Analysis of Frame Error Rate (FER) and Bit Error Rate (BER) of Viterbi Decoding with Periodic Puncturing

Shakeh Kalantarmoradian[1], Alethea Sung-Miller[1]

[1]ECE Department, UCLA; Communications Systems Laboratory (CSL)

Research Mentors: Richard D. Wesel, Alexander Baldauf

Introduction

When transmitting data over a noisy channel, the noise can cause information bits to be distorted at the receiver, resulting in the receiver selecting the wrong codeword. Error correcting codes add bits for redundancy to enable detection and correction of errors at the receiver.

This project uses a rate-$\frac{1}{3}$ 64-state 8PSK-modulated trellis code, which sends three encoded bits as one symbol per information bit. Periodic puncturing – intentional omission of certain symbols – has been implemented to reduce the number of symbols sent, where patterns are denoted as ones and zeros (zeros indicate omitted symbols).

This project has confirmed the results of previous papers by investigating the characteristics of the trellis encoder, as well as simulating the bit error rate in comparison to truncated and non-truncated bit error rate (BER) union bounds. Ongoing progress includes developing union bound equations for the frame error rate (FER).

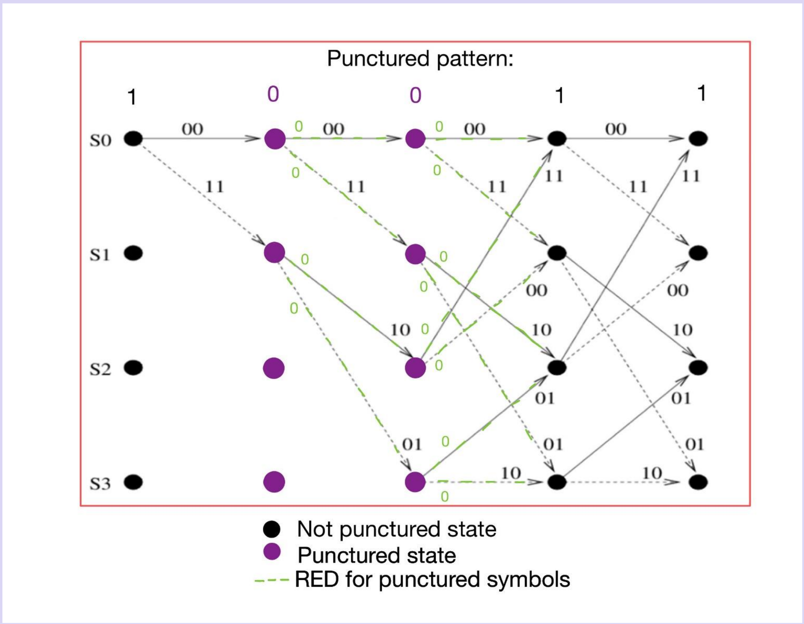

Fig. 1 – Four-state rate-$\frac{1}{2}$ punctured Viterbi diagram.

Method

Implement a Viterbi algorithm using C++.

Find residual Euclidean distance (minimum Euclidean distance between any two codewords of infinite length) for specific puncturing patterns.

Run simulations using UCLA's Hoffman2 Cluster to find BER vs. SNR for punctured patterns and FER vs. SNR for non-punctured pattern.

Plot BER truncated union bound for all patterns and true union bound for non-punctured pattern using MATLAB; compare with the simulation results.

The truncated union bound is $N_b(\tilde{a})Q\left(\sqrt{RED^2(\tilde{a})\varepsilon_x/(2N_0)}\right)$ where $N_b(\tilde{a})$ is the number of bits that are incorrect among the nearest neighbor paths (paths that share the same residual Euclidean distance, or RED). ε_x/N_0 is the magnitude of SNR.

The union bound equation used for the non-punctured pattern is $\frac{1}{kp}\,Q(\sqrt{RED^2(\tilde{a})\varepsilon_x/(2N_0)}e^{RED^2(\tilde{a})\varepsilon_x/(4N_0)}\,\frac{\partial T_p(W_1,\ldots,W_p,I)}{\partial I}$, where $I=1, W_j = e^{-|\tilde{a}_j|^2/(4N_0)}$.

Analyses

For lower SNRs, the data lies above the truncated union bound. This is expected, as the truncated union bound is an approximation of the true union bound. At higher SNRs, the truncated union bound converges to the true union bound and acts like an upper bound to the data. Interestingly, the truncated and true union bounds match better for more aggressive puncturing patterns. Similarly, the truncated union bounds and the data seem to approach each other much more quickly for more aggressively punctured patterns.

Results and Conclusion

Punctured Pattern	Code 13 RED	Code 14 RED
11111	4.11	4.21
01111	3.03	3.25
11110	3.43	3.63
11101	3.51	3.68
11011	3.66	3.25
10111	3.31	3.22
00111	2.08	2.08
01110	2.27	2.27
11100	3.03	3.03
11001	3.03	3.16
10011	3.03	2.57
01011	2.40	2.40
10110	2.79	2.79
01101	2.52	2.40
11010	2.93	2.93
10101	2.89	2.79
00101	1.33	1.33
01010	1.33	1.61
10100	1.78	2.27
01001	2.57	1.78
10010	2.27	2.14

Fig. 2 – Residual Euclidean distances of Codes 13 and 14 shown in columns 2 and 3 respectively for each of the punctured patterns listed under column 1.

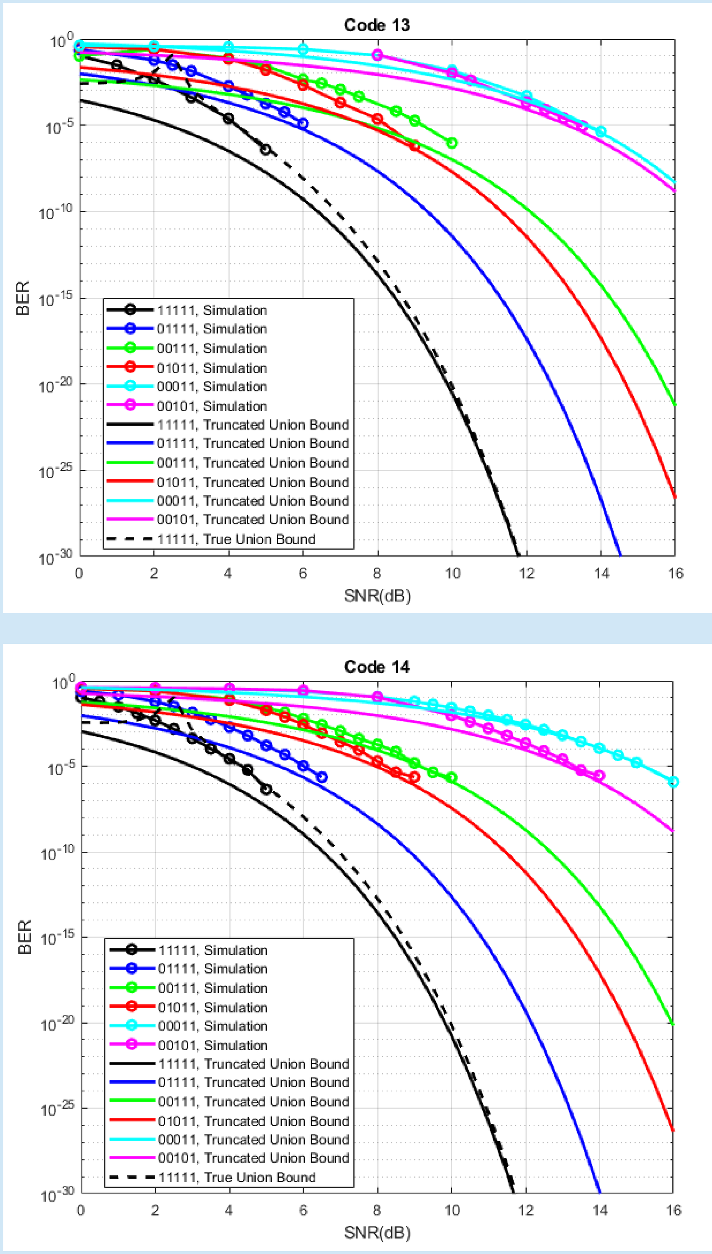

Fig. 3 – Truncated union bound (solid lines) vs. collected data (lines with circular points) vs. true union bound for non-punctured pattern (dashed line) for a) Code 13 and b) Code 14.

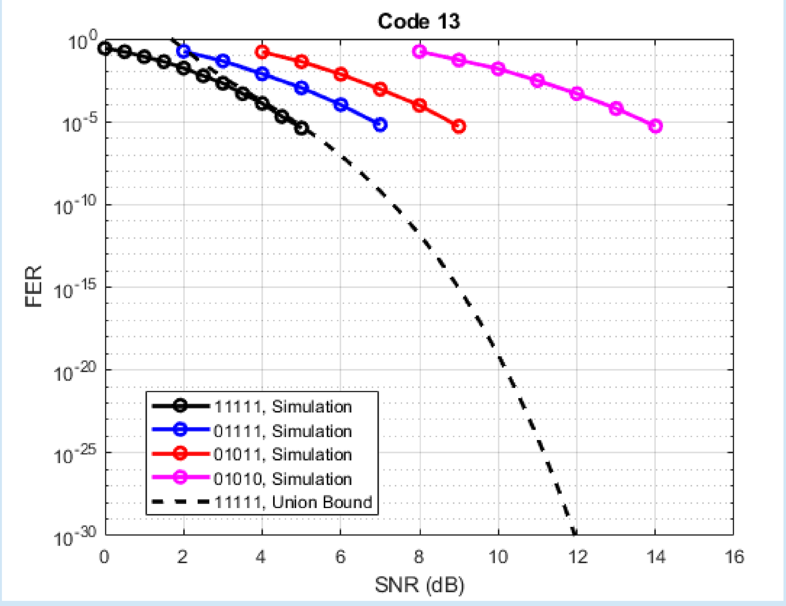

Fig. 4 – Collected data (lines with circular points) and FER equation for non-punctured pattern (dashed line) for Code 13.

Code 13 vs. Code 14:

- Rate-$\frac{1}{3}$ 8PSK codes with period 5.
- Code 13 maximizes $\sum_j \log(RED_j^{\,2})$ for each periodic pattern j considered in the channel [4].
- Code 14 is designed to perform better over progressive puncturing patterns (11111 01111 01011 01010).

Key findings:

- Residual Euclidean distances for both codes 13 and 14 match values reported in [4].
- As the signal to noise ratio (SNR) increases, the bit error rate (BER) curves from our simulations and the truncated union bound curves converge for both codes 13 and 14.
- Code 13 performs as well or better than Code 14, except for 01111. Code 13 is better for 00111 and 00011, but neither of these appear in our progressive puncturing patterns.

These results indicate:

- Though the literature notes that Code 14 performs slightly better than Code 13 for progressive puncturing patterns, the advantages of Code 14 are minor.
- The truncated union bound provides a good expectation of the true union bound at high SNRs. At lower SNRs the truncated union bound is considerably lower than the true union bound.
- Though the true union bound is a good match for the non-punctured pattern data, we are still finishing developing the true union bound for punctured patterns.

Developing the FER union bound:

- FER equations based on the bit error rate union bounds developed in [1], [2], and [3].
- Dan Song (CSL, UCLA) adapted the BER true union bound for FER.
- This project has adapted this general FER union bound to the case of puncturing, as follows:

$$FER \le Q(\sqrt{RED^2(\tilde{a})\varepsilon_x/(2N_0)}\; e^{RED^2(\tilde{a})\varepsilon_x/(4N_0)} \sum_{l=v+1}^{L} T_l(W)$$

$$T_l(\mathrm{W}) = \left[d_{\tilde{a}_0} c_{\tilde{a}_0}\right]\Big|_{W=e^{\frac{-|\tilde{a}_0|^2}{4N_0}}} \prod_{j=1}^{l-2}\left(\begin{bmatrix} d_{\tilde{a}_j} & c_{\tilde{a}_j} \\ b_{\tilde{a}_j} & A_{\tilde{a}_j}\end{bmatrix}\Bigg|_{W=e^{-\frac{|\tilde{a}_j|^2}{4N_0}}}\right)\begin{bmatrix} d^*_{\tilde{a}_{l-1}} \\ b_{\tilde{a}_{l-1}}\end{bmatrix}\Bigg|_{W=e^{\frac{|\tilde{a}_{l-1}|^2}{4N_0}}}$$

$$d^*_{\tilde{a}_j} = d_{\tilde{a}_j} - 1$$

$\tilde{a}_j = \begin{cases} 1, \textit{not punctured} \\ 0, \textit{punctured} \end{cases}$

B is a vector of punctured pattern symbols with length p

v is the encoder memory

$\begin{bmatrix} d & c \\ b & A \end{bmatrix}$ is the reduced state transition matrix

Next steps:

- Add feedback using the reliability output Viterbi algorithm (ROVA).
- Identify best order of punctured symbols to send during incremental retransmissions.
- **Add list decoding:**
  - List decoding ranks the most likely decoded sequences for the received codeword.
  - Correction involves comparing to all these options (which is inefficient).
- At some point, entries in the list are unnecessary: they create more errors than they correct.
- Imposing a maximum list length will avoid unnecessary entries, increasing efficiency.

References

- [1] R. D. Wesel and X. Liu, "Analytic Techniques for Periodic Trellis Codes," 36th Annual Allerton Conference on Communication, Control, and Computing, September 23-25, 1998.
- [2] R. D. Wesel, "Reduced-State Representations for Trellis Codes Using Constellation Symmetry," *IEEE Transactions on Communications*, vol. 52, no. 8, August 2004.
- [3] R. D. Wesel, "Reduced Complexity Trellis Code Transfer Function Computation," Communication Theory Mini-Conference at ICC '99, June 6-10, 1999.
- [4] R. D. Wesel, X. Liu, and W. Shi, "Trellis Codes for Periodic Erasures," *IEEE Transactions on Communications*, vol. 48, no. 6, June 2000.

Acknowledgments

We would like to give special thanks to Dean Wesel and Alexander Baldauf for their able guidance and support in conducting this research.

We would like to also thank the National Science Foundation (NSF) for funding our research project through the UCLA Summer Undergraduate Research Program (SURP).