An Impelmentation of Viterbi Detector based on AWGN Channel

Tao Liu[1,*]

[1] *University of Arizona, Department of Electrical and Computer Engineering, 1230 E. Speedway Blvd, Tucson, AZ 85721, USA*

*[*taoliu@email.arizona.edu](mailto:taoliu@email.arizona.edu)*

March 8th 2012

Abstract: The Viterbi algorithm is a dynamic programming algorithm in order to discover the most likely sequence of hidden states – called the Viterbi path – that results in a sequence of observed sequences, especially in the context of Markov information sources. We proposed an implementation of Viterbi Detector, which uses truncated Viterbi algorithm, based on C code in this paper. The simulation shows that it can achieve BER of 10^{-5} when SNR equals 11.3 while using the AWGN channel.

References

1. Sergio Benedetto and Ezio Bilglieri, "Principles of Digital Transmission", Information Technology: Transmission, Processing and Storage, 2002, 807-815, DOI: 10,1007/0-306-46961-8_20.
2. G. David Forney, Jr., "The Viterbi Algorithm: A Personal History" Viterbi Conference, March 8, 2005.
3. Chip Fleming, "A Tutorial on Convolutional Coding with Viterbi Decoding", Spectrum Applications, 2011.
4. Viterbi AJ (April 1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". IEEE Transactions on Information Theory 13 (2): 260–269. doi:10.1109/TIT.1967.1054010. (note: the Viterbi decoding algorithm is described in section IV.) Subscription required.B. R. Masters, "Three-dimensional confocal microscopy of the human optic nerve in vivo," Opt. Express 3, 356 (1998).

Index Terms — Viterbi algorithm, Finite state machine

1. Introduction

The Viterbi algorithm (VA) was originally proposed in 1967 for decoding convolutional codes. Shortly after its discovery, it was observed that the VA was based on the principles of dynamic programming, a general technique for solving extremum (that is, maximization or minimization) problems[1]. VA processor cycles is probably digital video broadcasting. A recent estimate at Qualcomm is that approximately 1015 bits per second are now being decoded by the VA in digital TV sets around the world, every second of every day[2]. Convolutional encoding with Viterbi decoding is a FEC technique that is particularly suited to a channel in which the transmitted signal is corrupted mainly by additive white gaussian noise (AWGN). In the past several years, convolutional coding with Viterbi decoding has begun to be supplemented in the geostationary satellite communication arena with Reed-Solomon coding. The two coding techniques are usually implemented as serially concatenated block and convolutional coding. Typically, the information to be transmitted is first encoded with the Reed-Solomon code, then with the convolutional code. On the receiving end, Viterbi decoding is performed first, followed by Reed-Solomon decoding. This is the technique that is used in most if not all of the direct-broadcast satellite (DBS) systems, and in several of the newer VSAT products as well. At least, that's what the vendors are advertising[3].

In this paper, we proposed a simple model, which consists of finite state machine, AWGN channel and Viterbi detector, in order to implement the truncated Viterbi detector. The finite state machine (FSM) accepts an infinitely long binary sequence m, and then produces the output sequence x. The output can be found by $x_i=O(s_i,m_i)$ at the i-th time interval, in which m_i is the input at time i, and s_i is the state of the finite state machine at time i. The next state is given as $s_{i+1}=N(s_i,m_i)$. The FSM states take values from the set S={0,1,…,M-

1}, which an be found in the matrix N, and the FSM outputs belong to a set X with q elements which can also be found in matrix O. The N and O can be changed to any 2×M matrix. Notice that we use truncated Viterbi Algorithm in this case. That is to say the Viterbi detector works in a sequential way, which means one symbol in and one bit our, instead of storing all the input and output sequences. The simulation shows that this model can achieve BER of 10^{-5} while SNR equals 11.3 when the truncation depth is 40.

2. Description of the Algorithm

The Viterbi algorithm operates on a state machine assumption, for example Markov chains. The bits being transmitted must be modeled in one of a finite number of states. That is, multiple pervious sequences of states (paths) can lead to a given state, at least one of them is a most likely path to that state, which is called the "survivor path". This is a fundamental assumption of the algorithm because the algorithm will check all the possible paths leading to a state and only keep the most likely one. The algorithm will keep only one pate with one state instead of keeping all the possible path. A second key assumption is that a transition from a previous state to next state is marked by an incremental metric, in our case it is a number and this transition is computed from the event. The third key assumption is that the events are cumulative over a path in some sense, in our case it is additive. So the crux of the algorithm is to keep a number for each state. When an event occurs, the algorithm moves forward to a new set of states by combining the metric of a possible previous state with the incremental metric of the transition due to the event and chooses the best based on the metric. The incremental metric associated with an event depends on the transition possibility from the old state to the new state. The Fig. 1 shows a brief example in which Viterbi Algorithm determines the path(the red arrow) in two states system.

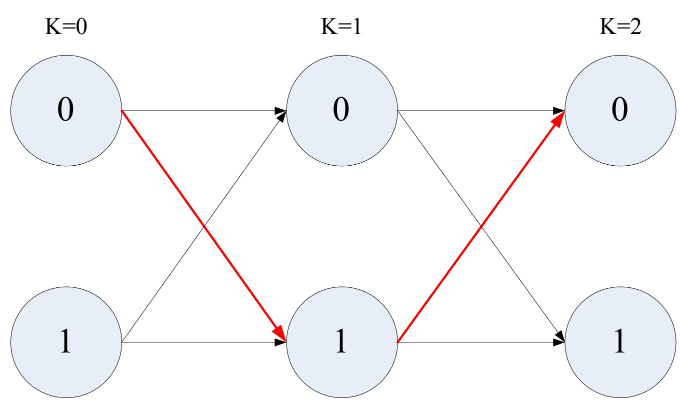

Fig. 1 The two states example of Viterbi Algorithm.

The Viterbi algorithm as described above doesn't consider the delay problem. In particular case, the algorithm cannot reach a decision about the maximum-metric sequence before check all the transmitted bits. In other words, the decision about the best sequence cannot be reached before scanning all the states from begin to the end. So that reducing the delay would necessarily be solved and the truncated Viterbi algorithm may be used for it. This consists of forcing decisions at *k-th* stage on all paths prior to the stage *k-D* and D is called truncation depth(or decision depth), which is usually equals 8 times number of states at each time. At first, the truncated Viterbi algorithm compares the partial path metrics for the paths at stage k, and noting which one is the largest. The branch chosen at this time is the one belonging to this path at time k-D. After a latency of D time instants, the truncated VA outputs one branch at one time. Because only the last D branches of the survivor paths must be kept in memory and this is a efficient way to reduce the memory. The loss of optimality is reduced when D is increased, because when D is large there is a high probability that all the surviving paths leading to any node have an initial part in common: so this initial path will be a part of the optimum one, and we say that a merge has taken place[1]. That is, one symbol in and one bit out, and the D length block moves along all the sequences. The truncated Viterbi algorithm detects the new states at end of the block and decode the symbol at start of the block at the same time.

3. Implementation

The simulation is done by the C code and the configuration of the system is shown as Fig 2. The finite state machine(FSM) is the way to implement the first assumption as discussed above. FSM is used to generate a sequence with memory use the matrix N and O. The output

Fig. 2 The implementation of the Viterbi Algorithm simulation.

can be found by $x_i=O(s_i,m_i)$ at the i-th time interval, in which m_i is the input at time i, and s_i is the state of the finite state machine at time i. The next state is given as $s_{i+1}=N(s_i,m_i)$. The

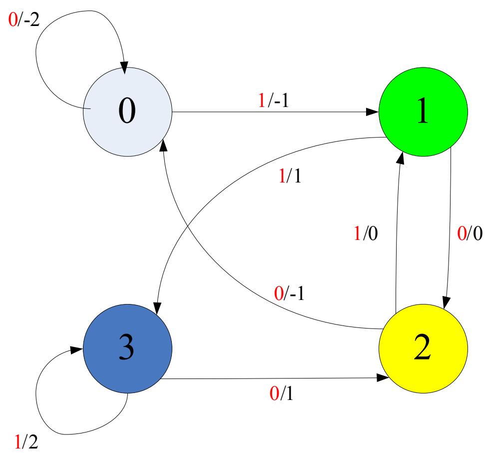

Fig. 3 The transition relationship between states..

transition relationship is presented by the Fig 3. Then the sequence pass through the AWGN channel in which the noise figure can be changed based on different SNR. At Viterbi Detector,

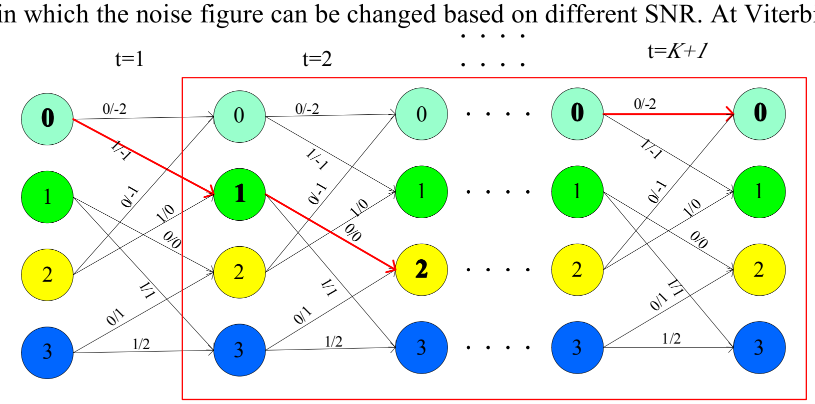

Fig. 4 The truncated Viterbi Algorithm detection..

we use the truncated Viterbi algorithm which is discussed above.

In C code implementation, the information bits are generated by random 1 and 0 numbers. And then we use N and O metric to generate sequences which will pass through AWGN channel. The AWGN channel is simulated by random Gaussian function. In the Viterbi detector, we first determine the first D states without output symbols and them started to move the block along the sequence with one bits in and one symbol out.At end of the channel, we compare the bits from start to end in order to calculate the BER with different SNR. The Viterbi Algorithm operation is shown in Fig 4 for first K states.

4. Conclusion

The result of the simulation is shown in the fig 5. The simulation shows that this model can achieve BER of 10^{-5} while SNR equals 11.3 when the truncation depth is 40. The performance for different truncation depth is shown by distinct color curves. It is clear to see that when the

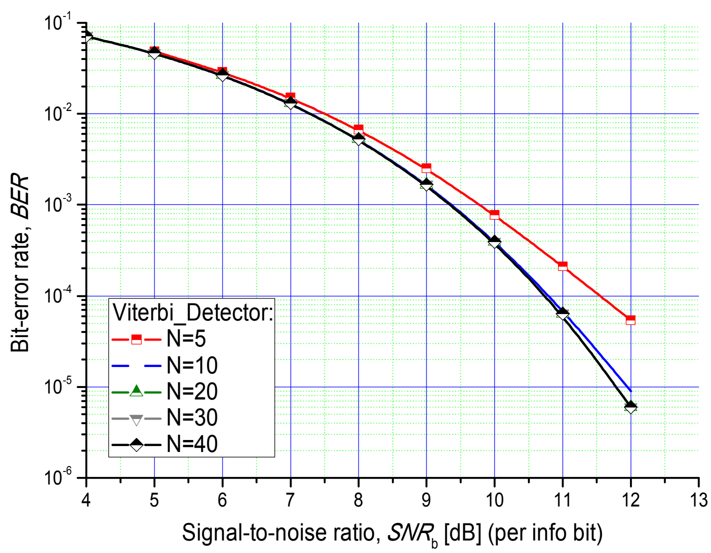

Fig. 5 The simulation result of Viterbi Algorithm in proposed system

truncation depth decreases the performance becomes bad and it is very obvious when the truncation depth equals 5 or 10. When the truncation depth increases to 20, it will not show much difference in the performance. So we can choose the least memory to achieve best performance based on the simulation.