A State-Reduction Viterbi Decoder for Convolutional Codes with Large Constraint Lengths

Presenter: Sheng-Shian Wang
Advisor: Po-Ning Chen
Dept. of Communications Engineering
National Chiao Tung University
2002/6/21

1

Outline

- Introduction
- Background
- A State-Reduction Viterbi decoding algorithm
- Simulation results
- Conclusion

2

Introduction

- With a proper design, the combination of convolutional encoders and Viterbi decoders can jointly provide an acceptable performance with feasible decoding complexity.
- For such a combination, a tradeoff between the error performance and the decoding complexity resides on the choice of the **code constraint length**.
- The maximum-likelihood error performance of convolutional codes with **very long constraint length** has been substantiated by *Important Sampling technique* to be close to the Shannon limit, although no feasible decoder can decode such codes.

3

Introduction

- In this thesis, the performance of a *state-reduction soft-decision Viterbi decoder* with constant complexity for use of decoding codes with large constraint length is examined.
- Idea behind the trials
  - Since the maximum-likelihood error performance of codes with large constraint length is quite good, a small degradation due to state reduction may still yield an acceptably good performance.

4

Introduction

- From certain aspect, *to reduce the decoding complexity of the Viterbi algorithm* is equivalent to *reducing the number of visited states* during the decoding process.
- Notably, the number of visited states (or specifically, the code constraint length) is one of the key factors that determine the complexity of Viterbi decoder.
- There has been several approaches proposed along this research line, such as *adaptive Viterbi algorithm* [Chan and Haccoun 1997] and *segment-match Viterbi algorithm* [Feldmann and Harris 1999].

5

Adaptive Viterbi algorithm

Adaptive Viterbi algorithm (Chan and Haccoun 1997)

- Adaptive Viterbi algorithm basically follows the Viterbi algorithm, except that it discards those successor paths whose metrics are larger than ($d_m + T$), where d_m is the metric of the most likely path at the previous level, and T is a pre-specified threshold.

6

Adaptive Viterbi algorithm

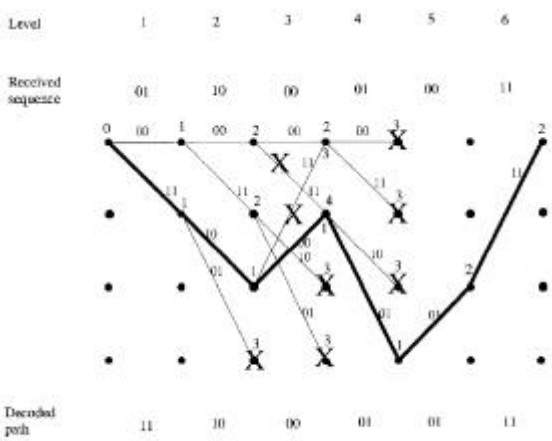


Fig. 1. Decoding a sequence with the AVA and a discarding threshold $\mathcal{T} = 1$. X denotes a discarded path. The last two branches correspond to the tail of the sequence.

7

Segment-match Viterbi algorithm

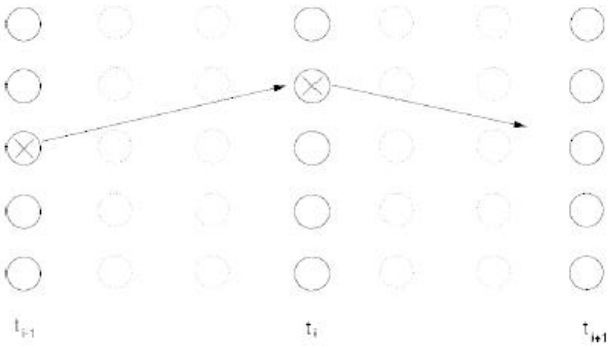


Fig. 1. Conditions of a path segment match. Circles with an X represent trellis states with maximum metric. States separated on the trellis by a constraint length, corresponding to times t_{i-1} (left), t_i, and t_{i+1}, are shown darkened. Arrows represent path segments over the intervals $[t_{i+1}, t_i]$ and $[t_i, t_{i-1}]$.

8

Pro and Con of adaptive and segment-match Viterbi algorithms

- None of these algorithms hard-ensure the amount of reduction in computational effort. In other words, the reduction in computational complexity is statistically dependent on the channel noise levels.
- These algorithms are model-parameter-sensitive. I.e., these algorithms achieve decent performance as the conventional Viterbi decoding algorithm with a marked reduction in computational complexity, only if proper system parameters have been selected.

9

M-algorithm

- A key question in this thesis is how to properly select visited states during the decoding process so that the computational complexity of the Viterbi algorithm can be a feasible constant, independent of the channel statistics.
- The basic principle of the M-algorithm is to keep the M most likely paths among those paths that end at the same level of a trellis, where a “more likely” path is the path with “larger path metric”.
- These M most likely paths, or equivalently states, are extended to the next level of the trellis, yielding at most $n \times M$ states for an (n,m,k) convolutional code.
- Only M most likely states are kept, and the remaining states are deleted.

10

M-algorithm

- At the end of the trellis, the M algorithm selects the best path with the largest path metric as the Viterbi algorithm does, and outputs the respective path labels of the located best path.

11

M-algorithm

- The M-algorithm is widely used in applications such as CPM, because of its feasible complexity and acceptable performance.
- The complexity of the M algorithm is determined by M.
- A certain number of papers on the M algorithm have been published. E.g., some papers focused on the choice of minimal M under which the M algorithm can asymptotically achieve the maximum-likelihood decoding performance.

12

A state-reduction Viterbi decoding algorithm

- Our proposed reduced-state Viterbi decoding algorithm happened to be the same as the M algorithm in its procedure.
- The main difference is that we are perhaps the first ones to apply the M algorithm to the soft-decision decoding of convolutional codes with very long constraint lengths.

13

A state-reduction Viterbi decoding algorithm

State-reduction Viterbi decoding algorithm

- Step 1. Set l=0, and set the single original state (or node) of the trellis as the only surviving state (or note) at level 0.
- Step 2. Extending all the surviving states (or nodes) at level l to the next level yields the set of visited states (or nodes) at level l+1.
- Step 3. For each visited state at level l+1, keep only one surviving path that enters the state (or node) by following the conventional Viterbi algorithm. Keep at most M states at level l+1, which correspond to the M largest surviving path metric.

14

A Reduced-stated Viterbi decoding algorithm

- Step 4. $l=l+1$.
- Step 5. If l is the last level of the trellis, output the information bits corresponding to the (single) surviving path, and stop the algorithm; else go to Step 2.

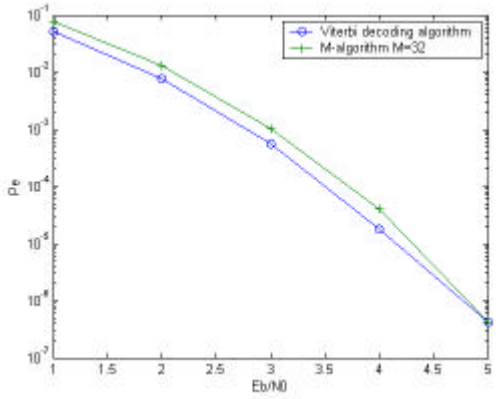
15

Simulation model

- We use two different convolutional codes.
  - One is a (2,1,23) code with generator 51202215, 66575563 (octal) and the free distance is 24.
  - Another is a (2,1,16) code with generator 715022, 514576 (octal) and the free distance is 18.
- The information length is 1200.

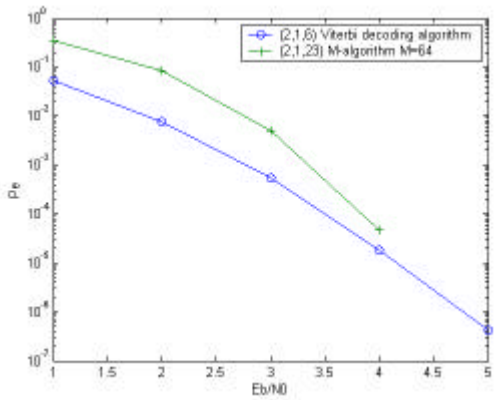
16

Simulation result



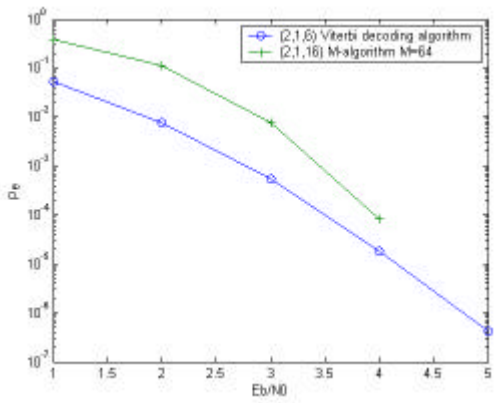
17

Simulation result



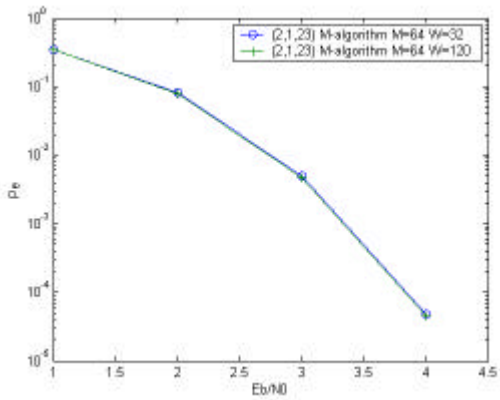
18

Simulation result



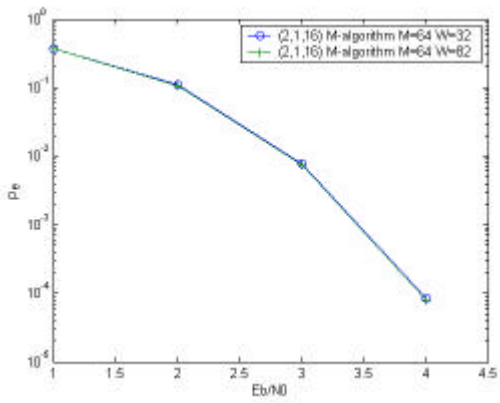
19

Simulation result
Impact of sliding window



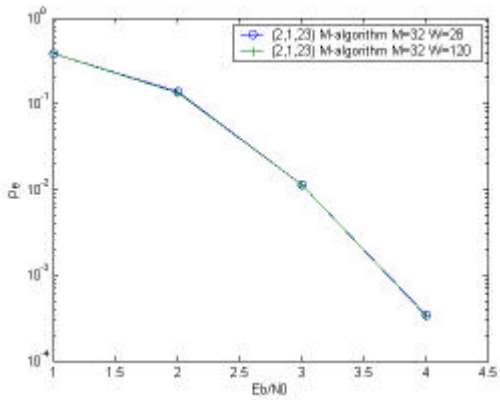
20

Simulation result
Impact of sliding window



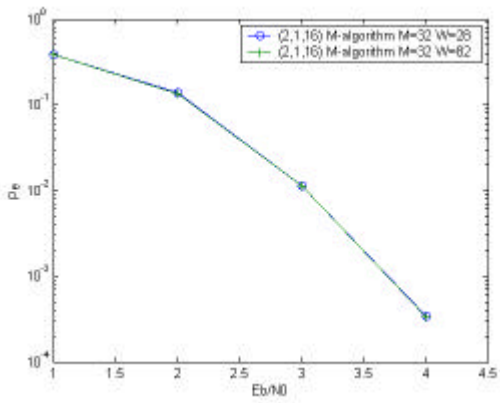
21

Simulation result
Impact of sliding window



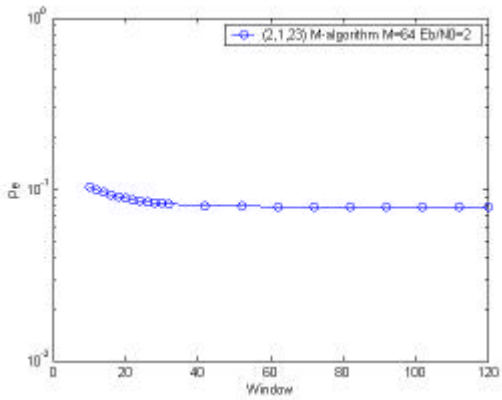
22

Simulation result
Impact of sliding window



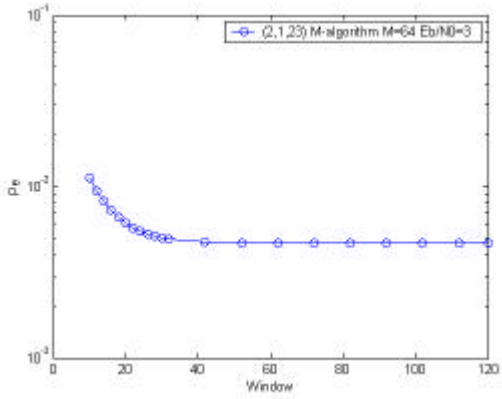
23

Simulation result
Impact of sliding window



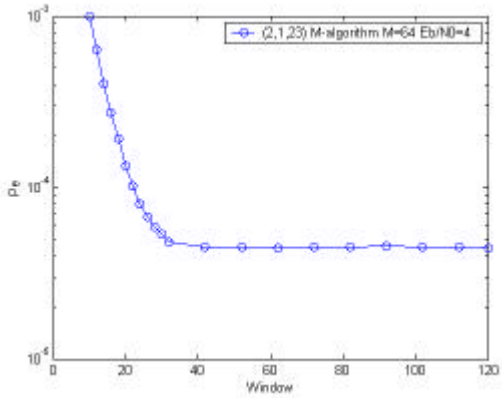
24

Simulation result
Impact of sliding window



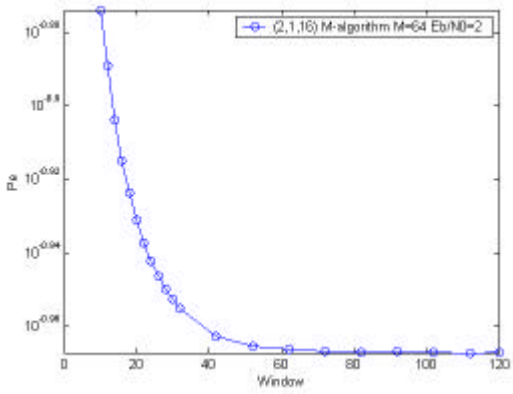
25

Simulation result
Impact of sliding window



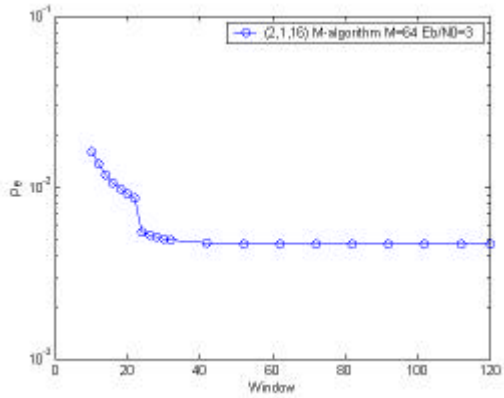
26

Simulation result
Impact of sliding window



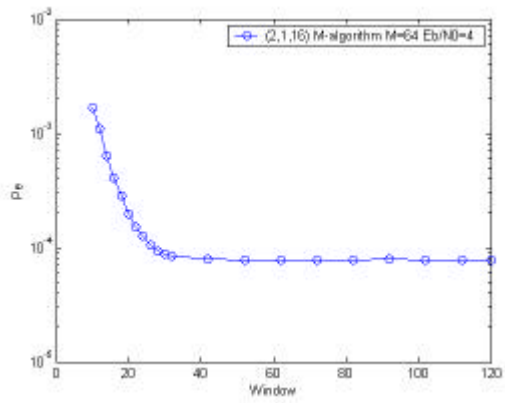
27

Simulation result
Impact of sliding window



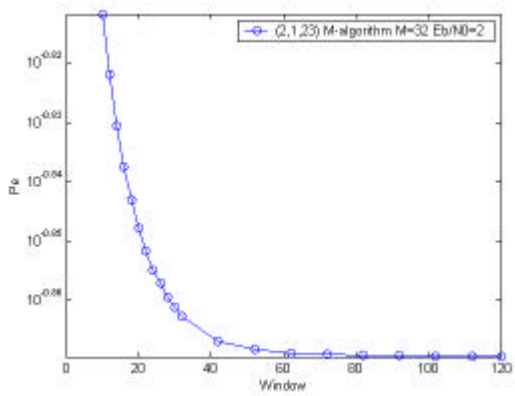
28

Simulation result
Impact of sliding window



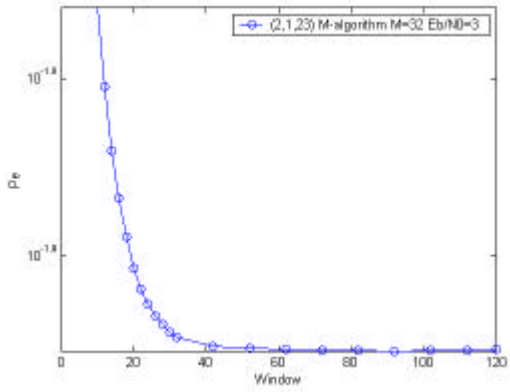
29

Simulation result
Impact of sliding window



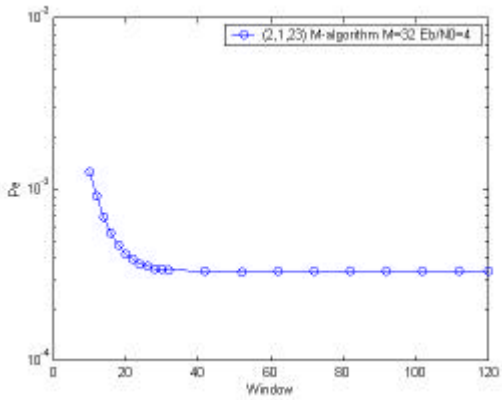
30

Simulation result
Impact of sliding window



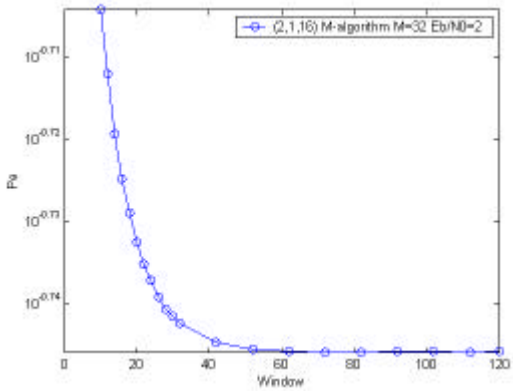
31

Simulation result
Impact of sliding window



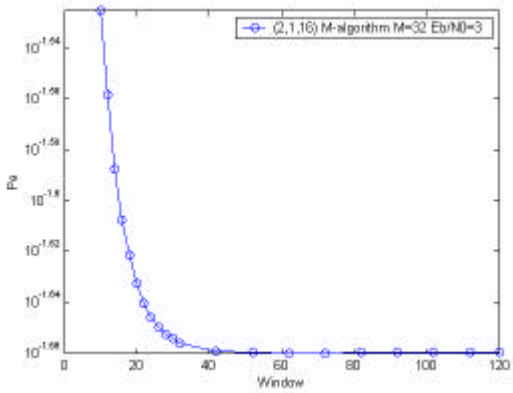
32

Simulation result
Impact of sliding window



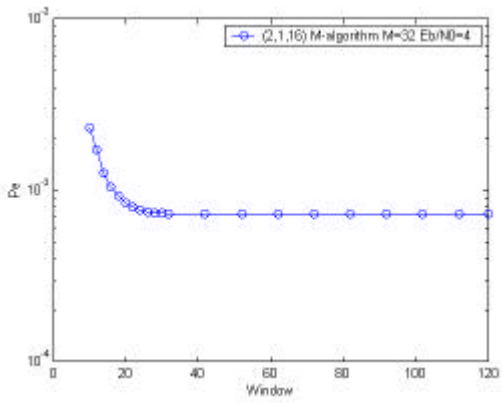
33

Simulation result
Impact of sliding window



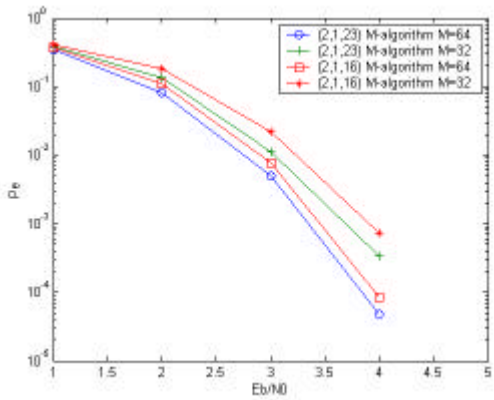
34

Simulation result
Impact of sliding window



35

Simulation result
Impact of constraint length



36

Conclusion

- The code constraint length has been a key tradeoff between the maximum-likelihood performance and complexity of Viterbi decoding.
- Due to the high decoding complexity of Viterbi for codes with long constraint length, conventional Viterbi decoding can not benefit from the seemingly low bit error rates that exponentially decay with code constraint lengths.
- Some algorithms have therefore been proposed to reduce the complexity of Viterbi, such as adaptive Viterbi and segment-match Viterbi. Their simulation results indicate that a small performance degradation can be achievable due to an adaptive state reduction; however, the results depend upon the chosen algorithmic parameters and noise level, inducing unexpected variation in system performance.

37

Conclusion

- Thus, our research is focused on the M-algorithm, whose complexity is fixed and independent of the noise level.
- Our simulation results show that by taking a sufficiently long code constraint length, the M-algorithm can achieve the maximum-likelihood performance of Viterbi decoding of comparable decoding complexity, especially at high SNR.
- Simulation results also imply that the sliding window that must be considered in practical implementation of M-algorithm is sufficiently taken to be five times larger than the order of the maintained states

38

Conclusion

- A future work of this thesis is to further extend the constraint length, and investigates the ultimate attainable performance of the M-algorithm.

39