Block Markov Superposition Transmission: Construction of Big Convolutional Codes from Short Codes

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

Joint work with: Chulong Liang, Kechao Huang, Qiutao Zhuang and Baoming Bai

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

Hong Kong, October, 2013

Outline

Outline

Construction of Long Codes from Short Codes

Short Convolutional Codes

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

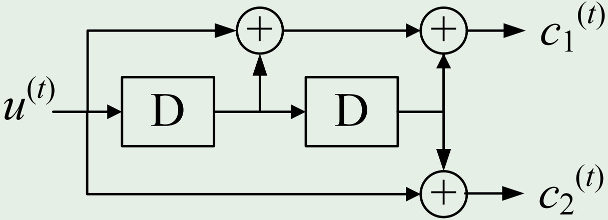


Figure: A $(2, 1, 2)$ convolutional code encoder.

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 $[2, 1, 2]^{5000}$, $[6, 5, 2]^{2000}$, $[7, 4, 3]^{2500}$;

Construction of Long Codes from Short Codes

Long Codes from Short Codes

- Product codes;
- Concatenated codes;
- Turbo codes: parallel concatenated convolutional codes (PCCC) and serial concatenated convolutional codes (SCCC);
- (Irregular) Repeat accumulate (RA) codes;
  Accumulate-repeat-accumulate (ARA) codes;
- Concatenated zigzag codes; Precoded concatenated zigzag codes;

Construction of Long Codes from Short Codes

Long Codes from Short Codes

- Product codes;
- Concatenated codes;
- Turbo codes: parallel concatenated convolutional codes (PCCC) and serial concatenated convolutional codes (SCCC);
- (Irregular) Repeat accumulate (RA) codes; Accumulate-repeat-accumulate (ARA) codes;
- Concatenated zigzag codes; Precoded concatenated zigzag codes;
- Low-density parity-check (LDPC) codes (either random construction or algebraic construction): From decoding aspect, they can be viewed as serially concatenated repeat codes with single parity-check codes;

Construction of Long Codes from Short Codes

Long Codes from Short Codes

- Product codes;
- Concatenated codes;
- Turbo codes: parallel concatenated convolutional codes (PCCC) and serial concatenated convolutional codes (SCCC);
- (Irregular) Repeat accumulate (RA) codes; Accumulate-repeat-accumulate (ARA) codes;
- Concatenated zigzag codes; Precoded concatenated zigzag codes;
- Low-density parity-check (LDPC) codes (either random construction or algebraic construction): From decoding aspect, they can be viewed as serially concatenated repeat codes with single parity-check codes;
- Convolutional LDPC codes;

Construction of Long Codes from Short Codes

Long Codes from Short Codes

- Product codes;
- Concatenated codes;
- Turbo codes: parallel concatenated convolutional codes (PCCC) and serial concatenated convolutional codes (SCCC);
- (Irregular) Repeat accumulate (RA) codes; Accumulate-repeat-accumulate (ARA) codes;
- Concatenated zigzag codes; Precoded concatenated zigzag codes;
- Low-density parity-check (LDPC) codes (either random construction or algebraic construction): From decoding aspect, they can be viewed as serially concatenated repeat codes with single parity-check codes;
- Convolutional LDPC codes;
- Polar codes: concatenation of a series of simple transformation;
- $\cdots$

Outline

Superposition Block Markov Encoding (SBME) in the Relay Channel

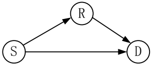


Figure: The Relay Channel.

Superposition Block Markov Encoding (SBME) in the Relay Channel

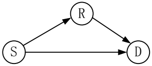


Figure: The Relay Channel.

- The data are equally grouped into B blocks;

Superposition Block Markov Encoding (SBME) in the Relay Channel

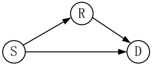


Figure: The Relay Channel.

- 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;

Superposition Block Markov Encoding (SBME) in the Relay Channel

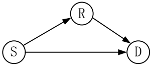


Figure: The Relay Channel.

- 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;

Superposition Block Markov Encoding (SBME) in the Relay Channel

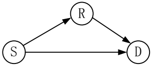


Figure: The Relay Channel.

- 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;

Superposition Block Markov Encoding (SBME) in the Relay Channel

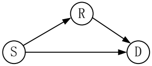


Figure: The Relay Channel.

- 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 the first data block from the two successive received blocks;

Superposition Block Markov Encoding (SBME) in the Relay Channel

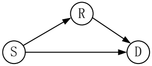


Figure: The Relay Channel.

- 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 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;

Superposition Block Markov Encoding (SBME) in the Relay Channel

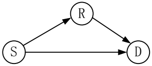


Figure: The Relay Channel.

- 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 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

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

- 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

- The data are equally grouped into B blocks;
- 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

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

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

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

BMST Scheme

- The data are equally grouped into B blocks;
- Initially, the transmitter sends a codeword from $\mathscr{C}$ that corresponds to the first data block;
- 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.
- In the meanwhile, a fresh codeword from $\mathscr{C}$ that corresponds to the second data block is superimposed on the second block transmission.
- Finally, the receiver recovers the first data block from the two successive received blocks.

Block Markov Superposition Transmission

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

BMST Scheme

- The data are equally grouped into B blocks;
- Initially, the transmitter sends a codeword from $\mathscr{C}$ that corresponds to the first data block;
- 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.
- In the meanwhile, a fresh codeword from $\mathscr{C}$ that corresponds to the second data block is superimposed on the second block transmission.
- Finally, the receiver recovers 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;

Block Markov Superposition Transmission

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

BMST Scheme

- The data are equally grouped into B blocks;
- Initially, the transmitter sends a codeword from $\mathscr{C}$ that corresponds to the first data block;
- 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.
- In the meanwhile, a fresh codeword from $\mathscr{C}$ that corresponds to the second data block is superimposed on the second block transmission.
- Finally, the receiver recovers 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.

Block Markov Superposition Transmission

Encoding

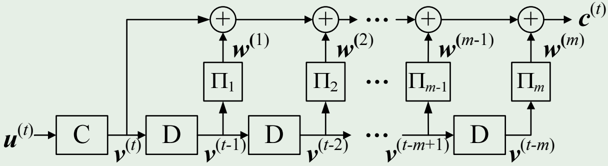


Figure: Encoding structure of BMST with memory m.

Recursive Encoding of BMST

1. *Initialization*: For $t < 0$, set $\mathbf{v}^{(t)} = \mathbf{0} \in \mathbb{F}_2^n$.
2. *Recursion*: For $t = 0, 1, \cdots, L-1$,
    - Encode $\mathbf{u}^{(t)}$ into $\mathbf{v}^{(t)} \in \mathbb{F}_2^n$ by the encoding algorithm of the basic code $\mathscr{C}$;
    - For $1 \le i \le m$, interleave $\mathbf{v}^{(t-i)}$ by the i-th interleaver Π_i into $\mathbf{w}^{(i)}$;
    - Compute $\mathbf{c}^{(t)} = \mathbf{v}^{(t)} + \sum_{1\le i\le m} \mathbf{w}^{(i)}$, which is taken as the t-th block of transmission.
3. *Termination*: For $t = L, L+1, \cdots, L+m-1$, set $\mathbf{u}^{(t)} = \mathbf{0} \in \mathbb{F}_2^k$ and compute $\mathbf{c}^{(t)}$ recursively.

Relation to Existing Codes

Relation to Existing Codes

- For a rate $R = k/n$ general convolutional code, information sequence $\boldsymbol{u} = \left(\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}, \cdots\right)$is encoded into code sequence $\boldsymbol{c} = \left(\boldsymbol{c}^{(0)}, \boldsymbol{c}^{(1)}, \cdots\right)$ by
$$\boldsymbol{c}^{(t)} = \boldsymbol{u}^{(t)}\boldsymbol{G}_0 + \boldsymbol{u}^{(t-1)}\boldsymbol{G}_1 + \cdots + \boldsymbol{u}^{(t-m)}\boldsymbol{G}_m, t \geq 0,$$
where $\boldsymbol{u}^{(t)} = 0$ for $t < 0$ and $\boldsymbol{G}_i$ $(0 \leq i \leq m)$ is a binary $k \times n$ matrix and m is called the encoder *memory*.

Block Markov Superposition Transmission

Relation to Existing Codes

- For a rate $R = k/n$ general convolutional code, information sequence $\boldsymbol{u} = \left(\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}, \cdots\right)$ is encoded into code sequence $\boldsymbol{c} = \left(\boldsymbol{c}^{(0)}, \boldsymbol{c}^{(1)}, \cdots\right)$ by
$$\boldsymbol{c}^{(t)} = \boldsymbol{u}^{(t)}\boldsymbol{G}_0 + \boldsymbol{u}^{(t-1)}\boldsymbol{G}_1 + \cdots + \boldsymbol{u}^{(t-m)}\boldsymbol{G}_m, t \geq 0,$$
where $\boldsymbol{u}^{(t)} = 0$ for $t < 0$ and $\boldsymbol{G}_i$ $(0 \leq i \leq m)$ is a binary $k \times n$ matrix and m is called the encoder *memory*.
- Let $\boldsymbol{G}$ be the generator matrix of the basic code. Let $\Pi_1, \cdots, \Pi_m$ be m permutation matrices. The generator matrix of the BMST has a simple form
$$G_{\text{BMST}} = \begin{bmatrix} \boldsymbol{G} & \boldsymbol{G\Pi}_1 & \cdots & \boldsymbol{G\Pi}_m & & & & & \\ & \boldsymbol{G} & \ddots & \vdots & \ddots & & & & \\ & & \ddots & \boldsymbol{G\Pi}_1 & \ddots & \boldsymbol{G\Pi}_m & & & \\ & & & \boldsymbol{G} & \ddots & \vdots & \ddots & & \\ & & & & \ddots & \boldsymbol{G\Pi}_1 & \ddots & \boldsymbol{G\Pi}_m & \\ & & & & & \boldsymbol{G} & \cdots & \boldsymbol{G\Pi}_{m-1} & \boldsymbol{G\Pi}_m \end{bmatrix}.$$

Relation to Existing Codes

- For a rate $R = k/n$ general convolutional code, information sequence $\boldsymbol{u} = \left(\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}, \cdots\right)$ is encoded into code sequence $\boldsymbol{c} = \left(\boldsymbol{c}^{(0)}, \boldsymbol{c}^{(1)}, \cdots\right)$ by
$$\boldsymbol{c}^{(t)} = \boldsymbol{u}^{(t)}\boldsymbol{G}_0 + \boldsymbol{u}^{(t-1)}\boldsymbol{G}_1 + \cdots + \boldsymbol{u}^{(t-m)}\boldsymbol{G}_m, t \geq 0,$$
where $\boldsymbol{u}^{(t)} = 0$ for $t < 0$ and $\boldsymbol{G}_i$ $(0 \leq i \leq m)$ is a binary $k \times n$ matrix and m is called the encoder *memory*.
- Let $\boldsymbol{G}$ be the generator matrix of the basic code. Let $\Pi_1, \cdots, \Pi_m$ be m permutation matrices. The generator matrix of the BMST has a simple form
$$G_{\text{BMST}} = \begin{bmatrix} \boldsymbol{G} & \boldsymbol{G\Pi}_1 & \cdots & \boldsymbol{G\Pi}_m & & & & \\ & \boldsymbol{G} & \ddots & \vdots & \ddots & & & \\ & & \ddots & \boldsymbol{G\Pi}_1 & \ddots & \boldsymbol{G\Pi}_m & & \\ & & & \boldsymbol{G} & \ddots & \vdots & \ddots & \\ & & & & \ddots & \boldsymbol{G\Pi}_1 & \ddots & \boldsymbol{G\Pi}_m \\ & & & & & \boldsymbol{G} & \cdots & \boldsymbol{G\Pi}_{m-1} & \boldsymbol{G\Pi}_m \end{bmatrix}.$$
- The BMST defines a *special* class of convolutional codes. Specialities: *possibly extremely* large constraint length; $\boldsymbol{G\Pi}_i$ instead of $\boldsymbol{G}_i$.

Block Markov Superposition Transmission

Normal Graph

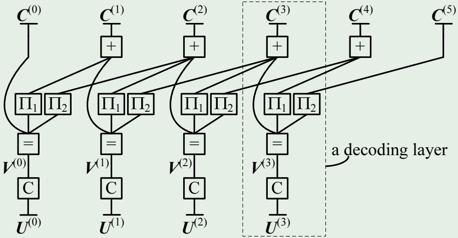


Figure: The normal graph of a BMST system with $L = 4$ and $m = 2$.

Decoding

- An iterative forward-backward decoding schedule is used for basic codes with small L;
- An iterative sliding-window decoding schedule is used for basic codes with large L;
- Four types of nodes: C, $=$, $+$, and $\prod$;
- Messages are processed and passed through different decoding layers forward and backward over the normal graph.

Outline

Coding Gain Analysis of the BMST

Genie-Aided Lower Bound on BER

- Imagine that $\mathbf{u}' = \{\mathbf{u}^{(i)}, t - m \le i \le t + m, i \ne t\}$ are known at the receiver.
- This is equivalent to transmitting $\boldsymbol{u}^{(t)}$ for $m + 1$ times.
- The coding gain of the BMST can not be larger than

$$10 \log_{10}(m + 1) - 10 \log_{10}(1 + m/L) \text{ dB}.$$

- Noticing that $\Pr\{\boldsymbol{u}'|\boldsymbol{y}\} \approx 1$ in the low error rate region, we can expect that the maximal coding gain $10 \log_{10}(m + 1) - 10 \log_{10}(1 + m/L)$ dB.

Upper Bound on BER

- The input-output weight enumerating function (IOWEF) of the BMST system can be computed from that of the basic code.
- The BER can be upper-bounded by an improved union bound.
- Notice that an incomplete (truncated) IOWEF is sufficient for upper bounds. (See Xiao Ma, Jia Liu and Baoming T-COMM 2013).

Outline

Simulation Results

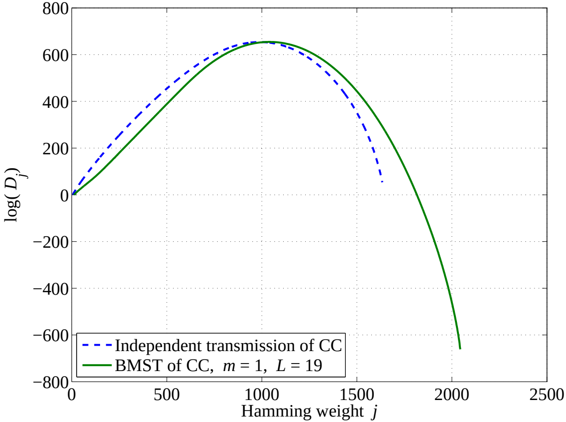


Figure: Comparison of the weight spectrum between the independent transmission system and the ensemble of the BMST system. The basic code is a terminated systematic encoded 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1, (1 + D + D^2)/(1 + D^2)]$. The BMST system encodes $L = 19$ sub-blocks of data with memory $m = 1$.

Simulation Results

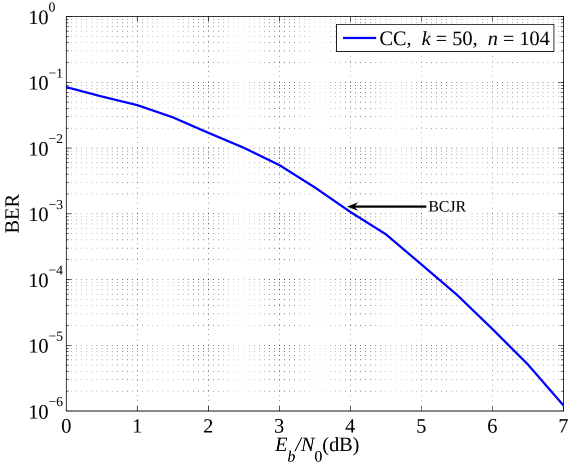


Figure: Coding gain analysis of the BMST system. The basic code is a terminated convolutional code (CC) with the polynomial generator matrix $[1, \frac{1+D+D^2}{1+D^2}]$. The coding parameters of the BMST system are $m = 1$, $L = 19$, $d = 19$, and $I_{\max} = 18$. The decoding algorithm is performed after all 20 transmitted sub-blocks are received (forward-backward schedule).

Simulation Results

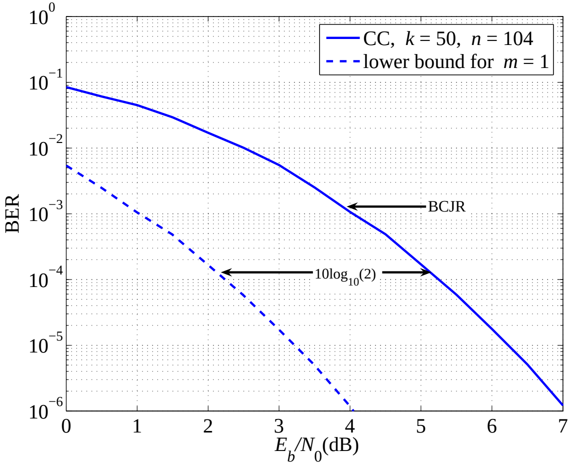


Figure: Coding gain analysis of the BMST system. The basic code is a terminated convolutional code (CC) with the polynomial generator matrix $[1, \frac{1+D+D^2}{1+D^2}]$. The coding parameters of the BMST system are $m = 1$, $L = 19$, $d = 19$, and $I_{\text{max}} = 18$. The decoding algorithm is performed after all 20 transmitted sub-blocks are received (forward-backward schedule).

Simulation Results

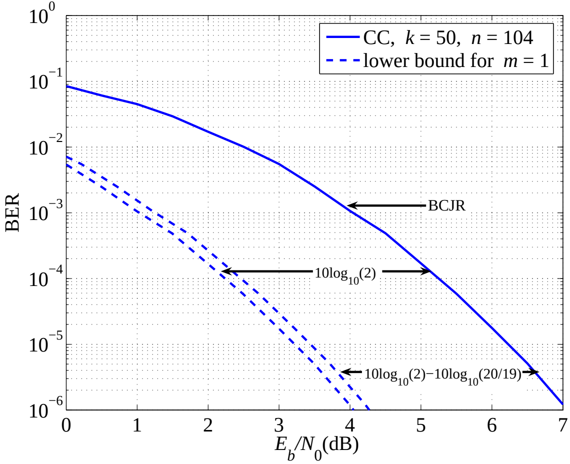


Figure: Coding gain analysis of the BMST system. The basic code is a terminated convolutional code (CC) with the polynomial generator matrix $[1, \frac{1+D+D^2}{1+D^2}]$. The coding parameters of the BMST system are $m = 1$, $L = 19$, $d = 19$, and $I_{\max} = 18$. The decoding algorithm is performed after all 20 transmitted sub-blocks are received (forward-backward schedule).

Simulation Results

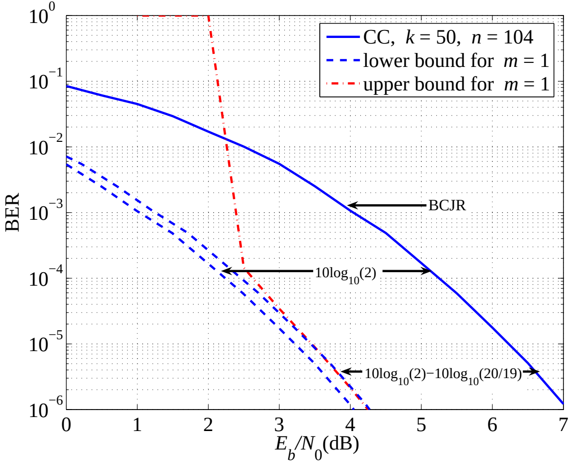


Figure: Coding gain analysis of the BMST system. The basic code is a terminated convolutional code (CC) with the polynomial generator matrix $[1, \frac{1+D+D^2}{1+D^2}]$. The coding parameters of the BMST system are $m = 1$, $L = 19$, $d = 19$, and $I_{\max} = 18$. The decoding algorithm is performed after all 20 transmitted sub-blocks are received (forward-backward schedule).

Simulation Results

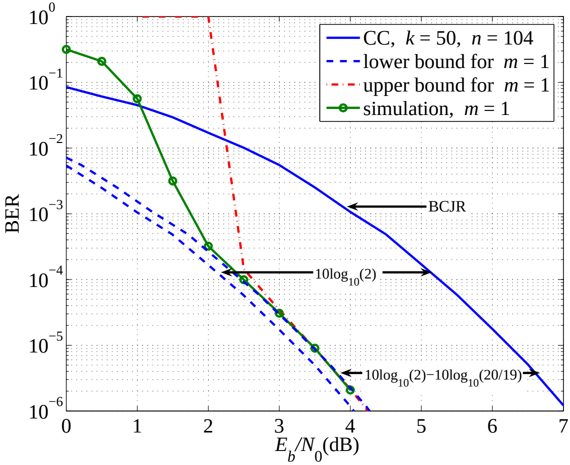


Figure: Coding gain analysis of the BMST system. The basic code is a terminated convolutional code (CC) with the polynomial generator matrix $[1, \frac{1+D+D^2}{1+D^2}]$. The coding parameters of the BMST system are $m = 1$, $L = 19$, $d = 19$, and $I_{\max} = 18$. The decoding algorithm is performed after all 20 transmitted sub-blocks are received (forward-backward schedule).

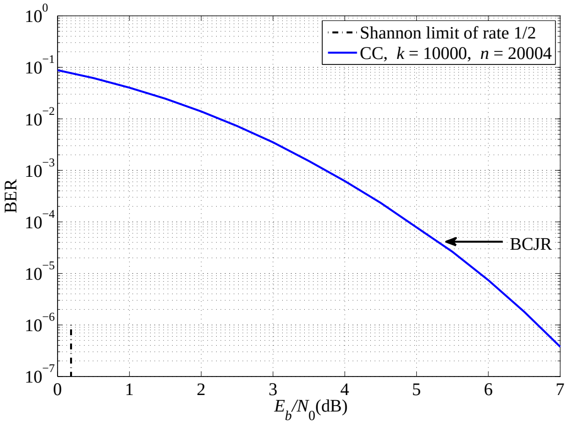


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and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

Simulation Results

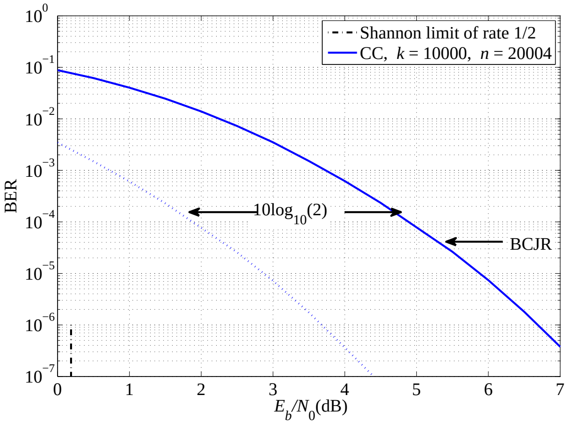


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and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

Simulation Results

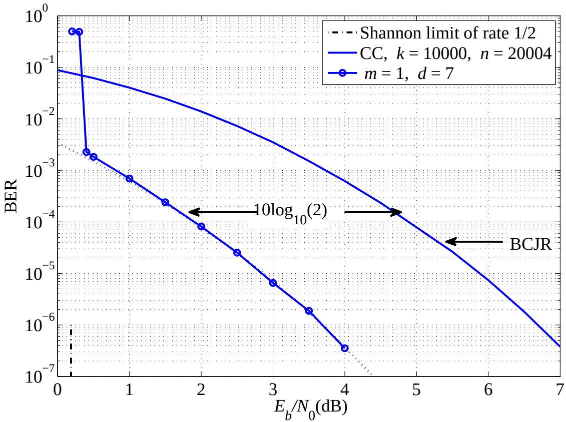


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and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

Simulation Results

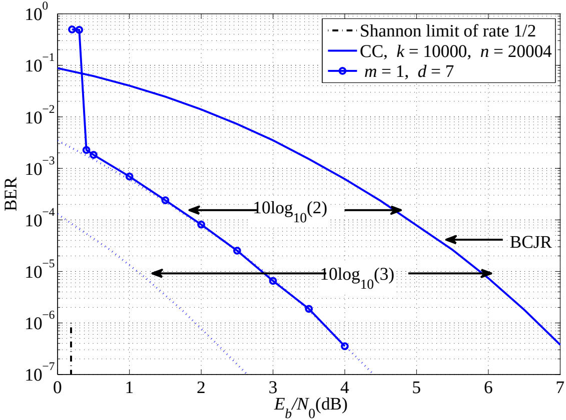


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and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

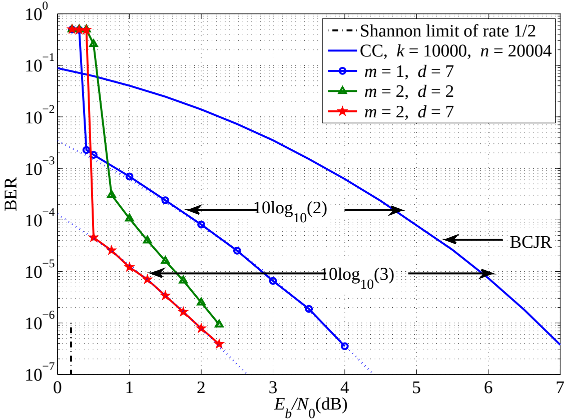


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and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

Simulation Results

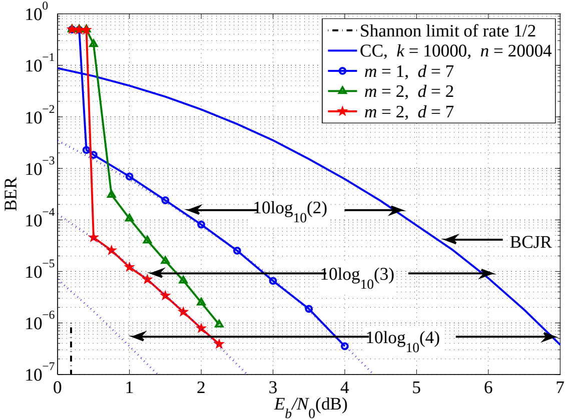


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and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

Simulation Results

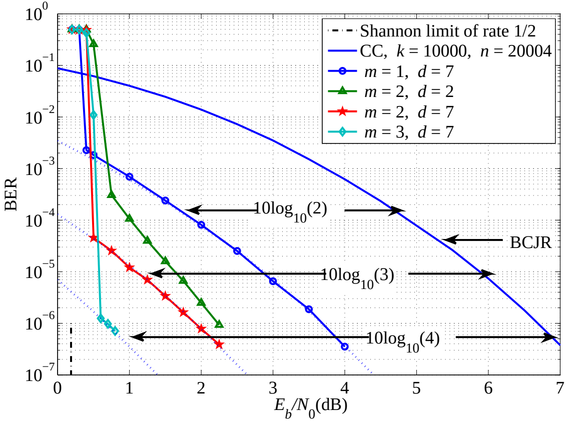


Figure: The basic code is a terminated 4-state $(2, 1, 2)$ convolutional code defined by the polynomial generator matrix $G(D) = [1 + D^2, 1 + D + D^2]$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and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

Simulation Results

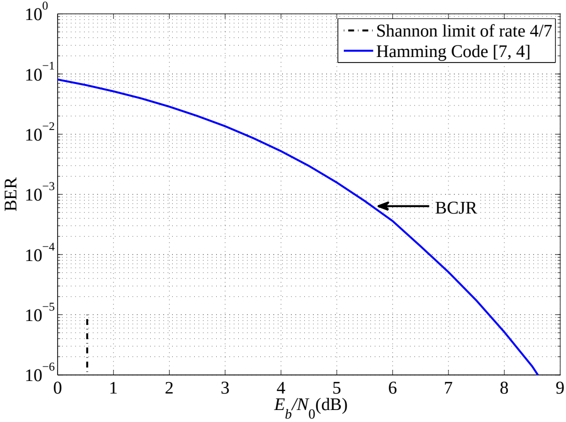


Figure: The basic code is the Cartesian product of Hamming code $[7, 4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

Simulation Results

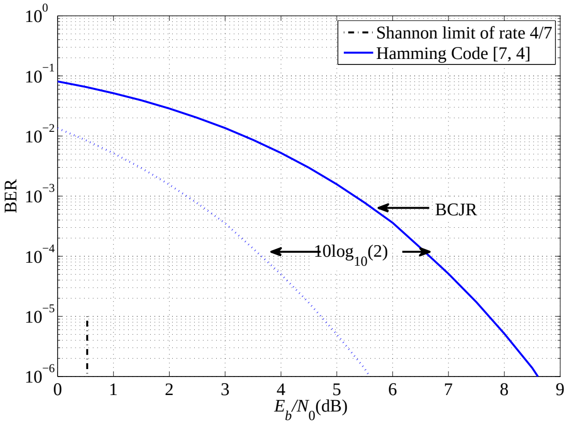


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

Simulation Results

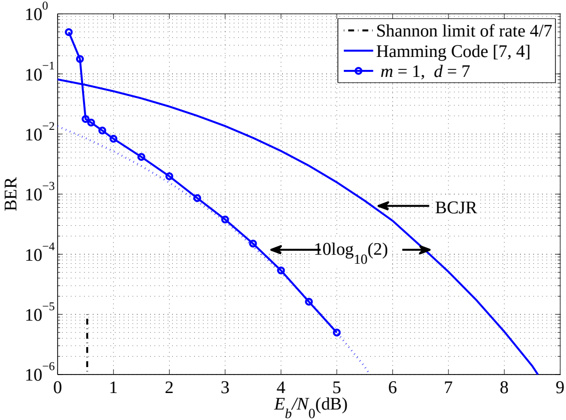


Figure: The basic code is the Cartesian product of Hamming code $[7, 4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

Simulation Results

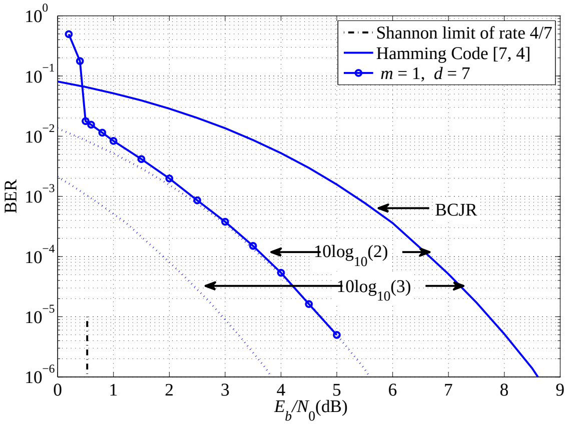


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

Simulation Results

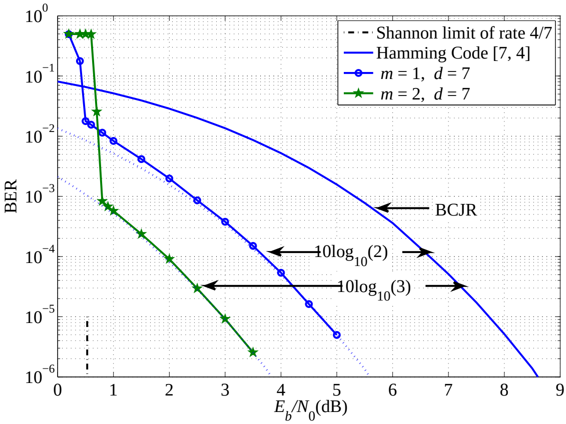


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

Simulation Results

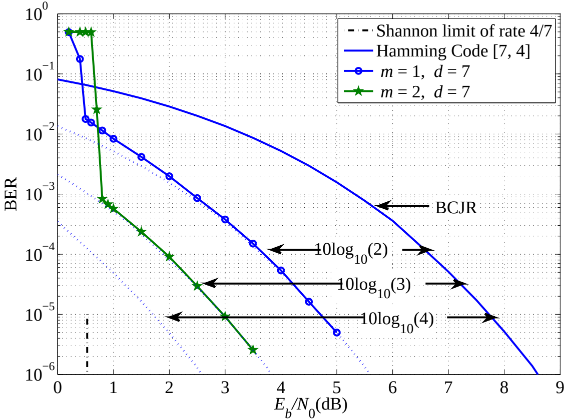


Figure: The basic code is the Cartesian product of Hamming code $[7, 4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

Simulation Results

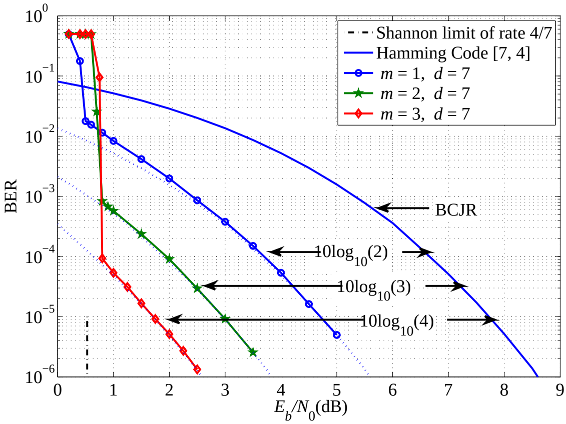


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

Simulation Results

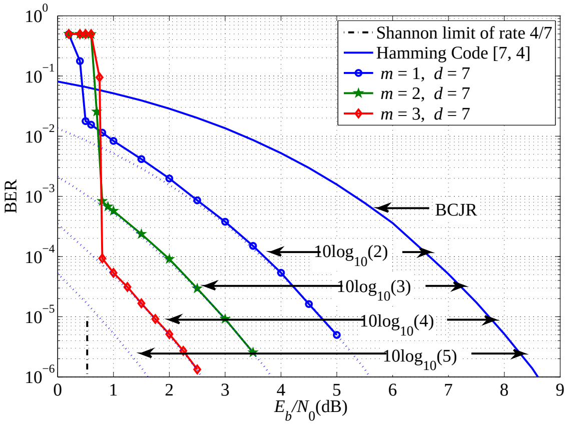


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

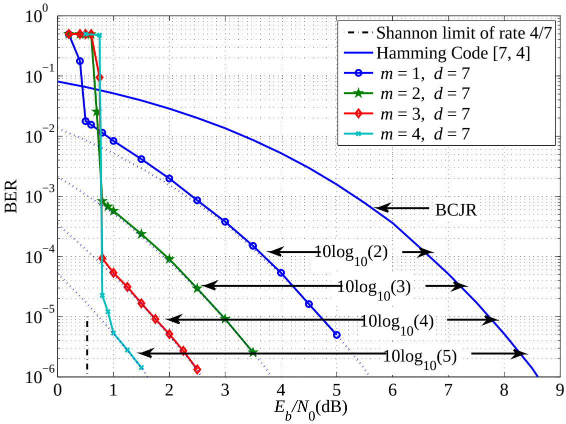


Figure: The basic code is the Cartesian product of Hamming code $[7,4]^{2500}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in the legends.

Simulation Results

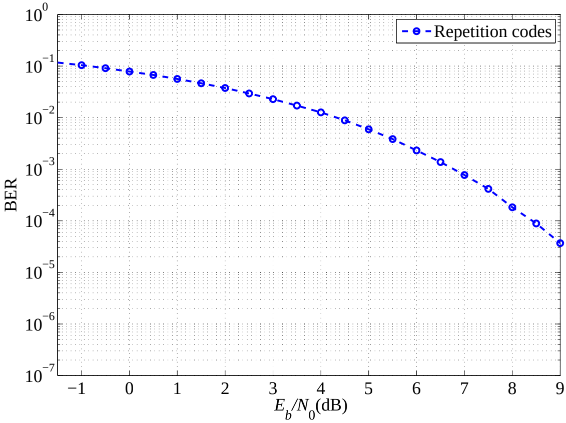


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

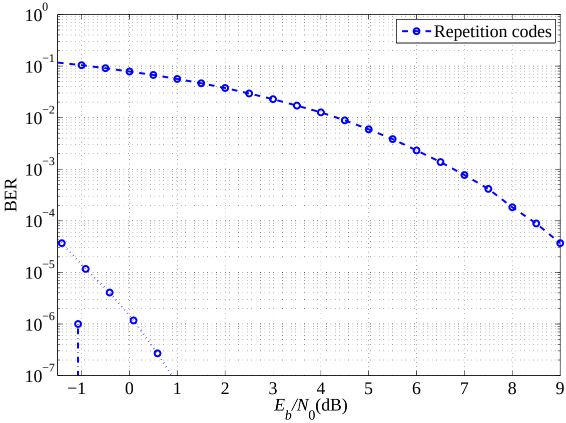


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

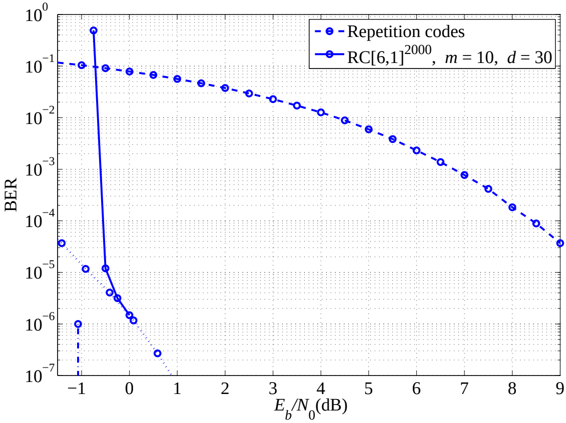


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

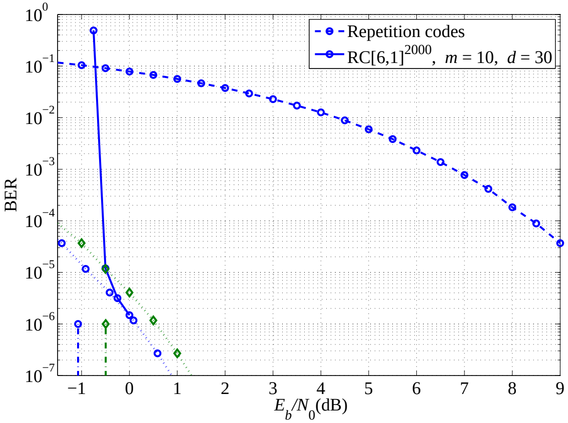


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

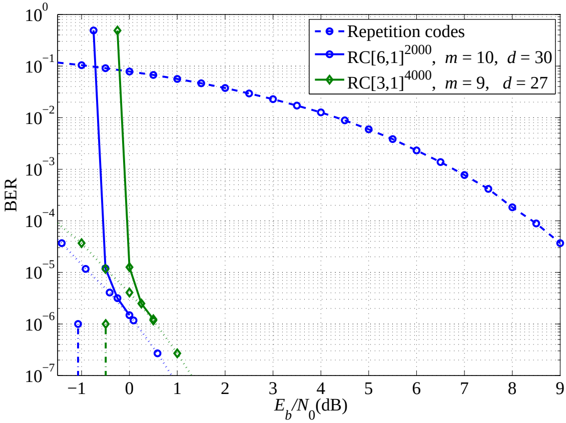


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

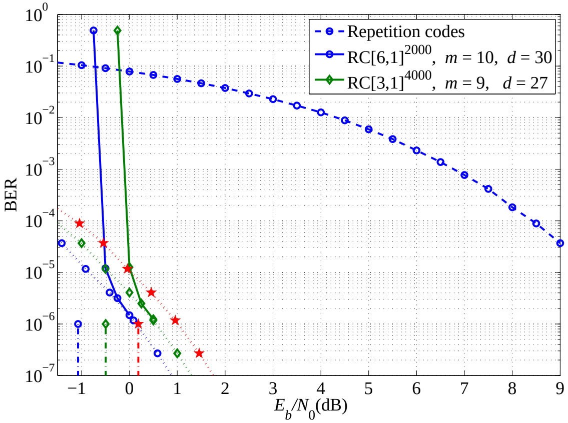


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

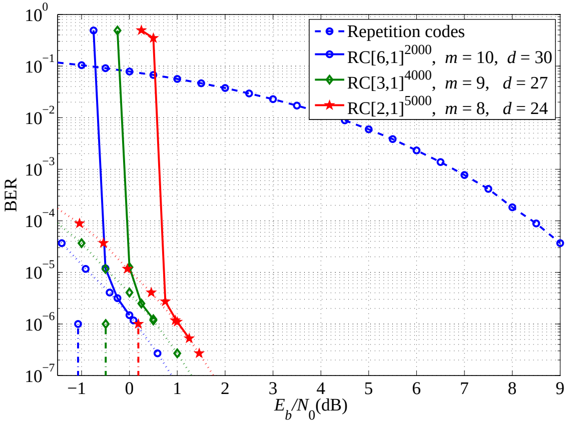


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

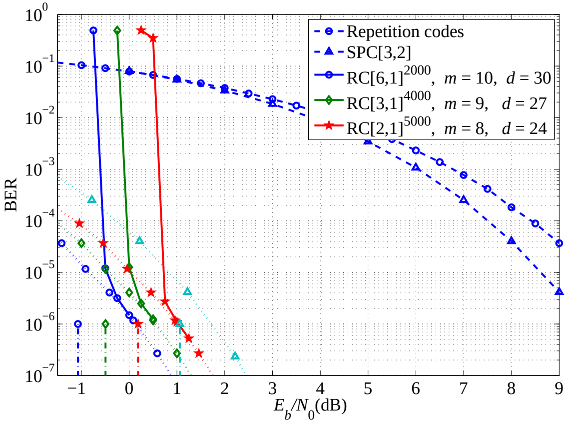


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

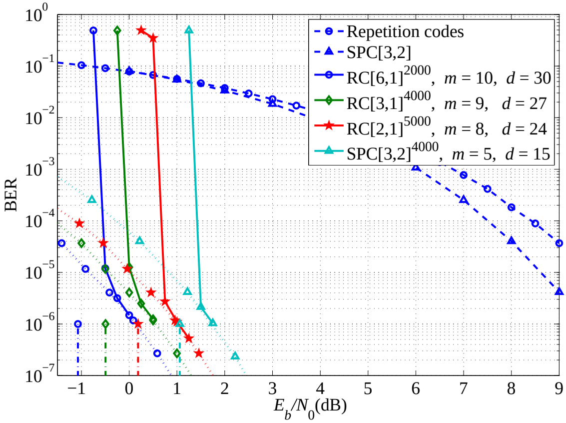


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

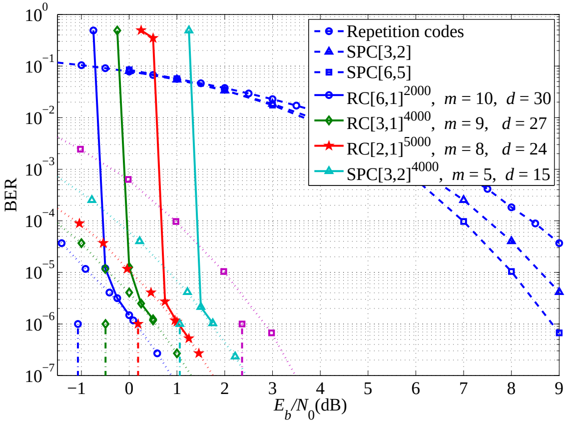


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

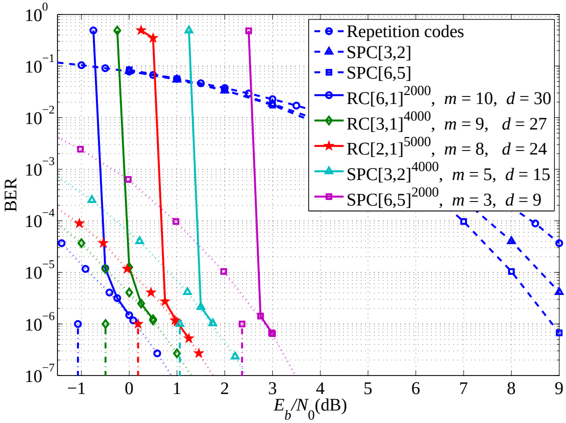


Figure: The basic code is either the Cartesian product of a repetition code or the Cartesian product of a single parity-check code. All systems encode $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are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

Simulation Results

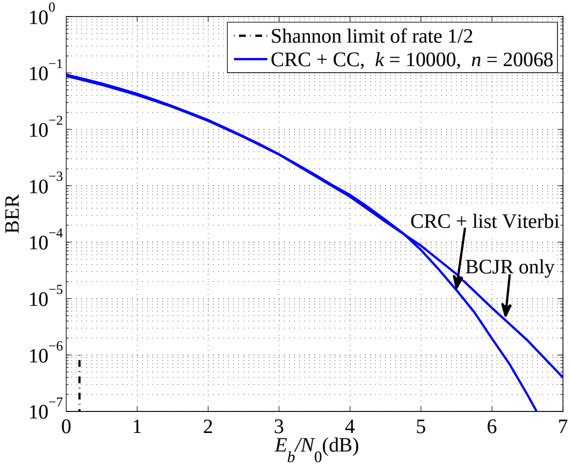


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 10000$ and length $n = 20068$, where the outer code is a 32-bit CRC code and the inner code is a terminated convolutional code with the polynomial generator matrix $[1 + D^2,\ 1 + D + D^2]$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

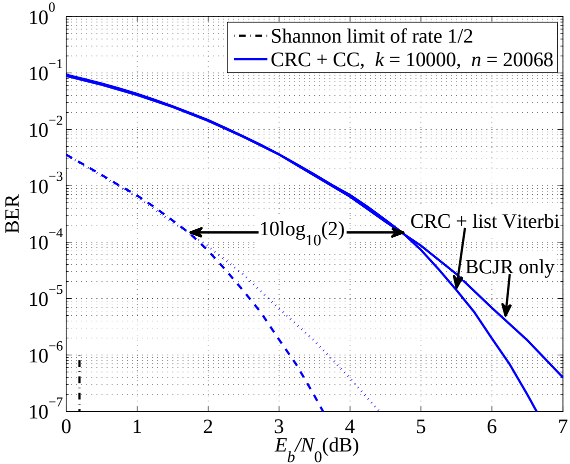


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 10000$ and length $n = 20068$, where the outer code is a 32-bit CRC code and the inner code is a terminated convolutional code with the polynomial generator matrix $[1 + D^2,\ 1 + D + D^2]$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

Simulation Results

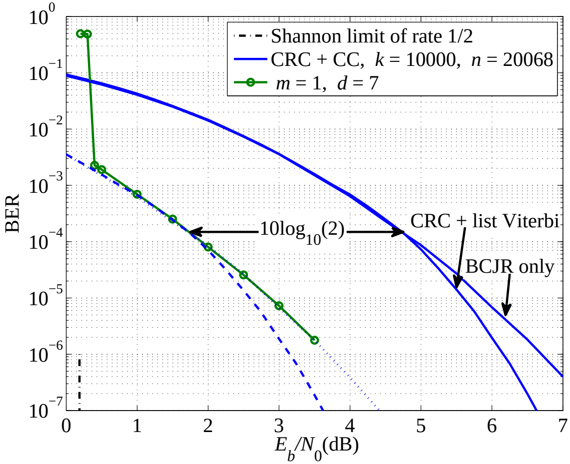


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 10000$ and length $n = 20068$, where the outer code is a 32-bit CRC code and the inner code is a terminated convolutional code with the polynomial generator matrix $[1 + D^2,\ 1 + D + D^2]$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

Simulation Results

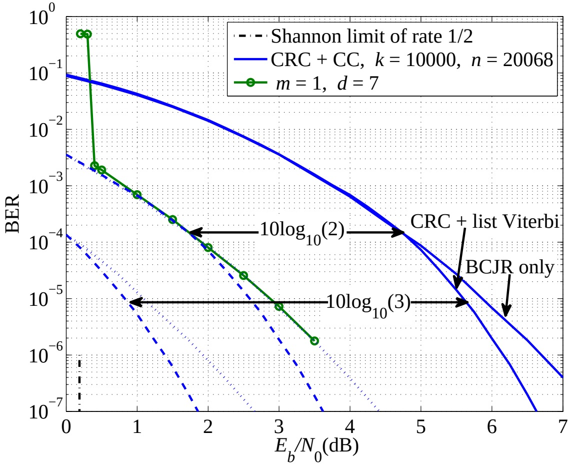


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 10000$ and length $n = 20068$, where the outer code is a 32-bit CRC code and the inner code is a terminated convolutional code with the polynomial generator matrix $[1 + D^2,\ 1 + D + D^2]$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

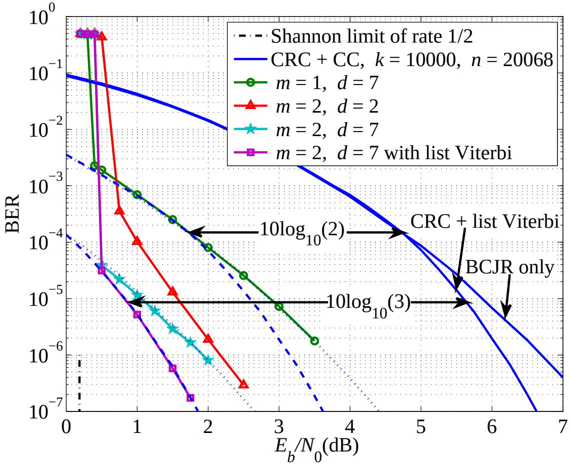


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 10000$ and length $n = 20068$, where the outer code is a 32-bit CRC code and the inner code is a terminated convolutional code with the polynomial generator matrix $[1 + D^2,\ 1 + D + D^2]$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

Simulation Results

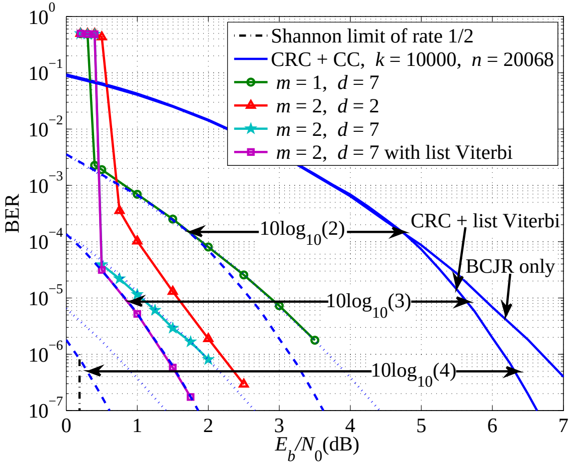


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 10000$ and length $n = 20068$, where the outer code is a 32-bit CRC code and the inner code is a terminated convolutional code with the polynomial generator matrix $[1 + D^2,\ 1 + D + D^2]$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

Simulation Results

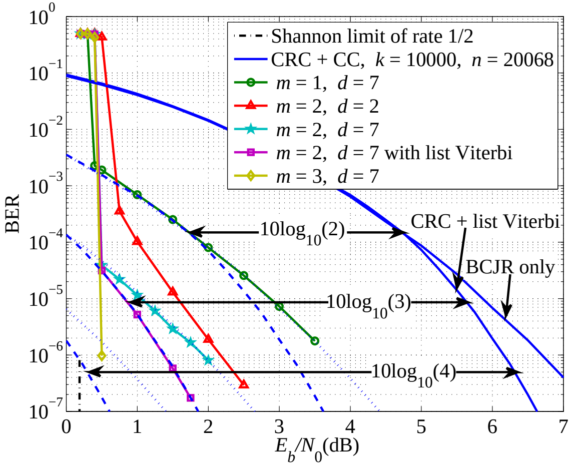


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 10000$ and length $n = 20068$, where the outer code is a 32-bit CRC code and the inner code is a terminated convolutional code with the polynomial generator matrix $[1 + D^2,\ 1 + D + D^2]$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

Simulation Results

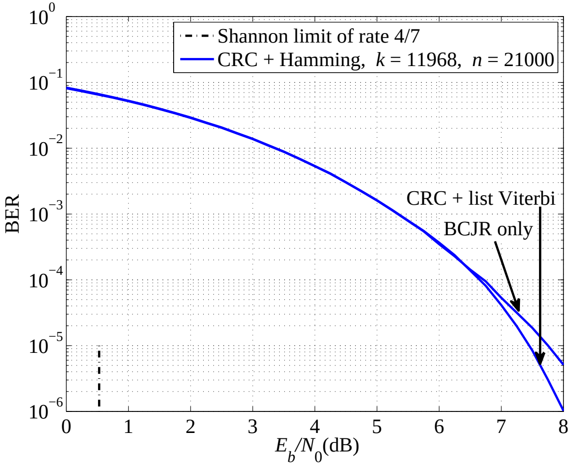


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 11968$ and length $n = 21000$, where the outer code is a 32-bit CRC code and the inner code is the Cartesian product of Hamming code $[7, 4]^{3000}$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\text{max}} = 18$.

Simulation Results

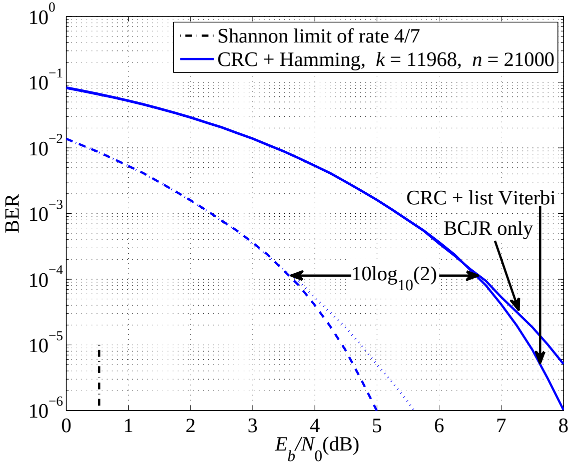


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 11968$ and length $n = 21000$, where the outer code is a 32-bit CRC code and the inner code is the Cartesian product of Hamming code $[7, 4]^{3000}$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

Simulation Results

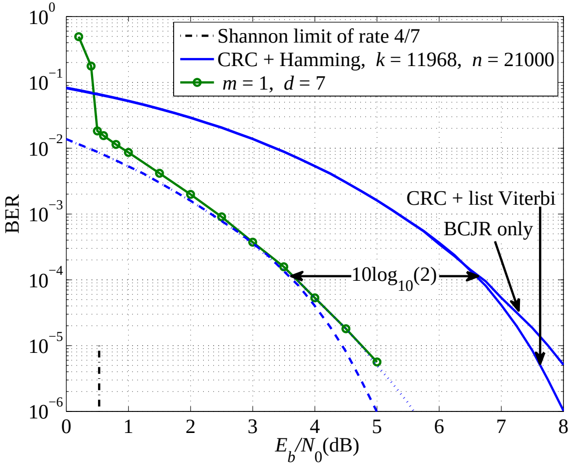


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 11968$ and length $n = 21000$, where the outer code is a 32-bit CRC code and the inner code is the Cartesian product of Hamming code $[7, 4]^{3000}$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\text{max}} = 18$.

Simulation Results

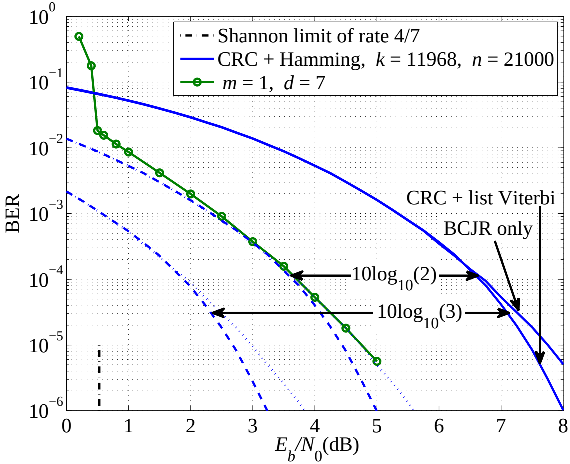


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 11968$ and length $n = 21000$, where the outer code is a 32-bit CRC code and the inner code is the Cartesian product of Hamming code $[7,4]^{3000}$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

Simulation Results

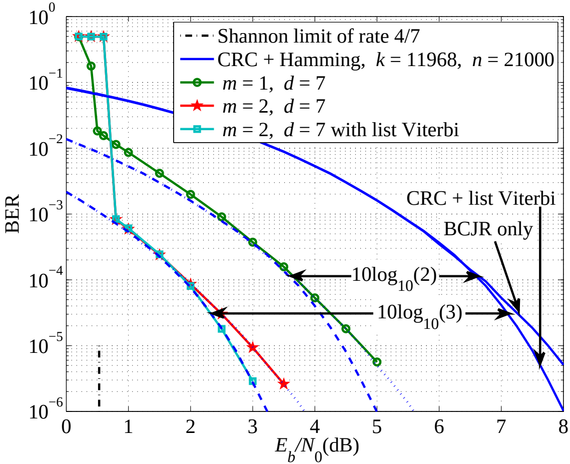


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 11968$ and length $n = 21000$, where the outer code is a 32-bit CRC code and the inner code is the Cartesian product of Hamming code $[7, 4]^{3000}$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\text{max}} = 18$.

Simulation Results

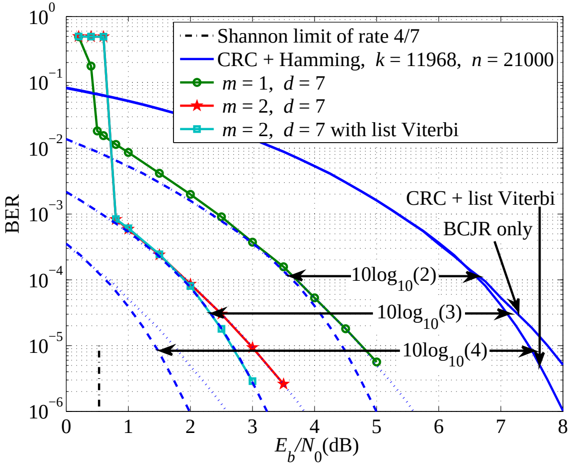


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 11968$ and length $n = 21000$, where the outer code is a 32-bit CRC code and the inner code is the Cartesian product of Hamming code $[7, 4]^{3000}$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\text{max}} = 18$.

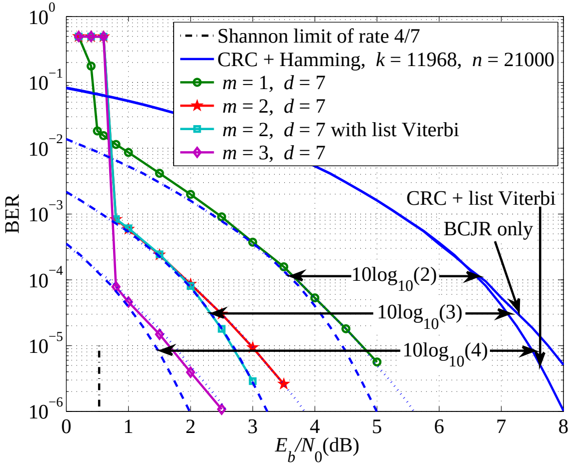


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 11968$ and length $n = 21000$, where the outer code is a 32-bit CRC code and the inner code is the Cartesian product of Hamming code $[7,4]^{3000}$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

Simulation Results

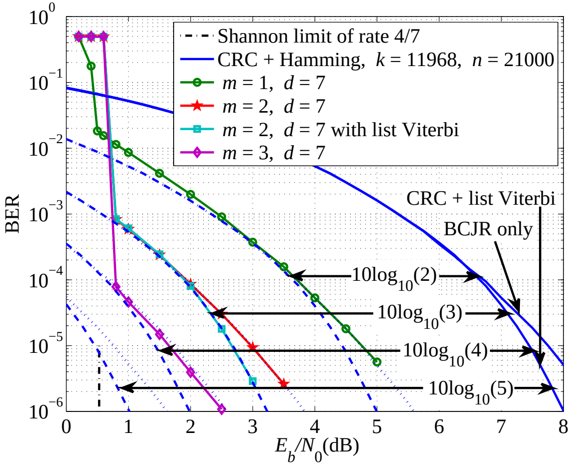


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 11968$ and length $n = 21000$, where the outer code is a 32-bit CRC code and the inner code is the Cartesian product of Hamming code $[7,4]^{3000}$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

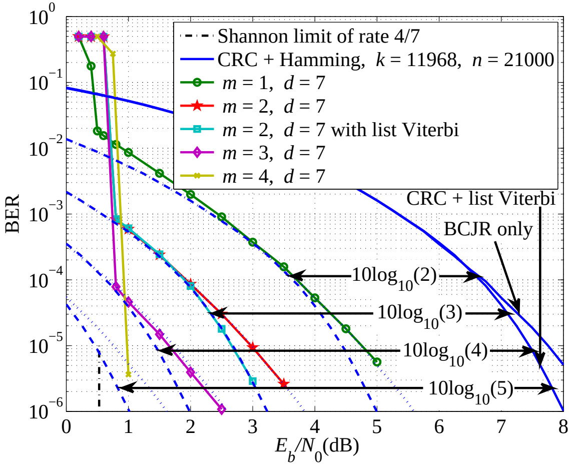
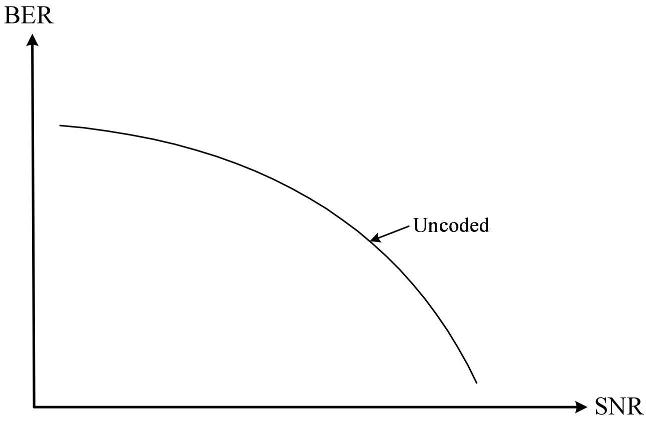


Figure: The basic code $\mathscr{C}$ is a concatenated code of dimension $k = 11968$ and length $n = 21000$, where the outer code is a 32-bit CRC code and the inner code is the Cartesian product of Hamming code $[7,4]^{3000}$. Other coding parameters of the BMST system are $L = 1000$ and $I_{\max} = 18$.

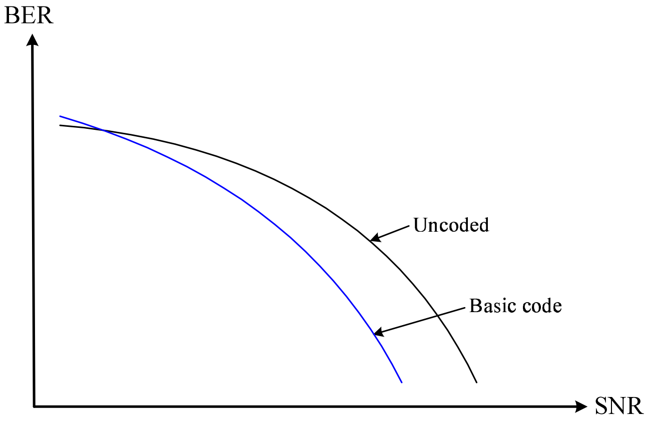
Outline

What does the performance curve look like?

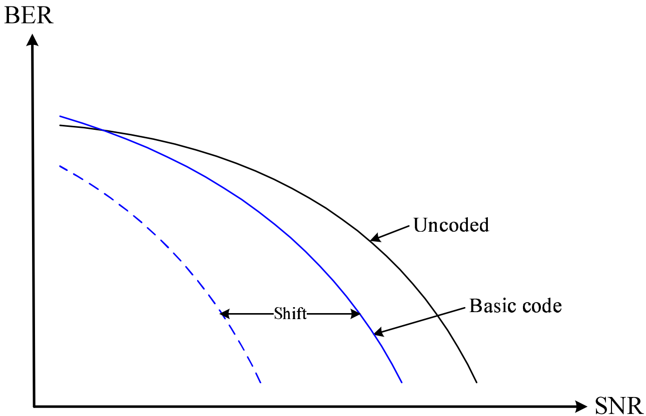
What does the performance curve look like?



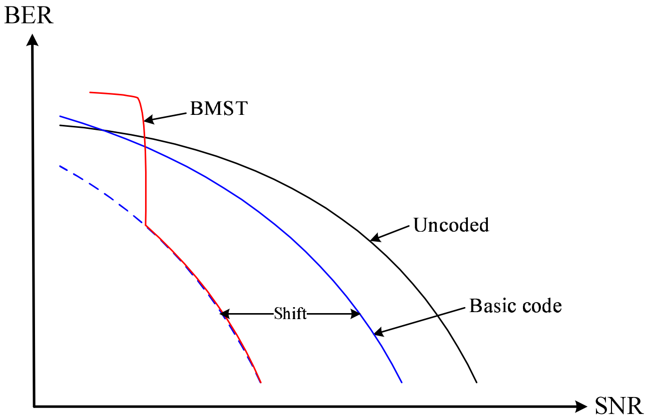
What does the performance curve look like?



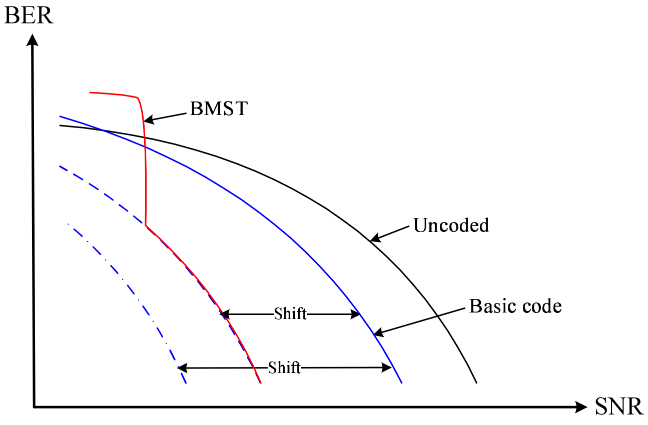
What does the performance curve look like?



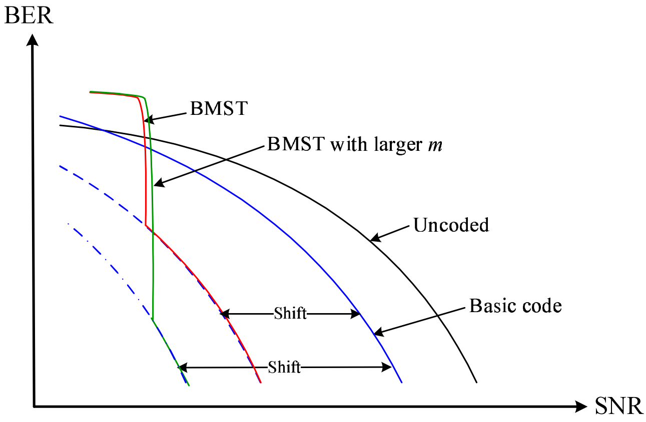
What does the performance curve look like?



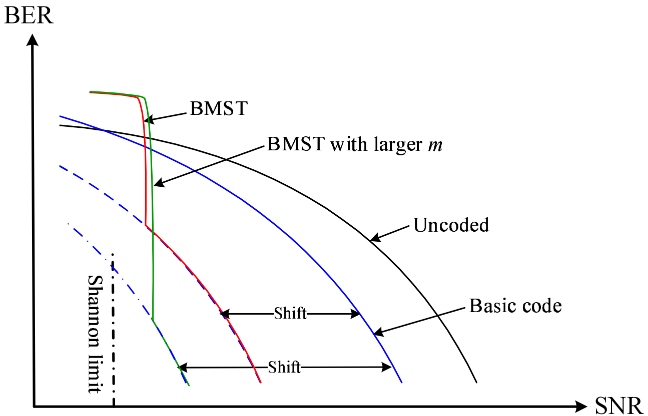
What does the performance curve look like?



What does the performance curve look like?



What does the performance curve look like?



What code can be the basic code?

What code can be the basic code?

- What do we mean by short code?

What code can be the basic code?

- What do we mean by short code?
- Can a random-generated linear code $[32, 16]$ be the basic code?

What code can be the basic code?

- What do we mean by short code?
- Can a random-generated linear code $[32, 16]$ be the basic code?
- Is BMST an LDPC code or a convolutional LDPC code?

Parity-check Matrix of the BMST System

Parity-check Matrix of the BMST System

Parity-check Matrix of the BMST System

- Define a sequence of matrices $\boldsymbol{P}_0 = \begin{cases} \mathbf{0}, & t \leq -1, \\ \boldsymbol{I} & t = 0, \\ \sum_{1 \leq \ell \leq m} \boldsymbol{P}_{t-\ell} \boldsymbol{\Pi}_\ell & t \geq 1, \end{cases}$ where $\boldsymbol{I}$ is the identity matrix of order n and $\mathbf{0}$ is the zero matrix of order n.

Parity-check Matrix of the BMST System

Parity-check Matrix of the BMST System

- Define a sequence of matrices $\boldsymbol{P}_0 = \begin{cases} \mathbf{0}, & t \leq -1, \\ \boldsymbol{I} & t = 0, \\ \sum_{1 \leq \ell \leq m} \boldsymbol{P}_{t-\ell} \boldsymbol{\Pi}_\ell & t \geq 1, \end{cases}$ where $\boldsymbol{I}$ is the identity matrix of order n and $\mathbf{0}$ is the zero matrix of order n.
- The parity-check matrix of the BMST system is given by

$$\boldsymbol{H}_{\text{BMST}} = \text{diag}\{\underbrace{\boldsymbol{H}, \cdots, \boldsymbol{H}}_{L}, \underbrace{\boldsymbol{I}, \cdots, \boldsymbol{I}}_{m}\} \boldsymbol{P}^{\mathrm{T}},$$

where, the superscript T denotes “transpose” and

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{P}_1 & \boldsymbol{P}_2 & \cdots & \boldsymbol{P}_{L+m-1} \\ & \boldsymbol{I} & \boldsymbol{P}_1 & \cdots & \boldsymbol{P}_{L+m-2} \\ & & \ddots & \ddots & \vdots \\ & & & \boldsymbol{I} & \boldsymbol{P}_1 \\ & & & & \boldsymbol{I} \end{bmatrix}.$$

What code can be the basic code?

- What do we mean by short code?
- Can a random-generated linear code $[32, 16]$ be the basic code?
- Is BMST an LDPC code or a convolutional LDPC code?

What code can be the basic code?

- What do we mean by short code?
- Can a random-generated linear code $[32, 16]$ be the basic code?
- Is BMST an LDPC code or a convolutional LDPC code?
- Actually, we care about neither the generator matrix nor the parity-check matrix. The basic code can even be a *non-linear* code.

What code can be the basic code?

- What do we mean by short code?
- Can a random-generated linear code $[32, 16]$ be the basic code?
- Is BMST an LDPC code or a convolutional LDPC code?
- Actually, we care about neither the generator matrix nor the parity-check matrix. The basic code can even be a *non-linear* code.
- What do we really care about?

What we really care about is whether or not the basic code has efficient encoding/decoding algorithms.

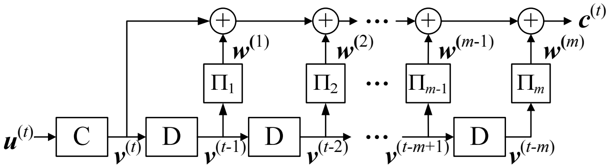


Figure: Encoding of BMST with memory m.

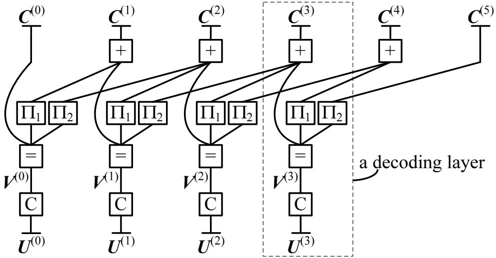


Figure: Sliding-window decoding over the normal graph.

Simulation Results

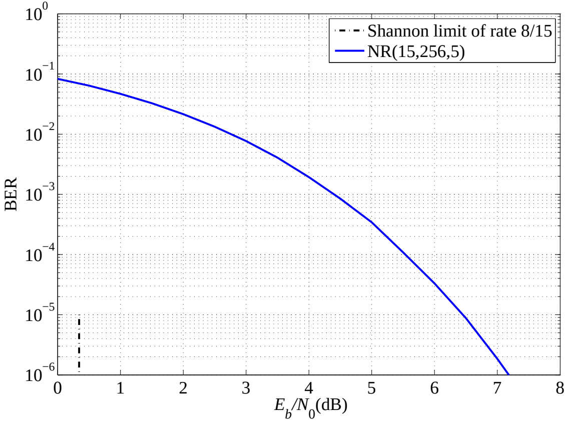


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in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

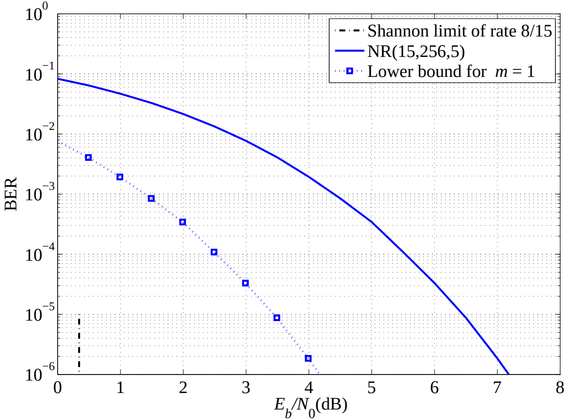


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in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

Simulation Results

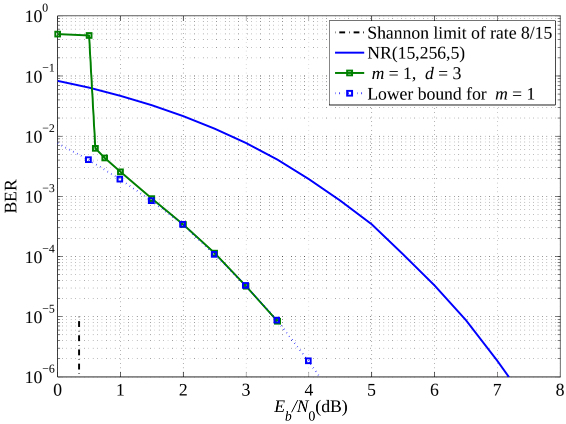


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in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

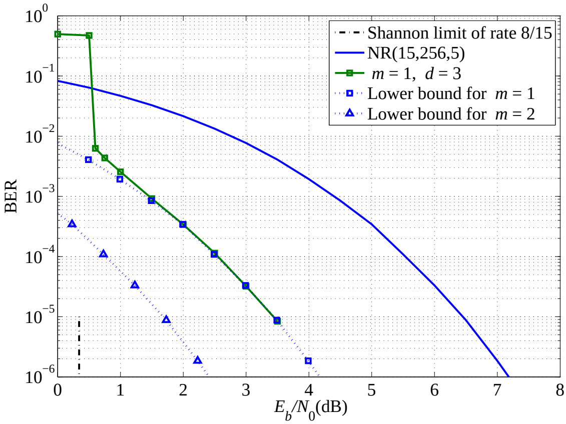


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in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

Simulation Results

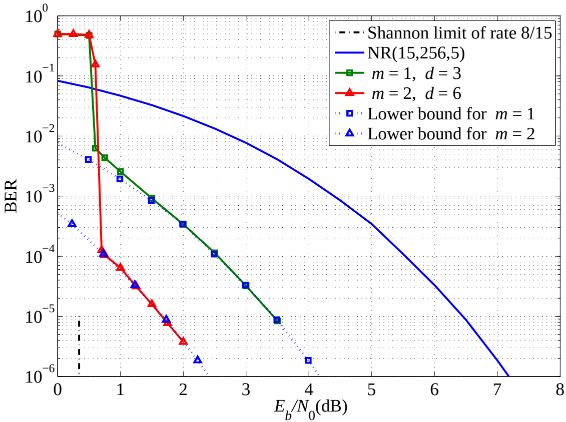


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in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

Simulation Results

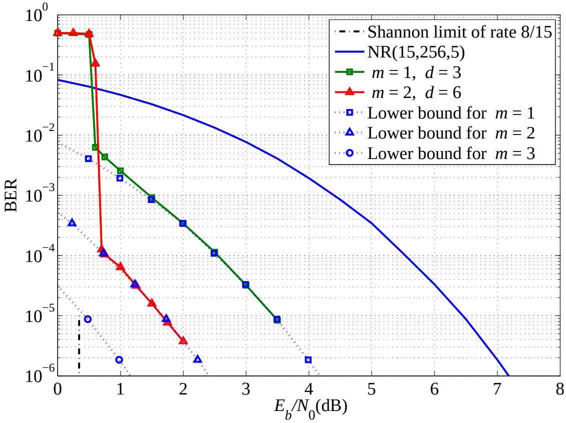


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in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

Simulation Results

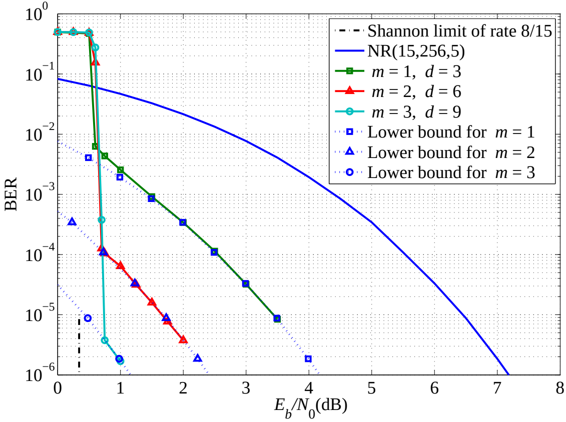


Figure: The basic code is the Cartesian product of the optimum Nordstrom-Robinson nonlinear code $(15, 256, 5)^{800}$. The system encodes $L = 1000$ sub-blocks of data and the iterative sliding-window decoding algorithm is performed, where the encoding memories and the decoding delays are specified in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

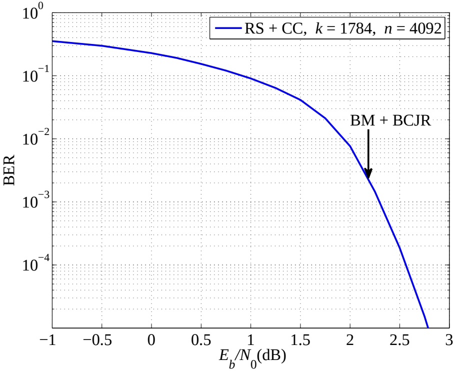


Figure: The basic code $\mathscr{C}$ is the Consultative Committee on Space Data System (CCSDS) standard code of dimension $k = 1784$ and length $n = 4092$, where the outer code is a [255, 223] Reed-Solomon (RS) code over $\mathbb{F}_{256}$ and the inner code is a terminated convolutional code with the 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 $I_{\max} = 18$.

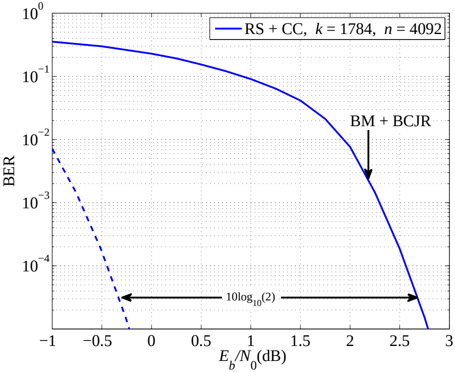


Figure: The basic code $\mathscr{C}$ is the Consultative Committee on Space Data System (CCSDS) standard code of dimension $k = 1784$ and length $n = 4092$, where the outer code is a [255, 223] Reed-Solomon (RS) code over $\mathbb{F}_{256}$ and the inner code is a terminated convolutional code with the 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 $I_{\max} = 18$.

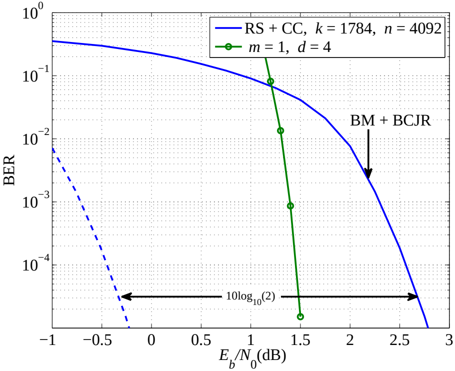


Figure: The basic code $\mathscr{C}$ is the Consultative Committee on Space Data System (CCSDS) standard code of dimension $k = 1784$ and length $n = 4092$, where the outer code is a [255, 223] Reed-Solomon (RS) code over $\mathbb{F}_{256}$ and the inner code is a terminated convolutional code with the 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 $I_{\max} = 18$.

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 it has efficient encoding algorithm and (exact or approximated) soft-in soft-out (SISO) decoding algorithm.

Acknowledgements

- This work is supported by the 973 Program (No.2012CB316100) and the NSF (No.61172082) of China.

Thank You for Your Attention!