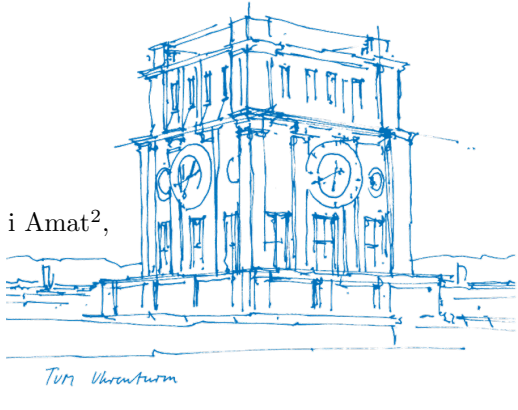
Sequential Decoding of Multiple Traces Over the Syndrome Trellis for Synchronization Errors

Anisha Banerjee[1]

Joint work with Lorenz Welter[1], Alexandre Graell i Amat[2], Antonia Wachter-Zeh[1] and Eirik Rosnes[3]

[1]Technical University of Munich, Germany
[2]Chalmers University of Technology, Sweden
[3]Simula UiB, Norway

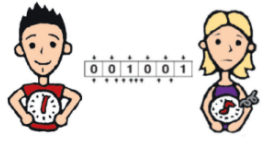
April 11, 2025



Outline

Motivation

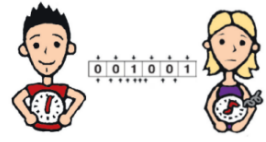
- Insertions and deletion errors occur
  - due to improper synchronization.



[HMG19] R. Heckel *et al.*, "A Characterization of the DNA Data Storage Channel," en, *Scientific Reports*, 2019
[F63] R. Fano, "A heuristic discussion of probabilistic decoding," *TIT*, 1963
[J69] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development*, 1969

Motivation

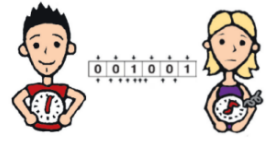
- Insertions and deletion errors occur
  - due to improper synchronization.
  - in molecular storage paradigms, e.g. DNA storage [HMG19].



[HMG19] R. Heckel *et al.*, "A Characterization of the DNA Data Storage Channel," en, *Scientific Reports*, 2019
[F63] R. Fano, "A heuristic discussion of probabilistic decoding," *TIT*, 1963
[J69] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development*, 1969

Motivation

- Insertions and deletion errors occur
  - due to improper synchronization.
  - in molecular storage paradigms, e.g. DNA storage [HMG19].

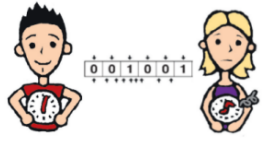


- May use **convolutional codes** for error correction.

[HMG19] R. Heckel *et al.*, "A Characterization of the DNA Data Storage Channel," en, *Scientific Reports*, 2019
[F63] R. Fano, "A heuristic discussion of probabilistic decoding," *TIT*, 1963
[J69] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development*, 1969

Motivation

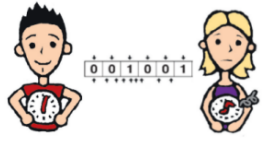
- Insertions and deletion errors occur
  - due to improper synchronization.
  - in molecular storage paradigms, e.g. DNA storage [HMG19].
- May use **convolutional codes** for error correction.
- Problem: High complexity of conventional decoders.



[HMG19] R. Heckel *et al.*, "A Characterization of the DNA Data Storage Channel," en, *Scientific Reports*, 2019
[F63] R. Fano, "A heuristic discussion of probabilistic decoding," *TIT*, 1963
[J69] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development*, 1969

Motivation

- Insertions and deletion errors occur
  - due to improper synchronization.
  - in molecular storage paradigms, e.g. DNA storage [HMG19].



- May use **convolutional codes** for error correction.

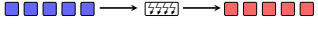
- Problem: High complexity of conventional decoders.

- Solution: Use ***sequential decoders***! [F63, J69]
  - Greedy-ish: only examine 'promising' codewords.
  - Sub-optimal, but fast.

[HMG19] R. Heckel *et al.*, "A Characterization of the DNA Data Storage Channel," en, *Scientific Reports*, 2019
[F63] R. Fano, "A heuristic discussion of probabilistic decoding," *TIT*, 1963
[J69] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development*, 1969

Channel Coding

- Most channels corrupt the transmitted message.



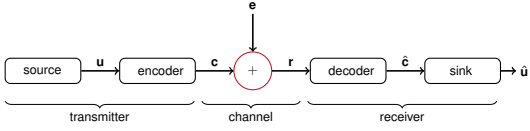
house	houke	error correctable
house	mouse	error undetectable

Channel Coding

- Most channels corrupt the transmitted message.

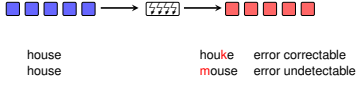
house	houke	error correctable
house	mouse	error undetectable

- Error-correcting codes to detect & correct errors at receiver.
  - Add redundancy to message intended for transmission.

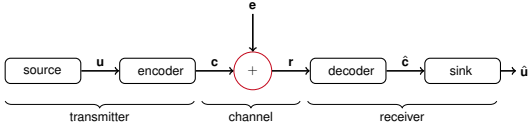


Channel Coding

- Most channels corrupt the transmitted message.



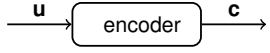
- Error-correcting codes to detect & correct errors at receiver.
  - Add redundancy to message intended for transmission.



- Ideal: low encoder & decoder complexity!

Convolutional Codes

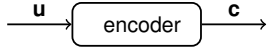
- Subclass of linear codes:



$$\boldsymbol{c} = \boldsymbol{u} \cdot \boldsymbol{G}$$

Convolutional Codes

- Subclass of linear codes:

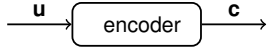


- The encoder of a (c, b, m) convolutional code

$$\boldsymbol{c} = \boldsymbol{u} \cdot \boldsymbol{G}$$

Convolutional Codes

- Subclass of linear codes:

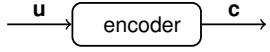


$$\boldsymbol{c} = \boldsymbol{u} \cdot \boldsymbol{G}$$

- The encoder of a (c, b, m) convolutional code
  - takes b input bits and

Convolutional Codes

- Subclass of linear codes:

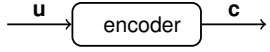


$$\boldsymbol{c} = \boldsymbol{u} \cdot \boldsymbol{G}$$

- The encoder of a (c, b, m) convolutional code
  - takes b input bits and
  - produces c output bits,

Convolutional Codes

- Subclass of linear codes:

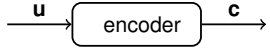


$$\boldsymbol{c} = \boldsymbol{u} \cdot \boldsymbol{G}$$

- The encoder of a (c, b, m) convolutional code
  - takes b input bits and
  - produces c output bits,
  - which depend on last $b(m+1)$ input bits (memory).

Convolutional Codes

- Subclass of linear codes:

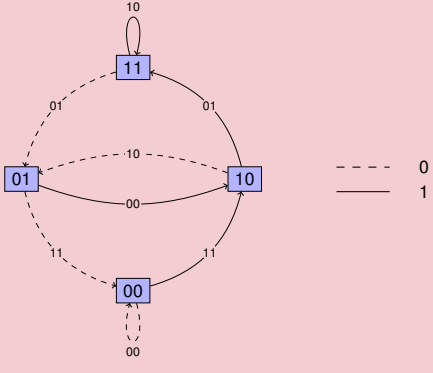


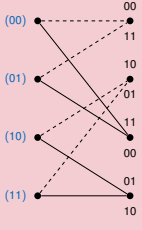
$$\boldsymbol{c} = \boldsymbol{u} \cdot \boldsymbol{G}$$

- The encoder of a (c, b, m) convolutional code
  - takes b input bits and
  - produces c output bits,
  - which depend on last $b(m+1)$ input bits (memory).
- Representable as a finite state machine with 2^m states.

Convolutional Codes: Encoding

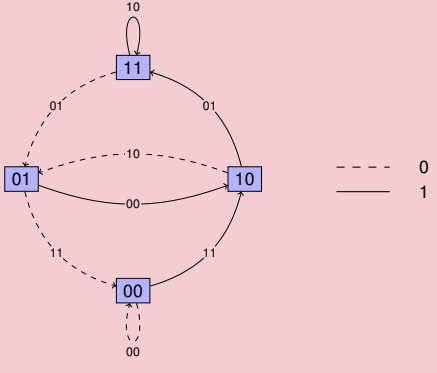
Example: [c=2,b=1,m=2] convolutional code

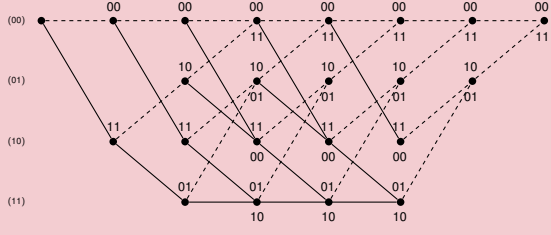




Convolutional Codes: Encoding

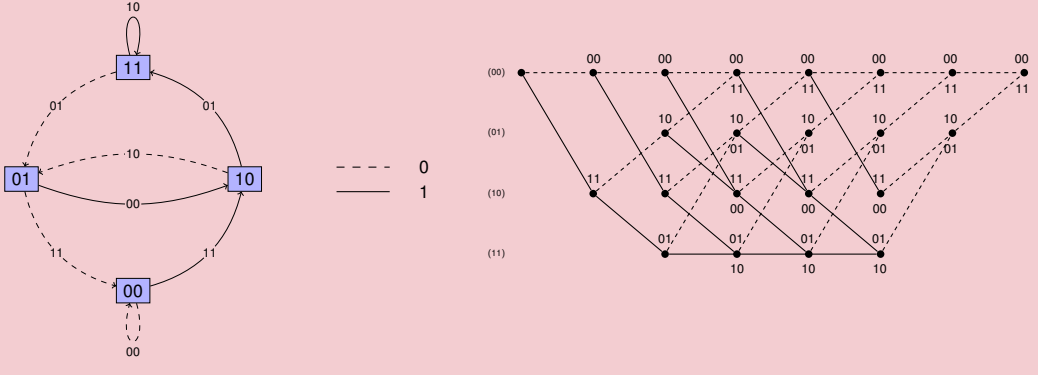
Example: [c=2,b=1,m=2] convolutional code





Convolutional Codes: Encoding

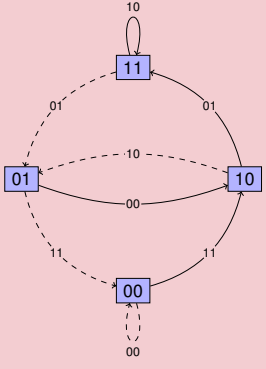
Example: [c=2,b=1,m=2] convolutional code

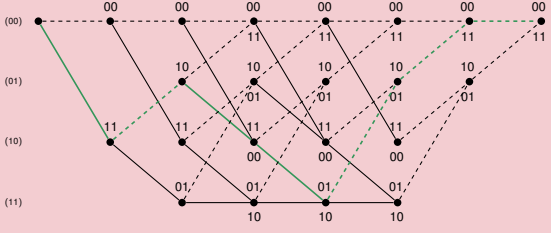


- Every path in trellis is a unique codeword.

Convolutional Codes: Encoding

Example: [c=2,b=1,m=2] convolutional code





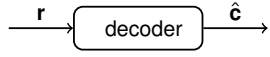
$$\boldsymbol{u} = (\ 1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0\ )$$

$$\boldsymbol{c} = (\ 11\ \ 10\ \ 00\ \ 01\ \ 01\ \ 11\ \ 00\ )$$

- Every path in trellis is a unique codeword.

Convolutional Codes: Decoding

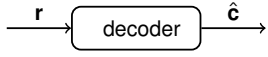
- Task of decoder: find $\hat{\boldsymbol{c}}$ closest to $\boldsymbol{r}$.



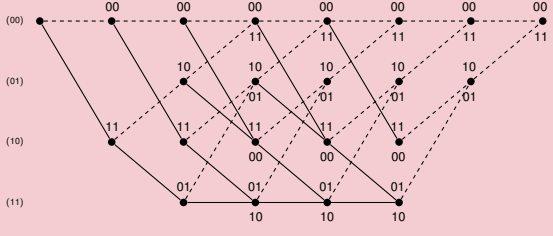
[BCJR74] L. Bahl *et al.*, "Optimal decoding of linear codes for minimizing symbol error rate," *TIT*, 1974

Convolutional Codes: Decoding

- Task of decoder: find $\hat{\boldsymbol{c}}$ closest to $\boldsymbol{r}$.



Example: [c=2,b=1,m=2] convolutional code

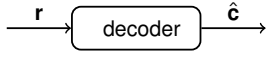


$$\boldsymbol{r} = (\ 11\ \ 10\ \ 00\ \ 00\ \ 01\ \ 11\ \ 00\ )$$

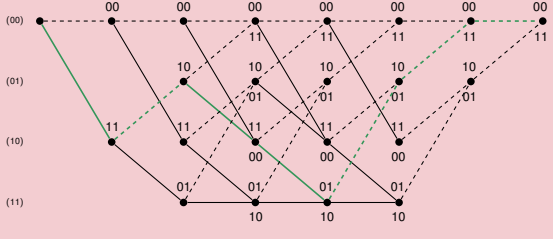
[BCJR74] L. Bahl *et al.*, "Optimal decoding of linear codes for minimizing symbol error rate," *TIT*, 1974

Convolutional Codes: Decoding

- Task of decoder: find $\hat{\boldsymbol{c}}$ closest to $\boldsymbol{r}$.



Example: [c=2,b=1,m=2] convolutional code



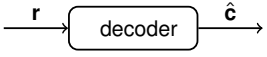
$$\boldsymbol{r} = (\ 11\ \ 10\ \ 00\ \ \underline{00}\ \ 01\ \ 11\ \ 00\ )$$

$$\hat{\boldsymbol{c}} = (\ 11\ \ 10\ \ 00\ \ \underline{01}\ \ 01\ \ 11\ \ 00\ )$$

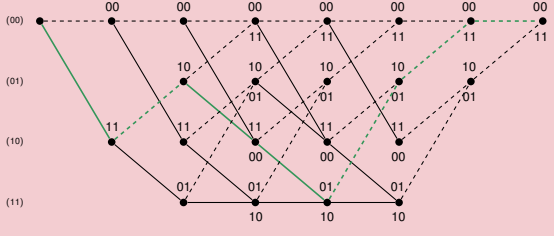
[BCJR74] L. Bahl *et al.*, "Optimal decoding of linear codes for minimizing symbol error rate," *TIT*, 1974

Convolutional Codes: Decoding

- Task of decoder: find $\hat{\boldsymbol{c}}$ closest to $\boldsymbol{r}$.



Example: [c=2,b=1,m=2] convolutional code



$$\boldsymbol{r} = (\ 11\ \ 10\ \ 00\ \ \underline{00}\ \ 01\ \ 11\ \ 00\ )$$

$$\hat{\boldsymbol{c}} = (\ 11\ \ 10\ \ 00\ \ \underline{01}\ \ 01\ \ 11\ \ 00\ )$$

- BCJR decoder yields $P(c_i|\boldsymbol{r})$.

$$\text{Complexity} \propto (\#\text{ Trellis states}) \cdot (\#\text{ Outgoing edges per state})$$
$$\propto 2^m \cdot 2^b$$

[BCJR74] L. Bahl *et al.*, "Optimal decoding of linear codes for minimizing symbol error rate," *TIT*, 1974

Channel Model

- Permits insertions, deletions and substitutions at random positions.

  → Denote as '**IDS channel**'.

[DM01] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *TIT*, 2001

Channel Model

- Permits insertions, deletions and substitutions at random positions.

  $\rightarrow$ Denote as '**IDS channel**'.

- Specified by insertion, deletion and substitution probabilities, i.e., p_{I}, p_{D} and p_{S}.

[DM01] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *TIT*, 2001

Channel Model

- Permits insertions, deletions and substitutions at random positions.

  $\rightarrow$ Denote as '**IDS channel**'.

- Specified by insertion, deletion and substitution probabilities, i.e., p_{I}, p_{D} and p_{S}.
- Modeled as **finite-state machine** [DM01].
  - Let channel input be $\boldsymbol{x} = (x_1, \ldots, x_T)$.
  - When x_i awaits transmission, four events possible.

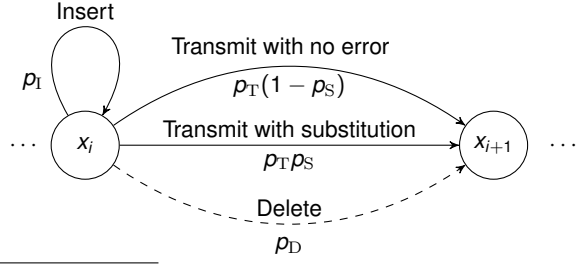
[DM01] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *TIT*, 2001

Channel Model

- Permits insertions, deletions and substitutions at random positions.

  $\rightarrow$ Denote as '**IDS channel**'.

- Specified by insertion, deletion and substitution probabilities, i.e., p_{I}, p_{D} and p_{S}.
- Modeled as **finite-state machine** [DM01].
  - Let channel input be $\boldsymbol{x} = (x_1, \ldots, x_T)$.
  - When x_i awaits transmission, four events possible.



[DM01] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *TIT*, 2001

Adapting decoders

- Aim: To account for insertions & deletions, use '**drift state**' [DM01, BF15].
  $\rightarrow$ Let d_t = #received bits $-$ #transmitted bits, at time t.

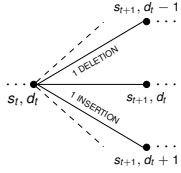
[DM01] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *TIT*, 2001

[BF15] V. Buttigieg and N. Farrugia, "Improved bit error rate performance of convolutional codes with synchronization errors," in *Proc. Int. Conf. Comm.*, 2015

[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023

Adapting decoders

- Aim: To account for insertions & deletions, use '**drift state**' [DM01, BF15].
  → Let d_t = #received bits − #transmitted bits, at time t.



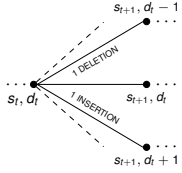
[DM01] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *TIT*, 2001

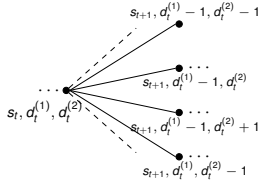
[BF15] V. Buttigieg and N. Farrugia, "Improved bit error rate performance of convolutional codes with synchronization errors," in *Proc. Int. Conf. Comm.*, 2015

[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023

Adapting decoders

- Aim: To account for insertions & deletions, use '**drift state**' [DM01, BF15].
  → Let d_t = #received bits − #transmitted bits, at time t.





- Decoding $M > 1$ received sequences jointly. [MLWRA23]

[DM01] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *TIT*, 2001

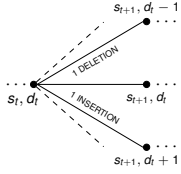
[BF15] V. Buttigieg and N. Farrugia, "Improved bit error rate performance of convolutional codes with synchronization errors," in *Proc. Int. Conf. Comm.*, 2015

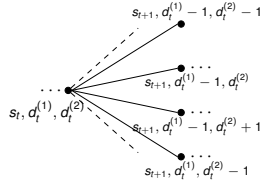
[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023

Adapting decoders

- Aim: To account for insertions & deletions, use '**drift state**' [DM01, BF15].

  $\rightarrow$ Let d_t = #received bits $-$ #transmitted bits, at time t.





- Decoding $M > 1$ received sequences jointly. [MLWRA23]

$$\text{Complexity of BCJR decoding} \propto (\#\text{ Trellis states}) \cdot (\#\text{ Outgoing edges per state})$$
$$\propto (2^m D^M) \cdot (2^b \delta^M),$$

where $D = \#$drift states and $\delta = \#$insdels per edge.

[DM01] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *TIT*, 2001

[BF15] V. Buttigieg and N. Farrugia, "Improved bit error rate performance of convolutional codes with synchronization errors," in *Proc. Int. Conf. Comm.*, 2015

[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023

Reducing decoder complexity

$$\text{Complexity} \propto 2^{m+b} \cdot D^M \cdot \delta^M$$

[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023
[BLW24] A. Banerjee *et al.*, "Sequential decoding of multiple sequences for synchronization errors," *TCOMM*, 2024

Reducing decoder complexity

$$\text{Complexity} \propto 2^{m+b} \cdot D^M \cdot \delta^M$$

- **Separate-BCJR** [MLWRA23]
  - For each received sequence $\boldsymbol{y}_i$, computes $P(c_j|\boldsymbol{y}_i)$.
  - Aggregates output of all M decoding operations.

$$\text{Complexity} \propto M \cdot 2^{m+b} \cdot D \cdot \delta$$

[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023
[BLW24] A. Banerjee *et al.*, "Sequential decoding of multiple sequences for synchronization errors," *TCOMM*, 2024

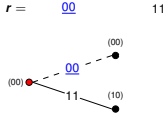
Reducing decoder complexity

$$\text{Complexity} \propto 2^{m+b} \cdot D^M \cdot \delta^M$$

- **Separate-BCJR** [MLWRA23]
  - For each received sequence $\boldsymbol{y}_i$, computes $P(c_j|\boldsymbol{y}_i)$.
  - Aggregates output of all M decoding operations.

$$\text{Complexity} \propto M \cdot 2^{m+b} \cdot D \cdot \delta$$

- **Joint sequential decoding** [BLW24]
  - Suboptimal, greedy search in a *tree*.



[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023

[BLW24] A. Banerjee *et al.*, "Sequential decoding of multiple sequences for synchronization errors," *TCOMM*, 2024

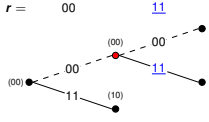
Reducing decoder complexity

$$\text{Complexity} \propto 2^{m+b} \cdot D^M \cdot \delta^M$$

- **Separate-BCJR** [MLWRA23]
  - For each received sequence $\boldsymbol{y}_i$, computes $P(c_j|\boldsymbol{y}_i)$.
  - Aggregates output of all M decoding operations.

$$\text{Complexity} \propto M \cdot 2^{m+b} \cdot D \cdot \delta$$

- **Joint sequential decoding** [BLW24]
  - Suboptimal, greedy search in a *tree*.



[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023

[BLW24] A. Banerjee *et al.*, "Sequential decoding of multiple sequences for synchronization errors," *TCOMM*, 2024

Reducing decoder complexity

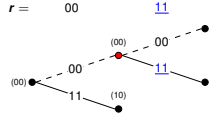
$$\text{Complexity} \propto 2^{m+b} \cdot D^M \cdot \delta^M$$

- **Separate-BCJR** [MLWRA23]
  - For each received sequence $\boldsymbol{y}_i$, computes $P(c_j|\boldsymbol{y}_i)$.
  - Aggregates output of all M decoding operations.

$$\text{Complexity} \propto M \cdot 2^{m+b} \cdot D \cdot \delta$$

- **Joint sequential decoding** [BLW24]
  - Suboptimal, greedy search in a *tree*.

$$\text{Complexity} \propto 2^b \cdot \delta^M$$



[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023

[BLW24] A. Banerjee *et al.*, "Sequential decoding of multiple sequences for synchronization errors," *TCOMM*, 2024

Reducing decoder complexity

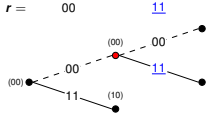
$$\text{Complexity} \propto 2^{m+b} \cdot D^M \cdot \delta^M$$

- **Separate-BCJR** [MLWRA23]
  - For each received sequence $\boldsymbol{y}_i$, computes $P(c_j|\boldsymbol{y}_i)$.
  - Aggregates output of all M decoding operations.

$$\text{Complexity} \propto M \cdot 2^{m+b} \cdot D \cdot \delta$$

- **Joint sequential decoding** [BLW24]
  - Suboptimal, greedy search in a *tree*.

$$\text{Complexity} \propto 2^b \cdot \delta^M$$



- Problem: Too complex for high-rate codes.
  - Solution: Alternate representation of tree/trellis!

[MLWRA23] I. Maarouf *et al.*, "Concatenated codes for multiple reads of a dna sequence," *TIT*, 2023
[BLW24] A. Banerjee *et al.*, "Sequential decoding of multiple sequences for synchronization errors," *TCOMM*, 2024

Syndrome Trellis

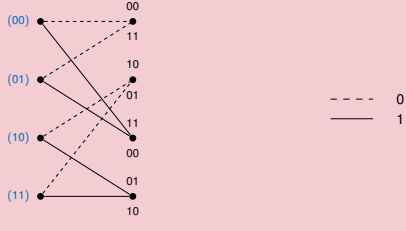
- Based on the dual space of the convolutional code. [S94]
  - Each edge denotes one codeword symbol $\rightarrow \leq 2$ outgoing edges per node.

[S94] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *TIT*, 1994

Syndrome Trellis

- Based on the dual space of the convolutional code. [S94]
  - Each edge denotes one codeword symbol $\rightarrow \leq 2$ outgoing edges per node.

Example: [c=2,b=1,m=2] convolutional code

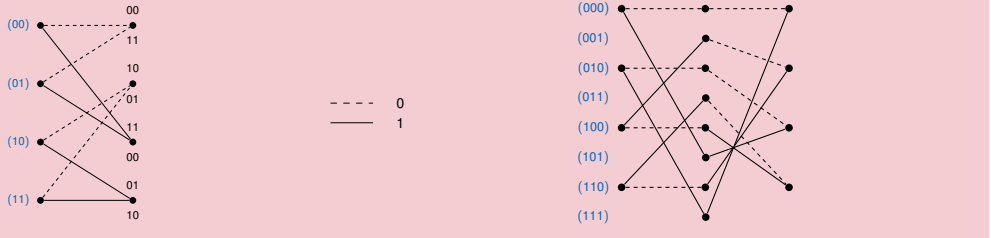


[S94] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *TIT*, 1994

Syndrome Trellis

- Based on the dual space of the convolutional code. [S94]
  - Each edge denotes one codeword symbol $\rightarrow \leq 2$ outgoing edges per node.

Example: [c=2,b=1,m=2] convolutional code

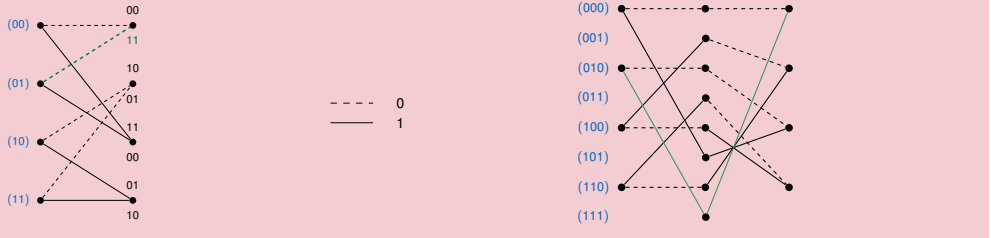


[S94] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *TIT*, 1994

Syndrome Trellis

- Based on the dual space of the convolutional code. [S94]
  - Each edge denotes one codeword symbol $\rightarrow \leq 2$ outgoing edges per node.

Example: [c=2,b=1,m=2] convolutional code

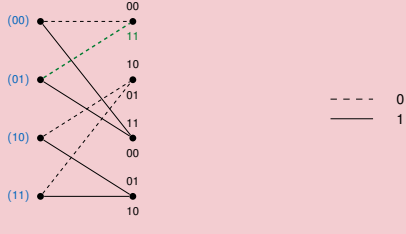


[S94] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *TIT*, 1994

Syndrome Trellis

- Based on the dual space of the convolutional code. [S94]
  - Each edge denotes one codeword symbol $\rightarrow \leq 2$ outgoing edges per node.

Example: [c=2,b=1,m=2] convolutional code



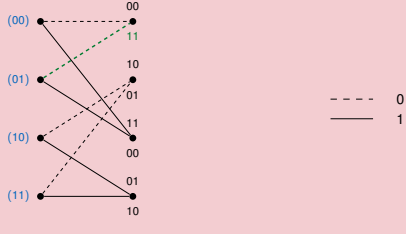
- Complexity comparison
  - **Separate-BCJR**: $M \cdot 2^{m+b} \cdot D \cdot \delta$
  - **Joint sequential decoding**: $2^b \cdot \delta^M$

[S94] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *TIT*, 1994

Syndrome Trellis

- Based on the dual space of the convolutional code. [S94]
  - Each edge denotes one codeword symbol $\rightarrow \leq 2$ outgoing edges per node.

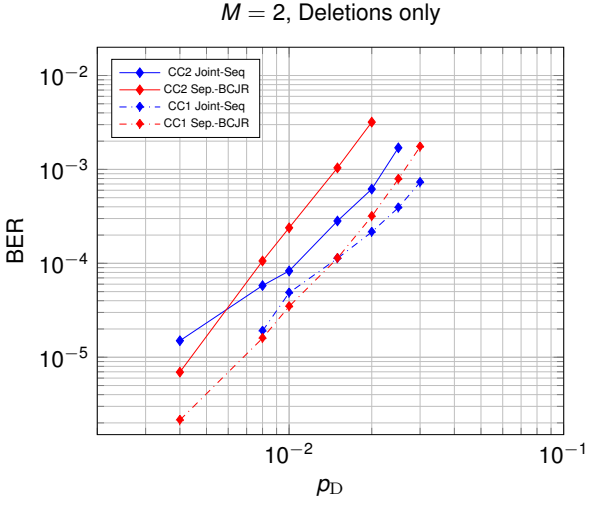
Example: [c=2,b=1,m=2] convolutional code



- Complexity comparison
  - **Separate-BCJR**: $M \cdot 2^{m+b} \cdot D \cdot \delta \rightarrow M \cdot 2^{m'+1} \cdot D \cdot \delta$
  - **Joint sequential decoding**: $2^b \cdot \delta^M \rightarrow 2 \cdot \delta^M$

[S94] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *TIT*, 1994

Bit Error Rates (BER)

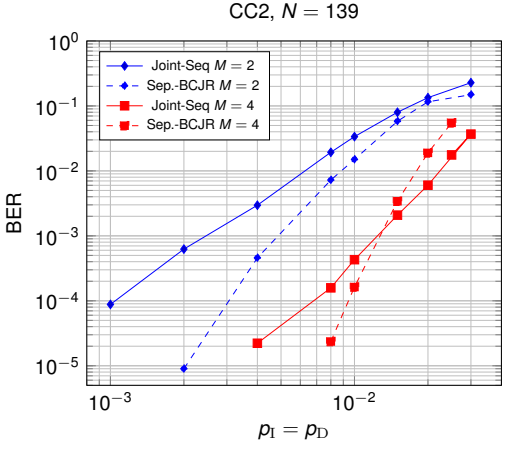


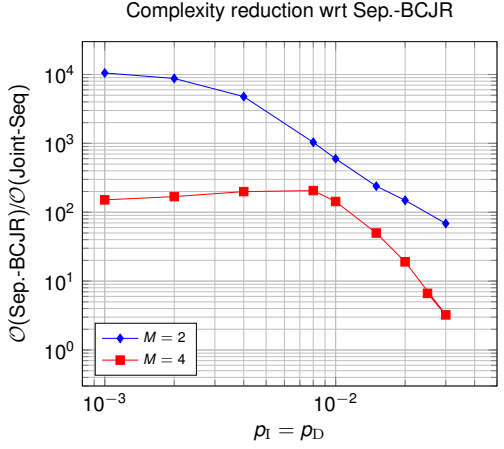
Code	$[c, b]$	d_{free}
CC1	$[10, 7]$	6
CC2	$[11, 9]$	5

Table: Code Parameters [RY04]

[RY04] E. Rosnes and Ø. Ytrehus, "On maximum length convolutional codes under a trellis complexity constraint," *Journal of Complexity*, 2004

BER & Complexity





Conclusion

- Joint sequential decoding v/s separate-BCJR decoding.
  - Under low-noise, the latter wins.
  - At higher noise levels, the former can win.

Conclusion

- Joint sequential decoding v/s separate-BCJR decoding.
  - Under low-noise, the latter wins.
  - At higher noise levels, the former can win.
- Future work: theoretical analysis of computational complexity.

Conclusion

- Joint sequential decoding v/s separate-BCJR decoding.
  - Under low-noise, the latter wins.
  - At higher noise levels, the former can win.
- Future work: theoretical analysis of computational complexity.

Thank you!

References I

[1] R. Heckel, G. Mikutis, and R. N. Grass, "A Characterization of the DNA Data Storage Channel," en, *Scientific Reports*, vol. 9, no. 1, p. 9663, Dec. 2019.

[2] R. Fano, "A heuristic discussion of probabilistic decoding," *TIT*, vol. 9, no. 2, pp. 64–74, Apr. 1963.

[3] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development*, vol. 13, no. 6, pp. 675–685, Nov. 1969.

[4] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *TIT*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[5] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *TIT*, vol. 47, no. 2, pp. 687–698, Feb. 2001.

[6] V. Buttigieg and N. Farrugia, "Improved bit error rate performance of convolutional codes with synchronization errors," in *Proc. Int. Conf. Comm.*, Jun. 2015, pp. 4077–4082.

[7] I. Maarouf, A. Lenz, L. Welter, A. Wachter-Zeh, E. Rosnes, and A. G. i Amat, "Concatenated codes for multiple reads of a dna sequence," *TIT*, vol. 69, no. 2, pp. 910–927, Feb. 2023.

References II

[8] A. Banerjee, A. Lenz, and A. Wachter-Zeh, "Sequential decoding of multiple sequences for synchronization errors," *TCOMM*, vol. 72, no. 11, pp. 6660–6676, Nov. 2024.

[9] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *TIT*, vol. 40, no. 5, pp. 1663–1666, Sep. 1994.

[10] E. Rosnes and Ø. Ytrehus, "On maximum length convolutional codes under a trellis complexity constraint," *Journal of Complexity*, vol. 20, no. 2-3, pp. 372–403, Apr. 2004.