KING ABDULLAH UNIVERSITY OF SCIENCE & TECHNOLOGY

ELECTRICAL ENGINEERING DEPARTMENT

EE-242 Digital Communications & Coding

FALL 2009

Final Report

Maximum Likelihood Sequence Detector & the Viterbi Algorithm

Team Members:

- Miguel Ángel Galicia Martínez
- Ismail AlQerm

December 12th, 2009

Maximum Likelihood Sequence Detector & the Viterbi Algorithm

Galicia, Miguel, AlQerm, Ismail

Electrical Engineering Department

KAUST, Thuwal, Saudi Arabia

miguel.galicia@kaust.edu.sa

ismail.,querm@kaust.edu.sa

***Abstract*—** When a sequence of symbols is transmitted in a communication system, they are usually detected at the receiver symbol by symbol (i.e. independently). This is not always the optimal solution and therefore there is a need to detect the complete sequence as a whole. This is usually done using what is called Maximum Likelihood Sequence Detection (MLSD) in AWGN Channels by using the Viterbi Algorithm (VA).

The scope of this work is to present the MLSD method based on the VA, which eliminates signal sequences as new data are received from the demodulator. We start providing a brief introduction of the topic, continued by the explanation of the VA followed by a real-life implementation and concluding with the explanation of the MLSD. Some examples will be provided in order to achieve a deeper understanding of the topic.

***Keywords*—** **Markov Chain, State Diagram, Viterbi Algorithm, Baye's Rule.**

I. INTRODUCTION

In the early days of trellis coding/decoding, channel symbols were modeled as numbers from a finite field and the channel described as a discrete memoryless channel (DMC). The decoding problem was attacked by heuristic methods at that time.

The first approach was taken by Viterbi who formulated a dynamic programming algorithm for decoding of convolutional codes. Later on it was found that the algorithm actually performs maximum likelihood sequence detection (MLSD) of the states the sequential machine is traversing. This famous procedure is known as the Viterbi Algorithm (VA) and has had a tremendous impact on both digital transmission theory and application

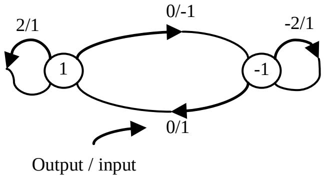
The Viterbi algorithm is used to find the most likely noiseless sequence of finite-state signals generated from a state diagram corrupted by noise. It searches for the least branch metric (Euclidean or Hamming distance) between the noisy received sequence and the original transmitted signal, which is called the survivor path of a certain state. When Euclidean distance is used as the metric, the VA is the optimal MLD method in AWGN since it is the least complex.

Two tools can be used on the MLSD algorithm; one is the Viterbi algorithm and the Markov Chain. The Viterbi algorithm is an efficient technique for solving the *minimum-distance* sequence detection for any Markov signal generator and any noise generator with independent noise components.

II. THE VITERBI ALGORITHM

A. Viterbi Algorithm (VA) - Background

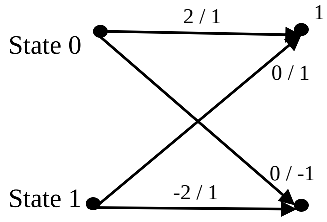
It is an algorithm implemented using dynamic programming to detect and estimate sequence of symbols in digital communication and signal processing by finding the most likely noiseless sequence. This algorithm also is used for speech recognition for speech symbols modeled by hidden Markov models. For detection using Viterbi Algorithm, the signals are generated from finite state diagram. The initial state is determined and the input decides to which state the transition will proceed.



State

In the figure shown above the initial state is 1 and input is -1, then the next state is -1 and the output is zero.

The Trellis is an indexed time diagram is used to represent the finite state diagram to make the inputs and outputs clearer. An example of trellis is the one that represent the state diagram below:



A trellis

B. Viterbi Algorithm (VA) - Description

The Algorithm is optimal in maximum likelihood sequence detection for symbols affected by additive white Gaussian noise AWGN. It is based on the principal of finding a noiseless output sequence with minimum distance from the detected noisy sequence of symbols. Viterbi is the best algorithm if Euclidean distance is used which is not always the case in practice.

A way to find the MLS

The VA calculates recursively the survivor path for each state in the state diagram. The survivor path is input sequence for each state that is the closest to the detected noisy sequence of each state. After VA has found all the survivor paths, it will compare the path metric for them. The path metric is the distance between the survivor path and the detected noisy sequence. The path with minimum metric will be chosen to be the globally detected path. For example, if x_1, x_2, $x_3 \ldots x_n$ are the noisy detected sequence, VA will compute the survivor path for each state based on its metrics at each recursion and update the metrics of the survivor path. The metric of a path coming into state is calculated by adding all branch metrics in that path. The branch metric is the distance between the noisy symbol X_n and the maximum likelihood symbol in the ideal sequence. Assume there are two states i and j: the branch metric at stage n is equal to $B_{i,j,n} = (X_n - C_{i,j})^2$ where $C_{i,j}$ is the output of the transition from state I to j.

The survivor path for state j at stage n is equal to: $M_{j,n} = \min[M_{i,n-1} + B_{i,j,n}]$ for all states I that have transitions to state j. After all the stages when the survivor path for state j has been calculated and the path metrics have been updated for all states, the survivor path with minimum metrics is selected to be the most likely path and this path can be traced from its state backwards. However, the number of recursion will go to infinity since the noisy symbols will not stop entering to the states. So an infinite amount of time is needed to recover the detected sequence. The solution for that is convergence of all the survivor paths for all states to the minimum metric path. Thus, all states will have the identical survivor path sequence prior to stage *n*. Therefore, we can trace back from any state and obtain the most likely sequence up to stage *n*. The number of recursion required to implement VA is called survivor path length L.

At each recursion, there are three main steps that must be done to achieve VA:

- *Branch Metric Generation:* This step is to calculate the metrics for all the transitions that enter each state.
- *Survivor path and path metrics update:* The branch metrics are added for each state with the previous path metrics. Then, the path with minimum sum is selected to be the survivor path for that state.
- *Most likely path traced back:* the survivor path for a given state is traced back l stages to determine the most likely symbols in the sequence.

C. Example: MLSD using VA

Using the trellis form the figure below, if the received noisy sequence is Xn= (0.05, 2.05, -1.05, -2.00, -0.05). The first step at stage 1 is to calculate the branch metrics: (5 stages needed)

B1,1,1 = (X1 – C1,1)2= (0.05 -2)2= 3.8025

B-1,-1,1 = (X1 -C-1,-1)2 =(0.05+2)2 =4.2025

B1,-1,1 = B-1,1,1 =(0.05)2 =0.0025

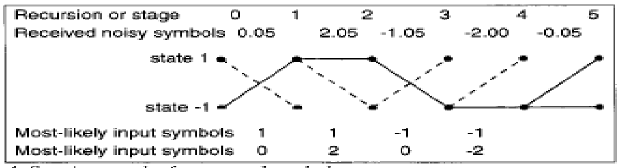
The second step is to update the survivor path for each sate and survivor metrics:

From the trellis, there are two possible transitions to each state so i=2. At the beginning, we assume that at the beginning of the recursions, all the path metrics are zero M1,0 = M-1,0 = 0). Thus, for state 1 at recursion 1

M1,1= min[Mi,0+Bi,1]

([M1,0 + B1,1] = 3.8025) > ([M-1,0 + B-1,1]=0+0.0025)

So the survivor path for state 1 is the transition from state -1 to state 1. Similarly the survivor path for state -1 is calculated. After 5 recursions, the received sequence will have 5 survivor paths which could be traced back to find the maximum likelihood sequence that represents the noisy symbols as shown in the figure below:



4. Survivor paths for states 1 and -1.

D. Real Time implementation of VA

The implementation of VA varies from one type of coding and the other and between trellis and other. It is usually implemented for convolutional codes and trellis codes which called sequential codes. VA is optimal for AWGN channels.

Branch metrics for convolutional codes

In Convolutional-Codes, the input and outputs are binary digits with N states and 2^k transitions for each state. If n bits code is sent in an AWGN channel, the corrupted received waveforms are quantized into m-bits symbol per transmitted bit. The quantized symbols then are sent to Viterbi decoder to calculate the branch metrics for each state. The implemented VA has two kinds of decision decoding:

- *Hard Decision Decoding*

The branch metric here is calculated based on hamming distance which is the number of bits that are different in the quantized sequence and the coded bit sequence of transition.

- Soft Decision Decoding

In this decoding no quantization is used to keep the content of the noisy symbols. The branch metrics are calculated using Euclidean distance between the noisy symbols and the coded bits of the given transition.

Branch metrics for Trellis Codes

Trellis are characterized as multi-dimensional which means that each codeword of a given transition consists of more than codeword. To find the branch metrics, the closest codeword in distance to the received noisy symbol should be selected to calculate the branch metric. The squared distance between this codeword and the received noisy symbol is the branch metric of the transition. After that the same procedure explained above is followed to find the maximum likelihood sequence.

III. MARKOV CHAIN MODEL IN MLSD FOR FREE-SPACE OPTICAL COMMUNICATION IN ATMOSPHERIC TURBULENCE CHANNELS

In the optical communication environment, the transmitted signals suffer from signal fading as a result of the atmospheric turbulences. Consequently, this makes the signal aperture D0 less than the correlation length of the fading signal d0. A maximum likelihood sequence detection MLSD is proposed based on the statistical properties of the turbulences and signals fading. However, this MLSD is very complicated to find since it requires multidimensional integration. A less complexity MLSD algorithm is derived based on single step Markov Chain model for fading correlation.

A. MLSD for Binary system

This MLSD is optimal for decoding i.i.d uniform sequence of bits but it has high complexity which is n. 2^n because it requires computing an n-dimensional integral for each of 2^n bit sequences. Assume the 2^n possible bit sequence S= {s_1,s_2…..S_n}, the received sequence R={r_1…..R_n} so the MLSD= S'= ArgMax.

$$\int_{\vec{x}} f(\vec{x}) \exp\left[-\sum_{i=1}^{n} \frac{(r_i - \eta s_i I_0 e^{2x_i})^2}{N}\right] d\vec{x}.$$

B. MLSD based on SMC fading correlation using PSP

The idea is to use the correlation between consecutive received ON bits to perform MLSD with a modification on the metric function. This MLSD technique is better than the previous in terms of complexity which is n^2 here.

Assume the transmitted sequence S= {s_1…..s_n} and ON bits symbols SON= {n_i=(1,2,3…n) S_n=1} i=1 to m, OFF bits sequence SOFF= {n_i=(1,2,3…n) S_n=0} i=1to n-m

The likelihood function is:

$$L(\vec{s}) = \int_{\vec{x}} f(\vec{x}) \exp\left[-\sum_{i=1}^{n} \frac{\left(r_i - \eta s_i I_0 e^{2x_i}\right)^2}{N_i}\right] d\vec{x}$$

$$= \exp\left[-\sum_{\substack{i=1\\ l_i \in S_{Off}}}^{n-m} \frac{r_{l_i}^2}{N_{l_i}}\right] \cdot \int_{\vec{x}_{On}} f(\vec{x}_{On}) \times \exp\left[\sum_{\substack{i=1\\ l_i \in S_{On}}}^{m} \frac{\left(r_{n_i} - \eta I_0 e^{2x_{n_i}}\right)^2}{N_{n_i}}\right] d\vec{x}_{On}$$

Based on the SMC model:

$$L(\vec{s}) \cong \exp\left[-\sum_{\substack{i=1\\ l_i \in S_{Off}}}^{n-m} \frac{r_{l_i}^2}{N_{l_i}}\right] \cdot \int_{\vec{x}_{On}} f(x_{n_1}) f(x_{n_2}|x_{n_1}) \ldots$$

$$f(x_{n_m}|x_{n_{m-1}}) \exp\left[-\sum_{\substack{i=1\\ l_i \in S_{On}}}^{m} \frac{\left(r_{n_i} - \eta I_0 e^{2x_{n_i}}\right)^2}{N_{n_i}}\right] d\vec{x}_{On}.$$

By decoupling the integral:

$$L(\vec{s}) = \frac{\left[\prod_{\substack{i=1\\ n_i \in S_{On}}}^{m-1} \int f(x_{n_i}, x_{n_{i+1}}) \Phi(r_{n_i}, r_{n_{i+1}}, x_{n_i}, x_{n_{i+1}}) dx_{n_i} dx_{n_{i+1}}\right] \exp\left[-\prod_{\substack{i=1\\ n_i \in S_{Off}}}^{n-m} \frac{r_{l_i}^2}{N_{l_i}}\right]}{\prod_{\substack{i=1\\ n_i \in S_{On}}}^{m-1} \int_{-\infty}^{\infty} f(x_{n_i}) \exp\left[-\frac{(r_{n_i} - \eta I_0 e^{2x_{n_i}})^2}{N_{n_i}}\right] dx_{n_i}}$$

The branch metrics can be found as follows:

$$BM_k(\mathbf{On}) = \frac{\iint f(x_{n_{i-1}}, x_{n_i}) \Phi(r_{n_{i-1}}, r_{n_i}, x_{n_{i-1}}, x_{n_i}) dx_{n_{i-1}} dx_{n_i}}{\int f(x_{n_{i-1}}) \exp\left[-\frac{\left(r_{n_{i-1}} - \eta I_0 e^{2x_{n_{i-1}}}\right)^2}{N_{n_{i-1}}}\right] dx_{n_{1-1}}}$$

$$BM_k(\mathbf{Off}) = \exp\left(-\frac{r_k^2}{N_k}\right)$$

In terms of branch metrics MLSD can be expressed as:

$$\vec{s} = \arg\max_{\vec{s}} \prod_{\substack{i=2\\ n_i \in S_{On}}}^{m} BM_{n_i}(\mathbf{On}) \prod_{\substack{i=1\\ l_i \in S_{Off}}}^{n-m} BM_{l_i}(\mathbf{Off})$$

* *Note:* the MLSD is the product of the branch metrics

C. MLSD based on SMC fading temporal correlation using Viterbi Algorithm

This MLSD has the least complexity which is n^2/2 since it requires only 2D integration. If the most recent bit is ON, the computation of branch metric is easy since only one survivor path. Otherwise we must keep track of all survivor paths whose last bit is OFF. The non survivor paths can only be eliminated if the most recent bit is ON. The complexity can be reduced further by making the first and last bit of each sequence to be ON. The disadvantage of this technique is that a large size memory is needed to keep track of the survivor path information and some bit overhead is required for the implementation of this algorithm.

IV. MLSD ALGORITHM

Modulation systems with memory can be modeled as finite-state machines that might be represented by a trellis. If it is assumed that the transmitted signal has duration of K symbol intervals, each path of length K through the trellis a message signal.

The number of messages in this case is equal to the number of paths through the trellis, and a Maximum Likelihood Sequence Detection (MLSD) algorithm selects the most likely path or sequence corresponding to the received signal **r(t)** over the K signaling interval.

As it has been handled ML detection selects a path of K signals through the trellis by minimizing the Euclidean distance between that path and **r(t)**. Noting that since

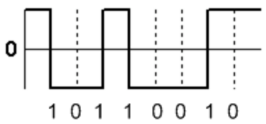
$$\int_0^{KTs} |r(t) - s(t)|^2 dt \;=\; \sum_{K=1}^{K} \int_{(K-1)Ts}^{KTs} |r(t) - s(t)|^2 dt$$

The optimal detection rule becomes:

$$\left(\hat{s}^{(1)}, \hat{s}^{(1)}, \ldots \hat{s}^{(K)}\right) = ArgMin_{\left(\hat{s}^{(1)}, \hat{s}^{(1)}, \ldots \hat{s}^{(K)}\right) \epsilon \gamma} \sum_{K=1}^{K} \left\| r^{(K)} - s^{(K)} \right\|^2 =$$

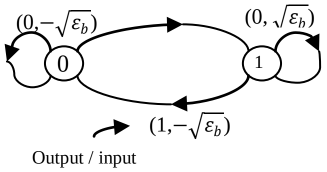
$$ArgMin_{\left(\hat{s}^{(1)}, \hat{s}^{(1)}, \ldots \hat{s}^{(K)}\right) \epsilon \gamma} \sum_{K=1}^{K} D\left(r^{(K)}, s^{(K)}\right)$$; where γ is the trellis

As an example of the algorithm consider the NRZI signal characterized by the trellis in the figure below. The signal transmitted in each interval is binary PAM, and therefore are two possible transmitted signals corresponding to the points S1 = -S2 $=\sqrt{\varepsilon_b}$, where ε_b is the energy per bit.



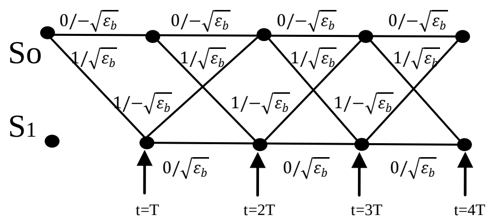
NRZI

Now it is needed to calculate the Euclidean distance for all possible sequences. For NRZI based on binary modulation with 2^k sequences we can reduce the number of sequences in the trellis search by using the Viterbi algorithm to eliminate sequences as new data are received.



State Diagram

The Viterbi algorithm is seen as a sequential trellis search algorithm for ML sequence detection. It is assumed that that the search process begins initially at state S_0. Then, at time t=T, we receive $r_1 = S_1^{(m)} + n_1$ from the demodulator, and at time t=2T it is received $r_2 = S_2^{(m)} + n_2$ and so on. Since the signal memory is 1 bit (L=1) it can be observed that the trellis reaches its regular (steady-state) after two transitions. After the reception of r_2 at t=2T and onwards it is observed that there are two signal paths entering and two signal paths leaving at each node. The two paths entering node S_0 at t=2T correspond to bits (0,0) & (1,1) or seen as signal points $(-\sqrt{\varepsilon_b}, -\sqrt{\varepsilon_b})$ & $(\sqrt{\varepsilon_b}, -\sqrt{\varepsilon_b})$. Respectively in node S_1 at t=2T we have (0,1) & (1,0) or $(-\sqrt{\varepsilon_b}, \sqrt{\varepsilon_b})$ & $(\sqrt{\varepsilon_b}, \sqrt{\varepsilon_b})$ as it is shown below:



For the two paths entering node S_0 the Euclidean distance metrics are calculated as follows by using the outputs r_1 and r_2 from the demodulator:

$$D_0(0,0) = \left(r_1 + \sqrt{\varepsilon_b}\right)^2 + \left(r_2 + \sqrt{\varepsilon_b}\right)^2$$
$$D_0(1,1) = \left(r_1 - \sqrt{\varepsilon_b}\right)^2 + \left(r_2 + \sqrt{\varepsilon_b}\right)^2$$

The Viterbi algorithm then compares these two metrics and discards the path with the largest (distance) metric and the other path (so called the survivor) is saved at t=2T. The elimination of one of the two paths does not compromise the optimality of the Trellis search, because any extension of the path with the larger distance beyond t=2T will always have a larger metric than the survivor that is extend along the same path beyond t=2T.

The Euclidean distances of the two paths entering at node S_1 at t=2T are:

$$D_1(0,1) = \left(r_1 + \sqrt{\varepsilon_b}\right)^2 + \left(r_2 - \sqrt{\varepsilon_b}\right)^2$$
$$D_1(1,0) = \left(r_1 - \sqrt{\varepsilon_b}\right)^2 + \left(r_2 - \sqrt{\varepsilon_b}\right)^2$$

by using the outputs r_1 and r_2 from the demodulator.

The two metrics are compared and the signal path with the larger metric is eliminated. This leaves two survivor paths at t=2T, one at node S_0 and the other at S_1 with their metrics. The signal paths are extended along the two survivors ($D_0(1,1)$ & $D_1(1,0)$), and upon the reception of r_3 at T=3T the paths (1,1) at S_0 and (1,0) at S_1. Thus the two metrics for the paths entering S_0 are:

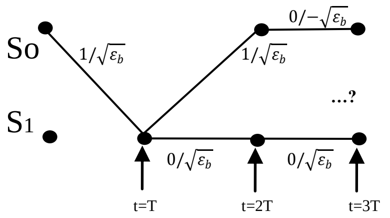
$$D_0(1,1,0) = D_0(1,1) + \left(r_3 + \sqrt{\varepsilon_b}\right)^2$$
$$D_0(1,0,1) = D_1(1,0) + \left(r_3 + \sqrt{\varepsilon_b}\right)^2$$

These two metrics are compared, and the path with the larger distance metric is eliminated. Similarly for the two paths entering S_1 at t=3T are:

$$D_1(1,1,1) = D_0(1,1) + \left(r_3 - \sqrt{\varepsilon_b}\right)^2$$
$$D_1(1,0,0) = D_1(1,0) + \left(r_3 - \sqrt{\varepsilon_b}\right)^2$$

Again the two metrics are compared and the one with greater distance is eliminated. This process is continued as a new signal sample is received from the demodulator.

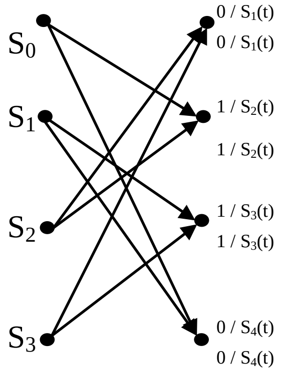
The VA computes two metrics for the two paths entering a node at each node and the two survivors are extended to the next state. Therefore the number of paths searched in the trellis is reduced by a factor of 2 at each stage.



As a second example, in order to extend the trellis search of the VA for M-ary modulation consider a system employing M=4 signals, characterized by the four-state trellis shown in the figure below where two signal paths enter and other two leave each node. Therefore the VA will have four survivors at each stage and their corresponding metrics.

Two metrics corresponding to the two paths computed at each node, and one of the two signal paths entering the node is eliminated at each stage of the trellis. Thus, the VA minimizes the number of trellis paths searched in performing MLSD.

If we had advanced to a stage K, where K>>L in the trellis, and we compare the surviving sequences they will be identical in bit or symbol position K-5L and less. In practical implementation of the VA, decisions on each information bit (or symbol) are delayed 5L bits (or symbols). Thus, a variable delay in bit or symbol detection is avoided, and the loss in performance from the optimum detection procedure is negligible if delay is at least 5L. This approach of the VA is called *Path Memory Truncation.*



One-stage Trellis diagram for delay modulation

V. Conclusion

The MLSD is a powerful way for sequence detection at the receiver compared to bit by bit detection. The VA, a widely used tool for sequence detection was explained in order to provide a background and a better understanding of the presented topic.

The MLSD algorithm finds the ML sequence in a recursive way to find the survivor path for each state and in turn the complete sequence is detected. The other algorithm stated for maximum likelihood sequence detection was based on single step Markov-Chain can be implemented to find MLSD if the temporal correlation of fading is known. Two sub optimal schemes were derived based on that. When the VA is merged with the second sub optimal scheme, it gave the lowest possible complexity which is $n^2/2$.

.

References

[1] Tor M. Aulin, *Breadth-First Maximum Likelihood Sequence Detection: Basics*, IEEE Transactions on Communications, Vol. 47, No. 2, February 1999.

[2] Hui-Ling Lou, *Implementing the Viterbi Algorithm: Fundamental and real-time issues for processor designers*, IEEE Signal Processing Magazine, September 1995.

[3] X. Zhu and J. M. Kahn, *Markov Chain Model in Maximum Likelihood Sequence Detection for Free-Space Optical Communication Through Atmospheric Turbulence Channels*, IEEE Transactions on Communications, Vol. 51, No. 3, March 2003.

[4] J. R. Barry, E. A. Lee and David G. Messershmitt, *Digital Communication*, 3rd ed., Ed. NY, USA: Springer-Verlag, 2004.

[5] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed., Mc-Graw Hill, 2008.