An Analysis of Optimal and Sub-Optimal Sequence Estimation Algorithms for the Wireless Channel

A Project Report

submitted by

AKSHAYAA MAGESH

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2018

THESIS CERTIFICATE

This is to certify that the thesis titled **An Analysis of Optimal and Sub-Optimal Sequence Estimation Algorithms for the Wireless Channel**, submitted by **Akshayaa Magesh**, to the Indian Institute of Technology, Madras, for the award of the degree of **B.Tech**, is a bona fide record of the research work done by her under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Giridhar Krishnamurthy
Research Guide,
Professor,
Dept. of Electrical Engineering,
IIT-Madras, 600 036

Place: Chennai

Date: 14th May 2018

ACKNOWLEDGEMENTS

I am deeply grateful to my research guide Dr. Giridhar Krishnamurthy for his unwavering support throughout the course of my project. Successful completion of this project would not have been possible without his valuable inputs and guidance.

I would like to thank my lab mate Krishna Madan for his most helpful notes and suggestions. His constant support was of great value in the process of fixing problems throughout the course of this project. I would also like to thank Manideep Dunna for all the support he provided.

I would also like to thank my parents and my sister for their unequivocal love and support.

ABSTRACT

KEYWORDS: **ISI, Viterbi, MLSE, Delayed Decision Feedback Sequence Estimation (DDFSE), Channel tracking, Per Survivor Processing**

The aim of this project is to study the performance of sequence estimation algorithms in the presence of dispersive (ISI), time varying and noisy channels. We first look at the Viterbi algorithm, which is the optimal technique for Maximum Likelihood Sequence Estimation under the assumption that all the parameters characterizing the channel are known at the receiver. Due to the very high complexity of the Viterbi algorithm for channels with a long channel response and because it is inapplicable for channels with infinite channel response, we look at the Delayed Decision Feedback Sequence Estimation (DDFSE) algorithm. The DDFSE algorithm is a reduced state Viterbi algorithm with feedback incorporated into the structure of path metric computations. We then look at one possible realization of the Per Survivor Processing technique - the adaptive MLSE technique to deal with a time varying Rayleigh fading channel with ISI. In this we use Least Mean Square (LMS) for the channel tracking in the realization of adaptive MLSE decoders. We then extend this application of PSP to an adaptive DDFSE scenario to reduce computational complexity.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

BER	Bit Error rate
ML	Maximum Likelihood
ISI	Inter Symbol Interference
AWGN	Additive White Gaussian Noise
LE	Linear Equalizer
DFE	Decision Feedback Equalizer
DDFSE	Delayed Decision Feedback Sequence Estimation
MLSE	Maximum Likelihood Sequence Estimation
PSP	Per Survivor Processing
LMS	Least Mean Squares
PDP	Power Delay Profile
QPSK	Quadrature Phase Shift Keying
SNR	Signal to Noise Ratio
CM	Cumulative Metric

NOTATION

$\mathbf{x}$	Bold face letters denote column vectors or matrices
$\mathbf{s}$	Transmitted symbols
L	Length of channel impulse response
N	Number of symbols transmitted
u	Complexity parameter for DDFSE
$\mathbf{r}$	Received sequence at the receiver
$\mathbf{h}$	Channel impulse response
M	Modulation constellation size
$\mathbf{x_k^l}$	Vector at time instant k and corresponding to state l
$x[i]$	ith element of vector $\mathbf{x}$

CHAPTER 1

Introduction

The medium for wireless communications is the radio channel between the transmitter (Tx) and the receiver (Rx). There can exist a direct Line-of-Sight (LOS) path between the Tx and the Rx or the signal may reach the Rx by being reflected by different and possibly moving Interacting Objects (IOs) or it can be a superposition of both. Each of these multipath components have a different amplitude and delay. Thus, mobile radio communications have time varying channels, characterized by small-scale fading effects such as multipath fading leading to Inter Symbol Interference (ISI) in addition to the presence of AWGN noise.

The cascade of the encoder and transmission channel impulse response (assuming it is finite) may be described as a finite state machine with associated state diagrams and trellis diagrams. Data detection approaches can be divided into symbol-by-symbol detection methods and sequence estimation techniques. While symbol-by-symbol detection methods have very low complexity, they have undesirably high error rates in the wireless environment. The Viterbi algorithm was initially proposed in 1967 as a method to decode convolutional codes. Since then, among other applications, it has been a primary candidate for maximum likelihood sequence estimation. Under the assumption that the receiver has exact channel state information, the Viterbi algorithm gives the maximum likelihood solution for signal corrupted with ISI and AWGN by searching for the path with minimum cost in the trellis diagram.

However with growth in channel length and modulation constellation size, the complexity of the Viterbi algorithm increases exponentially and is not practically implementable. The algorithm is also not applicable in cases with infinite channel response. Various methods have been suggested to deal with increasing channel response length. One of them is the truncation of the channel response. This method leads to last few components of ISI not being considered and the error propagation is catastrophic. Another approach to reduce complexity is to use a LE (Linear Equaliser) or DFE (Decision Feedback Equaliser) to cancel the tail of ISI part before sending it to the Viterbi algorithm. But pre-filtering still causes significant error propagation and high BER.

To get a better BER performance with reduced complexity, we look at Delayed Decision Feedback Sequence Estimation (DDFSE). The complexity of the algorithm is varied by a factor u, which can be varied from 0 to $L-1$ (where L is the number of taps in the channel response) for minimum to maximum complexity respectively. It is based on a trellis with the number of states exponential in u. When $u = 0$, it reduces to the Decision Feedback Equalizer (DFE). If channel memory is finite, at u equal to $L-1$, it is equivalent to the Viterbi algorithm. For intermediate values of u, it can be described as a reduced state Viterbi algorithm with feedback incorporated in the structure of path metric calculations.

The above mentioned sequence estimation algorithms are suited for time invariant ISI channels. In order to deal with fast time varying channels, we look one instance of the Per Survivor Processing (PSP) principle as the adaptive MLSE. In this particular realization of PSP, channels are updated independently for each survival trellis path and the decision feedback is retrieved from each individual trellis path with no decision feedback delay. We use Least Mean Squares (LMS) method for the per-survivor channel tracking. The PSP further increases the complexity of the MLSE algorithm and as the next step we look at applying the PSP principle to DDFSE to reduce complexity and study the trade-off between complexity and performance.

The remaining of this thesis is organized as follows. Chapter 2 defines the system model and the notation we use for the remainder of the thesis. Chapter 3 look at the structure of the Viterbi algorithm and its performance under various values for decoding delays. Chapter 4 details the DDFSE algorithm and its workings and we compare its performance to the Viterbi algorithm. Chapter 5 looks at PSP and its realization with Viterbi and DDFSE and future work in the area.

CHAPTER 2

System and Channel Models

2.1 System Model

The complex symbols to be transmitted is denoted by the vector $\mathbf{s}$ of length N corresponding to symbols transmitted from time 1 to N. The radio channel in a wireless communication system is often characterized by a multipath propagation model. The propagation environment is aptly modelled by a few dominant paths (usually 3 to 6). For Chapters 3 and 4 where we explore the Viterbi and DDFSE algorithm, we assume a time invariant channel model with only multipath components. The multipath channel model used here is modelled using the PED-B power delay profile (PDP). The channel impulse response is denoted by the vector $\mathbf{h}$ and has L taps. In Chapter 5 we look at a time varying Rayleigh channel. The channel response at the time instant k ($1 <= k <= N$) is denoted by the vector $\mathbf{h_k}$ having L taps and the fading across time is characterized by Rayleigh fading.

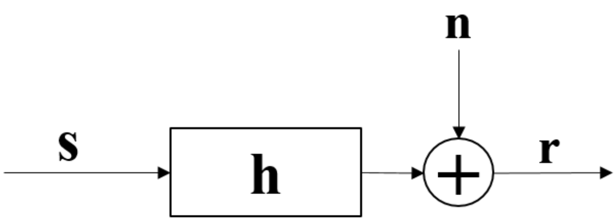

Figure 2.1: Received signal with ISI and AWGN

The discretized received signal at the receiver is denoted by a vector $\mathbf{r}$ of length N. Due to multiple taps being present in the channel response, the received sequence is a convolution of the transmitted symbols and channel response as shown in Figure 2.1. The received symbol at kth time instant is given by:

$$r_k = \sum_{i=0}^{L-1} s_{k-i} h_i + n_k \tag{2.1}$$

where r_k is the kth received symbol, s_k is the kth transmitted symbol, h_i is the ith tap of channel response and n_k is the AWGN noise term, a circular Gaussian random variable with real and imaginary parts each having variance $N0/2$.

2.2 Channel models

The channel impulse response used in Chapters 3 and 4 follow the PED-B PDP. The PED-B channel model is one of the commonly used set of empirical channel models in the ITU-R recommendation for outdoor-to-indoor pedestrian for medium delay spread [6]. The relative delay and average power for this channel model is as in Table 2.1.

Tap	Relative delay (ns)	Average Power (db)
1	0	0
2	200	-0.9
3	800	-4.9
4	1200	-8.0
5	2300	-7.8
6	3700	-23.9

Table 2.1: Caption

In Chapter 5, we look at a channel model with Rayleigh fading. To generate channel impulse responses with Rayleigh fading, the modified Jakes model [4] is used. The modified Jakes model employs the use of Walsh-Hadamard matrices (WH) in order to produce several uncorrelated waveforms. For N_0 as a power of 2, the waveform is generated with $N_0 + 1$ oscillators and the jth fading waveform is generated as:

$$T(k,j) = \sqrt{\frac{2}{N_0}} \sum_{n=0}^{N_0} A_j(n) \times ([cos(\beta_n) + jsin(\beta_n)]cos(\omega_n k + \theta_n)) \tag{2.2}$$

with $N = 4 * N_0$ rays arriving at angles $\alpha_n = 2\pi(n - 0.5)/N$ such that ray n experiences a Doppler shift of $\omega_n = \omega_D cos(\alpha_n)$, where ω_D is the maximum Doppler shift corresponding to the Doppler frequency. By using $\beta_n = \pi n/N_0$, the real and imaginary parts of the waveform have equal power and are uncorrelated and the variable θ_n is randomised to give different realisations of the waveforms. The parameter $A_j(n)$ is the jth WH code sequence in n. The WH codewords are orthogonal vectors and give zero inner product with one another. Thus the Rayleigh fading waveforms generated using this method are uncorrelated. L waveforms are generated for a channel impulse response of length L and the each tap is multiplied by the corresponding amplitude according to the required power delay profile.

CHAPTER 3

MLSE - Viterbi

3.1 Introduction

The Viterbi algorithm [1] operates on the trellis model to provide the optimal maximum likelihood solution for a data sequence tampered by ISI and AWGN. In the trellis, each node corresponds to a distinct state at a given time, and each branch represents a transition to some new state at the next instant of time. Its most important property is that to every possible state sequence, there exists a unique path through the trellis and vice versa. If the size of the modulation constellation is M (for example it is 4 for QPSK), the number of states at each instant of the trellis grows as M^{L-1} where L is the number of taps in the channel response. At each instant of time, the algorithm takes decisions recursively to get the transition from previous state to the next state.

3.2 The Viterbi Algorithm

At each time instant k, the trellis has M^{L-1} states and the lth state is denoted by the vector $\mathbf{x_k}$ of length $L-1$ where

$$\mathbf{x_k^l} = [x(k-L+2), ..., x(k-1), x(k)] \tag{3.1}$$

Each element of the state vector $\mathbf{x_k^l}$ can be an element from the modulation constellation, thus giving a total of M^{L-1} states. Each of these states store a survivor sequence and its corresponding cumulative metric upto that point of time. The survivor sequence of each state consists of the decoded symbols upto that node at each point of time and the cumulative metric keeps a track of the corresponding error of that survivor sequence as compared to the received symbols. The inputs to the algorithm are the received vector $\mathbf{r}$ and the channel response vector $\mathbf{h}$. The survivor sequence is updated symbol by symbol at each time instant from 1 to N by looking at the cumulative metrics of each of its previous states and branch metrics of the transition from previous state. At time instant k, each state can be accessed from M previous states at time $k-1$. For each state l, the branch metric from previous state m is calculated as

$$bm_{k-1,k}^{m,l} = |r[k] - (\sum_{i=1}^{L-1} x_{k-1}^{m}[i]h[L+1-i] + x_k^l[L-1]h[1])|^2 \quad (3.2)$$

The survivor sequence is chosen according to the least cumulative metric.

3.2.1 Inputs

(a) Received symbol vector $\mathbf{r}$ of length N

(b) Channel impulse response $\mathbf{h}$ of length L

3.2.2 Storage and initialization

(a) Survivor sequence: Each state stores a survivor sequence. The survivor sequence of state l is denoted by the vector $\mathbf{SP^l}$, a vector of length N and at the end of time k, each survivor sequence is updated to its kth element.

(b) Metrics: Each state stores a cumulative metric and the cumulative metric of state l is denoted by CM^l. At time 0, all the cumulative metrics are initialized to 0. The branch metric from state m to l from time $k-1$ to k is denoted by $bm_{k-1,k}^{m,l}$ and is calculated as mentioned above.

(c) Till time $k=0$, the bits transmitted are assumed to be 0

(d) The last $L-1$ symbols to be transmitted are forced to a known symbol in order to force the trellis to a single state at the end of the algorithm in order to get a zero error floor in the presence of no noise or very high SNR.

3.2.3 Steps of the Algorithm

(a) At each time instant k
   (a) For each state l from $1...M^{L-1}$ (for time instants k from $N-L+2$ to N the number of allowed states decrease as a factor of M as we are forcing last $L-1$ symbols to known bits), find the survivor path and $\mathbf{SP^l}$ and the cumulative metric CM^l as follows
      i. Each state l has M previous possible states. Calculate the branch metric from each of the previous states as mentioned above. The previous state m whose $CM^m + bm_{k-1,k}^{m,l}$ is the least has its survivor sequence chosen for state l
      ii. The survivor sequence of the state l is updated by appending the survivor sequence of the state m with $x_k^l[L-1]$

iii. The CM^l is updated as $CM^m + bm^{m,l}_{k-1,k}$

(b) Since the last $L-1$ symbols are forced to known values, the trellis collapses to a known state. The survivor sequence of this known state is the decoded sequence

3.3 Decoding delay

Certain modifications are necessary while implementing this in practice. For very long transmitted sequences, it is impractical to wait for the whole sequence to get decoded. And observing the trellis pattern can reveal that the survivor sequences of all the states merge after a certain delay. In other words, it is possible to make a decision at time k for a symbol of index $k-\delta$. In general, if the decoding delay δ is chosen large enough, the BER performance is not affected much. We look at 3 possible strategies for choosing the decoded symbol at time k:

(a) Minimum Cumulative Metric (CM): At time k, we choose the $(k-\delta)$th element of the survivor sequence with the minimum cumulative metric and give that as the decoded symbol.

(b) Majority rule: At time k, we look at the $(k-\delta)$th elements of all M^{L-1} survivors and choose the symbol that is in majority number

(c) Random Selection: At time k, we choose the $(k-\delta)$th element of the survivor sequence of a random state.

3.4 Simulation Results

We run the Viterbi algorithm for a channel response length of $L = 3$. The symbols are QPSK gray coded, thus giving a constellation size of $M = 4$. The channel response $\mathbf{h}$ follows PED-B power delay profile. The power of the vector $\mathbf{h}$ is normalised to 1, so the symbol energy Es has a mean value of 1. For QPSK modulation, the following relation holds : $Es/N0 = 2 * Eb/N0$. The simulation is done for range of Eb/N0 from 0dB to 14dB. The plot of BER vs Eb/N0 for the Viterbi algorithm with channel length $L = 3$ (without the application of any decoding delay, ie. the decoded sequence is taken at the end of the algorithm) is given in Figure 3.1

The performance of the Viterbi algorithm under the presence of decoding delays was also studied. The three methods mentioned in the previous section were implemented.

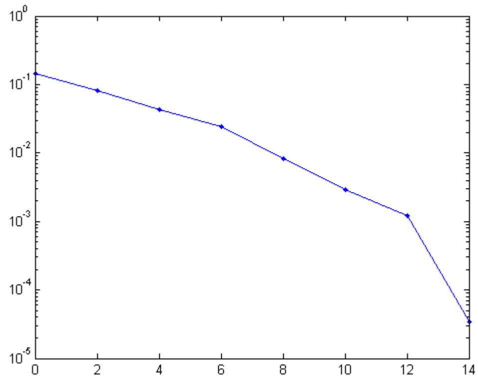

Figure 3.1: BER vs Eb/N0 for L = 3

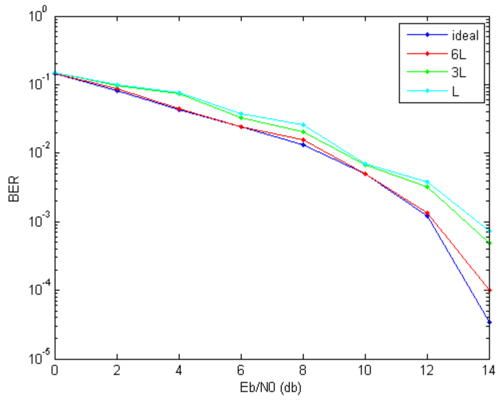

Figure 3.2: BER vs Eb/N0 for different decoding delays for minimum CM method

The BER curves for decoding delays of $\delta = 6L$, $\delta = 3L$ and $\delta = L$ against Eb/N0 for minimum cumulative metric method is as shown in Figure 3.2 along with the ideal BER curve (decoded signal taken after end of the algorithm). The BER curves for the same decoding delay values for the majority rule method is as shown in Figure 3.3. The performance for the random selection method is as in Figure 3.4.

The performance for $\delta = 6L$ almost matches the performance of the ideal Viterbi algorithm for all three methods. Hence using a decoding delay of $6L$ (18 symbols in this case) or even $5L$ (15 symbols) can give very good BER values.

As decoding delay decreases, the performance in all three methods deteriorate. However the deterioration as we move to $\delta = 3L$ and $\delta = L$ (9 and 3 symbols respectively) in the minimum CM method is much lesser than in the majority rule method or the random selection method. The random selection method performs the worst as we move to lower values of decoding delays. Thus, in cases where lower values of decoding delays are required, the minimum CM is the best way to go.

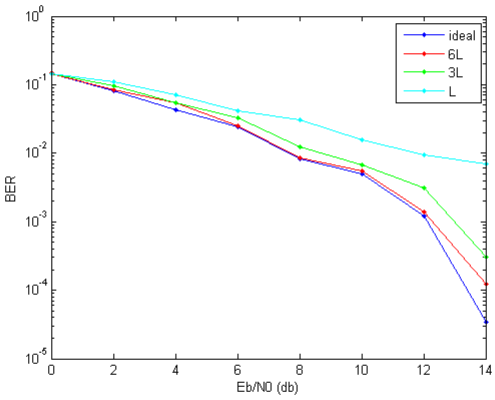

Figure 3.3: BER vs Eb/N0 for different decoding delays for majority rule method

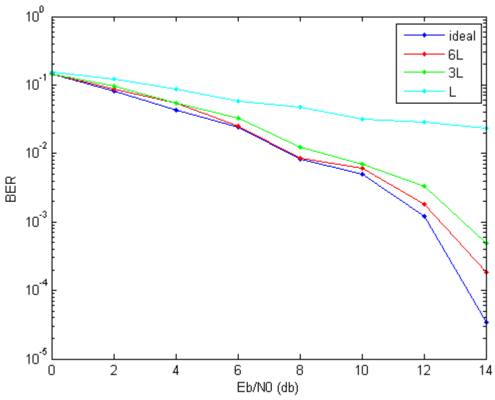

Figure 3.4: BER vs Eb/N0 for different decoding delays for random selection method

CHAPTER 4

Delayed Decision Feedback Sequence Estimation

4.1 Introduction

The DDFSE algorithm [2] is a method to reduce the complexity of the Viterbi algorithm with some trade off on the performance. The complexity of the algorithm is controlled by a factor u which can be varied from 0 to $L-1$ where L is the number of taps in the channel response. The parameter u serves as a way to define the tradeoff between complexity and performance of the algorithm. As in the Viterbi algorithm, at each step, the states describe all possible values taken on by a finite number u of previous inputs. While the parameter u is at its minimum value of 0, the DDFSE algorithm is equivalent to a simple decision feedback equaliser and when u is $L-1$, it is equivalent to the Viterbi algorithm.

4.2 U-V decomposition of the channel

The discrete channel impulse response is assumed to have L taps. This discrete time channel is specified by a causal, rational transfer function $H(D) = \sum_{i=0}^{L-1} h_i D^{-i}$, assuming the channel impulse response is finite. The state machine defined using this transfer function gives a state space $\mathbf{S}$ where $|\mathbf{S}| = M^{L-1}$, since $L-1$ is the degree of $H(D)$. The state space $\mathbf{S}$ can be decomposed into $\mathbf{U} \times \mathbf{V}$ where $|\mathbf{U}| = M^u$ and u serves as the reduced memory of the channel. The decomposition is obtained by representing the transfer function as

$$H(D) = H_u(D) + D^{u+1} H^+(D) \tag{4.1}$$

where $H_u(D) = \sum_{i=0}^{u} h_i D^i$ and $H^+(D) = \sum_{i=u+1}^{L-1} h_i D^{i-u-1}$.

Defining w_k as

$$w_k = \sum_{i=0}^{L-u-2} h_{i+u+1} s_{k-i} \tag{4.2}$$

we can define the received sequence as a result of ISI and AWGN as

$$r_k = \sum_{i=0}^{u} h_i s_{k-i} + w_{k-u-1} + n_k \tag{4.3}$$

Thus the state of the system at time k can be divided into a reduced state $\mathbf{u_k} = [x_{k-u+1}, ...x_k]$ and partial state $\mathbf{v_k} = [x_{k-L+2}, ...x_{k-u}]$. The reduced state is used to build the trellis and the partial state is estimated at each time for every state to be used in the branch metric computations. The partial state $\mathbf{v_{k+1}}$ at time $k+1$ is a function of $\mathbf{u_k}$ and $\mathbf{v_k}$.

4.3 The DDFSE Algorithm

The DDFSE has a trellis with reduced number of states. The number of states in the trellis is M^u where M is the constellation size and u is the complexity parameter. The reduced states form the trellis as the complete state space did in the Viterbi algorithm. The algorithm proceeds the same as in the Viterbi algorithm with the updation of survivor sequence and cumulative metric using the reduced states and the branch metric from state m at time $k-1$ to state l at time k is calculated as

$$bm_{k-1,k}^{m,l} = |r[k] - (\sum_{i=0}^{u} u_{k-1}^{m}[i]h[u+1-i] + u_k^l[u]h[1] + w_{k-u-1})|^2 \tag{4.4}$$

where w_{k-u-1} is calculated using the estimated partial state $\mathbf{v_{k-1}}$ at time $k-1$.

As in the Viterbi algorithm, the last $u-1$ symbols of the transmitted signal are forced to known values in order to force the trellis to a known state at the end of the algorithm. The storage required for survivor sequences and cumulative metrics also reduce drastically with reduction in the parameter u. M^u survivor sequences of length N and cumulative metrics corresponding to the same are stored.

For each time instant k, the survivor sequence and the cumulative metric for each state l from $1...M^u$ are updated as follows:

(a) Each state l has M previous states. The branch metrics $bm_{k-1,k}^{m,l}$ for all possible previous states are calculated (this would involve the partial states stored in the previous states). The previous state m with the least $CM^m + bm_{k-1,k}^{m,l}$ has its survivor metric chosen for state l

(b) The survivor sequence of state l is updated by appending the survivor sequence of state m with $u_k^l[L-1]$

(c) The CM^l is updated as $CM^m + bm_{k-1,k}^{m,l}$

(d) The partial state is chosen as a function of the previous states of m with the minimum branch metric. The estimate v_k^l is a function of u_{k-1}^m and v_{k-1}^m.

Thus in this manner, the DDFSE algorithm combines the Viterbi algorithm and the Decision Feedback Equaliser. The estimate of the partial state is what stores the feedback information extracted from the best path. By analogy with the Decision Feedback Equaliser, this feedback is used to cancel interference from past inputs greater than u samples in the past. With $u = 0$, this reduces to only the DFE.

4.4 Simulation Results

We run the DDFSE algorithm for a channel response length of $L = 5$ which follows the PED-B profile. QPSK gray coded modulation is followed giving $M = 4$. The DDFSE is simulated for values of $u = 4$ (full Viterbi realisation), $u = 3$ and $u = 2$. The BER performance for Eb/N0 in the range 0dB to 14dB can be seen in Figure 4.1

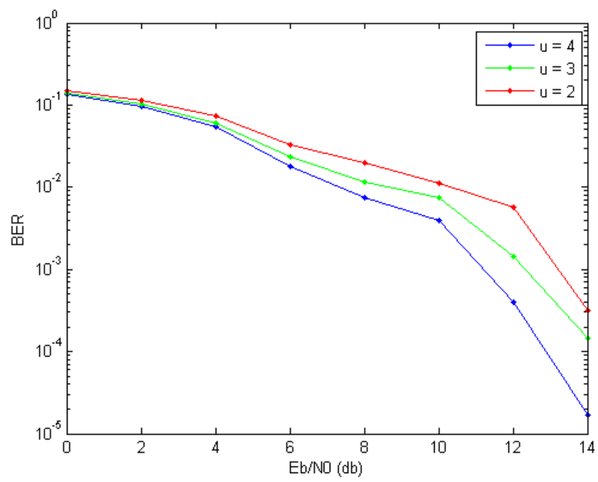

Figure 4.1: BER performance of DDFSE for different values of u and $L = 3$

The complexity of the algorithm reduces drastically from $u = 4$ as 4^4 to $u = 3$ as 4^3 to $u = 2$ as 4^2 and this reflects in the drastic reduction in computation time required. However, the reduction in performance is not as drastic in terms of the BER performance of the algorithm.

Thus, the DDFSE gives a viable alternative for the Viterbi algorithm to get a good BER performance while also reducing the complexity for channels with large impulse responses. By altering the definition of the transfer function U-V decomposition and thus subsequently the definition of the partial state v_k, this algorithm can also be applied to channel impulse responses with infinite length.

CHAPTER 5

Per Survivor Processing

5.1 Motivation

The Viterbi algorithm and the reduced complexity approximations to the Viterbi algorithm give the optimal performance in the case that the channel parameters are exactly known. However in many practical communication systems, these channel parameters may not be known and needs to be decoded along with the data sequence or even with good initial channel estimates, the channel parameters may be varying due to a fast fading environment which is the case in many wireless environments. Thus, data aided channel tracking is a favourable option for such channels. The Per Survivor Processing [3] principle encompasses a variety of techniques that provide a general framework for using the code or survivor sequence associated with each state in the trellis as the data-aiding sequence for the per-survivor estimate of the unknown channel parameters. Thus each state in the trellis, along with a survivor sequence, also has its own channel estimate.

The intuitive rationale behind this method is that since the channel estimates at a particular time are not known exactly, we calculate the transition metrics based on estimates of the channel parameters which is based on the survivor data sequence leading to that transition. Thus, if a particular survivor sequence is right, the channel estimates corresponding to that particular survivor sequence is evaluated using the correct data sequence. Thus at each point of time, since we don't know which survivor is correct, we extend each survivor based on estimates obtained using its associated date sequence. By proceeding to update the channel estimates of each survivor in this manner, the best survivors are extended using the best data sequence available.

Uncertainties in sequence estimation are usually due to imperfect knowledge of some channel parameters such as carrier phase or timing epoch or the impulse response itself. PSP [3] provides a common and unifying approach to deal with all these cases. An appealing aspect of PSP is that the per-survivor channel parameters estimator associated with the best survivor is derived from data information that can be perceived as high quality and with zero-delay, making it appealing for fast fading channels. Also since many possible data sequences are considered, the estimation of channel parameters without a training sequence works better.

In this chapter we look at two instances of the application of PSP principles. The first is adaptive MLSE. In this, the Viterbi algorithm is carried out normally, with each survivor state having a channel estimate, updated through the LMS algorithm aided by previous data sequences. The second is adaptive DDFSE. To reduce the complexity and the memory required to store the channel response estimates, we apply the channel updation principles of PSP to DDFSE.

5.2 Adaptive MLSE

The steps in the algorithm for adaptive MLSE are essentially the same as in the Viterbi algorithm, except for a small change that at each instant of time k, the channel impulse response used for calculation of the metrics is no longer a constant vector $\mathbf{h}$, but a vector specific to that state and the estimate at the previous instant, denoted by $\mathbf{h^l_{k-1}}$. Thus the computation of the branch metric from previous state m to state l, from time instant $k-1$ to k is now calculated as

$$bm^{m,l}_{k-1,k} = |r[k] - (\sum_{i=1}^{L} x^m_{k-1}[i]h^l_{k-1}[L+1-i] + x^l_k[L-1]h^l_{k-1}[1])|^2 \qquad (5.1)$$

The input to the algorithm is now an initial estimate of the channel response. The storage also consists of the channel response for each state in the trellis and is updated at each instant of time. The channel response at time k for state l is denoted by $\mathbf{h^l_{k-1}}$ which is a vector of length L. At $k = 0$ the channel response for all states are initialised to the initial channel estimate and thus at $k = 1$ the initial channel estimates are used for computation of metrics.

At each instant k, for each state l, after the survivor sequence and the cumulative metrics are updated according to the previous state m having the minimum $CM^m + bm^{m,l}_{k-1,k}$, the channel response of state l is updated according the previous channel estimate of state m and the survivor data sequence of l. The updation step goes as follows:

$$\mathbf{h_k^l} = \mathbf{h_{k-1}^m} + \beta * err * \mathbf{a_k} \tag{5.2}$$

where err is defined as

$$err = r[k] - (\sum_{i=1}^{L} x_{k-1}^m[i] h_{k-1}^l[L+1-i] + x_k^l[L-1] h_{k-1}^l[1]) \tag{5.3}$$

and $\mathbf{a_k}$ is a vector of length L and is derived from the survivor sequence of state l, $\mathbf{SP^l}$ as

$$\mathbf{a_k} = [SP^l[k], SP^l[k-1], ..., SP^l[k-L+1]] \tag{5.4}$$

and β is chosen as a tradeoff between the tracking capability and excess Mean Square Error (MSE) as in the traditional LMS algorithm.

5.3 Adaptive DDFSE

The complexity of the Viterbi algorithm increases exponentially with increase in channel length and constellation size. And with the introduction of PSP principles, the memory required and the complexity increases further. Thus, we move to the DDFSE approach and introduce channel tracking as in adaptive MLSE to deal with the fast fading channel.

The U-V decomposition of the channel described in the DDFSE algorithm is applied to the channel estimate of each state. As in the DDFSE algorithm, the number of states in the trellis is M^u where M is the constellation size and u is the complexity parameter. The input to the algorithm is an initial estimate of the channel response. The channel response at time k for state l is denoted by $\mathbf{h_{k-1}^l}$ which is a vector of length L. At $k = 0$ the channel response for all states are initialised to the initial channel estimate and thus at $k = 1$ the initial channel estimates are used for computation of metrics. The estimation of the partial states $\mathbf{v_k}$ is carried out the same way as in DDFSE.

The branch metric from state m at time $k-1$ to state l at time k is calculated using the previous channel estimate of state l as:

$$bm_{k-1,k}^{m,l} = |r[k] - (\sum_{i=0}^{u} u_{k-1}^{m}[i]h_{k-1}^{l}[u+1-i] + u_{k}^{l}[u]h_{k-1}^{l}[1] + w_{k-u-1})|^2 \quad (5.5)$$

where w_k is defined as :

$$w_k = \sum_{i=0}^{L-u-2} h_{k-1}^{l}[i+u+2]s[k-i] \quad (5.6)$$

At each instant k, for each state l (from 1 to M^u) the survivor sequence, the cumulative metrics and the partial states are updated according to the previous state m having the minimum $CM^m + bm_{k-1,k}^{m,l}$ using the branch metric calculation in Equation (5.6). The channel response of state l is updated according the previous channel estimate of state m and the survivor data sequence of l. It is carried out as:

$$\mathbf{h_k^l} = \mathbf{h_{k-1}^m} + \beta * err * \mathbf{a_k} \quad (5.7)$$

where err is calculates as:

$$err = r[k] - (\sum_{i=0}^{u} u_{k-1}^{m}[i]h_{k-1}^{l}[u+1-i] + u_{k}^{l}[u]h_{k-1}^{l}[1] + w_{k-u-1}) \quad (5.8)$$

and β and the vector $\mathbf{a_k}$ are calculated as in adaptive MLSE.

5.4 Simulation Results

The simulation results are for a channel model with impulse response length $L = 3$. The channel impulse response varies over time with Rayleigh fading. The Rayleigh fading model is generated according to the modified Jakes model and the power delay profile used is the PED-B profile according to ITU-R specifications. The Doppler frequency used for Jakes model is 150 Hz. Modulation used is QPSK, giving $M = 4$. All simulations are done for a range of Eb/N0 from 2dB to 12dB.

The optimum performance is obtained when the algorithm at every time instant knows the exact channel response. That is referred to as the 'known channel response' performance in the BER curves. The adaptive MLSE simulation has a full Viterbi realization. It can also be seen as DDFSE with $u = L - 1$ ($u = 2$ in this case). The adaptive DDFSE for $u = 1$ is also carried out. The BER vs Eb/N0 curves for all the three methods mentioned above are as shown in Figure 5.1.

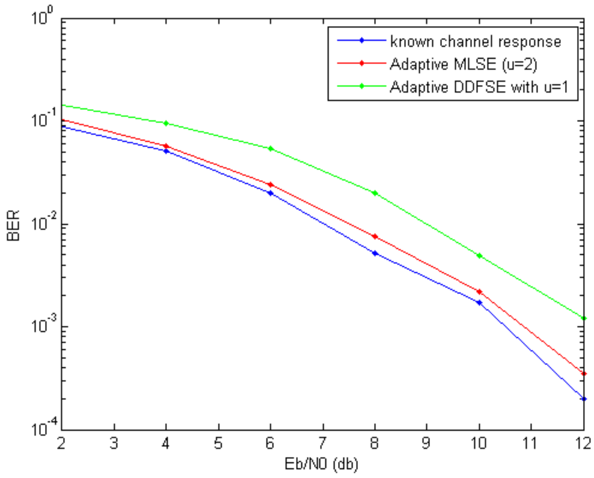

Figure 5.1: BER curves for Viterbi with exact channel estimates known at each instant (blue), adaptive MLSE (red) and adaptive DDFSE with u=1 (green)

The performance of adaptive MLSE matches closely with that of the performance with known channel estimates. There is a degradation of only a fraction of a dB for the adaptive MLSE. As we move to adaptive DDFSE with $u = 1$, that is reduction of one in the complexity parameter, the degradation is about 2dB. As we move to higher Doppler frequencies, the degradation in both adaptive MLSE and adaptive DDFSE worsen due to the nature of the fast fading channels.

5.5 Future work

All the algorithms mentioned above have been studied in a symbol-spaced (T spaced) scenario. In future work, we plan to study the performance of these channel tracking sequence estimation algorithms in a T/2 spaced scenario.

REFERENCES

[1] G. David Forney Jr. (1973). *The Viterbi Algorithm*. Proceedings of the IEEE (Volume: 61, Issue: 3, March 1973).

[2] A. Duel-Hallen and C. Heegard (1989). *Delayed decision-feedback sequence estimation*. IEEE Transactions on Communications (Volume: 37, Issue: 5, May 1989).

[3] R. Raheli, A. Polydoros and Ching-Kae Tzou (1995). *Per-Survivor Processing: a general approach to MLSE in uncertain environments*. IEEE Transactions on Communications (Volume: 43, Issue: 234, Feb./March/April 1995)

[4] Dent, P.G. Bottomley and T.Croft (1993). *Jakes Fading Model Revisited*. Electronics Letters, 29(13), 1162. ISSN 00135194. URL `http://digital-library.theiet.org/content/journals/10.1049/el{_}19930777`.

[5] John G. Proakis . *Digital Communications*. Fourth Edition, McGraw-Hill, 2001.

[6] ITU-R Recommendation M.1225. *Guidelines for evaluation of radio transmission technologies for IMT-2000*. 1997.

CURRICULUM VITAE

1. NAME : Akshayaa Magesh
2. DATE OF BIRTH : 26th Dec 1996
3. E-MAIL : akshayaamagesh@gmail.com
4. EDUCATIONAL QUALIFICATIONS :

   2018 Bachelor of Technology (B.Tech), Electrical Engineering
   Indian Institute of Technology Madras, Chennai