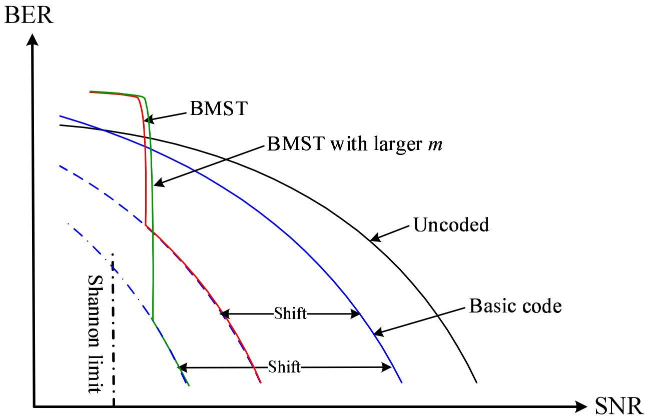
What does the performance curve look like?



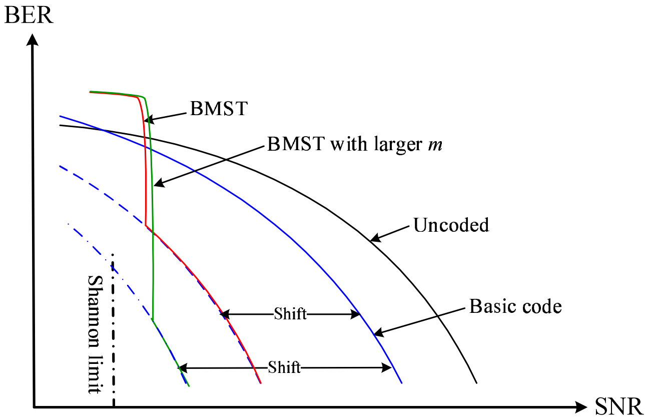
What does the performance curve look like?



What does the performance curve look like?



What does the performance curve look like?



The maximal coding gain for a BMST system with memory m compared with the basic code can be $10\log_{10}(m+1)$.

Outline

BMST with BPSK over AWGN channels

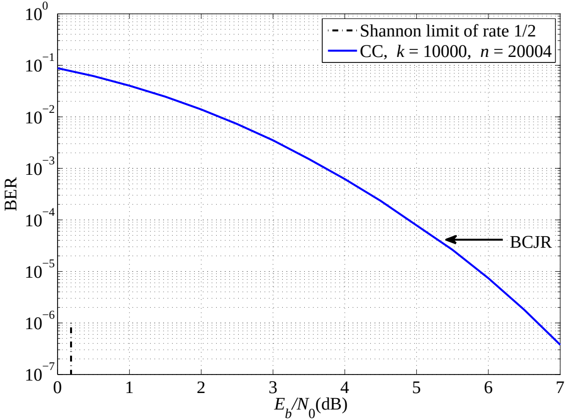


Figure: The basic code is a terminated 4-state $(2,1,2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

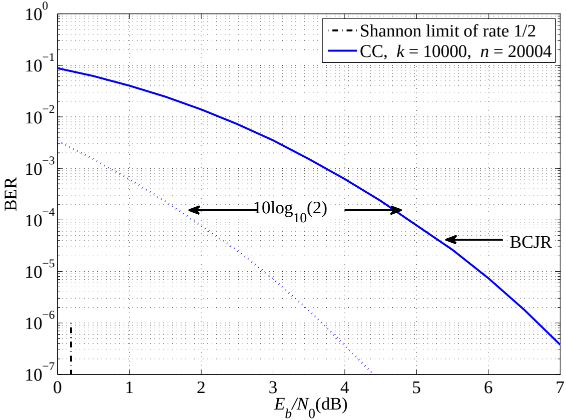


Figure: The basic code is a terminated 4-state $(2,1,2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

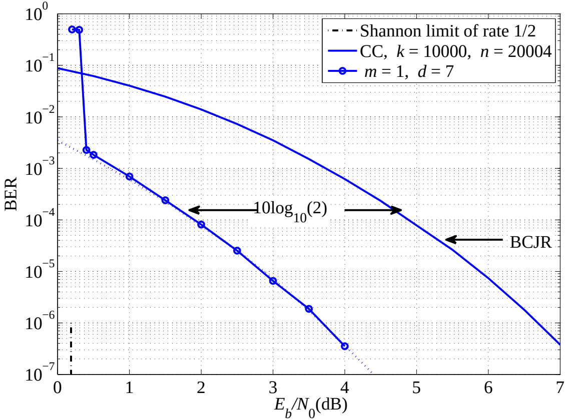


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

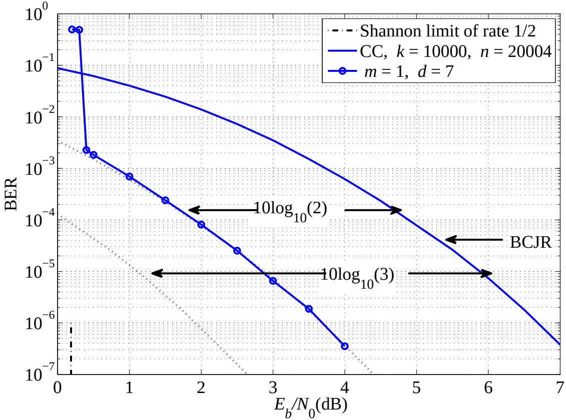


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

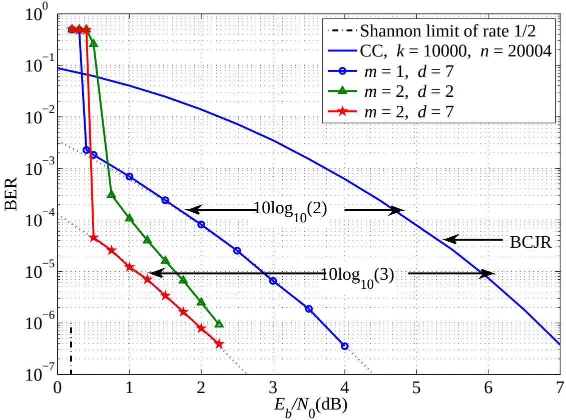


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

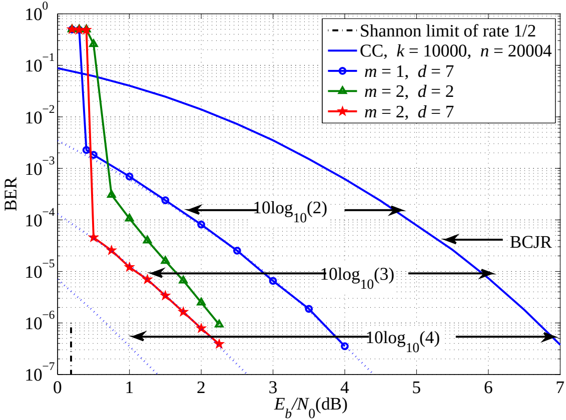


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

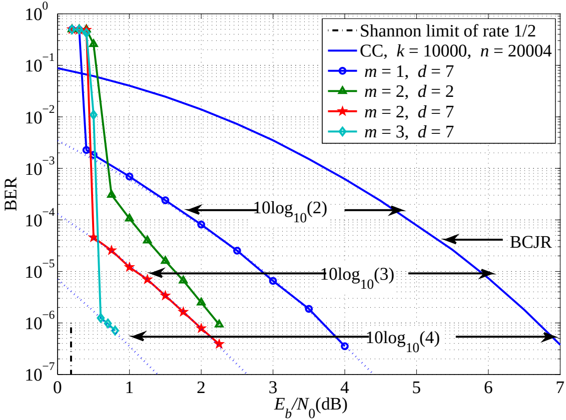


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

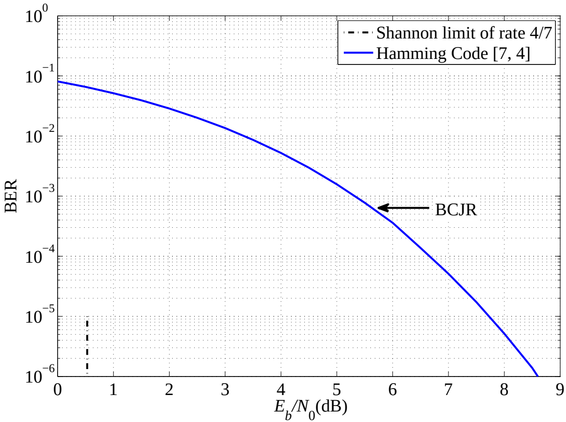


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

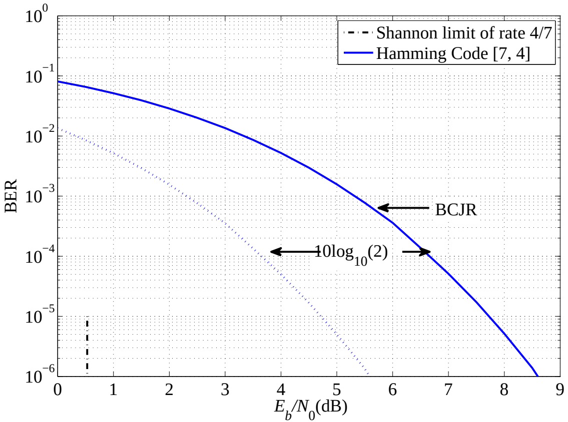


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

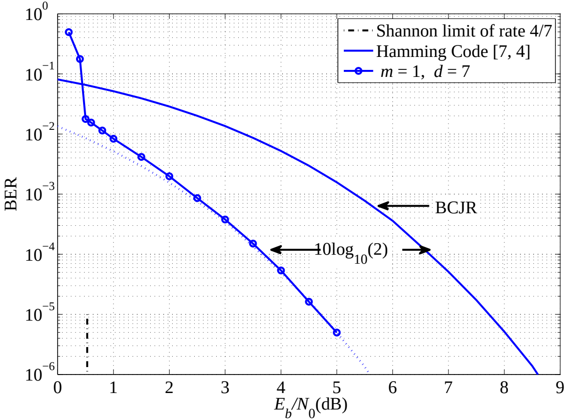


Figure: The basic code is the Cartesian product of Hamming code $[7, 4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

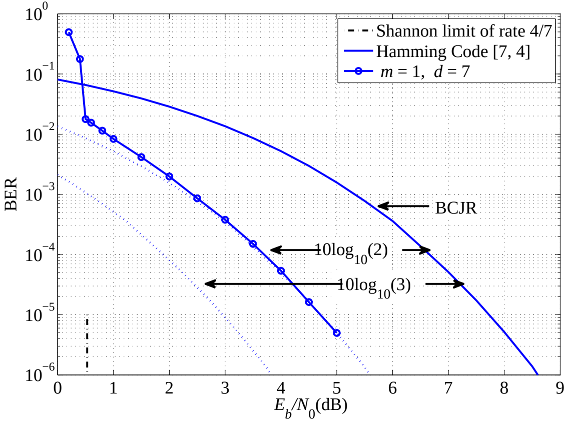


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

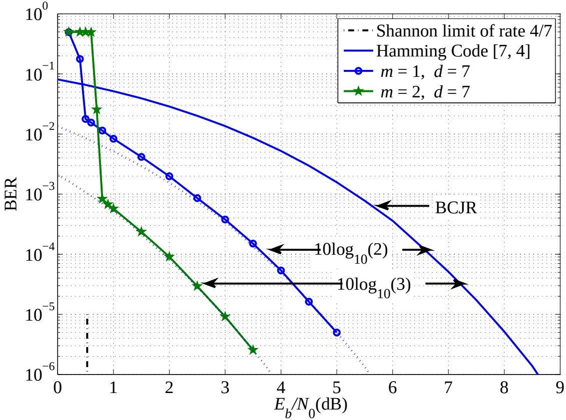


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

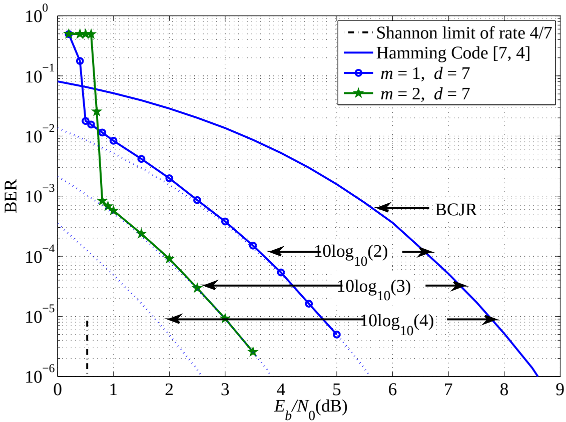


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

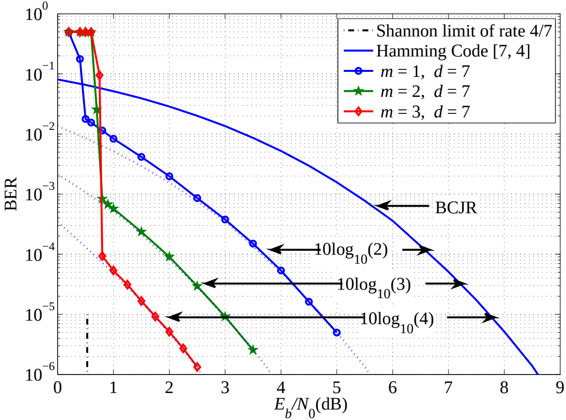


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

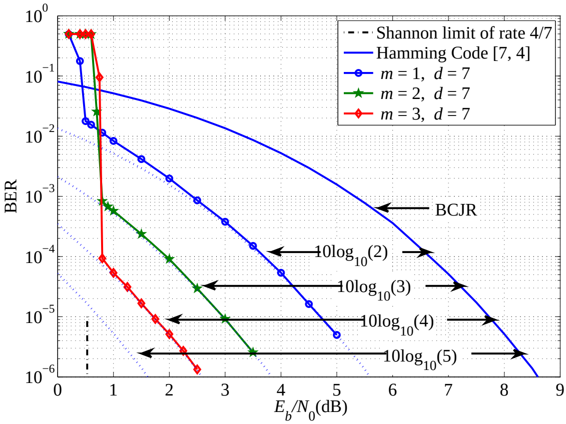


Figure: The basic code is the Cartesian product of Hamming code $[7, 4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with BPSK over AWGN channels

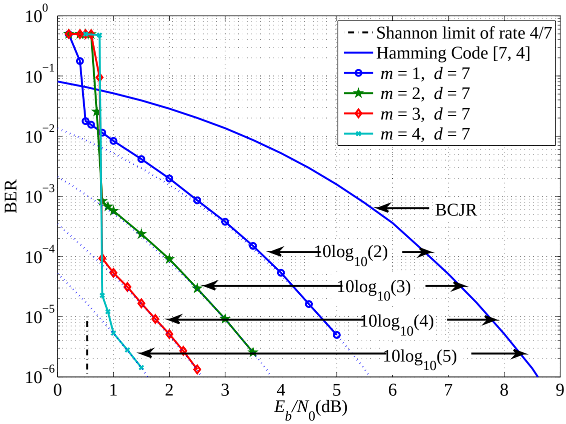


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with different code rates using BPSK over AWGN channels

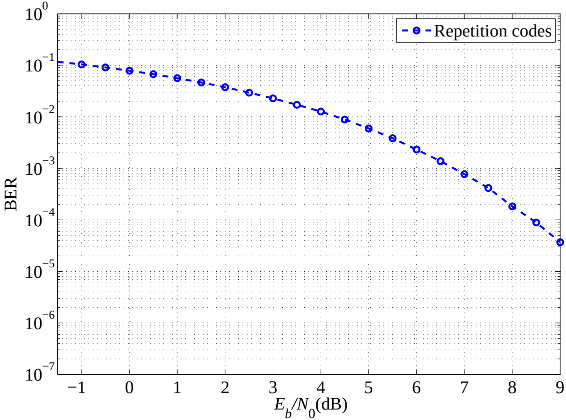


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

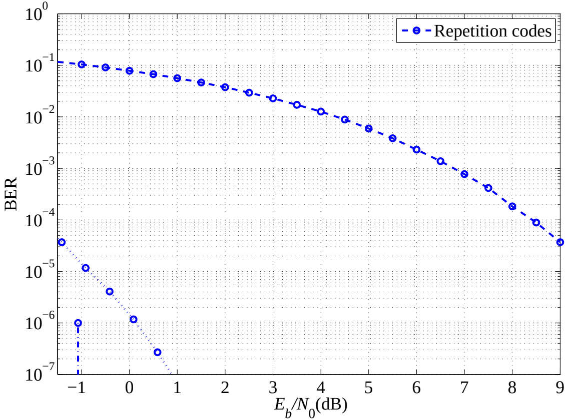


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

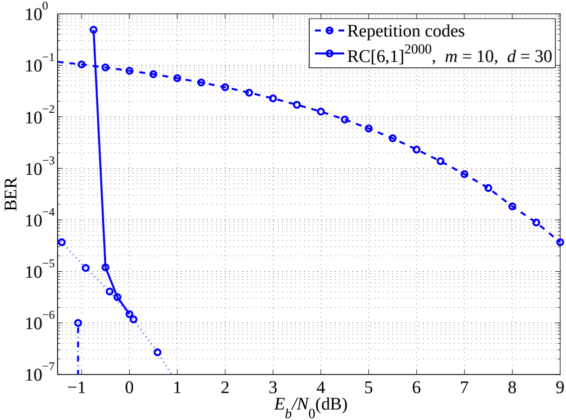


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

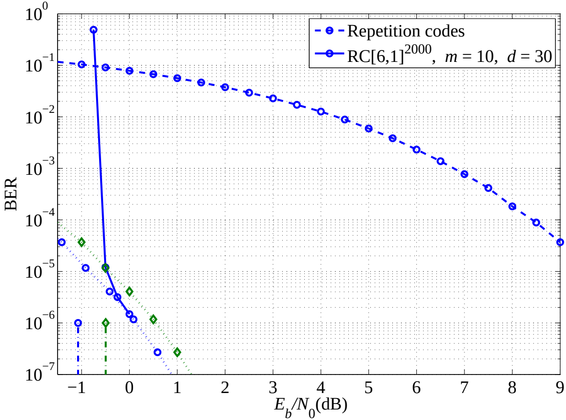


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

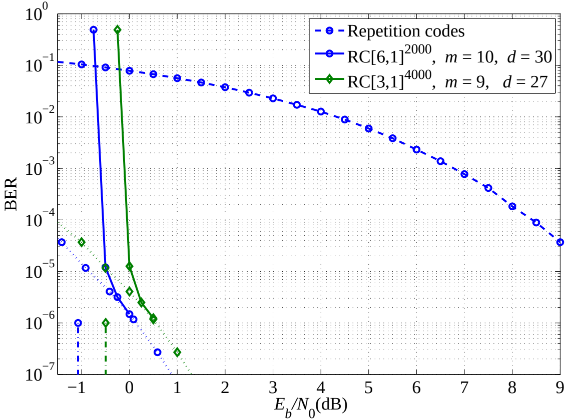


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

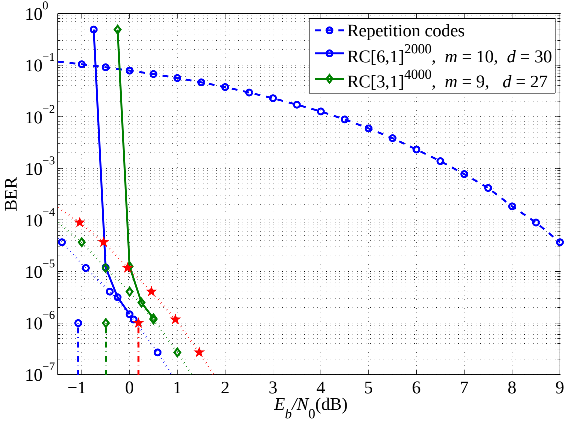


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

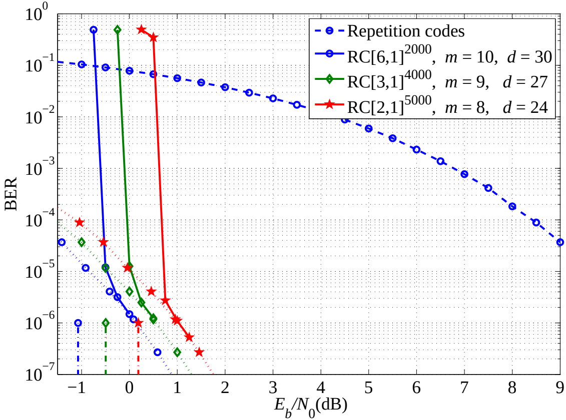


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

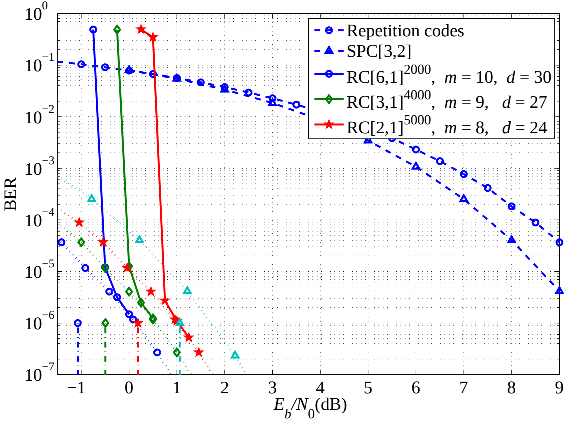


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

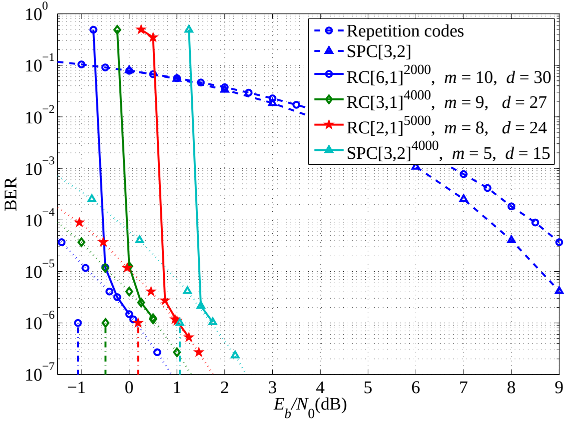


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

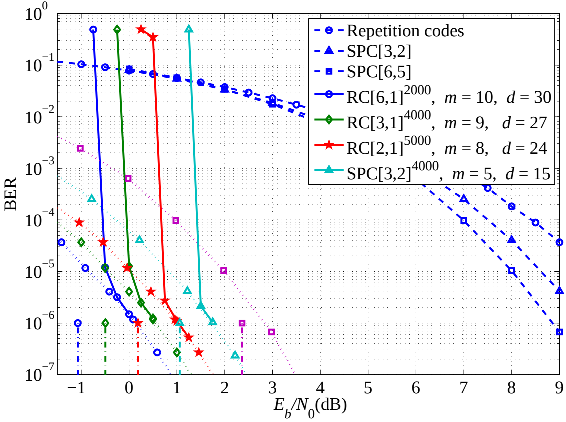


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with different code rates using BPSK over AWGN channels

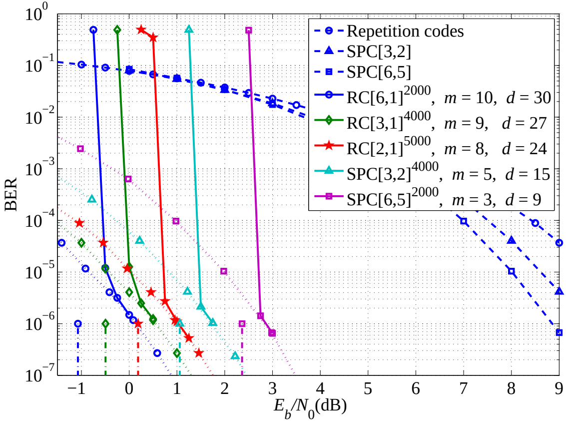


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends. The vertical dashed lines correspond to the respective Shannon limits.

BMST with Nonlinear Basic Codes

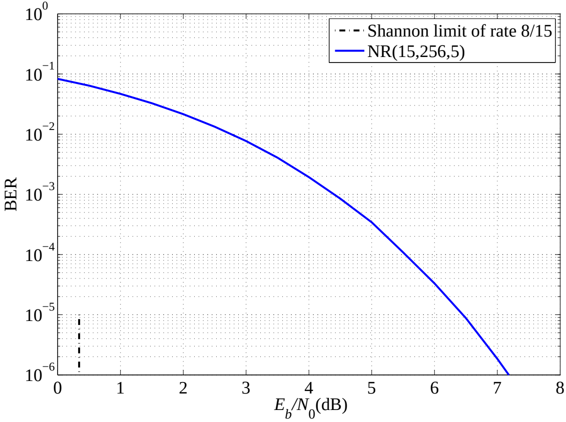


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with Nonlinear Basic Codes

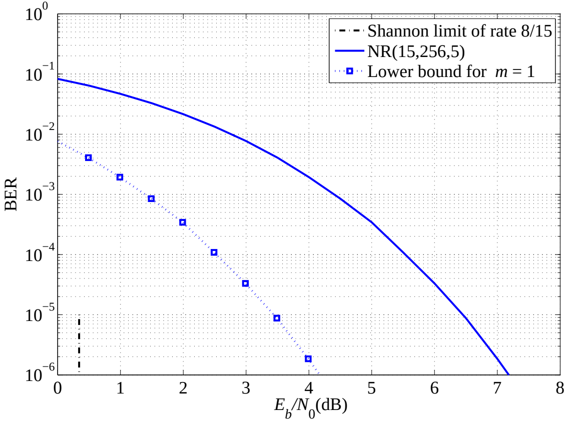


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with Nonlinear Basic Codes

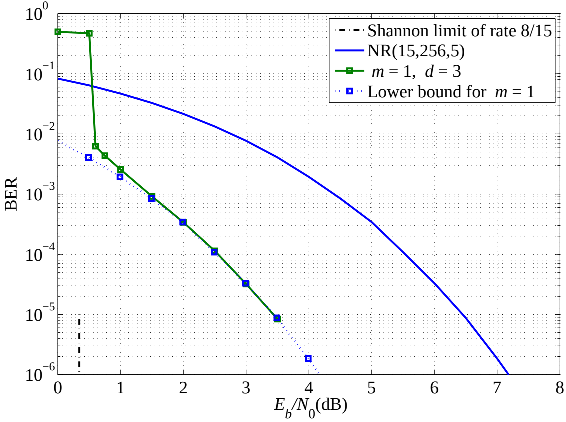


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with Nonlinear Basic Codes

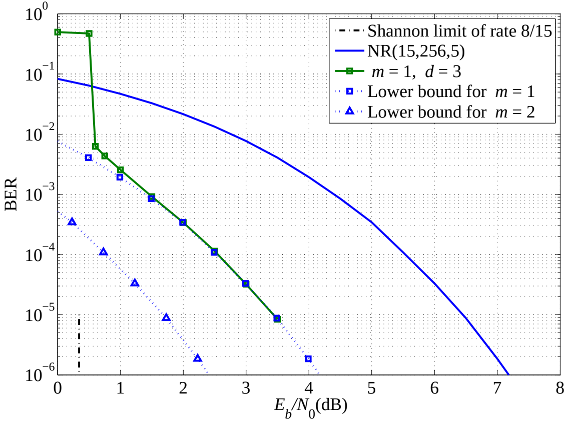


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with Nonlinear Basic Codes

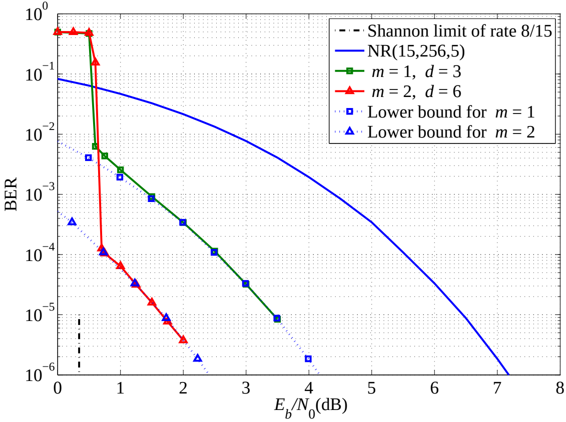


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with Nonlinear Basic Codes

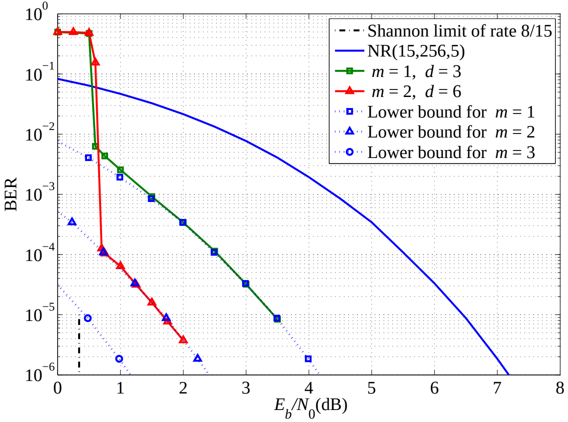


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with Nonlinear Basic Codes

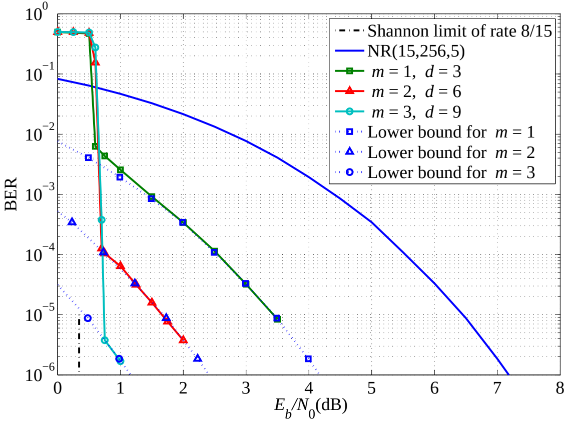


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with Long Basic Codes

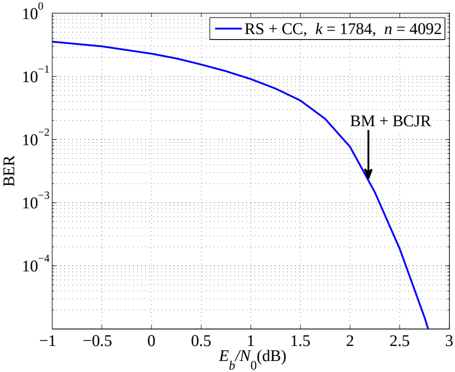


Figure: The basic code $\mathscr{C}$ is the Consultative Committee on Space Data System (CCSDS) standard code of dimension $k = 1784$ and length $n = 4092$, where the outer code is a [255, 223] Reed-Solomon (RS) code over $\mathbb{F}_{256}$ and the inner code is a terminated convolutional code with the polynomial generator matrix $G(D) = [1 + D + D^2 + D^3 + D^6,\ 1 + D^2 + D^3 + D^5 + D^6]$. Other coding parameters of the BMST system are $L = 100$ and $I_{\text{max}} = 18$.

BMST with Long Basic Codes

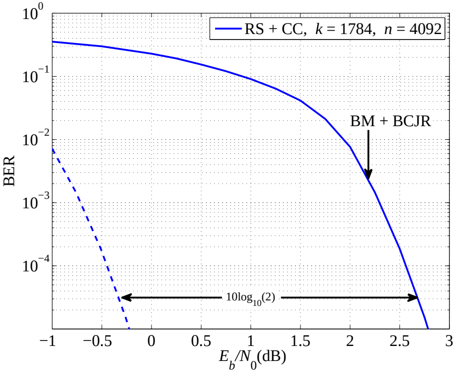


Figure: The basic code $\mathscr{C}$ is the Consultative Committee on Space Data System (CCSDS) standard code of dimension $k = 1784$ and length $n = 4092$, where the outer code is a [255, 223] Reed-Solomon (RS) code over $\mathbb{F}_{256}$ and the inner code is a terminated convolutional code with the polynomial generator matrix $G(D) = [1 + D + D^2 + D^3 + D^6,\ 1 + D^2 + D^3 + D^5 + D^6]$. Other coding parameters of the BMST system are $L = 100$ and $I_{\text{max}} = 18$.

BMST with Long Basic Codes

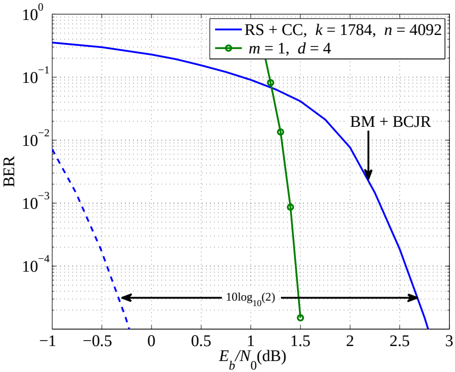


Figure: The basic code $\mathscr{C}$ is the Consultative Committee on Space Data System (CCSDS) standard code of dimension $k = 1784$ and length $n = 4092$, where the outer code is a [255, 223] Reed-Solomon (RS) code over $\mathbb{F}_{256}$ and the inner code is a terminated convolutional code with the polynomial generator matrix $G(D) = [1 + D + D^2 + D^3 + D^6,\ 1 + D^2 + D^3 + D^5 + D^6]$. Other coding parameters of the BMST system are $L = 100$ and $I_{\max} = 18$.

A General Procedure to Design BMST

Given a target code rate R_{target} and a target BER p_{target}, the general procedure for designing a BMST system to approach the Shannon limit is as follows.

A General Procedure to Design BMST

Given a target code rate R_{target} and a target BER p_{target}, the general procedure for designing a BMST system to approach the Shannon limit is as follows.

- Find k and n as small as possible such that $R_{\text{target}} = k/n$.

A General Procedure to Design BMST

Given a target code rate R_{target} and a target BER p_{target}, the general procedure for designing a BMST system to approach the Shannon limit is as follows.

- Find k and n as small as possible such that $R_{\text{target}} = k/n$.
- Find a code $\mathscr{C}[n, k]$, which can be linear, nonlinear or even randomly generated.

A General Procedure to Design BMST

Given a target code rate R_{target} and a target BER p_{target}, the general procedure for designing a BMST system to approach the Shannon limit is as follows.

- Find k and n as small as possible such that $R_{\text{target}} = k/n$.
- Find a code $\mathscr{C}[n, k]$, which can be linear, nonlinear or even randomly generated.
- Find the performance curve $p_b = f_{\text{basic}}(\gamma_b)$ of the code $\mathscr{C}[n, k]$, where p_b is the BER and $\gamma_b \triangleq E_b/N_0$ in dB.

A General Procedure to Design BMST

Given a target code rate R_{target} and a target BER p_{target}, the general procedure for designing a BMST system to approach the Shannon limit is as follows.

- Find k and n as small as possible such that $R_{\text{target}} = k/n$.
- Find a code $\mathscr{C}[n,k]$, which can be linear, nonlinear or even randomly generated.
- Find the performance curve $p_b = f_{\text{basic}}(\gamma_b)$ of the code $\mathscr{C}[n,k]$, where p_b is the BER and $\gamma_b \triangleq E_b/N_0$ in dB.
- From the performance curve, find the required E_b/N_0 to achieve the target BER. That is, find γ_{target} such that $f_{\text{basic}}(\gamma_{\text{target}}) \leq p_{\text{target}}$;

A General Procedure to Design BMST

Given a target code rate R_{target} and a target BER p_{target}, the general procedure for designing a BMST system to approach the Shannon limit is as follows.

- Find k and n as small as possible such that $R_{\text{target}} = k/n$.
- Find a code $\mathscr{C}[n,k]$, which can be linear, nonlinear or even randomly generated.
- Find the performance curve $p_b = f_{\text{basic}}(\gamma_b)$ of the code $\mathscr{C}[n,k]$, where p_b is the BER and $\gamma_b \triangleq E_b/N_0$ in dB.
- From the performance curve, find the required E_b/N_0 to achieve the target BER. That is, find γ_{target} such that $f_{\text{basic}}(\gamma_{\text{target}}) \leq p_{\text{target}}$;
- Find the Shannon limit for the code rate R, denoted by $\gamma_{\lim}$;

A General Procedure to Design BMST

Given a target code rate R_{target} and a target BER p_{target}, the general procedure for designing a BMST system to approach the Shannon limit is as follows.

- Find k and n as small as possible such that $R_{\text{target}} = k/n$.
- Find a code $\mathscr{C}[n, k]$, which can be linear, nonlinear or even randomly generated.
- Find the performance curve $p_b = f_{\text{basic}}(\gamma_b)$ of the code $\mathscr{C}[n, k]$, where p_b is the BER and $\gamma_b \triangleq E_b/N_0$ in dB.
- From the performance curve, find the required E_b/N_0 to achieve the target BER. That is, find γ_{target} such that $f_{\text{basic}}(\gamma_{\text{target}}) \le p_{\text{target}}$;
- Find the Shannon limit for the code rate R, denoted by $\gamma_{\lim}$;
- Determine the encoding memory m by $10\log_{10}(m+1) \ge \gamma_{\text{target}} - \gamma_{\lim}$. That is, $m = \left\lceil 10^{\frac{\gamma_{\text{target}} - \gamma_{\lim}}{10}} - 1 \right\rceil$, where $\lceil x \rceil$ stands for the minimum integer greater than or equal to x;

A General Procedure to Design BMST

Given a target code rate R_{target} and a target BER p_{target}, the general procedure for designing a BMST system to approach the Shannon limit is as follows.

- Find k and n as small as possible such that $R_{\text{target}} = k/n$.
- Find a code $\mathscr{C}[n,k]$, which can be linear, nonlinear or even randomly generated.
- Find the performance curve $p_b = f_{\text{basic}}(\gamma_b)$ of the code $\mathscr{C}[n,k]$, where p_b is the BER and $\gamma_b \triangleq E_b/N_0$ in dB.
- From the performance curve, find the required E_b/N_0 to achieve the target BER. That is, find γ_{target} such that $f_{\text{basic}}(\gamma_{\text{target}}) \le p_{\text{target}}$;
- Find the Shannon limit for the code rate R, denoted by $\gamma_{\lim}$;
- Determine the encoding memory m by $10\log_{10}(m+1) \ge \gamma_{\text{target}} - \gamma_{\lim}$. That is, $m = \left\lceil 10^{\frac{\gamma_{\text{target}}-\gamma_{\lim}}{10}} - 1 \right\rceil$, where $\lceil x \rceil$ stands for the minimum integer greater than or equal to x;
- Take the B-fold Cartesian product of the code $\mathscr{C}[n,k]^B$ as the basic code. To approach the Shannon limit, we set $nB \ge 10000$.

A General Procedure to Design BMST

Given a target code rate R_{target} and a target BER p_{target}, the general procedure for designing a BMST system to approach the Shannon limit is as follows.

- Find k and n as small as possible such that $R_{\text{target}} = k/n$.
- Find a code $\mathscr{C}[n,k]$, which can be linear, nonlinear or even randomly generated.
- Find the performance curve $p_b = f_{\text{basic}}(\gamma_b)$ of the code $\mathscr{C}[n,k]$, where p_b is the BER and $\gamma_b \triangleq E_b/N_0$ in dB.
- From the performance curve, find the required E_b/N_0 to achieve the target BER. That is, find γ_{target} such that $f_{\text{basic}}(\gamma_{\text{target}}) \le p_{\text{target}}$;
- Find the Shannon limit for the code rate R, denoted by γ_{lim};
- Determine the encoding memory m by $10\log_{10}(m+1) \ge \gamma_{\text{target}} - \gamma_{\text{lim}}$. That is, $m = \left\lceil 10^{\frac{\gamma_{\text{target}}-\gamma_{\text{lim}}}{10}} - 1 \right\rceil$, where $\lceil x \rceil$ stands for the minimum integer greater than or equal to x;
- Take the B-fold Cartesian product of the code $\mathscr{C}[n,k]^B$ as the basic code. To approach the Shannon limit, we set $nB \ge 10000$.
- Generate $m+1$ interleavers randomly.

A Construction Example

- For a target code rate $R_{\mathrm{target}} = 0.5$ and different target BERs p_{target}, the repetition code $[2,1](k=1, n=2)$ with $0 \to 00$ and $1 \to 11$ is chosen.
- The encoding memories required to approach the Shannon limit using the BMST of RC $[2,1]^B$ is listed in the table.

p_{target}	3×10^{-3}	10^{-5}	10^{-15}
γ_{target} (dB)	5.78	9.59	14.99
γ_{lim} (dB)	0.19	0.19	0.19
Gap (dB)	6.59	9.40	14.80
m	3	8	30

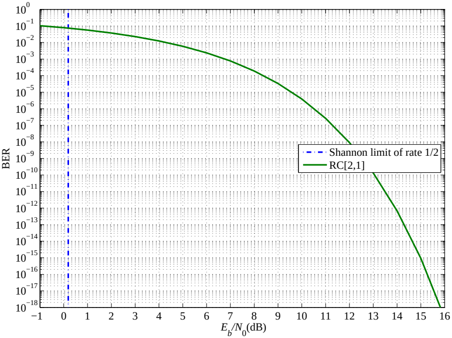


Figure: The basic code is the 5000-fold Cartesian product of the repetition code $[2,1]^{5000}$. The system encodes $L = 100000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$.

A Construction Example

- For a target code rate $R_{\text{target}} = 0.5$ and different target BERs p_{target}, the repetition code $[2,1](k=1, n=2)$ with $0 \to 00$ and $1 \to 11$ is chosen.
- The encoding memories required to approach the Shannon limit using the BMST of RC $[2,1]^B$ is listed in the table.

p_{target}	3×10^{-3}	10^{-5}	10^{-15}
γ_{target} (dB)	5.78	9.59	14.99
γ_{lim} (dB)	0.19	0.19	0.19
Gap (dB)	6.59	9.40	14.80
m	3	8	30

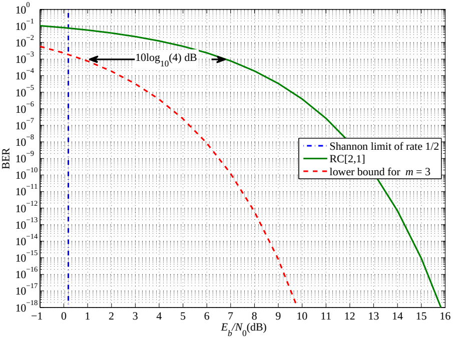


Figure: The basic code is the 5000-fold Cartesian product of the repetition code $[2,1]^{5000}$. The system encodes $L = 100000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$.

A Construction Example

- For a target code rate $R_{\text{target}} = 0.5$ and different target BERs p_{target}, the repetition code $[2,1](k=1, n=2)$ with $0 \to 00$ and $1 \to 11$ is chosen.
- The encoding memories required to approach the Shannon limit using the BMST of RC $[2,1]^B$ is listed in the table.

p_{target}	3×10^{-3}	10^{-5}	10^{-15}
γ_{target} (dB)	5.78	9.59	14.99
$\gamma_{\lim}$ (dB)	0.19	0.19	0.19
Gap (dB)	6.59	9.40	14.80
m	3	8	30

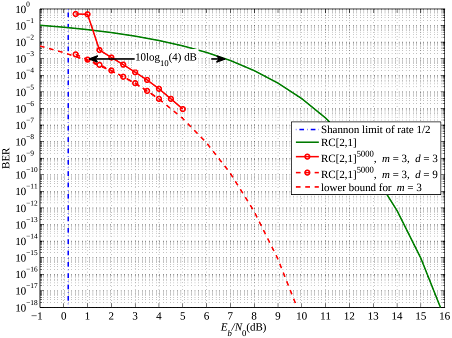


Figure: The basic code is the 5000-fold Cartesian product of the repetition code $[2,1]^{5000}$. The system encodes $L = 100000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$.

A Construction Example

- For a target code rate $R_{\text{target}} = 0.5$ and different target BERs p_{target}, the repetition code $[2,1](k=1, n=2)$ with $0 \to 00$ and $1 \to 11$ is chosen.
- The encoding memories required to approach the Shannon limit using the BMST of RC $[2,1]^B$ is listed in the table.

p_{target}	3×10^{-3}	10^{-5}	10^{-15}
γ_{target} (dB)	5.78	9.59	14.99
γ_{lim} (dB)	0.19	0.19	0.19
Gap (dB)	6.59	9.40	14.80
m	3	8	30

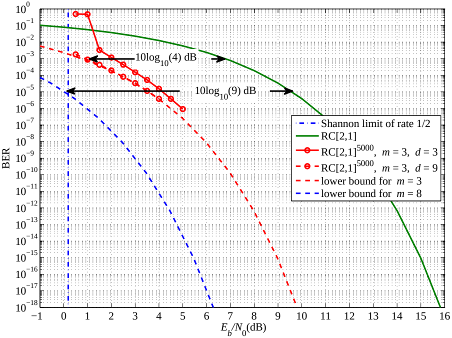


Figure: The basic code is the 5000-fold Cartesian product of the repetition code $[2,1]^{5000}$. The system encodes $L = 100000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$.

A Construction Example

- For a target code rate $R_{\text{target}} = 0.5$ and different target BERs p_{target}, the repetition code $[2,1](k=1, n=2)$ with $0 \to 00$ and $1 \to 11$ is chosen.
- The encoding memories required to approach the Shannon limit using the BMST of RC $[2,1]^B$ is listed in the table.

p_{target}	3×10^{-3}	10^{-5}	10^{-15}
γ_{target} (dB)	5.78	9.59	14.99
γ_{lim} (dB)	0.19	0.19	0.19
Gap (dB)	6.59	9.40	14.80
m	3	8	30

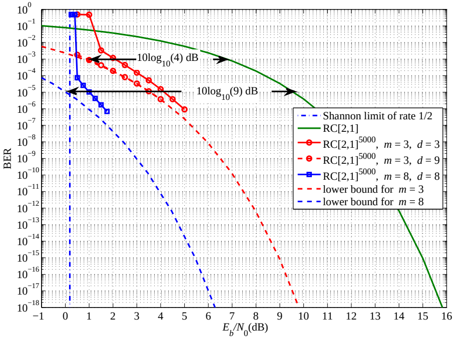


Figure: The basic code is the 5000-fold Cartesian product of the repetition code $[2,1]^{5000}$. The system encodes $L = 100000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$.

A Construction Example

- For a target code rate $R_{\text{target}} = 0.5$ and different target BERs p_{target}, the repetition code $[2,1](k=1, n=2)$ with $0 \to 00$ and $1 \to 11$ is chosen.
- The encoding memories required to approach the Shannon limit using the BMST of RC $[2,1]^B$ is listed in the table.

p_{target}	3×10^{-3}	10^{-5}	10^{-15}
γ_{target} (dB)	5.78	9.59	14.99
γ_{lim} (dB)	0.19	0.19	0.19
Gap (dB)	6.59	9.40	14.80
m	3	8	30

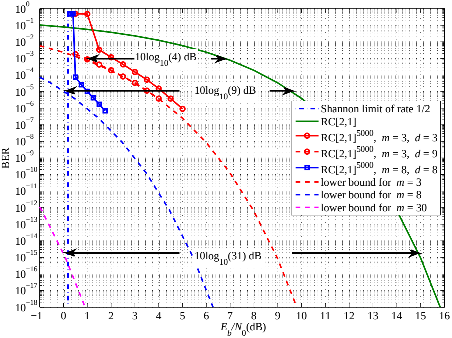


Figure: The basic code is the 5000-fold Cartesian product of the repetition code $[2,1]^{5000}$. The system encodes $L = 100000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$.

A Construction Example

- For a target code rate $R_{\text{target}} = 0.5$ and different target BERs p_{target}, the repetition code $[2,1](k = 1, n = 2)$ with $0 \rightarrow 00$ and $1 \rightarrow 11$ is chosen.
- The encoding memories required to approach the Shannon limit using the BMST of RC $[2,1]^B$ is listed in the table.

p_{target}	3×10^{-3}	10^{-5}	10^{-15}
γ_{target} (dB)	5.78	9.59	14.99
γ_{lim} (dB)	0.19	0.19	0.19
Gap (dB)	6.59	9.40	14.80
m	3	8	30

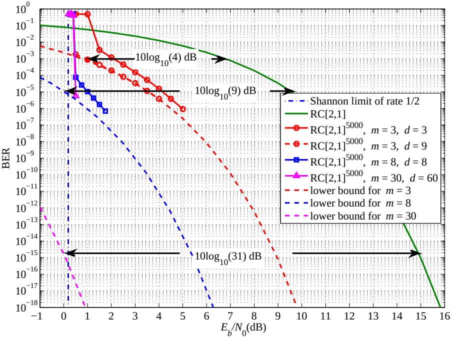


Figure: The basic code is the 5000-fold Cartesian product of the repetition code $[2,1]^{5000}$. The system encodes $L = 100000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$.

Outline

BMST Can also Combine with High-order Modulations

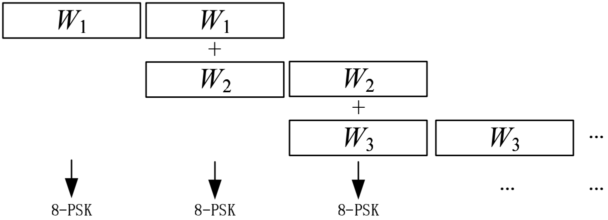


Figure: The BMST system with 8-PSK.

BMST with 8-PSK over AWGN channels

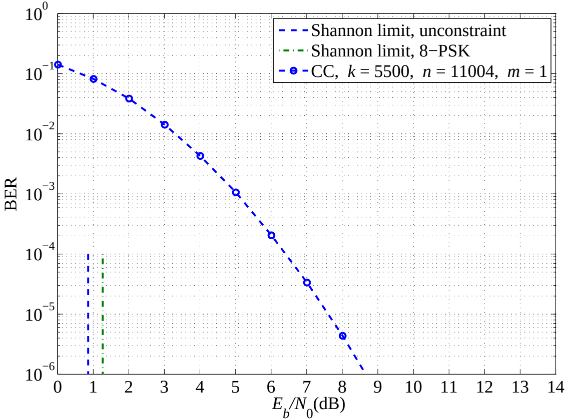


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over AWGN channels

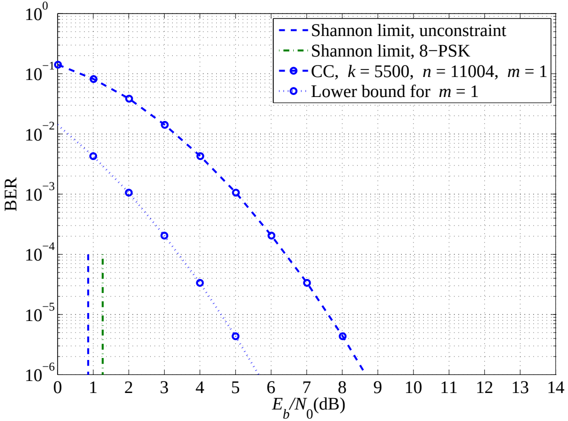


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over AWGN channels

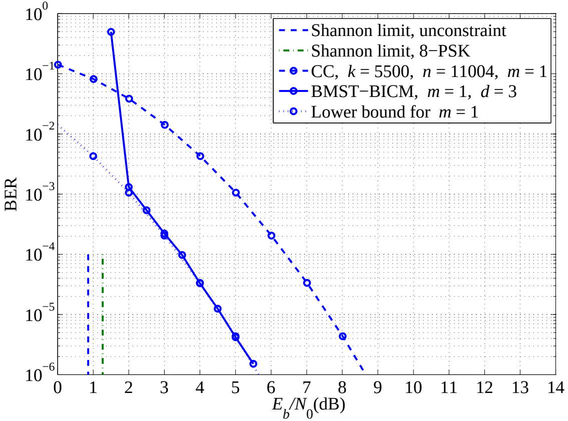


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over AWGN channels

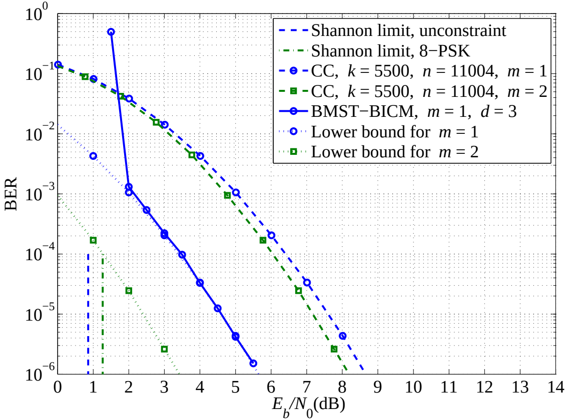


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over AWGN channels

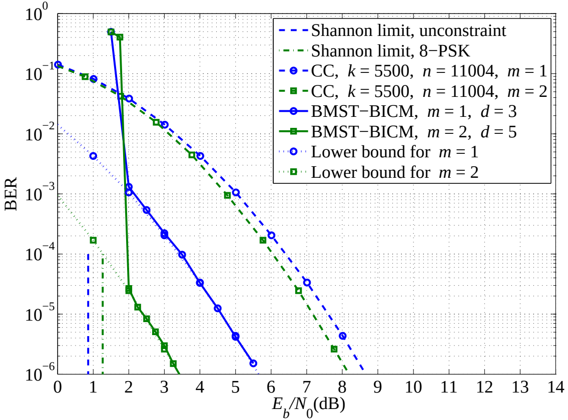


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over AWGN channels

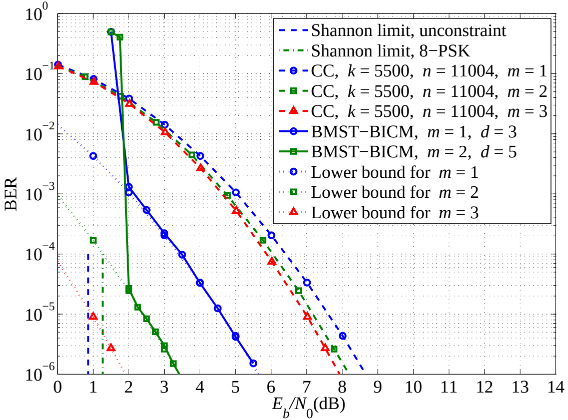


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over AWGN channels

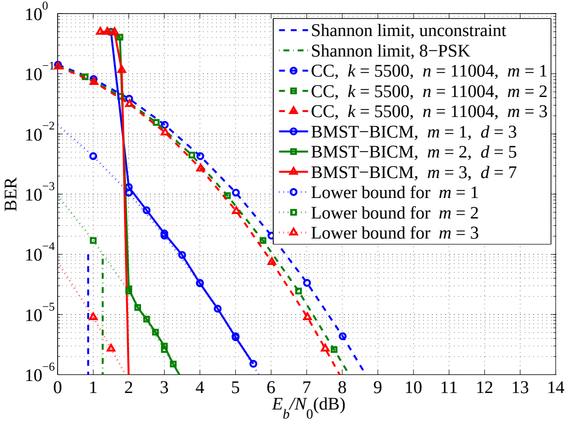


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over Rayleigh fading channels

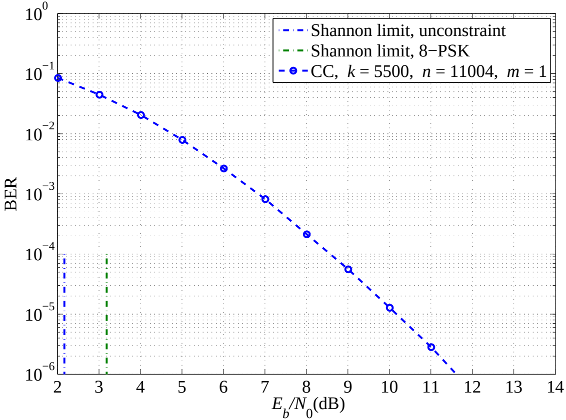


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over block Rayleigh fading channels with coherence period $B = 10$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over Rayleigh fading channels

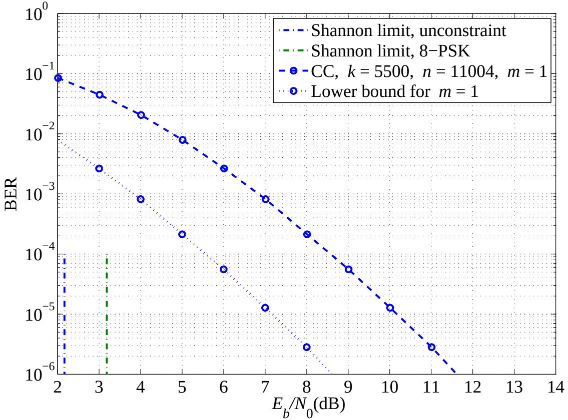


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over block Rayleigh fading channels with coherence period $B = 10$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over Rayleigh fading channels

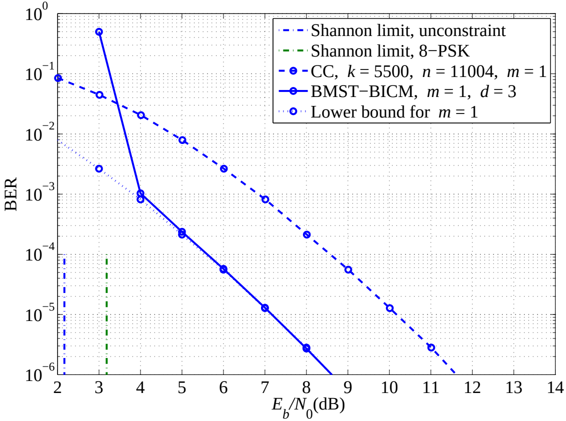


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over block Rayleigh fading channels with coherence period $B = 10$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over Rayleigh fading channels

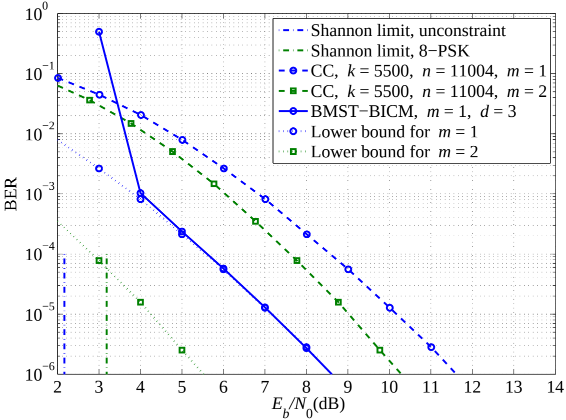


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over block Rayleigh fading channels with coherence period $B = 10$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over Rayleigh fading channels

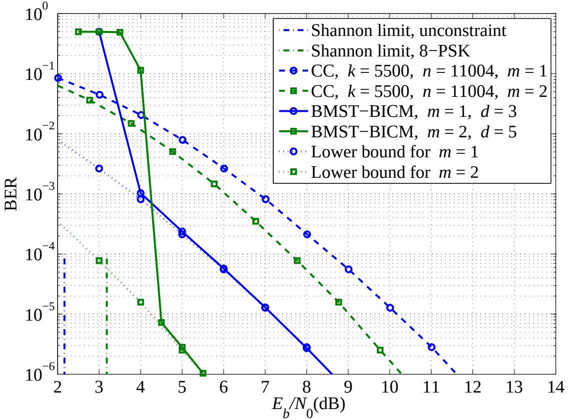


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over block Rayleigh fading channels with coherence period $B = 10$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over Rayleigh fading channels

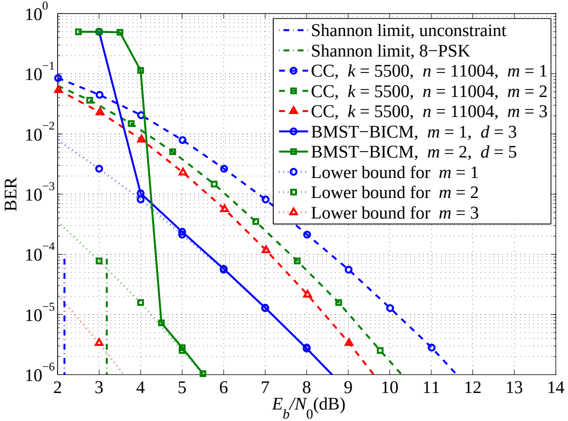


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over block Rayleigh fading channels with coherence period $B = 10$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

BMST with 8-PSK over Rayleigh fading channels

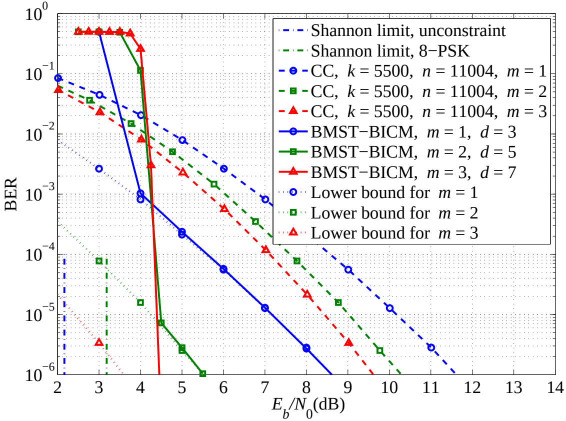


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 5500$ and $n = 11004$. Signals are transmitted using 8-PSK modulation with Gray mapping over block Rayleigh fading channels with coherence period $B = 10$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

Outline

BMST with Continuous phase modulation (CPM)

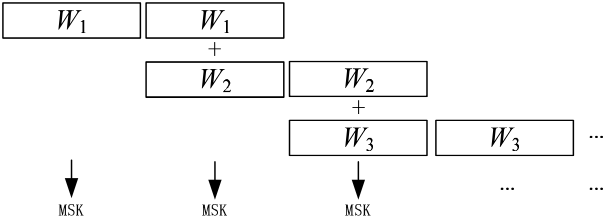


Figure: The BMST system with MSK.

BMST with MSK

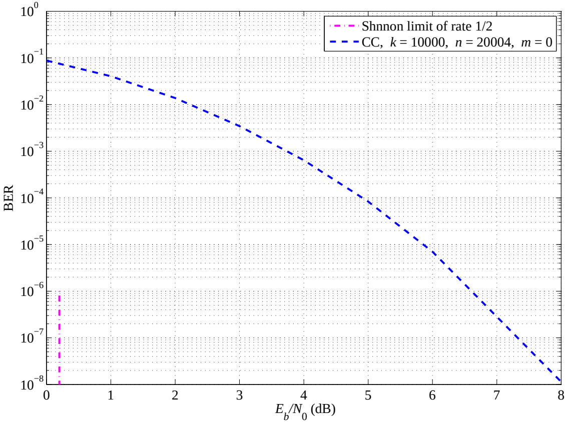


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 10000$ and $n = 20004$. Signals are transmitted using non-recursive MSK modulation over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $d = 7$ and $I_{\max} = 18$ is performed, where the encoding memories are specified in the legends.

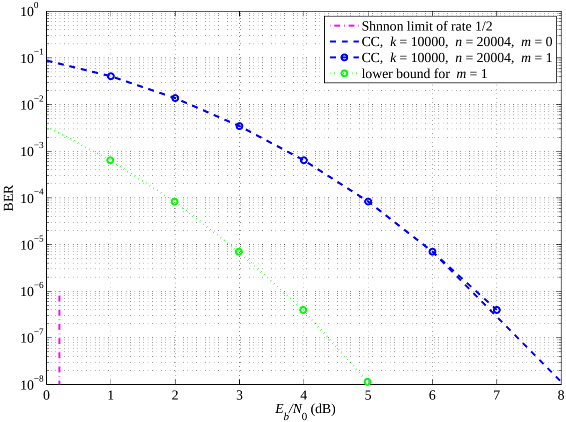


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 10000$ and $n = 20004$. Signals are transmitted using non-recursive MSK modulation over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $d = 7$ and $I_{\max} = 18$ is performed, where the encoding memories are specified in the legends.

BMST with MSK

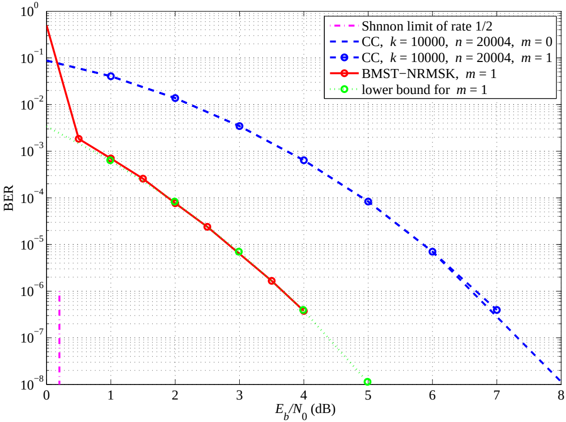


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 10000$ and $n = 20004$. Signals are transmitted using non-recursive MSK modulation over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $d = 7$ and $I_{\max} = 18$ is performed, where the encoding memories are specified in the legends.

BMST with MSK

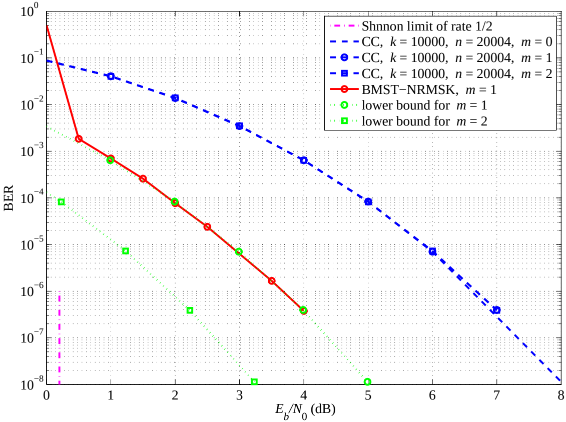


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 10000$ and $n = 20004$. Signals are transmitted using non-recursive MSK modulation over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $d = 7$ and $I_{\max} = 18$ is performed, where the encoding memories are specified in the legends.

BMST with MSK

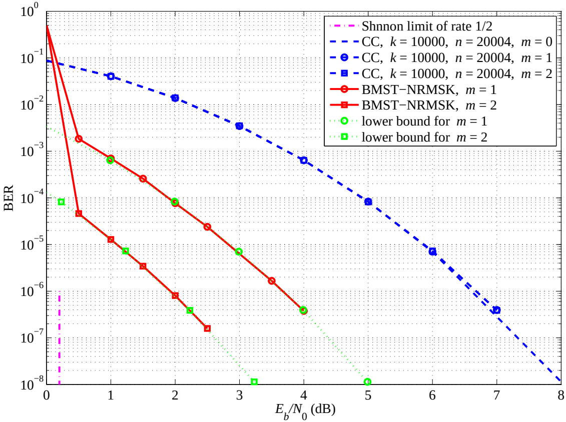


Figure: The basic code is a terminated 4-state $(2,1,2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 10000$ and $n = 20004$. Signals are transmitted using non-recursive MSK modulation over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $d = 7$ and $I_{\max} = 18$ is performed, where the encoding memories are specified in the legends.

BMST with MSK

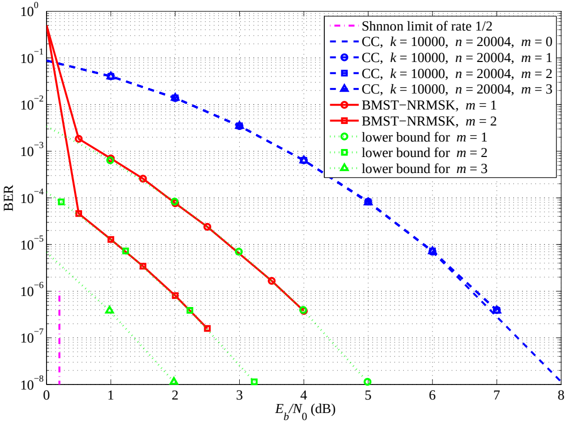


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 10000$ and $n = 20004$. Signals are transmitted using non-recursive MSK modulation over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $d = 7$ and $I_{\max} = 18$ is performed, where the encoding memories are specified in the legends.

BMST with MSK

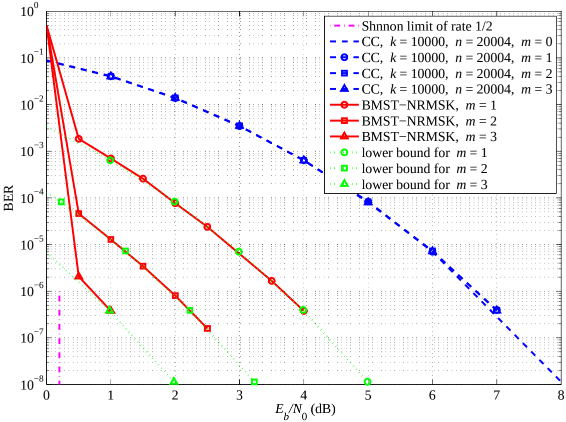


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 10000$ and $n = 20004$. Signals are transmitted using non-recursive MSK modulation over AWGN channels. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $d = 7$ and $I_{\max} = 18$ is performed, where the encoding memories are specified in the legends.

Outline

BMST with Spacial Modulation (SM)

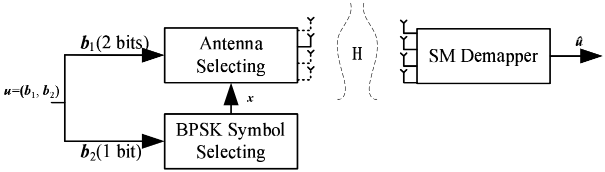


Figure: The spacial modulation with 4 transmitter antennas and 4 receiver antennas using BPSK modulation. Only one antenna is active for each transmission.

BMST with Spacial Modulation (SM)

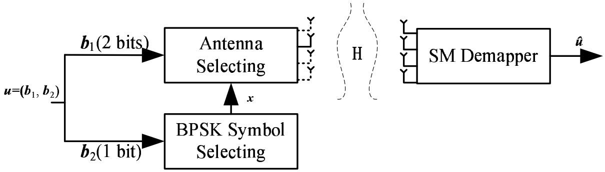


Figure: The spacial modulation with 4 transmitter antennas and 4 receiver antennas using BPSK modulation. Only one antenna is active for each transmission.

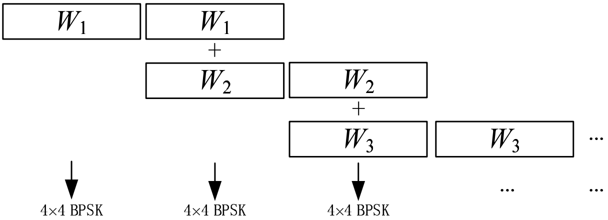


Figure: The BMST system with a 4×4 BPSK spacial modulation with only one antenna being active for each transimission.

BMST with Spacial Modulation over Rayleigh fading channel

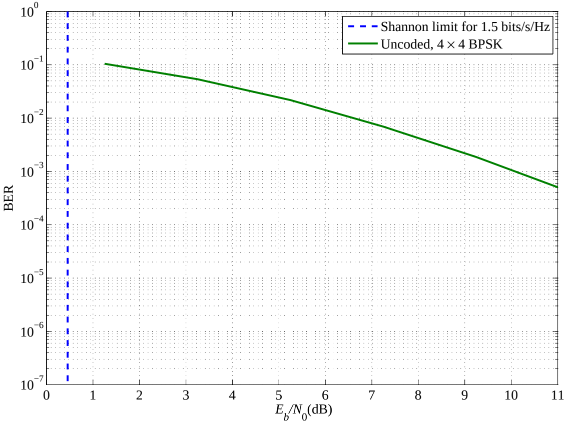


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 7000$ and $n = 14004$. Signals are transmitted using BPSK with 4 transmitter antennas and 4 receiver antennas over Rayleigh fading channels. Only one antenna is active for each BPSK symbol. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$ is performed.

BMST with Spacial Modulation over Rayleigh fading channel

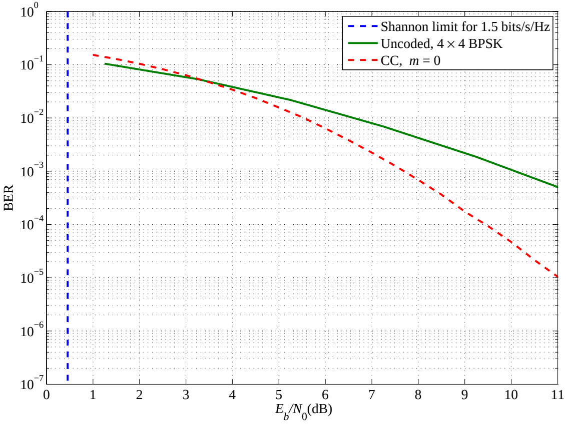


Figure: The basic code is a terminated 4-state $(2,1,2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 7000$ and $n = 14004$. Signals are transmitted using BPSK with 4 transmitter antennas and 4 receiver antennas over Rayleigh fading channels. Only one antenna is active for each BPSK symbol. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$ is performed.

BMST with Spacial Modulation over Rayleigh fading channel

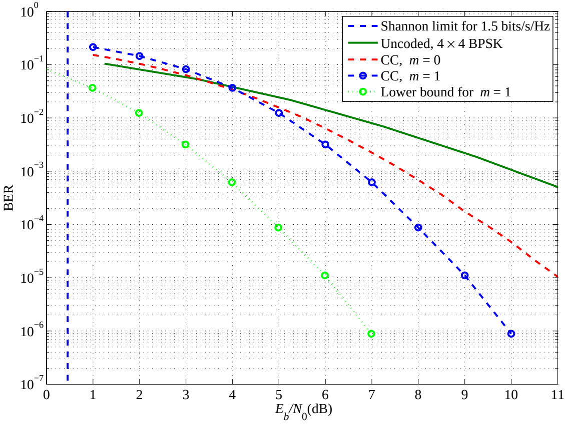


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 7000$ and $n = 14004$. Signals are transmitted using BPSK with 4 transmitter antennas and 4 receiver antennas over Rayleigh fading channels. Only one antenna is active for each BPSK symbol. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$ is performed.

BMST with Spacial Modulation over Rayleigh fading channel

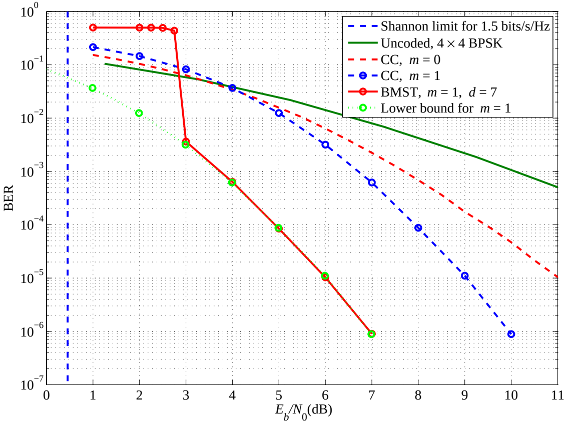


Figure: The basic code is a terminated 4-state $(2,1,2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 7000$ and $n = 14004$. Signals are transmitted using BPSK with 4 transmitter antennas and 4 receiver antennas over Rayleigh fading channels. Only one antenna is active for each BPSK symbol. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$ is performed.

BMST with Spacial Modulation over Rayleigh fading channel

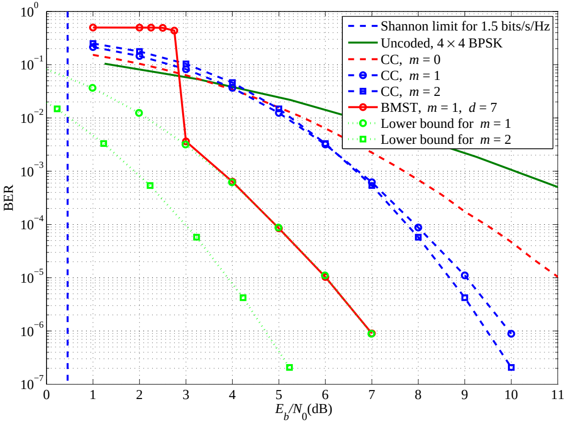


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 7000$ and $n = 14004$. Signals are transmitted using BPSK with 4 transmitter antennas and 4 receiver antennas over Rayleigh fading channels. Only one antenna is active for each BPSK symbol. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$ is performed.

BMST with Spacial Modulation over Rayleigh fading channel

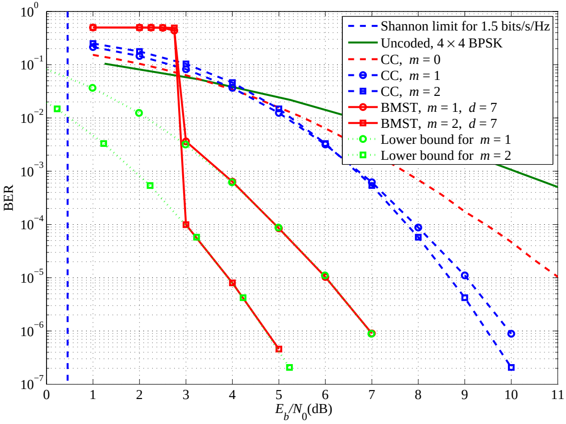


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 7000$ and $n = 14004$. Signals are transmitted using BPSK with 4 transmitter antennas and 4 receiver antennas over Rayleigh fading channels. Only one antenna is active for each BPSK symbol. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$ is performed.

BMST with Spacial Modulation over Rayleigh fading channel

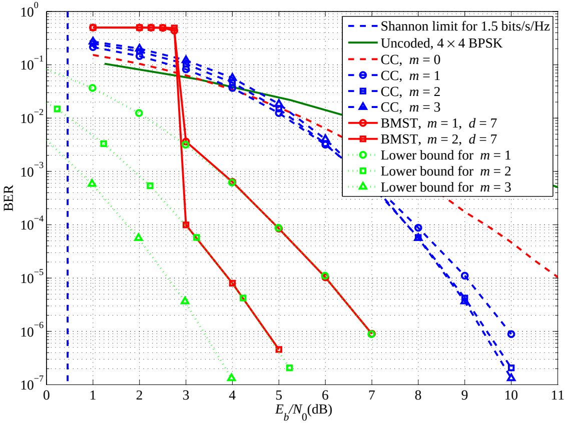


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 7000$ and $n = 14004$. Signals are transmitted using BPSK with 4 transmitter antennas and 4 receiver antennas over Rayleigh fading channels. Only one antenna is active for each BPSK symbol. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$ is performed.

BMST with Spacial Modulation over Rayleigh fading channel

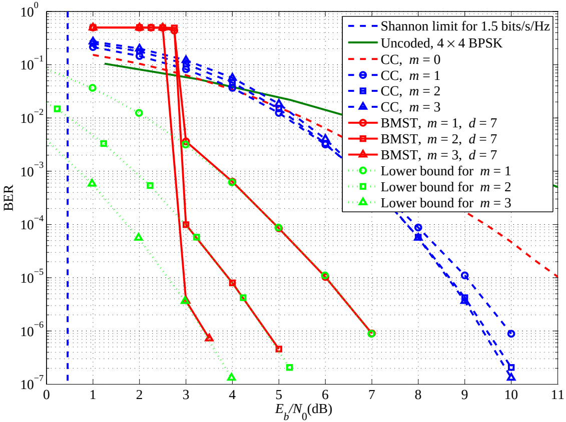


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$ with $k = 7000$ and $n = 14004$. Signals are transmitted using BPSK with 4 transmitter antennas and 4 receiver antennas over Rayleigh fading channels. Only one antenna is active for each BPSK symbol. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm with $I_{\max} = 18$ is performed.

Outline

Conclusions

Conclusions

- We presented a new method for constructing long codes from short codes;
- The encoding process can be as fast as the short code, while the decoding has a fixed but tunable delay.
- With an iterative sliding-window decoding algorithm, the performance of BMST can approach the derived lower bound in low error rate region;
- This scheme can be generalized, for example, to non-binary codes, lattice codes, and so on.
- In principle, any code can be the basic code as long as
  1. it is defined over a group (but not necessarily group code; (This is required by the *superposition* before transmission).
  2. it has an efficient encoding algorithm;
  3. it has an exact (or approximated) soft-in-soft-out (SISO) decoding algorithm.
- The BMST scheme is easy to combine with high-order modulation, continuous phase modulation (CPM), and even spacial modulation (SM) etc. and has a good performance.

Related Works

X. Ma, C. Liang, K. Huang, and Q. Zhuang,
Obtaining Extra Coding Gain for Short Codes by Block Markov Superposition Transmission.
in Proc. IEEE Int. Symp. Information Theory 2013, Istanbul, Turkey, Jul. 2013.

X. Ma, C. Liang, K. Huang, and Q. Zhuang,
Block Markov Superposition Transmission Construction of Big Convolutional Codes from Short Codes.
submitted to *IEEE Trans. Inf. Theory* .

C. Liang, K. Huang, X. Ma, and B. Bai,
Block Markov Superposition Transmission with Bit-Interleaved Coded Modulation.
accepted by *IEEE Commum. Lett.*.

X. Liu, C. Liang, and, X. Ma,
Block Markov Superposition Transmission of Convolutional Codes with MSK Signaling.
submitted to *IET Commun.*.

C. Liang, X. Ma, Q. Zhuang, and B. Bai,
Approaching the Channel Capacity within One dB at a BER of 10^{15}.
submitted to *Proc. IEEE Int. Symp. Information Theory 2014*.

J. Hu, C. Liang, X. Ma, and B. Bai,
Block Markov Superposition Transmission of Short Polar Codes: A New Class of Multiple-rate Codes.
submitted to *Proc. IEEE Int. Symp. Information Theory 2014*.

Thank You for Your Attention!

Acknowledgements

- This work is supported by the 973 Program (No. 2012CB316100) and the NSF (No. 61172082) of China.
- This work is joint with:
  - Prof. Baoming Bai from Xidian University.
  - Students: Chulong Liang, Jingnan Hu, Kechao Huang, Qiutao Zhuang, Xiaopei Xu, Xiying Liu, and Zhihua Yang.