Performance of Viterbi Sequence Detection in the presence of AWGN channel

Arush S. Sharma, *Graduate student in the Dept. of Electrical and Computer Engineering, University of Arizona,*

***Abstract*—This report documents the task that is required in the Project 2 of the Graduate Course ECE 535A in the context of analyzing the performance of Viterbi decoder. The output from finite state machine is passed through AWGN channel and is fed to the Viterbi decoder which outputs the sequence of decoded inputs. Monte carlo simulation is performed to estimate the probability of error of Viterbi decoder with varying signal to noise ratio.**

***Index Terms*—FSM, HMM, Viterbi, AWGN, Monte Carlo, SNR, BER.**

I. INTRODUCTION

THE Viterbi Algorithm is mainly used for detection and estimation of sequences in signal processing and digital communications. It is often used to detect signals having memory in communication channels. The Viterbi Algorithm (VA) finds the most likely input sequence in a state diagram given a sequence of output symbols. Often times, the signal is made memory dependent by following the given Finite state machine (FSM) model or hidden Markov Models (HMMs). The signal is then corrupted by passing it through Additive White Gaussian Noise (AWGN) channel. The VA receives the corrupted sequence and it recursively finds the most-likely noiseless transition coming into each state. Therefore, given an observation sequence of bits, the VA can be used to find the most-likely state sequence and its likelihood in a given FSM.

Figure 1 shows an example of the state transition diagram of the FSM in which there are four states (state 0,...,state 3) and the state transitions are labeled as input/output (x, y). This figure can be interpreted as follows. For example, if the initial state is 0, an input of 1 will cause a transition from state 0 to state 1 with an output symbol of 1. Likewise, an input of -1 will go back to state 0 with an output of -3. For better visualization, the state transitions are represented by time-indexed equivalent called a trellis which is explained in Section II. Thus, if a sequence of symbols, generated from a trellis is corrupted by AWGN, the VA will find the most-likely original noiseless bits. In this case study noise is simulated by considering different Signal to Noise Ratio (SNR) values. The VA then decodes the received signal following the maximum-likelihood criterion which is explained in Section III. The probability of error or Bit Error Rate (BER) is computed by taking the ratio of number of bit errors divided by the total number of transmitted bits. The source code is written in C++ programming language. In this simulation, the source sends 1 million bits to destination in total. Monte Carlo simulation is then performed 1000 times with different time-based seed to have more randomness and the data is then averaged.

arushsharma@email.arizona.edu

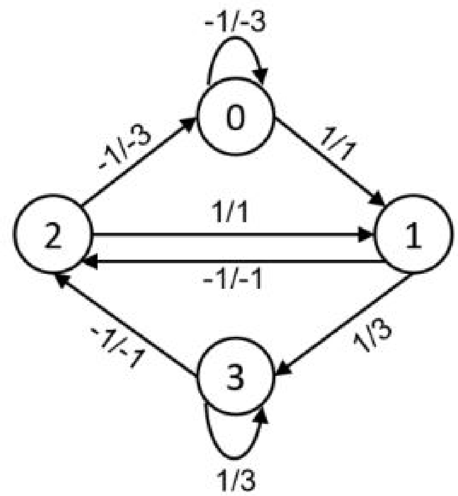

Fig. 1. State transition diagram of the FSM

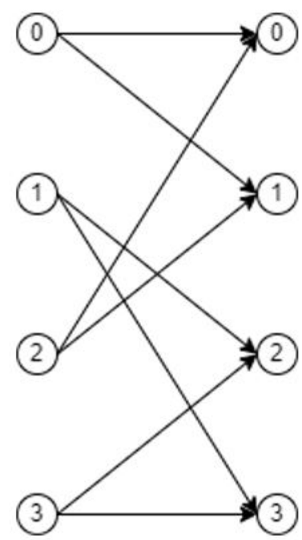

Fig. 2. Trellis of FSM following Figure 1

The rest of the section is as follows. Section II shows the Viterbi detection with trellis diagram by considering an

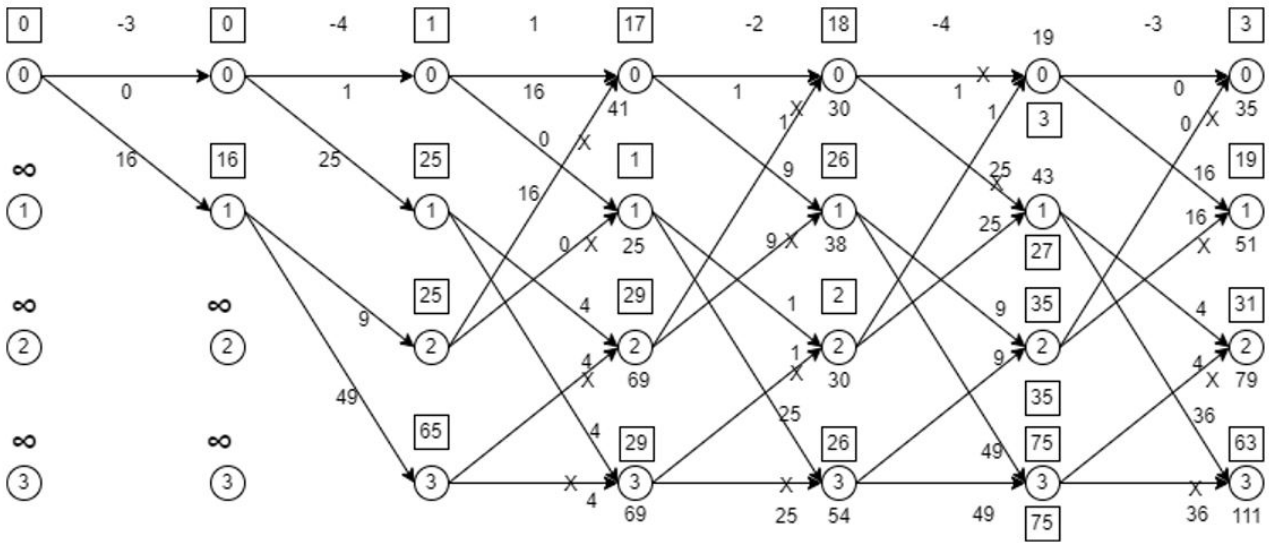

Fig. 3. Full trellis diagram considering the received sequence $\overline{r}$

example of finite received sequences. Section III explains the principle of the Maximum Likelihood Detector. Section IV describes the performance of VA with varying SNR which is achieved through Monte Carlo simulation. Section V summarizes the task(s) involved in the project.

II. Viterbi Detection

The VA finds the sequence of symbols in the given trellis that is closest in distance (Euclidean or Hamming) to the received sequence of noisy symbols. This sequence decoded by VA is the *most likely sequence*. In the project, Euclidean distance is considered as the performance metric among different states which makes VA as the optimal maximum-likelihood detector.

To find the most likely sequence, the VA first iteratively computes the survivor path entering each state. The survivor path for any given state is the sequence of symbols entering that state which is closest in distance to the received symbol. The distance between the survivor path and the sequence of noisy symbols is called path metric for that state. After the computation of the survivor paths for all state, the path having least path metric is considered to be the most likely path.

Assuming that the starting state is 0 and received noisy sequence is given by $\overline{r} = \{-3, -4, 1, -2, -4, -3\}$. At each iteration that corresponds to one stage of trellis, the VA computes the most likely transition coming into each state by computing the branch metrics and then updates the survivor path and path metric for that state. The survivor path is selected which has the least Euclidean distance. Branch metric is computed by taking the squared distances between the received symbols and ideal sequence symbols in that iteration.

Figure 3 depicts the computation of branch metrics and path metrics of each state in every iteration. Numbers in the square box against every state in the iteration depicts the final path metric. *X* represents that those branch metrics are ignored in computing the path metrics. Finally at the end of the iteration, the trace-back is done by choosing that state which has the least path metric and moving backwards until the starting state is reached as depicted in Figure 4. The input now can be decoded by referring to the FSM model which comes out to be $\overline{i} = \{-1, -1, 1, -1, -1, -1\}$.

III. Maximum likelihood Detector

Assuming that the source wishes to transmit sequence of bits which are independent of each other, i.e., memory-less system, the detector would like to make a decision on transmitted signal based on the observation of received vector, r. With this goal in mind, the decision rule based on the computation of the posterior probabilities is defined as [1]

$$P(signal\ s_m\ was\ transmitted \mid r),\ m = 1, 2, ..., M \quad (1)$$

The decision criteria is based on selecting the signal corresponding to the maximum of the set of posterior probabilities $P(s_m \mid r)$. This criterion is called the maximum a posteriori probability (MAP) criterion. Using Bayes' rule, the posterior probabilities may be expressed as

$$P(s_m \mid r) = \frac{p(r|s_m)p(s_m)}{p(r)} \quad (2)$$

where $p(r \mid s_m)$ is the conditional pdf of the observed vector given s_m, and $P(s_m)$ is the a priori probability of the mth signal being transmitted.

Some simplification occurs in the MAP criterion when the M signals are equally probable events, i.e., $P(s_m) = 1/M$. Furthermore the denominator in Equation 2 is independent of which signal is transmitted. Thus the decision rule based on finding the signal that maximizes $P(s_m|r)$ is equivalent to finding the signal that maximizes $p(r|s_m)$ which is usually called the likelihood function. The decision criterion based on the maximum of $p(r|s_m)$ over the M signals is called

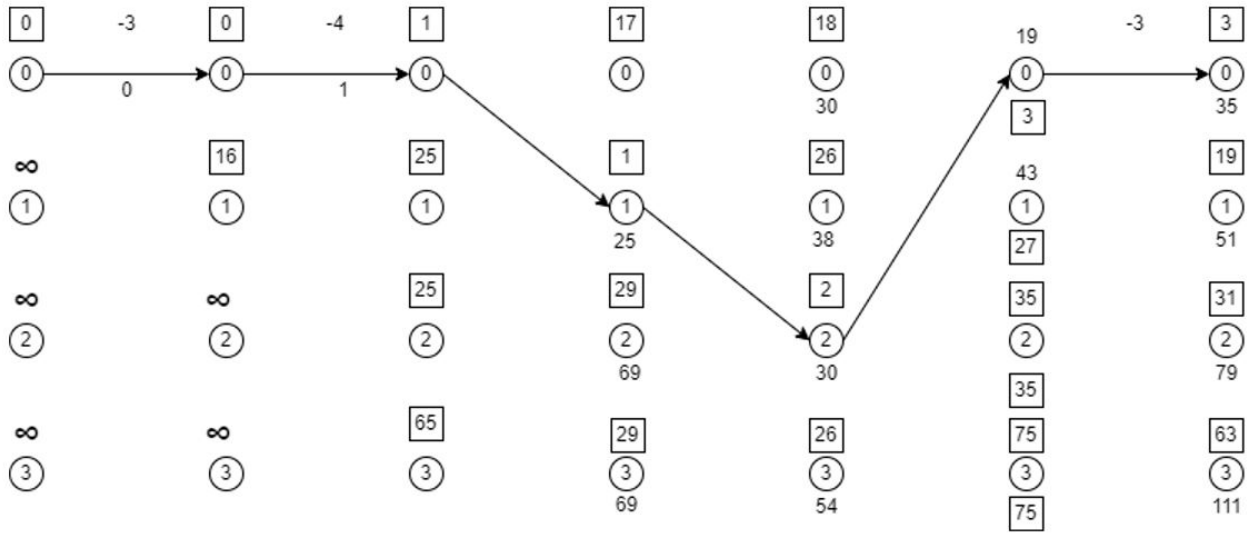

Fig. 4. Final decoded path considering the received sequence $\overline{r}$

the maximum-likelihood (ML) criterion. The detector based on MAP criterion makes the same decision as that of ML criterion as long as the signals are equally probable. In case of an AWGN channel, the likelihood function $p(r|s_m)$ is given as:

$$p(r|s_m) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[\frac{-(r-s_m)^2}{2\sigma^2}\right] \tag{3}$$

Now suppose the sequence of outputs $r_1, r_2, ..., r_k$ is observed. The joint pdf of $r_1, r_2, ..., r_k$ may be expressed as a product of K marginal pdfs, i.e.,

$$p(r_1, r_2, ..., r_k|s_m) = p(r_1|s_1)p(r_2|s_2)...p(r_k|s_m) \tag{4}$$

$$p(r_1, r_2, ..., r_k|s_m) = \prod_{i=1}^{k} p(r_i|s_i) \tag{5}$$

$$p(r_1, r_2, ..., r_k|s_m) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[\frac{-(r_i-s_i)^2}{2\sigma^2}\right] \tag{6}$$

$$p(r_1, r_2, ..., r_k|s_m) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^k exp\left[-\sum_{k=1}^{k}\frac{(r_k-s_k)^2}{2\sigma^2}\right] \tag{7}$$

By taking the log of Equation 7, the final expression is achieved which is given as follows:

$$= kln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{1}{2\sigma^2}\sum_{k=1}^{k}(r_k-s_k)^2 \tag{8}$$

By neglecting the terms that are constant in the above Equation 8, the ML sequence detector selects that sequence that minimizes the euclidean distance metric given as follows:

$$D(r, s_m) = \sum_{k=1}^{k}(r_k-s_k)^2 \tag{9}$$

The Equation 9 follows the Euclidean distance criteria of computing the branch metrics in every iteration of the VA. Hence VA achieves maximum likelihood sequence detection for the case of AWGN channel.

IV. Performance of BER with varying SNR

The simulation records the Probability of Error with SNR beginning from -500 dB to +55 dB with increments of 5. In the Figure 5, it can be seen that the performance of AWGN channel improves as the SNR increases. The unnecessary details from -500 dB to -20 dB and from 20 dB to 50 dB are omitted in the figure. The probability of error was close to 0.5 when the SNR was -500 dB and this was consistent until the SNR reached -20 dB. The probability of error then exponentially reduces with gradual increase in SNR. Onwards from 20 dB SNR, the probability of error is very much close to 0.

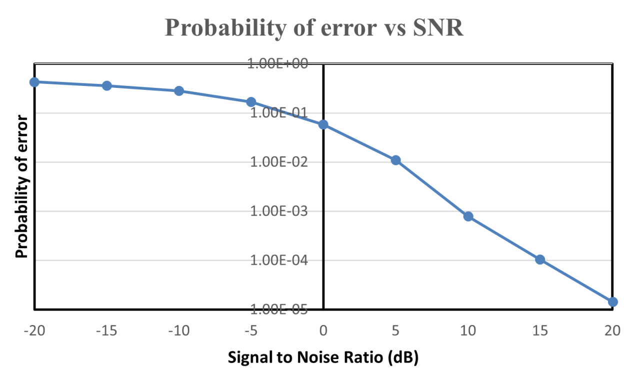

Fig. 5. Performance of BER with varying SNR using VA decoder

In the software, the program randomly selects any state as the starting state. It then generates one input bit following the FSM model and outputs one symbol in a sequential fashion. At every 100th iteration, the path metric register is normalized to avoid overflow since the registers are of finite length. The re-normalize approach [2] is carried out by subtracting a constant from path metrics of all state in that iteration. This normalization doesn't affect the results since the differences of the path metrics are not affected by subtracting a constant from all path metrics. The constant was calculated by finding the minimum path metric in that iteration.

V. Conclusion

1 Million bits with memory following FSM model are transmitted by source which are then corrupted by AWGN channel using varying SNR. These bits are transmitted sequentially and is sequentially decoded by VA. The path metrics are re-normalized at every 100th iteration to avoid overflow. The Monte Carlo simulation is run 1000 times with different values of seed and the data is then averaged. Performance of the VA detector in the presence of AWGN channel is evaluated in terms of BER with varying SNR written in C++ programming language.

References

[1] J. G. Proakis and M. Salehi, *Digital communications*. McGraw-hill New York, 2001, vol. 4.

[2] H.-L. Lou, "Implementing the viterbi algorithm," *IEEE Signal processing magazine*, vol. 12, no. 5, pp. 42–52, 1995.

VI. Appendix

-30	0.473288
-25	0.453461
-20	0.424912
-15	0.354429
-10	0.282577
-5	0.165821
0	0.057932
5	0.010947
10	0.00079
15	0.000104
20	1.42E-05
25	1.25E-05
30	1.27E-05
35	1.28E-05
40	1.31E-05
45	1.3E-05
50	1.25E-05
55	1.28E-05

Table showing the BER with varying SNR from -30 dB to +55 dB