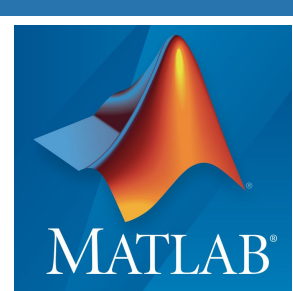


## Introduction

- Transmitting data across noisy channels using high rate convolutional codes and standard maximum likelihood decoding on the encoder-based trellis incurs high decoding costs. To improve efficiency, we implement the dual trellis proposed in Yamada et al. [1], which is based on the parity check matrix of the original code, and improves efficiency for high rate decoding by reducing the number of comparisons performed at each state in the trellis.
- To improve robustness, we built off the work of Liang et al. [5], who demonstrated that for low rate convolutional codes (of the form  $1/n$ ), using distance spectrum optimal cyclic redundancy checks (DSO CRCs), in conjunction with serial list decoding, provided significant improvements to signal to noise ratio (SNR) at a fairly low cost.
- Our work extends this approach to high rate codes (of the form  $(n-1)/n$ ), to identify if such an approach is viable for high rate codes as well.

## Materials



We implemented the dual trellis list Viterbi algorithm with DSO CRCs in C++, and plotted the results in Matlab.



## Methods

- We started by building the maximum likelihood high rate tail-biting decoder, then incorporated the dual trellis, and finally implemented list decoding.
- Fig. 1 on the right outlines the process of evaluating the system.
- For each SNR, we ran trials until we reached 200 frame errors, so our FER would have a statistically significant number of errors.

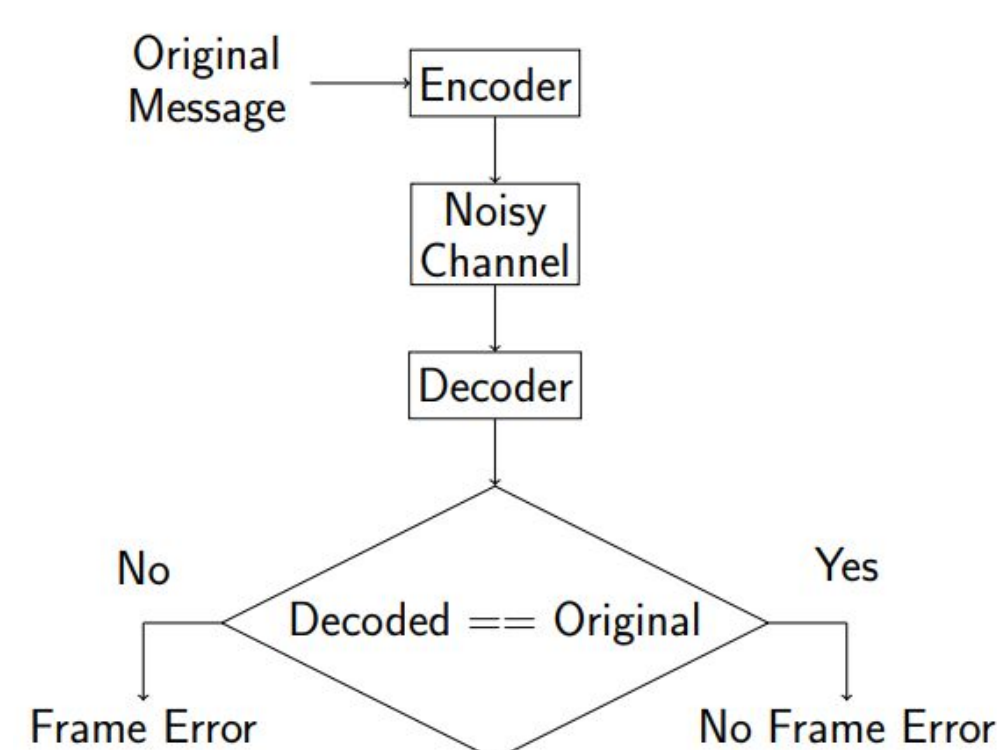


Figure 1: Evaluating performance of the system

## Convolutional Codes

- Convolutional codes use memory elements in the encoding process to add redundancy to the encoded signal, which can be used to correct errors in the received message.
- Since the outputs are a function of the memory elements and inputs, this can be viewed as a state machine, which we can arrange on a trellis to account for multiple input blocks. For brevity, only one state transition is shown in Fig. 2.

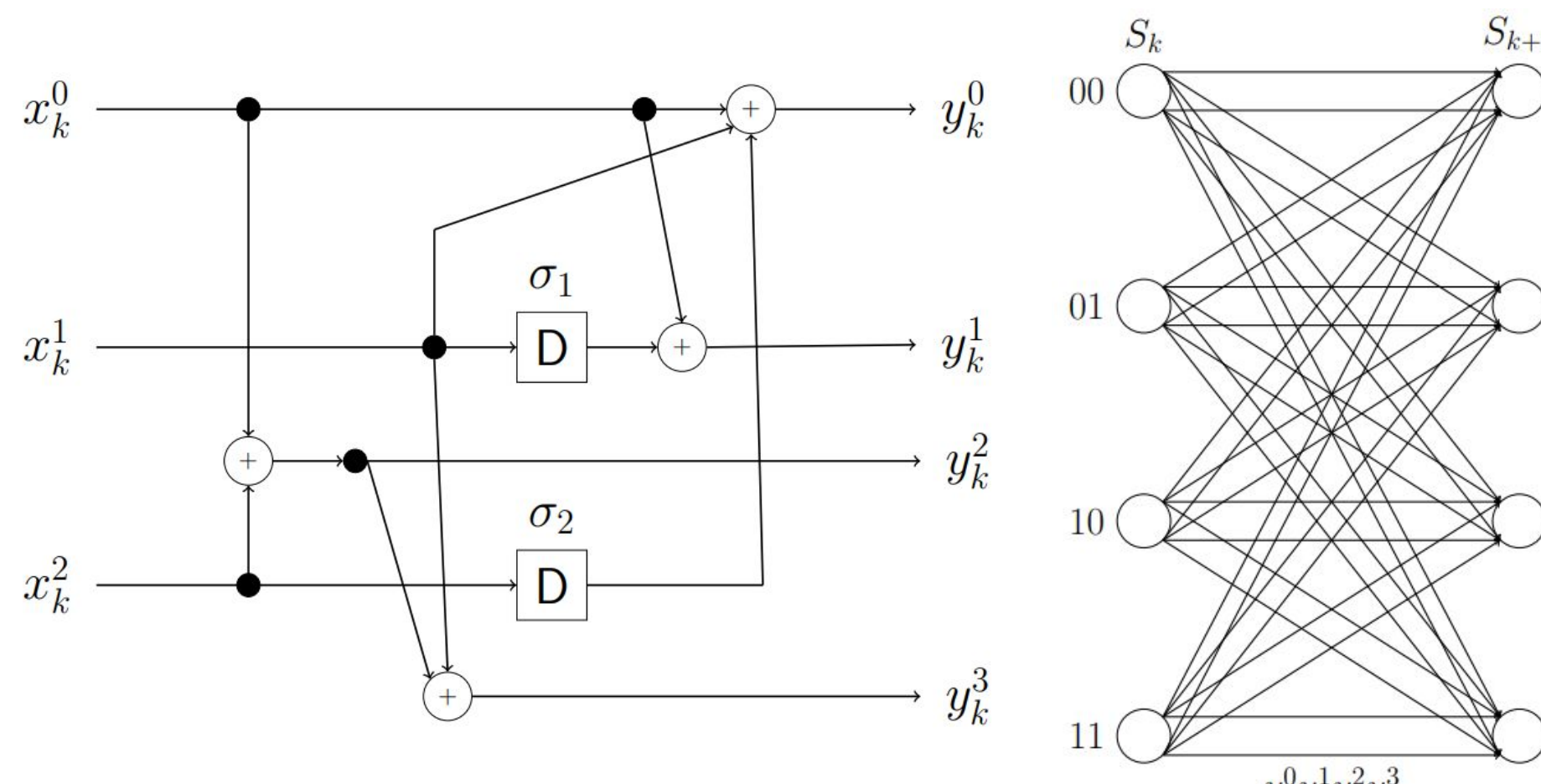


Figure 2: An example of a rate-3/4 convolutional encoder and the corresponding trellis. Each adder is modulo 2, so each value will remain a binary 1 or 0. This encoder has two memory elements, denoted by sigmas.

## Dual Trellis

- For standard maximum likelihood decoding, we want to find the message closest to the received message by comparing all possible paths through the trellis, however, directly enumerating them all is computationally intractable.
- The Viterbi algorithm efficiently finds this sequence by only storing the locally optimal path at each state, upper bounding the number of paths to keep track of at the number of states.

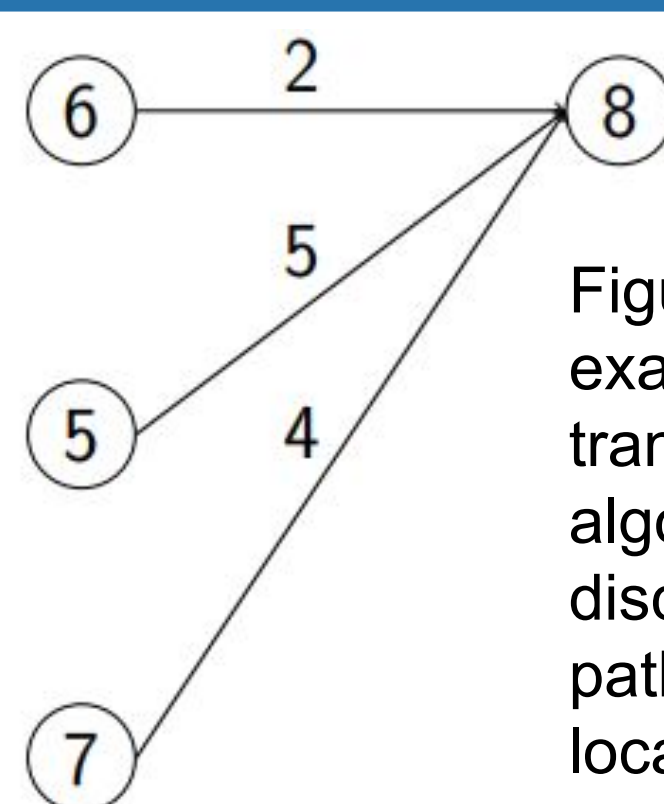


Figure 3: In this small example of one state transition, the Viterbi algorithm would discard the lower paths, since they are locally worse than the top path.

- While highly efficient relative to the naive approach, the Viterbi algorithm on the standard trellis still struggles with high rate codes due to the high number of comparisons needed at each state.
- Using the dual trellis, which is based on the parity check matrix of the code, reduces the number of comparisons needed at each state, improving efficiency despite the larger number of states and state transitions.

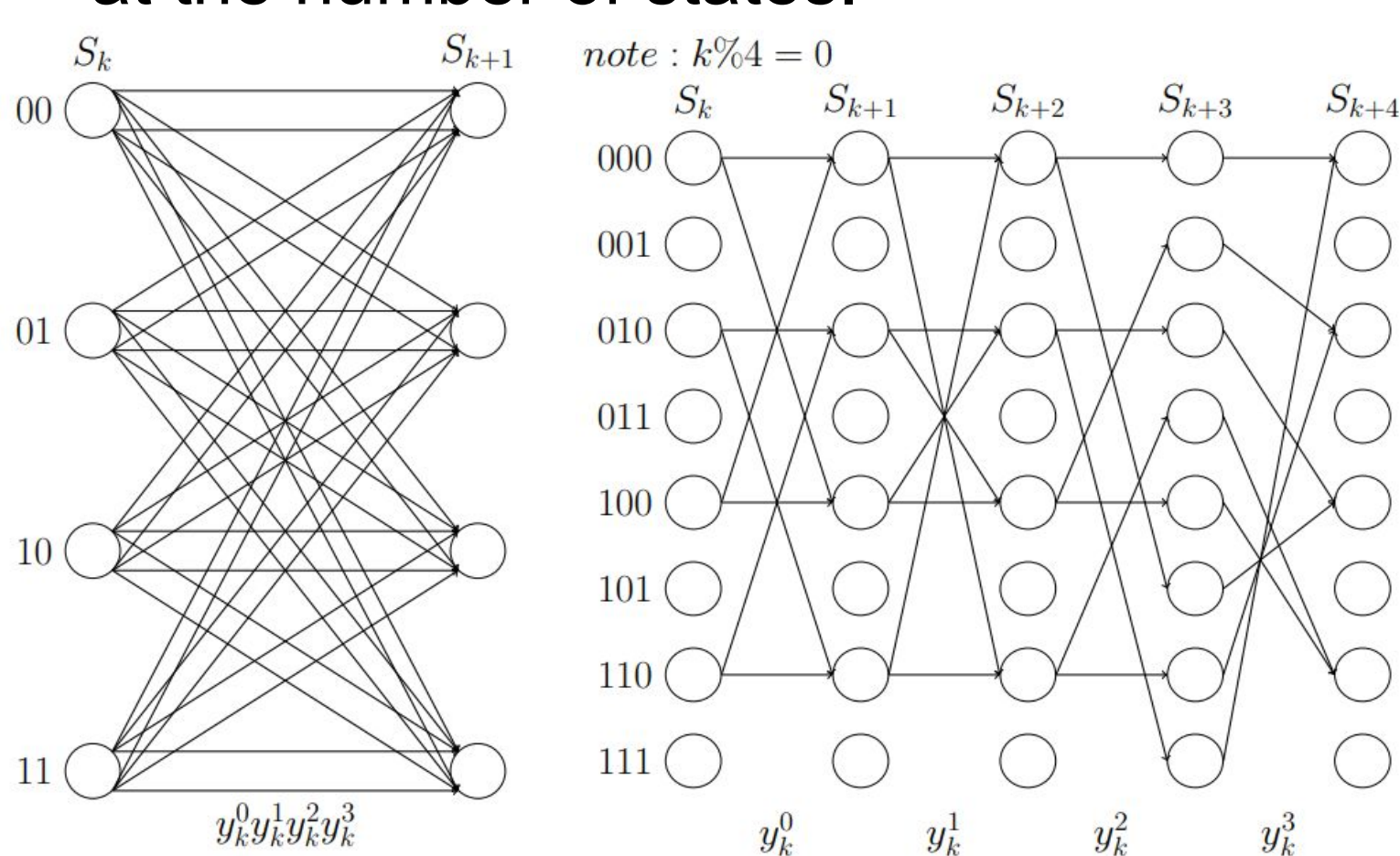


Figure 4: Standard (left) vs Dual Trellis (right)

## List Decoding

- Adding DSO CRCs provides a further layer of redundancy to rule out invalid decoded messages.
- If a decoded message fails the CRC check, serial list decoding computes the next most likely path by using the paths that were more likely, but failed the CRC.
- The locally second best path will either be one divergence from one of the locally best paths, or a path corresponding to a different starting state, since more than one divergence will always add additional weight compared to one divergence on its own.
- This can be efficiently implemented by using a red-black binary search tree, as described by Röder and Hamzaoui [5].
- To cap decoding complexity, we fix the list size, which is the maximum number of messages that can be checked.

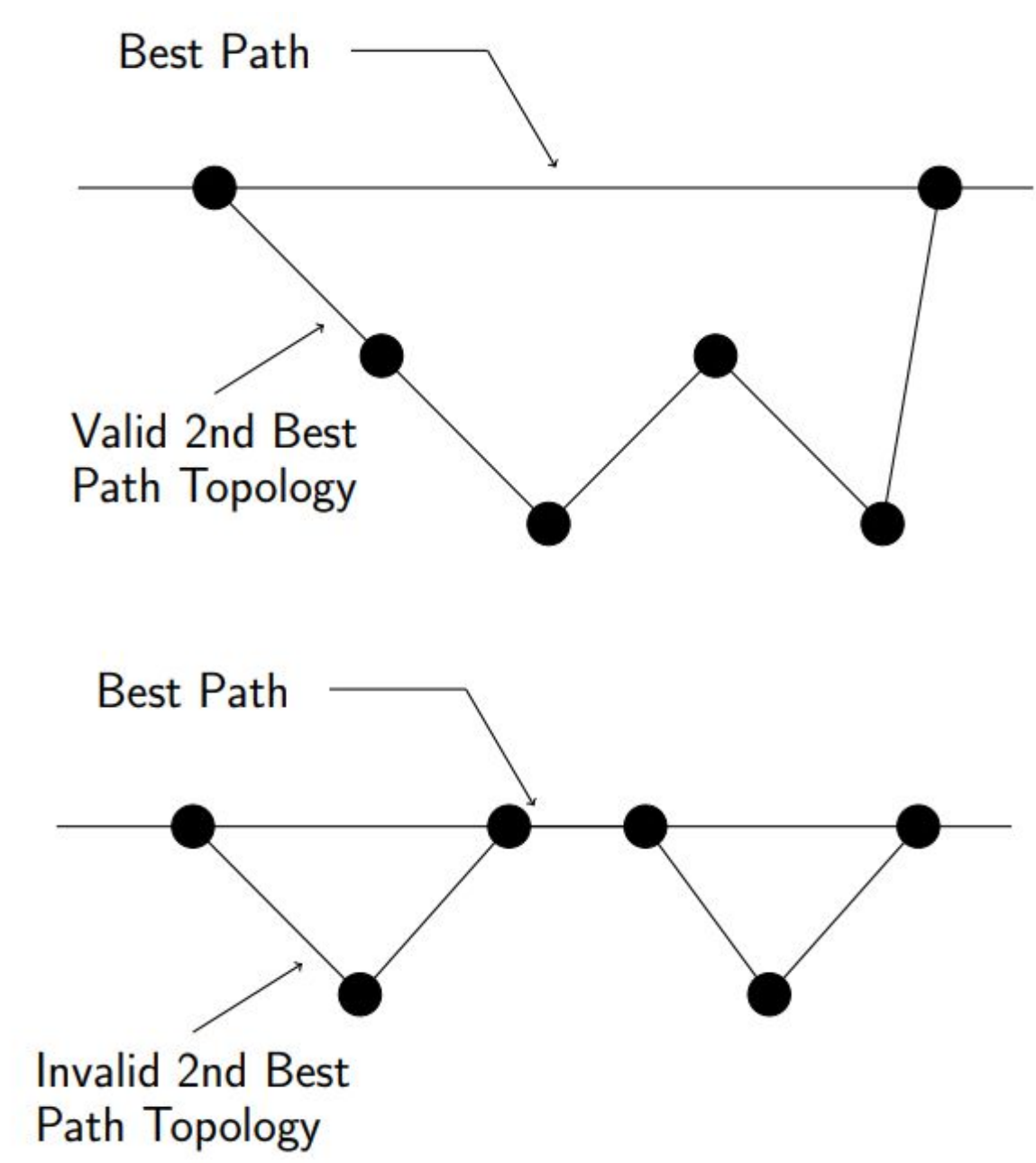


Figure 5: Valid and Invalid Next Best Path Topologies

## Results

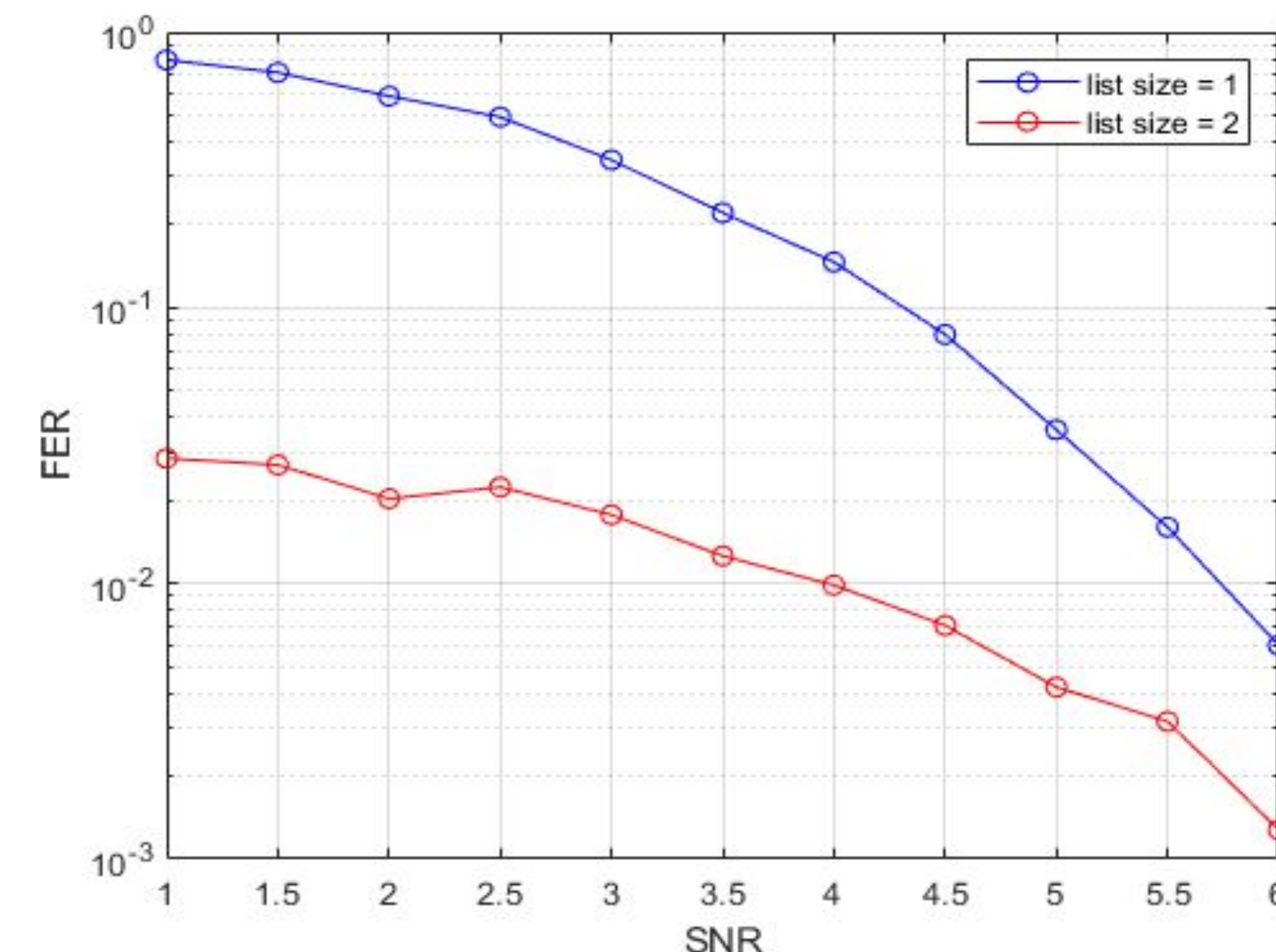


Figure 6: Frame error rate for list sizes of one and two, using the optimal rate-3/4 code with six memory elements, and a blocklength of 30 bits.

- As seen in Fig. 6, increasing list size offers significant improvements in FER for a given SNR.
- Use of the dual trellis reduced decoding complexity by 75% compared to the standard trellis for a rate-3/4 system with two memory elements.
- Larger list sizes produce less frame errors in the decoded message at the cost of increased decoding complexity.

## Conclusion, Future Works, and Limitations

- The improvement to FER provided by the list decoding algorithm, with minor increases in decoding complexity, indicates that the method of using S-LVA with DSO CRC for low rate codes presented in Liang et al. [5] generalizes well to high rate codes.
- We're still working on the software implementation, so this poster only includes data for small list sizes. When fully implemented, we plan to test longer list sizes, as well as varying blocklengths and code rates.
- A high rate tail-biting list decoder using a dual trellis accurately and efficiently decodes transmitted data. However, information theory provides theoretical limits on the performance of any communication system, regardless of decoding complexity.

## References

- [1] Yamada, T. et al. "A New Maximum Likelihood Decoding of High Rate Convolutional Codes using a Trellis".
- [2] Seshadri, N. and Sundberg, C.-E.W. "List Viterbi Decoding Algorithms with Applications".
- [3] Yang, H. et al. "CRC-Aided List Decoding of Convolutional Codes in the Short Blocklength Regime".
- [4] Röder, M. and Hamzaoui, R. "Fast Tree-Trellis List Viterbi Decoding".
- [5] Liang, E. et al. "List-Decoded Tail-Biting Convolutional Codes with Distance-Spectrum Optimal CRCs for 5G".

## Acknowledgements

We would like to thank Dean Wesel for his guidance and leadership throughout the research process. We are grateful to the National Science Foundation for funding, through the Summer Undergraduate Research Program and the UCLA Electrical and Computer Engineering Department. Finally, we would like to thank Beryl Sui, Hengjie Yang, Linfang Wang, and Ethan Liang for their assistance with the technical aspects of our project.