Adaptive Channel Tracking for Fading Mobile Radio Channels: Performance Evaluation on the TETRA Uplink Channel

A Project Report

submitted in partial fulfillment of the requirements
for the award of the degree of

Bachelor of Technology

in

Electrical Engineering

by

Suhas Mathur
EE00249

under the guidance of

Prof. David Koilpillai

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS

May 2004

Certificate

This is to certify that the project entitled **Adaptive Channel Tracking for Fading Mobile Radio Channels: Performance Evaluation on the TETRA uplink channel** by **Suhas Mathur** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology**, is a bona-fide record of work carried out by him under my supervision and guidance at the Department of Electrical Engineering, Indian Institute of Technology Madras.

Place: Chennai [Prof. David Koilpillai]
Date:

Acknowledgments

Prof. David Koilpillai has been a constant source of inspiration during this project and I am grateful to him for having given me an opportunity to work under him. I owe much more than a conventional debt of gratitude to my class mates Hari Balaji and Girish Redekar for the invaluable guidance and help they have given me throughout the course of my project and to Mr. Lars Lindbom for taking time to send me part of his licentiate thesis from Sweden. I would also like to thank my lab-mates at the Intel Lab for providing a vibrant atmosphere for research.

Place: Chennai [Suhas Mathur]
Date:

Abstract

The performance of a wireless communications system is largely guided by the Bit Error Rate that the system experiences on a given wireless channel. The issue of wireless channel equalization and tracking has attracted increased attention in recent years. These techniques are able to provide significant enhancements in performance in some of the most severe channel conditions with time-varying inter-symbol interference, thus opening up new avenues for high bit-rate applications on the mobile radio channel. This thesis deals with the application of adaptive filtering algorithms of low-complexity for channel equalization and tracking so as to counter the effects of multipath propagation and Rayleigh fading. These algorithms are based on the principles of stochastic internal modeling of time-varying coefficients of an FIR channel model. Channel estimators are based on simplified second order models and approximations of a Kalman Estimator. A novel averaging approach is used to replace the online update of a Riccati - equation with a constant matrix. This together with simplified modeling leads to algorithms with high performance at LMS complexity.

The TETRA (TErrestrial Trunked RAdio) Mobile Radio Standard proposed by the European Telecommunications Standards Institute has been used as a case study for the application of these channel tracking algorithms. The performance of these algorithms in conjunction with a Viterbi Decoder is simulated using the TETRA slot structure on the uplink channel for various types of fading channels. Further a simple algorithm is used for the estimation of the instantaneous Doppler spread using the channel estimates provided by the tracking algorithms. This online update from the Doppler estimator is in turn used in an adaptive receiver, that utilizes a priori information about the channel, to improve tracking performance. Receiver parameters such as channel tracker step size can be optimized based on Doppler spread information.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

When digital data is transmitted over a wireless communications channel, noise, intersymbol interference (ISI), and fading limit the achievable performance. If the intersymbol interference is caused by multipath propagation, one way to avoid it would be to decrease the symbol rate. Delayed signals would then have time to arrive before a decision on the next symbol is made. The price to be paid for decreasing the symbol rate is a channel more affected by fading. The increased delays would also limit the efficiency. A superior alternative would be to use an equalizer.

In several digital mobile radio standards including TETRA, the data slot structure and the symbol duration are such that a fading environment would cause severe degradation in BER performance. In such fading environments, an estimate of the channel from the training sequence alone is insufficient for satisfactory decoding of the data frame. In the TETRA standard, a relatively low symbol rate of 18 Kbaud and long data slots (14.166 ms) can cause severe fading, especially at high velocities. Furthermore large variations in fading rates and frequency selectivity may be encountered. Well designed channel estimators are therefore required for obtaining adequate performance. Estimates obtained by interpolating between training sequences would provide inadequate performance in such fading environments. The idea therefore is to design an algorithm that is able to track the variations in the channel as closely as possible while having as low a computational complexity as possible. Sometimes complexity must be traded for performance and vice-versa depending on the requirement of the channel tracking system.

1.2 Organization of this report

This thesis deals with the implementation of recently proposed stochastic modeling techniques in the development of a low complexity high-performance channel estimator and along with it a simple Doppler estimation system to support the channel tracker. The first part of the thesis describes in some detail, the wireless communication system design: the transmitter and the receiver along with the effects of the channel. It then goes into the formal problem formulation for adaptive channel equalization and channel tracking. Finally the simplified stochastic modeling technique, as applied to channel tracking, is presented, along with a simple Doppler estimation algorithm, based on the Maximum Likelihood principle. The report ends with a performance evaluation, through simulations, of the discussed stochastic channel tracking scheme and Doppler estimation algorithm.

CHAPTER 2

The Transmitter

Digital transmission of data over the wireless channel is performance limited by intersymbol interference, noise, fading and interference from other users. While designing the transmitter, these issues are kept in mind so as to minimize the undesirable effects of the above to the maximum possible degree using design parameters that are in our hands. The TETRA mobile transmitter makes use of specifications thus optimized [[?]] for performance given the expected wireless channel conditions under which TETRA systems are expected to operate. In our simulation study we make use of a software radio using the TETRA uplink specifications. Some of these specifications for the transmitter and TETRA uplink channel are described in the sections that follow.

2.1 The TETRA standard

The TETRA Mobile Communications Standard of the European Telecommunications Standards Institute [[?]] has been chosen for testing all the channel tracking algorithms in this thesis through simulation. TETRA is a powerful multi-function radio standard that provide a comprehensive tool kit from which system planners may choose in order to satisfy their requirements. The TETRA suite of radio specifications provide a radio capability encompassing trunked, non-trunked and direct terminal to terminal / system to system communication with a range of facilities including voice, circuit mode data, short data messages, packet mode data services, direct IP-Connectivity, pro-active priority and full encryption. TETRA supports an especially wide range of supplementary services, many of which are exclusive to TETRA. The TETRA platform and standards can also fully support, in a relatively economic way, the designing and implementation of a Public License GSM-like Mobile Telephony System, containing more than all the services and functionalities the GSM / GPRS operators and the market require to day.

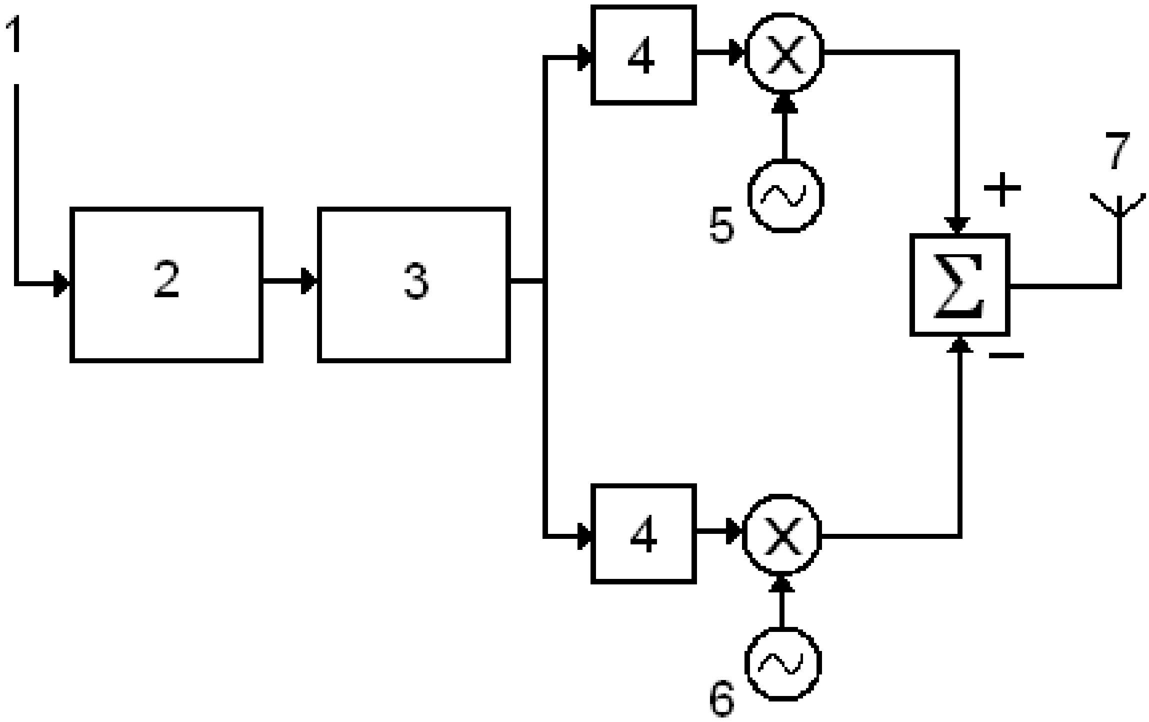

Figure 2.1: Conceptual bock diagram of Transmitter; 1: Raw data in bits; 2: TETRA framing and interleaving; 3: $\pi/4$-shifted DQPSK modulator; 4: SRRC pulse shaping filter; 5: Local Oscillator $\cos 2\pi f_c t$; 6: Local Oscillator $\sin 2\pi f_c t$; 7: Transmit signal

Due to its range of services and the fact that it provides a relatively inexpensive communications network for secure voice and data communications including direct mobile-to-mobile access, TETRA is being looked upon as an important option for the establishment of private communications networks for security and emergency services in European markets and other markets around the world including India.

Fig. 3.1 Shows the TETRA Slot structure and Fig 3.2 shows the TETRA uplink framing hierarchy in detail. The modulation used is $\pi/4$ - shifted DQPSK with square root raised cosine pulse shaping with a roll off factor of $\alpha = 0.35$. Table I summarizes the relevant transmission parameters of TETRA from an equalization point of view. It can be seen that at a vehicle speed of 200 Kmph there are on average more than two fades ($2f_dL_s$) (f_d is the maximum Doppler frequency and L_s is the slot duration.

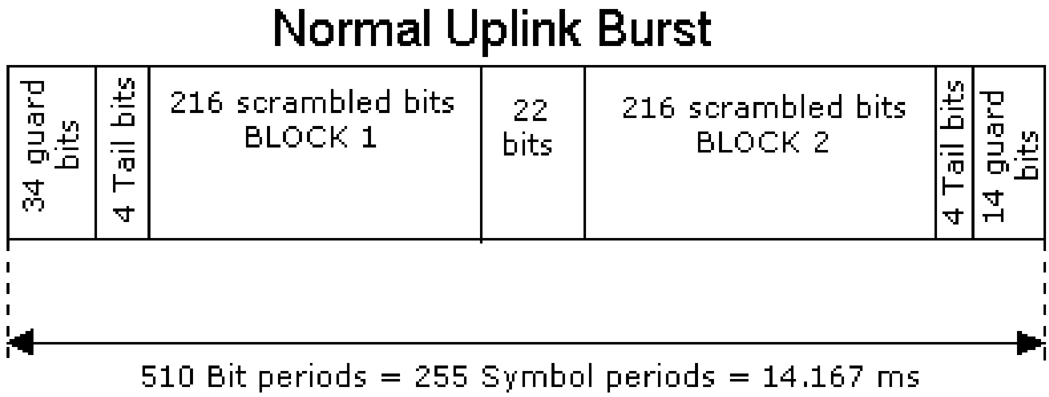

Figure 2.2: A single slot on the TETRA uplink

$2f_dL_s$ is a measure of the average number of "deep fades" the channel experiences in the a single slot) in signal strength per burst. The conclusion from the above table is that a fully adaptive equalizer that continuously tracks the rapid time variations of the channel is definitely needed. The preferred equalizer structure for the TETRA uplink channel is the MLSE rather than the decision feedback equalizer.

Table 2.1: TETRA transmission parameters

Carrier frequency, f_c [Mhz]	450
Channel Spacing, B_c [KHz]	30
Symbol Rate, R_s [kBaud]	18
Excess bandwidth ratio, $\gamma = B_c/R_c$	1.23
Slot length, L_s [ms]	14.167
Maximum Doppler frequency, f_d @ 200 Kmph [Hz]	83
Slot normalized fading rate, f_dL_s @ 200 Kmph	1.18
Symbol normalized fading rate, f_d/R_s @ 200 Kmph	4.6E-3
SRRC Pulse Roll-off factor, α	0.35

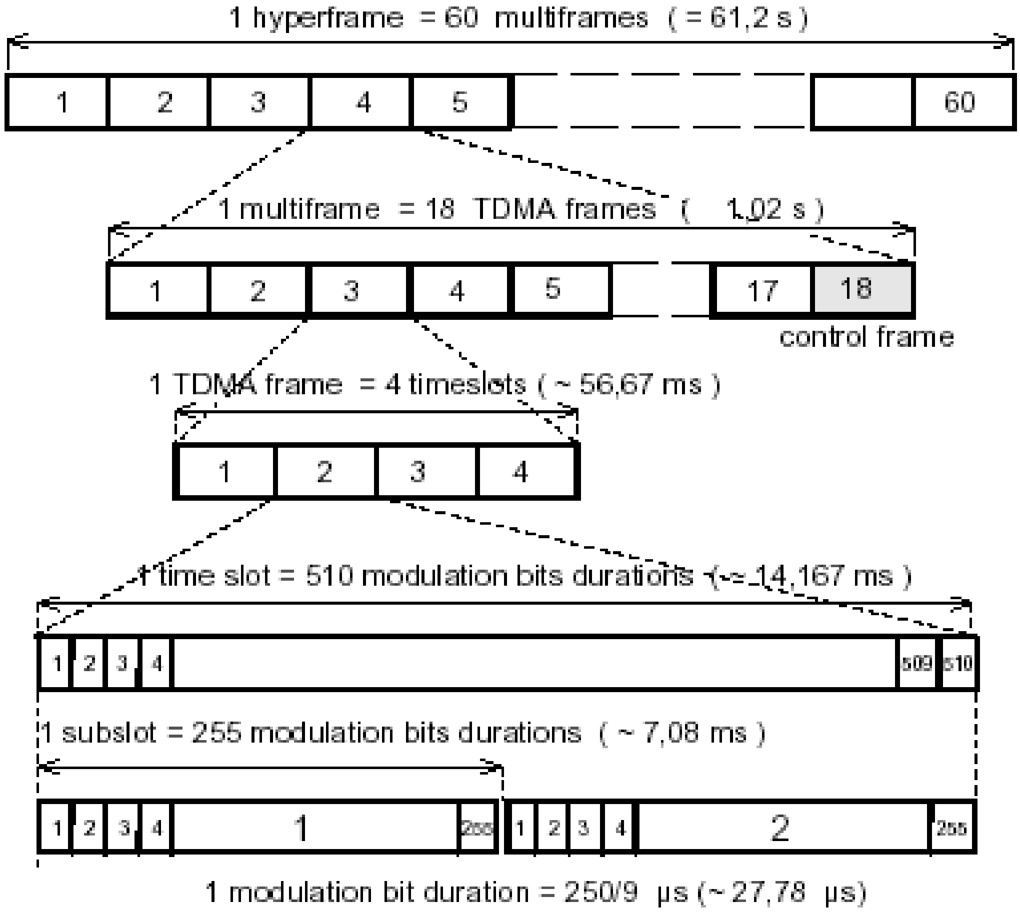

Figure 2.3: The framing hierarchy in TETRA

2.2 $\pi/4$-Shifted DQPSK

$\pi/4$-shifted DQPSK also known as symmetric differential phase exchange keying is a method of modulating raw data bits that is especially suited to the mobile radio channel. The $\pi/4$-shifted DQPSK constellation can be viewed as the superposition of two QPSK signal constellations offset by 45 degrees relative to each other, resulting in eight phases. Symbol phases are alternately selected from one of the QPSK constellations and then the other and, as a result successive symbols have a relative phase difference that is one of four angles $\pm\pi/4$ and $\pm 3\pi/4$. Differential coding of the symbol phases can be achieved by either differentially encoding the source bits and mapping them onto absolute phase angles of a $\pi/4$-shifted QPSK signal constellation or, alternately, by directly mapping the pairs of input bits onto the relative phases

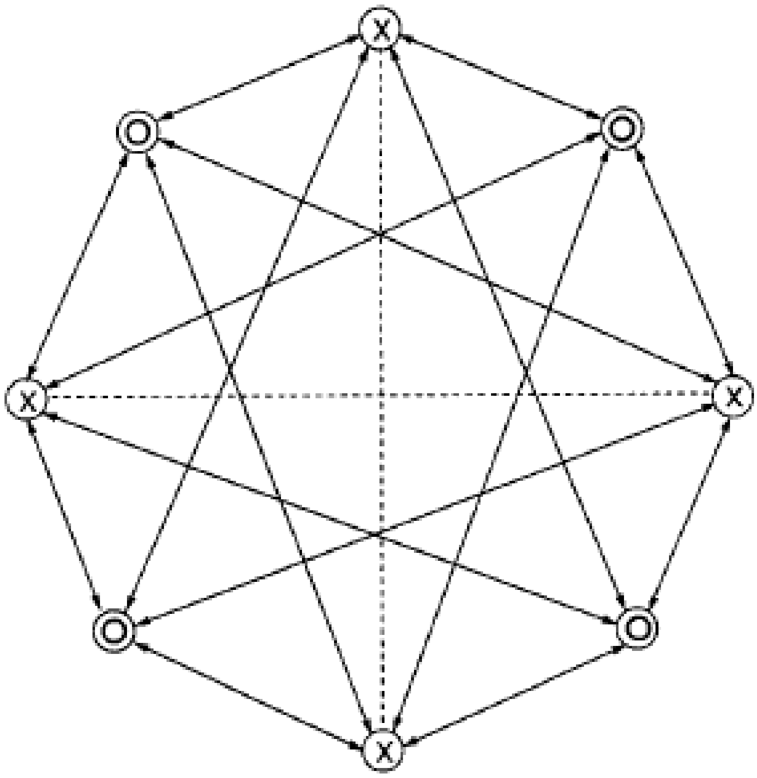

Figure 2.4: $\pi/4$ - shifted DQPSK signal constellation

$\pm\pi/4, \pm3\pi/4$. A feature of this type of modulation is that it can be detected using a non-coherrent detector: a differential detector, or a discriminator followed by an integrate-and-dump filter. The suitability of both differential detection and discriminator detection provides an advantage since both can be performed by low-complexity receiver structures [Che01R]. On the other hand coherent detection requires a more complex receiver than either differential or discriminator detection due to the carrier recovery process. Further, in fast fading environments, coherent detection results in higher irreducible BER than either differential or discriminator detection [Che01R]. In the presence of ISI however, the advantage of differential detection is lost and coherrent detection is required to handle ISI. Another advantage of $\pi/4$-shifted DQPSK is that the transitions in the signal constellation do not pass through the origin. As a result, the envelope of $\pi/4$-shifted DQPSK exhibits less variations and thus therefore has better output spectral characteristics.

2.3 SRRC Pulse Shaping

Pulse shaping of the modulated bit-stream is important from the point of view of minimizing the inter-symbol interference. According to the Nyquist pulse-shaping criterion [[Pro01B]] for zero ISI,

$$B(f) = \sum_{k=-\infty}^{+\infty} X\left(f - \frac{k}{T}\right) = T \tag{2.1}$$

which is derived from the simple condition that

$$x(kT) = \begin{cases} 0 & \text{if } k \neq 0 \\ 1 & \text{if } k = 0 \end{cases} \tag{2.2}$$

where $T =$ symbol rate of data transmission, $x(t)$ is the pulse shape in the time domain and $X(f)$ is the Fourier transform of the pulse shape. The above criterion allows only certain pulse shapes to be used if one is to have zero inter-symbol interference on an ideal transmission channel. The Raised Cosine pulse shape is one such shape that not only satisfies Nyquist's zero ISI criterion but can also be practically achieved through proper filter design. Let $X_{RC}(f)$ be the Fourier transform of the Raised cosine pulse shape. The raised cosine frequency characteristic [[Pro01B]] is given by

$$X_{RC}(f) = \begin{cases} T & \left(0 \leq |f| \leq \frac{1-\beta}{2T}\right) \\ \frac{T}{2}\left(1 + cos\left[\frac{\pi T}{\beta}\left(|f| - \frac{1-\beta}{2T}\right)\right]\right) & \left(\frac{1-\beta}{2T} \leq |f| \leq \frac{1+\beta}{2T}\right) \\ 0 & |f| \geq \frac{1+\beta}{2T} \end{cases} \tag{2.3}$$

Here β is a parameter that determines how 'smoothly' the frequency spectrum of the raised cosine pulse shape falls to zero and is called the 'roll-off factor'. Because of the smooth characteristics of the raised cosine spectrum it is possible to design practical filters for the transmitter and the receiver that approximate the overall desired frequency response. In the special case when the channel is ideal, i.e. $C(f) = 1$, $|f| \leq W$, we have

$$X_{RC}(f) = G_T(f).G_R(f) \tag{2.4}$$

where $G_T(f)$ and $G_R(f)$ are the frequency response of the two filters. In this case, if the receiver filter is matched to the transmitter filter, we have $X_{RC}(f) = G_T(f).G_R(f) = |G_T(f)|^2$. Ideally

$$G_T(f) = \sqrt{|X_{RC}(f)}e^{-j2\pi f t_0} \tag{2.5}$$

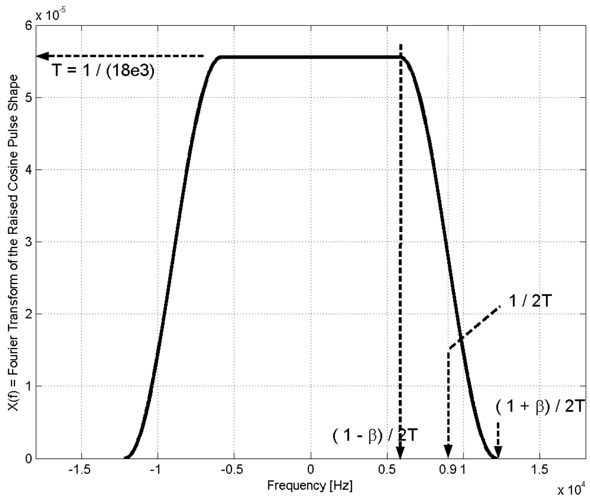

Figure 2.5: Frequency domain representation of the RC pulse amplitude

and $G_R(f) = G_T^*(f)$, where t_0 is some nominal delay that is required to ensure physical realizability of the filter. Thus the overall raised cosine spectral characteristic is split evenly between the receiver and the transmitter. If the raised cosine spectrum is split evenly in this manner, each of the filters is said to have a 'Square Root Raised Cosine' spectrum $X_{SRRC}(f)$. The impulse response of a filter having square-root raised cosine spectral characteristic [[Pro01B]] is given as

$$x_{SRRC}(t) = \frac{(4\beta t/T)cos[\pi(1+\beta)t/T] + sin[\pi(1-\beta)t/T]}{(\pi t/T)[1-(4\beta t/T)^2]} \tag{2.6}$$

This is the time domain representation of the pulse shape that is used to linearly convolve the output of the $\pi/4$-shifted DQPSK modulator. Note that while the ideal pulse extends from $-\infty$ to $+\infty$ on the time axis, in practice the pulse the truncated on either side after a certain number of side lobes. In our simulations, this truncation is set to four lobes on either side. If $\{d_n\}$ represents the sequence of symbols generated by the $\pi/4$ - shifted DQPSK modulator (for DQPSK modulation, each symbol represents 2 bits of information), then the baseband representation of the transmit symbol is given by:

$$s_b(t) = \sum_k d_k x(t - kT) \tag{2.7}$$

Finally the pulse-shaped signal is frequency shifted, i.e. modulated by a carrier, into a suitable frequency band for transmission. Note that $\{d_n\}$ is a complex valued sequence, and therefore so is the pulse shaped signal $s_b(t)$. The real and imaginary parts of the pulse shaped signal are multiplied by local oscillators offset by a phase of $\pi/2$ and the resultant signal added and transmitted by the transmitter antenna.

$$s_T(t) = Re\{s_b(t).e^{j2\pi f_c t}\} \tag{2.8}$$

$$\begin{aligned} s_b(t) = \sum_k d_k x(t-kT) &= \sum_k Re(d_k)x(t-kT) + j.\sum_k Im(d_k)x(t-kT) \\ &= Re(s_b(t)) + j.Im(s_b(t)) = s_1(t) + j.s_2(t) \end{aligned} \tag{2.9}$$

Therefore

$$\begin{aligned} s_T(t) &= Re\{(s_1(t) + j.s_2(t)).(\cos 2\pi f_c t + j.\sin 2\pi f_c t)\} \\ &= s_1(t).\cos 2\pi f_c t - s_2(t)\sin 2\pi f_c t. \end{aligned} \tag{2.10}$$

The above time domain relations result in the following relation in the frequency domain:

$$s_T(w) = \frac{1}{2}\left[S^*(w-w_c) + S(w+w_c)\right] \tag{2.11}$$

where

$$S(w) = S_1(w) + j.S_2(w) \tag{2.12}$$

where $S_1(w)$ and $S_2(w)$ are the Fourier transforms of $s_1(t)$ and $s_2(t)$ above, respectively.

The radio transmitter is a complex part of any radio communications system. Apart from the issues and specifications discussed above, there are a large number of other aspects pertaining to the transmitter such as specifications for emitted power, spectral properties of the mobile terminal emissions, etc. However will not delve into these aspects as they are not relevant to the main subject of this thesis. Having studied the features of the radio transmitter relevant to our study, we must now move on to a discussion on the wireless radio channel and how it is responsible for corruption of digitally transmitted data.

CHAPTER 3

Channel Effects on Transmitted Signal

Having studied the transmitter modeling in sufficient detail it is now advisable to venture into how the channel affects the transmitter signal. This understanding will help us design receiver algorithms at a later stage.

3.1 Large scale path loss :

The term *large scale* is used to denote that *relative* to other kinds of propagation loss, change in loss is slower with time. Large scale path loss models the loss in signal power as a function of distance between the transmitter and receiver as a result of the signal spreading into free space. This loss is usually simplistically modeled as per the *Friis* free space equation:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi^2) d^2 L} \tag{3.1}$$

where
d denotes the line of sight distance between transmitter and receiver
P_r is the signal power at the receiver
P_t is the signal power at the transmitter
G_t and G_r denote the antenna gains at the transmitter and receiver respectively
L denotes the system loss factor, and
λ is the carrier wavelength.

There exist many more efficient models for large scale path loss in terms of the log-normal path loss model and a huge array of models that take into account the environs of the transmitter and receiver to the finest detail. This thesis is however concerned with small scale losses (primarily multipath propagation).

3.2 Small scale path loss

As the name suggests it denotes a set of rapid fluctuations in signal phase, amplitude or multipath delay lasting for a relatively short period of time. Small scale path loss can be classified as multipath propagation that results in time dispersion and movement of mobile users that results in a Doppler shift and hence signal fluctuation or fading. The rest of the chapter discusses the nature of these losses, mathematical representation and modeling of these losses for simulation purposes.

3.2.1 Multipath propagation and time dispersion

In a wireless communications system, when a line of sight path exists between the transmitter and receiver, there are copies of the transmitted signal received after multiple reflections with some time delay in comparison to that received along the line of sight path, which is of high power and dominates the received signal. In the absence of such a direct path the received signal is wholly composed of such reflected components with different delays. Thus it becomes very important to be able to identify and separate at least one copy of the transmitted signal and preferably exploit the availability of multiple copies to improve the efficiency of the decoder. The effects of a multipath channel may be modeled as:

$$y(t) = \sum_{k=1}^{M} \left(\sum_{i=1}^{N} h_{i;k}(t) \right) s(t - \tau_k) \tag{3.2}$$

where N denotes the number of time delayed versions received and theoretically is close to infinity, τ_k denotes the time delay of the k^{th} such version, $\tau_i \in [0, \infty]$, $h_{i;k}(t)$ denotes a random time varying modulation coefficient that is a complex number. ***Coefficients with the same k subscript denote versions the same delay spread.*** $\sum_{i=1}^{N} h_{i;k}(t)$ is usually modeled as being composed of real and imaginary parts that are independent zero mean Gaussian random variables. $s(t)$ denotes the pulse-shaped, transmitted symbols and hence can be rewritten as $s(t) = c(t) * s[n], t \in (t, t + nT)$.

As the receiver under consideration works on sampled versions of the received signal its is advisable to continue modeling using equations that operate on samples. The above equation can be represented equivalently as :

$$y[n] = \sum_{k=o}^{m-1} h_k[n] s[n - k] \tag{3.3}$$

where $h_k[n]$ has been obtained by summing over all paths of the same delay i.e.

$$h_k[n] = \sum_{i=1}^{N} h_{i;k}[n] \tag{3.4}$$

In order to explore the nature of $h_k[n]$ we need to consider *Doppler Spread* first.

3.2.2 Mobility of user and Rayleigh Fading

In the previous section we modeled $h_k[n]$ as composed of real and imaginary Gaussian r.v.s. As a result the envelope of the resultant signal has a *Rayleigh* pdf. The variations in the envelope are referred to as *Rayleigh fading.* Th inherent assumption here is the absence of a line of sight component. In the presence of such a component the Gaussian variable would no longer have zero mean and the resultant envelope would follow a *Rician* pdf.

The wide sense stationary random process $h_k[n]$ ***as such is characterized by a flat autocorrelation function.*** This behaviour of $h_k[n]$ is affected by the mobility of the user and the mathematical analysis that follows discusses this aspect.

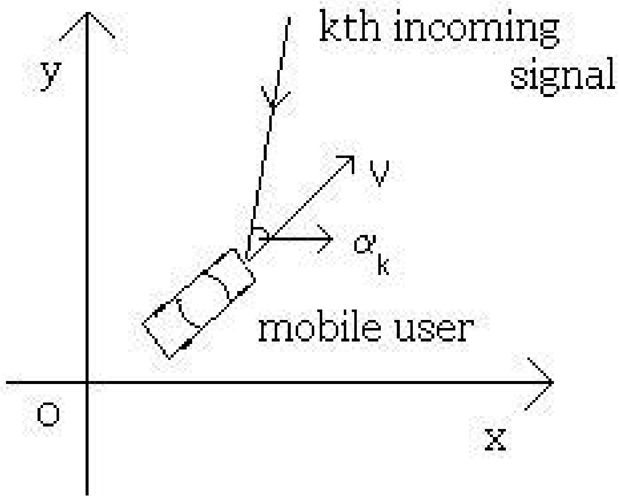

Figure 3.1: A component wave incident on mobile receiver

Consider a stationary mobile user receiving reflected versions of the transmitted signal from all directions. These signals are of random phases from 0 to 2π and there amplitudes are also statistically independent. Figure [3.1] shows a *plan* view of the situation. The motion of the user introduces a Doppler shift in each wave given by:

$$\omega_k = \beta v cos\alpha_n \tag{3.5}$$

where β= $2\pi/\lambda$, λ being the carrier frequency of the transmitted signal. Rewriting 3.4 to accommodate for the Doppler spread:

$$h_k[n] = \sum_{i=1}^{N} h_{i;k}[n]. \exp(\omega_i n T_s) \tag{3.6}$$

where T_s is the sampling frequency and all other symbols have their usual meanings.

Now, as before $|h_k[n]|$ has a Rayleigh pdf. However the autocorrelation of the process, $R_C(\tau)$ is altered to the form :

$$R_C(\tau) = AJ_0(\omega_d \tau) \tag{3.7}$$

where
A is a real positive constant that is dependent on signal power
J_0 denotes the Bessel function of the first kind and zeroth order, and
$\omega_d = \beta * v$ denotes the Doppler shift limit. The mathematics behind the above result is too detailed to be contained in this thesis.

3.3 Modeling of Small Scale Path Loss - the *Jakes'* model

Having considered the channel effects on the signal we now deal with the implementation of these effects into a channel model for simulation. One of the most effective models proposed to generate channel coefficients that possess a Rayleigh envelope while maintaining the autocorrelation function described in 3.7 is the *Jakes'* model [Jak01B]. The schematic of the model is represented in fig [3.2]

The set of N_0 oscillators depicted in the figure, produce signals of the form :

$$x_c(t) = 2\sum_{i=1}^{N} \cos\beta_i \cos\omega_i t + \sqrt{2}\cos\alpha\cos\omega_d t x_s(t) = 2\sum_{i=1}^{N} \sin\beta_i \cos\omega_i t + \sqrt{2}\sin\alpha\cos\omega_d t \tag{3.8}$$

where ω_i is an oscillator frequency equal to $\omega_d cos(2\pi n/N)$, $N = 4N_0 + 2$, α is chosen as $\pi/4$ and $\beta_i = \pi i/N_0$.

Finally the channel coefficients are obtained as :

$$h_k(t) = x_c(t) + jx_s(t); \quad for\ a\ given\ k \tag{3.9}$$

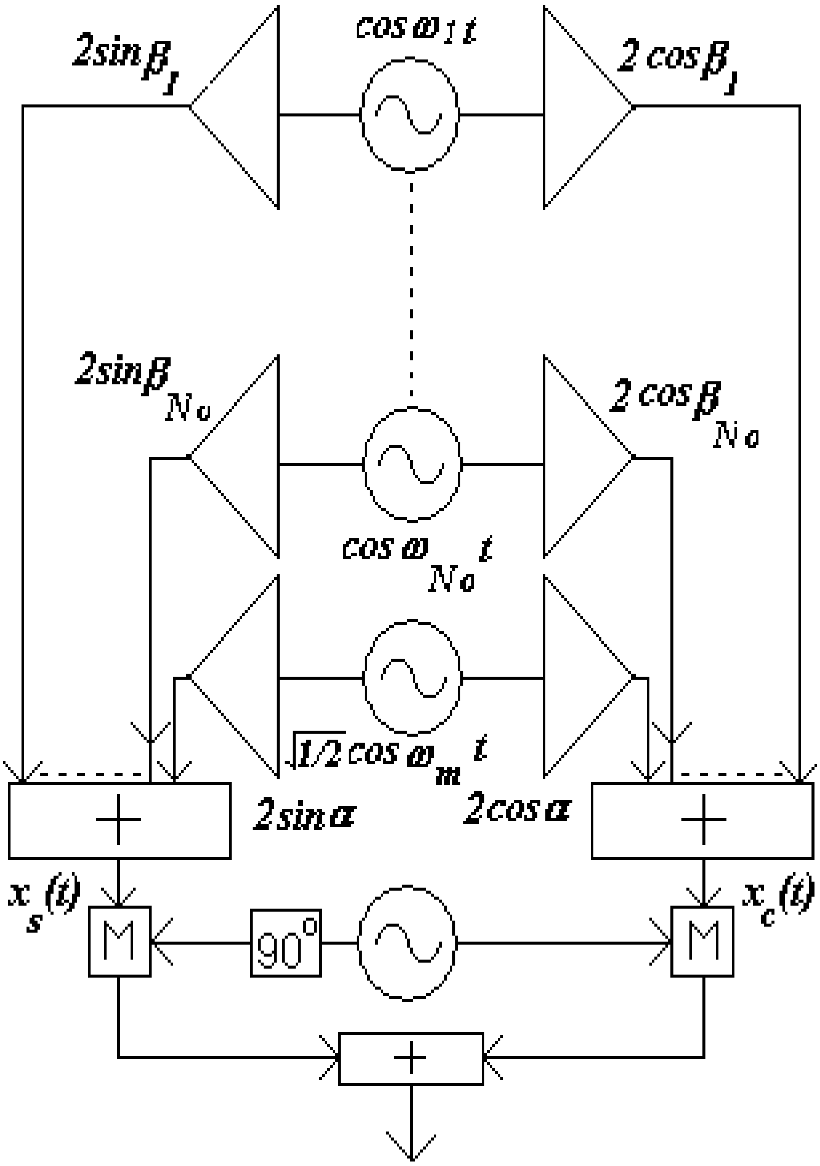

Figure 3.2: Schematic diagram showing the structure of the Jakes' fading variable generator

Uncorrelated channel coefficient sequences maybe generated by providing the generating oscillators with a random phase offset θ_i at the start of generation. The quality of generated signal is largely contingent on the choice of N_0. For greater precision a larger value of N_0 maybe used.

The Jakes' model has been used in all our simulations involving a fading channel. In the channel tracking algorithms discussed in the later chapters of this thesis, a finiter impulse response (FIR) channel model is assumed, with channel taps subjected to Rayleigh fading with identical statistical characteristics. In section [8.1.2] we present some of our simulation results that show the characteristics if our channel model.

CHAPTER 4

The Receiver

In the previous chapter we discussed in some detail, the multipath propagation effects of fading and dispersion on the received signal. In this chapter we will characterize the structure of the mobile radio receiver, designed to minimize the undesirable effects of the fading multipath channel. The block diagram below outlines the basic structure of the receiver to be discussed in this thesis.

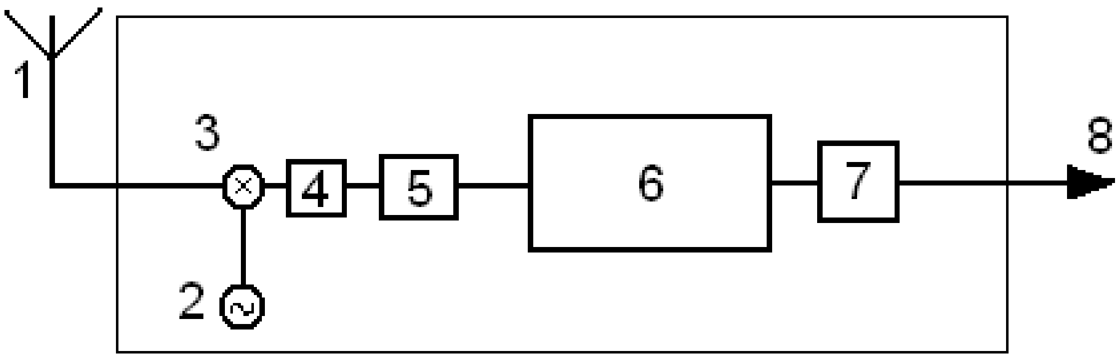

Figure 4.1: A basic wireless receiver structure. 1: Receiver antenna, 2: Local Oscillator ($f_{oscillator} = f_c$), 3: Mixer, 4: Matched SRRC Filter, 5: A/D converter, 6: Digital Decoding Block, 7: Reverse *Grey*coder, 8: Decoded bit stream

4.1 Recovery of baseband signal

The signal received at the antennae is of the form :

$$y_{antenna}(t) = y(t).exp(j2\pi f_c t); \tag{4.1}$$

where $y(t)$ is a baseband representation of the received signal, $y_{antenna}$ is a passband representation of the received signal, and f_c represents the carrier frequency in Hz. As indicated in the block diagram the $y(t)$ is extracted from $y_{antenna}(t)$ using the local oscillator followed by the receiver filter which is a low pass filter. The process has been diagrammatically represented in the frequency domain below :

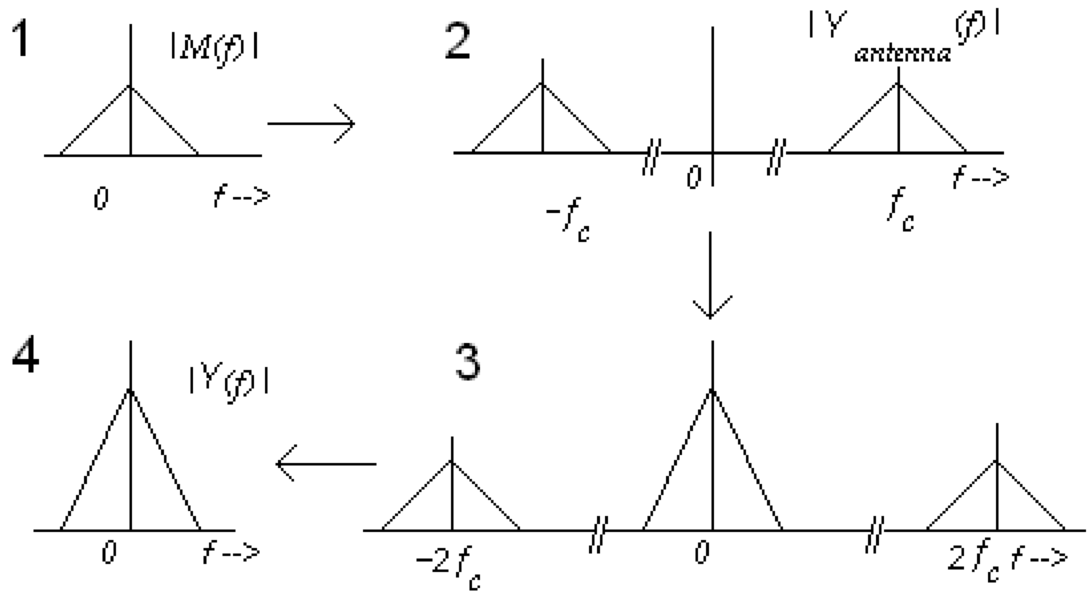

Figure 4.2: Recovery of the baseband signal: 1.At transmitter before modulation 2.Transmitted signal 3.After mixer 4.After receiver filter

The receiver filter is chosen to be a matched SRRC filter in order to maximize the SNR at the output of the filter (Ref. [Pro01B]) The output of the filter is then sampled at the pre-specified rate and fed into the digital decoder which estimates symbols from the samples. The rest of this chapter shall discuss this block as implemented in our simulations.

4.2 The Digital Decoder

Apart from channel modulated transmitted symbols, the front end of the receiver is also susceptible to thermal noise (ignoring interference for the present argument). Modeling this effect simplistically as additive white Gaussian noise (AWGN) we represent the received symbols as:

$$y(t) = \sum_{k=0}^{m-1} h_k(t)s[n-k] + v_{AWGN}(t); \qquad t \in (t - nT, nT) \tag{4.2}$$

or equivalently

$$r[n] = r_h[n] + v_{AWGN}[n]; \tag{4.3}$$

where

$$y_h[n] = \sum_{k=0}^{m-1} h_k[n]s[n-k]; \tag{4.4}$$

In the above equation:
(i) The received symbol is available to us.
(ii) As each frame is provided with a training sequence let us for the moment assume that the channel coefficients can be estimated/tracked satisfactorily over the frame.
(iii) The probability distribution of the random variable $v_{AWGN}[n]$ is available to us.
(iv) The symbols are unknown and need to be estimated from (i), (ii) and (iii) above.

We would like to estimate $R_h[n]$ using the above data.
Bayes' theorem tells us that:

$$P(y_h(k) = y_h|y(k) = y) = \frac{P(y(k) = y|y_h(k) = y_h).P(y_h(k) = y_h)}{P(y(k) = y)} \tag{4.5}$$

The premise of MLSE is that ***the transmitted symbols are those which maximize the above probability***. Simplying the above equation:

$$P(y_h(k) = y_h|y(k) = y) = \frac{1}{\pi N_0}.exp\left[-\frac{(y-y_h)'(y-y_h)}{N_0}\right]; \tag{4.6}$$

or on logarithmic simplification

$$\underbrace{P(y_h(k) = y_h|y(k) = y)}_{maximize} \equiv \underbrace{(y-y_h)'(y-y_h)}_{minimize} \tag{4.7}$$

The above can be used to make the decision on a single $y_h[n]$. This principle can be extended to an entire frame of received symbols using the ***Viterbi*** algorithm.

4.3 The Viterbi Algorithm

Consider the Maximum Likelihood Estimation of a time sequence of random variables **S**.

We require:

$$\underbrace{P(\mathbf{S} = \mathbf{s})}_{maximize} = \underbrace{P(S[1] = s[1]) \cap P(S[2] = s[2]).... \cap P(S[n] = s[n])}_{maximize} \tag{4.8}$$

Using the theorem of conditional probability:

$$\underbrace{P(\mathbf{S}=\mathbf{s})}_{maximize} =$$

$$\underbrace{P(S[1]=s[1]).P(S[2]=s[2]|S[1]=s[1])...P(S[n]=s[n]|S[1]=s[1]...S[n-1]=s[n-1])}_{maximize} \tag{4.9}$$

The above technique of decision is extremely cumbersome and not viable for large sequences. The Viterbi algorithm reinvents the problem using a *state-space representation* involving recursive stage by stage analysis. At each stage one of several states is chosen. The algorithm works on the assumption that once you are at a particular stage, the future states are *a function of only the current state and not the past states*. This sort of analysis reduces complexity while retaining optimality.

More importantly the Viterbi algorithm makes decisions based on the maximization (or minimization) of a cumulative *path - metric*. The path metric is the sum of branch metrics accumulated at each stage. Decision on the individual branch metrics is similar in nature to that represented in equation (4.7). Therefore in the simple case of channel impairment including no interference we would accumulate:

$$\underbrace{P(R_h(k)=r_h|R(k)=r)}_{maximize} \equiv \underbrace{(y-y_h)'(y-y_h)}_{minimize} \tag{4.10}$$

Further discussion on the the decoder block is carried out in [Skl01R] and [Pro01B].

The last block in the receiver structure is the reverse Gray coder which interprets the symbols estimated by the decoder block in terms of bits, based on the encoding algorithm used at the transmitter end.

CHAPTER 5

Channel Estimation and Adaptive Equalization

Consider the received sampled sequence $\{y(n)\}_{n=1}^{N}$, generated from the transmission of one data burst over an RF channel. It is represented as the output of a discrete - time complex - valued baseband channel model.

$$y(n) = h_0 d(n) + \sum_{k=1}^{m} h_k(n) d(n-k) + v(n) \quad n = 1, \dots, N \tag{5.1}$$

$$E|d(n)|^2 = \sigma_d^2 \quad E|v(n)|^2 = \sigma_v^2 \quad \rho = \frac{\sigma_v^2}{\sigma_d^2}$$

The transmitted symbol sequence $\{d(n)\}$ is white with zero mean. The symbols take a finite number of possible values which are equally likely. The measurements are corrupted by a zero - mean white noise $v(n)$. Symbols and noise are independent. The objective is to reconstruct the transmitted sequence $\{d(n)\}_{n=N_{tr}+1}^{N}$ from $\{y(n)\}_{n=1}^{N}$ and a known training sequence $\{d(n)\}_{n=1}^{N_{tr}}$.

If a transmit symbol $d(n)$ is to be reconstructed from the received sequence, the inter-symbol interference caused by the middle term of the right hand side of the equation [5.1], the *channel tail* has to be removed. If there is severe ISI, a simple symbol - by -symbol detector would lead to frequent incorrect decisions. Since ISI is caused mainly by multipath propagation, , the most straightforward way to avoid it would be to decrease the symbol transmission rate. Delayed symbols would then have time to arrive before the next symbol is received. This is however, unattractive from an efficiency point of view. A superior alternative would be to use an equalizer or a maximum likelihood sequence detector implemented via the Viterbi algorithm to compensate for the inter-symbol interference.

An equalizer is an input estimator consisting of one or several filters. The equalizer produces an estimate $\hat{d}(n)$ of the transmitted symbol. This estimate is fed into a decision device to obtain a decisioned symbol $\overline{d}(n)$. The decision device selects the

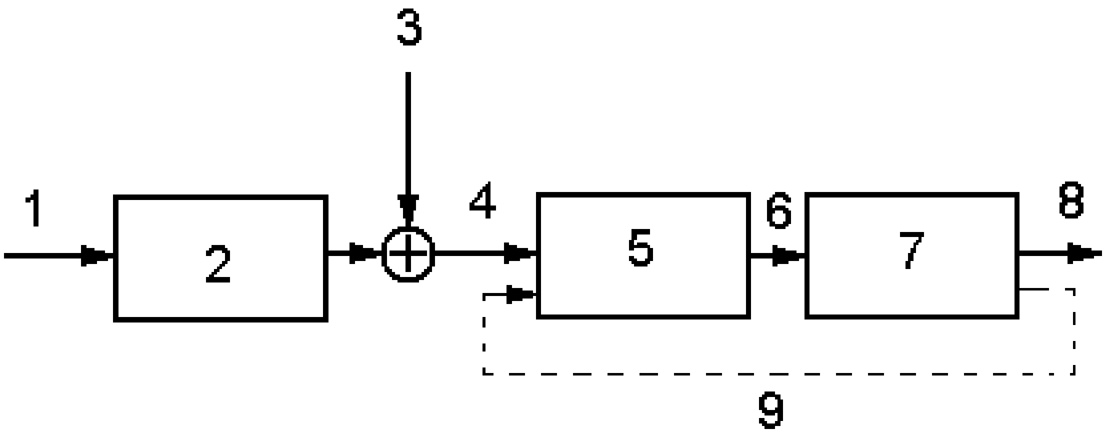

Figure 5.1: The general layout of an equalizer. 1: $d(n)$, 2: Channel, 3: $v(n)$, 4:$y(n)$, 5: Equalizer, 6: $\hat{d}(n)$, 7:Decision-device, 8: $\overline{d}(n)$, 9: Decision feedback. Signals in an equivalent baseband description. The transmitted symbols $d(n)$ propagate through a time dispersive channel, having a finite or infinite impulse response $\{h_k\}$. The received signal $y(n)$ is corrupted by noise $v(n)$. An equalizer gives an estimate $\hat{d}(n)$ and a decision device selects the symbol $\overline{d}(n)$.

symbol which is closest, in Euclidean distance, to the estimated input. In the case of complex valued signals, the complex plane is divided into decision regions.

In this thesis we assume discrete-time channel models with finite impulse response (FIR). A finite impulse response model with taps at symbol space is a fairly good model for a dispersive channel. Each tap represents a different path taken by the signal from the transmitter to the receiver. The two tap FIR channel model is a good approximation for most dispersive mobile radio channels and that other FIR models can be closely approximated by the two-tap model.

When the FIR channel cannot be modeled as a single $\delta(n)$, the channel is said to be non ideal. The job of the equalizer is in some sense to 'invert' the channel by acting as a system in cascade with the channel. The equalizer must 'learn' of the channel impulse response to do so and assumes that the channel is unchanging. However a real mobile radio channel is not static and must be modeled as a time varying FIR system. The equalizer must be supplied with the constantly updated information about the latest state of the channel impulse response. This is the job of

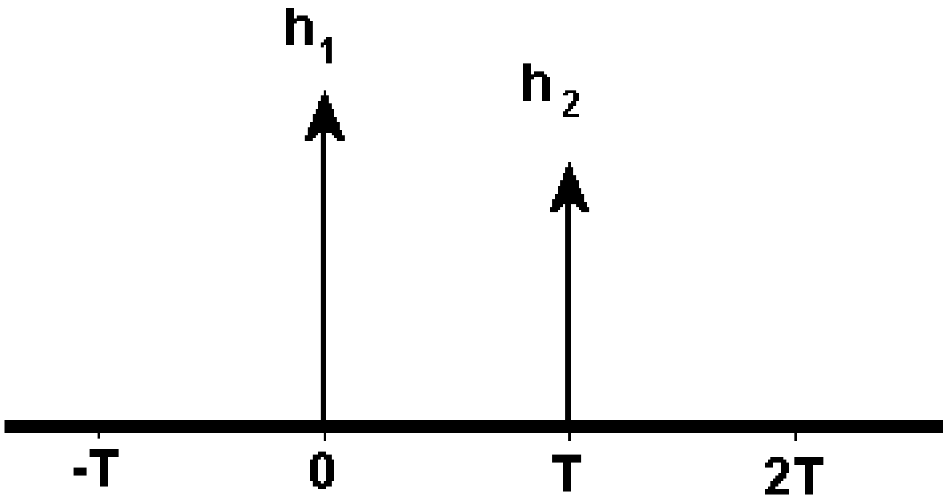

Figure 5.2: 2 - Tap channel model with time varying coefficients

a *channel tracking* algorithm or a *channel estimator* and the entire system constitutes an *adaptive equalizer*. In the sections that follow we describe certain types of equalizer structures and channel estimation techniques relevant to this thesis.

5.1 Adaptive equalizer structures

In this section we introduce and describe two approaches to adaptive equalization, direct and indirect. Further we will also show why the indirect method of adaptive equalization is a better and preferred method.

Messages are assumed to be transmitted in bursts, blocks of N symbols, where the first N_{tr} ($N_{tr} < N$) symbols are known to the receiver and constitute the /emph-training sequence. The adaptation procedure is thus divided into two modes:

1. Learning - directed mode, $\{y(n)\}_{n=1}^{N_{tr}}$; utilizes the training sequence to estimate the (time varying) parameters. Provides an initialization to the next mode.
2. Decision - directed mode, $\{y(n)\}_{n=N_{tr}+1}^{N}$; Decisioned symbols are used to update the time - varying parameters. In the decision - directed mode, $d(n)$ is replaced by $\tilde{d}(n)$.

The general (overall) structure of the adaptive equalizer / detection problem is depicted in figure [5.3]. It consists of two parts:

1. Estimation of a channel or equalizer parameter vector θ using a recursive algorithm. Inputs to the algorithm are the measurements $y(n)$ and either known symbols $\{d(n)\}_{n=1}^{N_{tr}}$ or decisioned symbols $\{\tilde{d}_n\}_{n=N_{tr}+1}^{N}$.
2. Estimation of symbols, using an equalizer or Viterbi detector.

As stated above, estimation of the parameter vector has to be in some way based on decisioned symbols $\tilde{d}(n)$. With incorrect decisions, there will be a potential risk ok loosing the ability to track the parameters, in particular if several consecutive erroneous decisions are made.

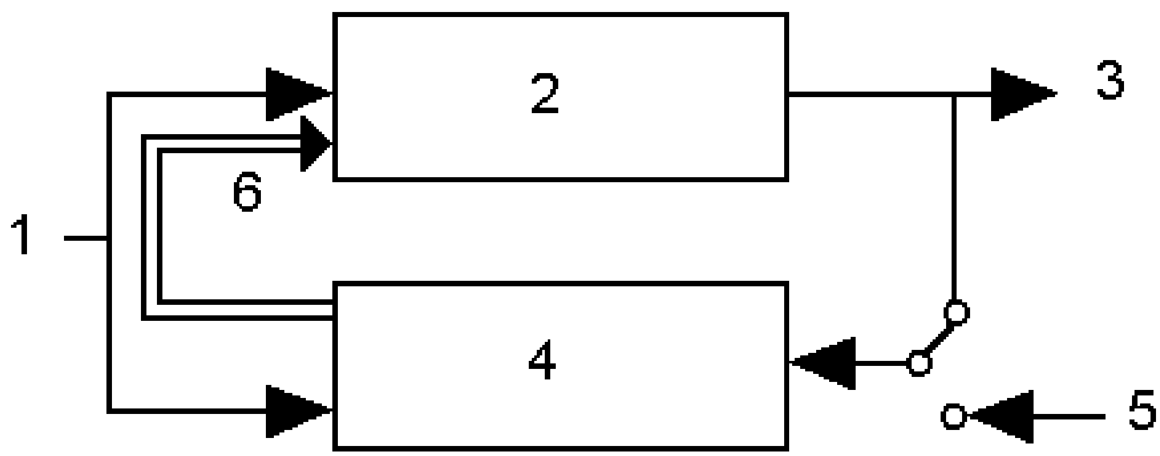

Figure 5.3: The overall structure of the adaptive equalizer / detection. 1: $y(n)$, 2: Estimation of symbols, 3: $\overline{d}(n)$, 4: Estimation of parameters, 5: Training sequence, 6: Estimated parameters θ

5.1.1 Direct adaptation of equalizer parameters

The traditional and the most straightforward way of adapting the co-efficients in an equalizer, is to update the co-efficients, $\theta_{eq}(n)$ *directly* from data. The adjustment of the co-efficients is based on recursive minimization of a function of the input prediction error. In training mode, this error is

$$\epsilon_d(n) = d(n)\hat{d}(n) \tag{5.2}$$

while in decision directed mode, decisioned symbols are used as substitutes for the unknown transmitted symbols so that $\overline{\epsilon}_d(n) = \overline{d}(n) - \hat{d}(n)$. Thee approach is illustrated in fig. [5.4]. At each time instant, the equalizer parameters are updated by a recursive algorithm. Inputs to the algorithm are the regression vector $\mathbf{\Phi}_{eq}(n)$ and the prediction error $\epsilon_d(n)$, or $\overline{\epsilon}_d(n)$. The output is $\widehat{\theta}_{eq}(n)$

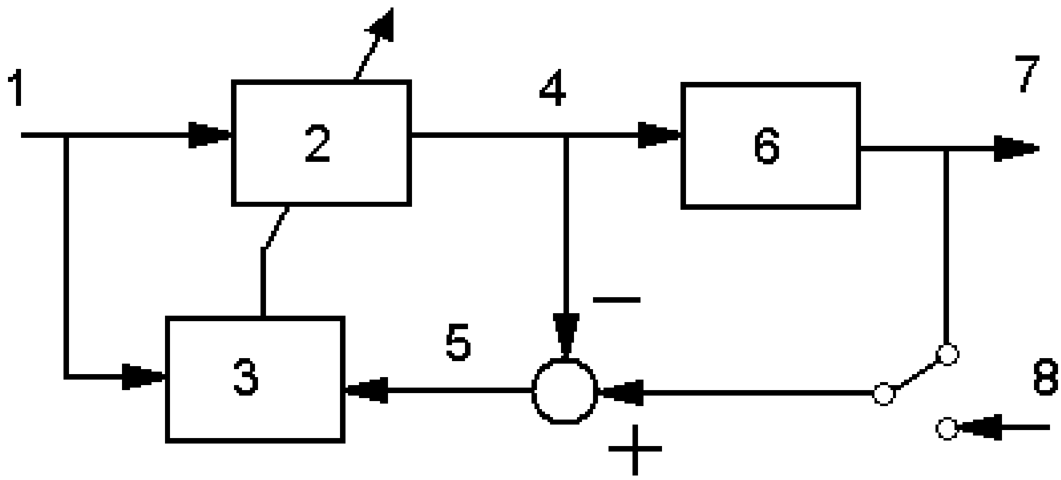

Figure 5.4: Signal flow of direct adaptation of equalizer parameters in decision directed mode. 1: $\mathbf{\Phi}_{eq}^H(n)$, 2: $\theta_{eq}(n)$, 3: Recursive algorithm, 4: $\hat{d}(n)$, 5: $\overline{\epsilon}_d(n)$, 6:Decision device, 7: $\overline{d}(n)$, 8: Training sequence. The regressor $\mathbf{\Phi}_{eq}^h(n)$ and the equalizer parameters, updated by a recursive algorithm, produce an estimate of the input $\hat{d}(n)$. This estimate is fed into a decision device, and a new prediction error $\overline{\epsilon}_d(n)$ can be formed. The equalizer parameters are then updated based on $\mathbf{\Phi}_{eq}^h(n)$ and $\overline{\epsilon}_d(n)$.

5.1.2 Indirect adaptation of the equalizer parameters

Indirect adaptation is based on recursive estimation of the *channel* coefficient vector

$$\theta^T(n) = (h_0(n) \dots h_m(n)) \tag{5.3}$$

Using the updated channel parameters θ, the equalizer parameters are re-adjusted periodically, possibly at every sample. Thus we perform a mapping $\widehat{\theta}_{eq}(n) = f(\widehat{\theta}(n))$. Channel parameter estimation is based on recursive minimization of a function of the *output* prediction error

$$\epsilon(n) = y(n) - \hat{y}(n) \tag{5.4}$$

The output prediction $\hat{y}$ is simply the estimate of the noise-free received signal

$$\hat{y}(n) = \mathbf{\Phi}^H(n)\theta(n) \tag{5.5}$$

where $\mathbf{\Phi}^H(n)$ is the channel regression vector. In decision directed mode, we are forced to use decisioned symbols. In an adaptive equalizer, we have to substitute channel co-efficient estimates, $\hat{\theta}(n)$ for the true channel co-efficients for the calculations. The indirect approach to equalization is illustrated in fig [5.5]. At each time instant, the channel model coefficients and the equalizer parameters are updated via an estimate $\hat{\theta}(n)$ produced by a channel estimator. Inputs to the channel estimator are the regressor $\mathbf{\Phi}(n)$, or $\overline{\mathbf{\Phi}}(n)$, and the output prediction error $\epsilon(n) = y(n) - \hat{y}(n)$ respectively. The output is $\hat{\theta}(n)$.

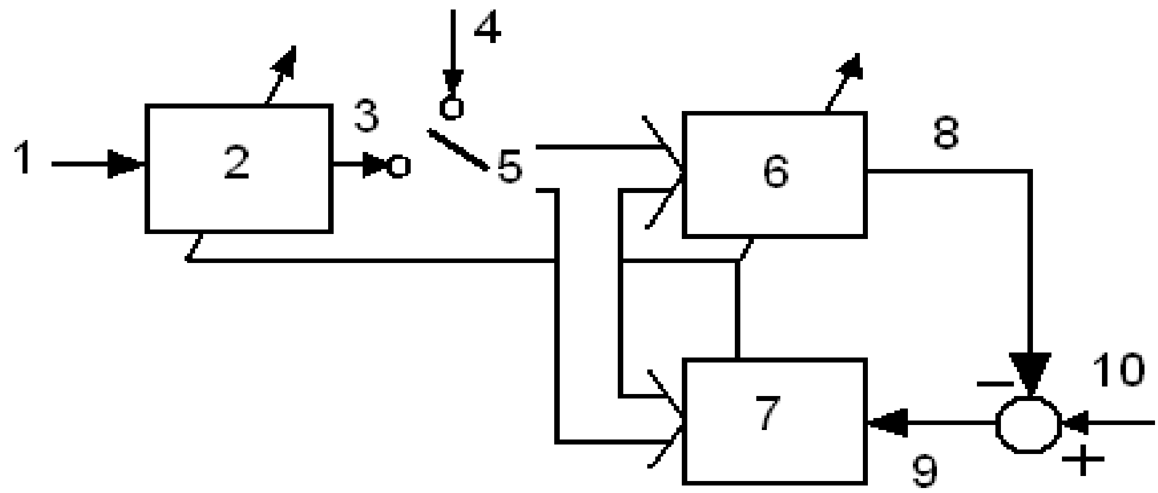

Figure 5.5: The structure of an indirect adaptive equalizer. 1: $\mathbf{\Phi}^H_{eq}(n)$, 2: $\theta_{eq}(n) = f(\theta)$, 3: $\overline{d}(n)$, 4: Training sequence, 5: $\mathbf{\Phi}^H(n)$, 6: $\theta(n)$, 7: Channel estimator, 8: $\hat{y}(n)$, 9: prediction error $\epsilon(n)$, 10: yn. The channel co-efficients are updated, based on $\mathbf{\Phi}^H(n)$ and $\epsilon(n)$, and used to update the equalizer parameters. In decision directed mode, decisioned symbols are used in $\mathbf{\Phi}^H(n)$

5.1.3 Why is indirect adaptation preferred

From the discussion above, it follows that the most straightforward approach to adaptive equalization is to use the direct form. However this approach leads to a very difficult tracking problem. A closer examination of the MSE - optimal equalizer parameters, indicates a non-linear relationship between the parameters and the channel co-efficients. This follows from the fact that an equalizer attains channel inversion in some way. The typical relation is $1/h_0$, where h_0 is the direct term in the channel impulse response. In a non-stationary environment, an MSE - optimal equalizer has to track a moving minimum point. A problem arises when $|h_0|(n)$ passes zero. Then large and rapid changes in the optimal equalizer parameters occur. These parameter jumps are the reason why direct adaptation of equalizer parameters will fail in the case of fast Rayleigh fading channels. This problem is avoided if the equalizer parameters are updated via channel estimates. The channel co-efficients change rapidly but without large jumps.

5.2 Per-survior Viterbi detection

In this section we consider implementation aspects of an adaptive Viterbi algorithm. Both indirect adaptation of equalizer parameters and adaptive Viterbi schemes are based on estimation of a channel model.

The basic theory behind the Viterbi algorithm was explained in section [4.3]. We recall that the Viterbi algorithm is designed to base its decision upon the *entire* sequence instead of making decisions on a symbol-by-symbol basis. We also recall from above that a channel estimator requires decisioned symbols, *at each time instant*, in decision-directed mode. At first glance, there appears to be a conflict between sequence estimation and simultaneous estimation of channel co-efficients. However there is a way to utilize the Viterbi algorithm in conjunction with a channel estimator in decision directed mode: The **Per survivor** implementation of the Viterbi algorithm solves this problem by updating channel co-efficients of the assumed channel model (using the the particular channel tracking law being used) along each of the survivor trellis paths. This means that every stage of the MLSE Viterbi detector through the trellis (assume that the trellis is moving forward from stage n to stage $n+1$) involves the following steps:

1. Calculation of all the branch metrics and resulting path metrics for each of the possible symbols in stage $n+1$.
2. Updation of the channnel co-efficients long the chosen optimal branch for each of the possible symbols in the stage $n+1$

Hence each stage of the Viterbi trellis has, associated with it, a vector of channel co-efficients. Below we describe the basic step involved in moving through the Viterbi trellis:

Let the current stage of the Viterbi trellis be labeled n. We are to propagate survivor paths to stage $n+1$. Let M be the size of the symbol alphabet (and therefore also the number of possible symbols in each stage of the Viterbi Trellis). Let the fading dispersive channel be modeled by a '$m+1$'- ray symbol spaced discrete time FIR model. Thus a complete channel estimate consists of the vector $\mathbf{\Phi}^H(n) = (h_0(n) \ldots h_m(n))$ (For most dispersive channels m = 2 is a fairly good approximation. In our simulation study we have used m = 1). In other words:

$$y(n) = h_0(n)d(n) + h_1(n)d(n-1) + v(n) \tag{5.6}$$

where the taps $h_0(n)$ and $h_1(n)$ are subject to independent Rayleigh fading and $v(n)$ is sampled AWGN. Let the vector $P_n = (p_0(n) \ldots p_{M-1}(n))$ represent the cumulative path metrics at the nth stage of the Viterbi trellis. Now each of the possible symbols $(d_0(n+1) \ldots d_{M-1}(n+1))$ in stage $n+1$ of the trellis must link up to one symbol in the stage n of the trellis. Consider the 'linking' of the symbol $d_j(n+1)$ to a symbol in the nth stage. The choice of which symbol to link to is made through minimization of the metric (for a choice of m = 2)

$$Min_{i=0}^{M-1} \left\{ |r(n+1) - \mathbf{\Phi}^H(n)_i.\theta_i(n+1)|^2 + p_i(n) \right\} \tag{5.7}$$

where $r(n+1)$ is the received value as output from the receiver filter, $\mathbf{\Phi}^H(n)_i$ represents the channel co-efficents vector stored in memory for the ith symbol in the nth stage of the Viterbi trellis and $\theta_i(n+1)$ is the vector containing the sequence of data symbols $(d_j(n+1), d_i(n+1))$ for the linking of the symbol $d_j(n+1)$ in the $(n+1)$th stage and for a hypothesis i in the nth stage. Notice that in the calculation of the metric for a hypothesis involving the ith symbol in stage n and jth symbol in stage $n+1$, the channel-coefficient vector *for the ith symbol* is used. Having chosen say $i = k$ from the the above minimization, the channel co-efficients stored for the kth state in the nth stage of the trellis must now be updates according to a pre-decided channel tracking law (We have used the NLMS and KLMS algorithms in our simulation study).

$$\mathbf{\Phi}_j^H(n+1) = f(\mathbf{\Phi}_i^H(n)) \tag{5.8}$$

where f is the function that updates the channel co-efficient vector, given as input, the old channel co-efficient vector and the forward prediction error

$$\epsilon(n) = r(n) - (d(n) \ldots, d(n-m)). \begin{pmatrix} \hat{h}_0(n) \\ \vdots \\ \hat{h}_m(n) \end{pmatrix} \tag{5.9}$$

Hence at any stage of the per - survivor Viterbi scheme, apart from cumulative path metrics, channel co-efficients corresponding to each possible symbol in that stage must also be stored in memory. However, this information need not be maintained in memory for the entire trellis as it is required only for the previous stage when moving forward in the trellis. Thus the per survivor implementation of the Viterbi algorithm is able to perform maximum likelihood sequence estimation as before but also provides a method of updation of channel co-efficients by the channel tracking algorithm that is working in conjunction.

5.3 Constant gain channel tracking using stochastic hypermodeling

Parameter tracking in severe Rayleigh fading environments (high Doppler shifts and low signalling rates) with RLS / LMS algorithms, often requires large adaptation gains, i.e. a short memory. Such algorithms become sensitive to noise and to incorrect decisions. By utilizing *a priori* information about the time-varying channel co-efficients, adaptive algorithms with larger memories can be designed without sacrificing tracking capability. It is evident that in fading environments, the co-efficients of an FIR channel model exhibit typical trend or quasi periodic behaviour.

The LMS and RLS algorithms (for a description of the LMS algorithm, see Appendix A) are perhaps the most obvious first choices as channel tracking algorithms for use in conjunction with an adaptive equalizer or a Viterbi detector. However, while they are low in complexity and simple to implement, the optimal design in both these algorithms presumes a random walk behaviour for the time - varying channel parameters. The channel tracking algorithms proposed in [Lin01] and studied in our simulation study utilize *a priori* information about the channel through simplified stochastic modeling. Here, stochastic modeling leads naturally to Kalman filtering. However the optimal Kalman filter design applied to our channel tracking algorithm, (Ref. Appendix A) being a time-variant gain algorithm requires constant update of the adaptive gain and therefore has a very high computational complexity. Models intended to describe the behaviour of the true time - varying co-efficients are sometimes called 'hypermodels'.

When transmit symbols are white and have constant modulus (as in our case study), a channel estimator with high performance and low complexity can be obtained using simplified stochastic internal modeling of the co-efficients of an FIR channel model and approximation of a Kalman predictor. [Lin01] It is related to the design of algorithms based on hypermodels and is less *ad hoc* than the use of co-efficient prediction filters combined with the LMS / RLS algorithms.

Consider the discrete time FIR channel model discussed earlier, now expressed as:

$$y(n) = \mathbf{\Phi}^H(n)\theta(n) + v(n) \quad n = 1, \ldots, N \tag{5.10}$$

$$\mathbf{\Phi}^H(n) = (d(n) \ldots d(n-m))$$

$$\theta^T(n) = (h_0(n) \ldots h_m(n))$$

$$E|d(n)|^2 = \sigma_d^2 \qquad E|v(n)|^2 = \sigma_v^2$$

When $v(n)$ is white noise, the predictor of the received signal $y(n)$ is given by $\hat{y}(n) = \mathbf{\Phi}^H(n)\theta(n)$.Then the output prediction error (or the *forward* prediction error) is $\epsilon(n) = y(n) - \hat{y}(n)$. The goal of the time-varying parameter estimation problem is to determine a sequence of estimates $\{\hat{\theta}(n)\}_1^N\}$ such that the cumulative squared prediction errors are minimized.

$$\{\hat{\theta}(n)\} = arg\left[Min_{\theta_n}\left(\sum_{n=1}^{N}|\epsilon(n)|^2\right)\right] \tag{5.11}$$

In order to minimize this criterion, parameters of time-varying systems are often estimated by recursive algorithms, having the structure

$$\hat{\theta}(n) = \hat{\theta}(n-1) + K(n)\epsilon(n) \tag{5.12}$$

$$\epsilon(n) = y(n) - \mathbf{\Phi}^H(n)\hat{\theta}(n-1) \tag{5.13}$$

The choice of the gain vector sequence differs in different algorithms. A common feature of all recursive algorithms is that some design variable is to be tuned. For the particular case of a linear regression model, we choose to express the gain as

$$K(n) = P(n)\mathbf{\Phi}(n) \tag{5.14}$$

where $P(n)$ is a positive definite $m+1|m+1$ matrix. Note that the parameter update is made in the direction of $P(n)\mathbf{\Phi}(n)$ The design variable is then $P(n)$. In the following chapter we will show how the time - varying adaptive gain tracking formulated here can be converted to a channel tracking algorithm with a constant gain using *a priori* information about the channel (discussed later in this section).

5.3.1 Stochastic modeling of the channel coefficients

In the case of multipath fading channels, prior knowledge about the time-variation of the channel exists.

- The channel co-efficients $h_k(n)$ are such that $|h_k(n)|$, $k = 0, \ldots, m$ are Rayleigh distributed.
- $\mathrm{Re}\{h_k(n)\}$ and $\mathrm{Im}\{h_k(n)\}$are mutually independent Gaussian processes with a normalized autocovariance function given by

$$R(l) = J_0(w_m l) \quad l = 0, \pm 1, \ldots \tag{5.15}$$

$$w_m = 2\pi \frac{f_m}{f_s}$$

  Here $J_0(.)$ is the Bessel function of the first kind and of order zero, f_m is the maximum Doppler frequency and f_s is the sampling rate. The maximum Doppler frequency is determined by the speed of the vehicle, as discussed in section [3.2.2]. This autocovariance tells us that the variation of the co-efficients is similar to that of narrow band Gaussian noise. The co-efficients can be generated by filtering white complex-valued Gaussian noise with a filter of high order (However, in our case study we have used the Jakes' Rayleigh fading simulator).
- The channel co-efficients change independently thus

$$E[h_j(n)h_k(n)]_{j\neq k} = E[h_j(n)]E[h_k(n)] \tag{5.16}$$

In carrying out stochastic modeling of the channel co-efficients, we will consider linear stochastic models of the following type:

$$h_k(n) = \frac{C(q^{-1})}{A(q^{-1})} e_k(n) \quad k = 0, \ldots, m \tag{5.17}$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_{n_a-1} q^{-n_a+1}$$

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + c_{n_a} q^{-n_a}$$

where $C(q^{-1})$ and $A(q^{-1})$ are polynomials with real valued coe-efficients in the backward shift operator q^{-1}. The sequence $\{e_k(n)\}$ is white complex valued Gaussian noise with zero-mean. The sequances $\{e_0(n)\}, \ldots, \{e_m(n)\}$are assumed mutually independent. Thus we *model* the channel co-efficients as being generated by independent white noise sources filtered through a bank of identical real-valued linear filters. The white Gaussian noise sequences $\{e_k(n)\}$ have independent real and imaginary parts. Linearity gives

$$h_k^{(r)}(n) = \frac{C(q^{-1})}{A(q^{-1})} Re\{e_k(n)\} \quad h_k^{(i)}(n) = \frac{C(q^{-1})}{A(q^{-1})} Im\{e_k(n)\}$$

Hence the real and imaginary parts of each channel co-efficient are uncorrelated. This corresponds to a modeling assumption of channel co-efficients subjected to Rayleigh fading. (Appendix B contains details of the various stochastic models that can be practically applied to model the variation of channel co-efficients under different conditions. One of these models, the Autoregressive 2nd order model is described in detail in the following chapter)

CHAPTER 6

Simplified Kalman Filtering Algorithm applied to Channel Tracking

The Simplified Kalman filtering approach to channel tracking is based on stochastic modeling of the time-varying channel coefficients using *a priori* information. In the case of a Rayleigh fading environment, this a-priori information is in the form of the statistical properties of the variation of the channel coefficients, for eg. A U-shaped power spectral density that peaks at the maximum Doppler frequency. Hence we know *a priori* that we can model these channel coefficients by passing a complex valued Gaussian noise-like sequence through a filter that has a transfer function similar to the U-shaped PSD of the Jakes Spectrum.

The motivation behind simplifying the optimal Kalman estimator is that:

1. The computational complexity of the optimal Kalman estimator is very high.
2. It requires precise estimation of the maximum Doppler frequency as its working depends crucially on the maximum Doppler Frequency.

6.1 The optimal Kalman estimator applied to Channel Tracking

The optimal channel estimator for the channel tracking problem, defined and discussed in the previous chapter, is the Kalman estimator, if the channel co-efficients are modelled by linear stochastic models of the type:

$$h_k(n) = \frac{C(q^{-1})}{A(q^{-1})}.e_k(n) \quad k = 0, 1, \dots m \tag{6.1}$$

$$C(q^{-1}) = 1 + c_1 q^{-1} \dots + c_{n_a-1} q^{-n_a+1}$$

$$A(q^{-1}) = 1 + a_1 q^{-1} \dots + a_{n_a} q^{-n_a}$$

where $C(q^{-1})$ and $A(q^{-1})$ are polynomials with real-valued co-efficients in the backward shift operator. The sequence $\{e_k(n)\}$ is white complex-valued Gaussian noise with zero-mean. The sequences $\{e_0(n)\}, \ldots, \{e_m(n)\}$ are assumed to be mutually independent. Thus we model the channel co-efficients as being generated by *independent* white Gaussian sources filtered through a bank of identical real-valued linear filters. We define R_e as the covariance matrix of the noise vector $e^T(n) = (e_0(n) \ldots e_m(n))$:

$$\mathbf{R}_e(n) = E[e(n)e^H] = diag(\sigma_{e_0}^2 \ldots \sigma_{e_0}^2)$$

Re-writing our stochastic model to emphasize the fact that it is dependent on the instantaneous Doppler frequency and that we are modeling the optimal estimator,

$$h_k^o(n) = \frac{C(\omega_m, q^{-1})}{A(\omega_m, q^{-1})}.e_k(n) \quad k = 0, 1, \ldots, m \tag{6.2}$$

where $A(\omega_m, q^{-1})$ has degree n_a^0 and $C(\omega_m, q^{-1})$ has degree less than n_a^0. The argument ω_m emphasizes the maximum Doppler frequency dependence of the filter co-efficients and the superscript "o" denotes the optimal model. In canonical state-space form (in the canonical representation we can see the recursive relationship between the channel co-efficients that we are modeling and the parameters of the stochastic model we are making use of), we may express the above system as:

$$x_k(n+1) = \mathbf{F}(\omega_m)x_k(n) + \mathbf{G}(\omega_m)e_k(n)$$

$$h_k^o(n) = \mathbf{H}x_k(n) \quad k = 0, 1, \ldots, m. \tag{6.3}$$

with state vectors $x_k(n)$. The matrices $\mathbf{F}(\omega_m)$, the vectors $\mathbf{G}(\omega_m)$ and $\mathbf{H}$ are given by

$$\mathbf{F}(\omega_\mathbf{m}) = \begin{pmatrix} -a_1(\omega_m) & 1 & \ldots & 0 \\ \vdots & 0 & \ddots & \\ \vdots & \vdots & \ddots & 1 \\ -a_{n_a^o}(\omega_m) & 0 & \ldots & 0 \end{pmatrix} \quad \mathbf{G}(\omega_\mathbf{m}) = \begin{pmatrix} 1 \\ c_1(\omega_m) \\ \vdots \\ c_{n_a^0-1}(\omega_m) \end{pmatrix}$$

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & \ldots & 0 \end{pmatrix} \tag{6.4}$$

Introduce the total state vector $X(n) = (x_0^T(n) \ldots x_m^T(n))$, of dimension $(m+1)n_0^a|1$. This gives the following state space description of the channel co-efficient vector:

$$\mathbf{X}(n+1) = \mathcal{F}(\omega_m)\mathbf{X}(n) + \mathcal{G}(\omega_m)e(n)$$

$$\theta^{(o)}(n) = \mathcal{H}\mathbf{X}(n) \tag{6.5}$$

where

$$\mathcal{F} = \begin{pmatrix} \mathbf{F}(\omega_m) & & 0 \\ & \ddots & \\ 0 & & \mathbf{F}(\omega_m) \end{pmatrix}$$

$$\mathcal{G} = \begin{pmatrix} \mathbf{G}(\omega_m) & & 0 \\ & \ddots & \\ 0 & & \mathbf{G}(\omega_m) \end{pmatrix}$$

$$\theta^{(0)}(n) = \begin{pmatrix} h_0^o(n) \\ \vdots \\ h_m^0 \end{pmatrix}$$

$$\mathcal{H} = \begin{pmatrix} \mathbf{H} & & 0 \\ & \ddots & \\ 0 & & \mathbf{H} \end{pmatrix}$$

A state-space desciption of the time-varying model can now be represented as

$$\mathbf{X}(n+1) = \mathcal{F}(\omega_m)\mathbf{X}(n) + \mathcal{G}(\omega_m)e(n)$$

$$y(n) = \mathbf{\Phi}^H(n)\mathbf{X}(n) + v(n) \tag{6.6}$$

where $\mathbf{\Phi}^H(n) = (d(n)\mathbf{H}, \ldots, d(n-m)\mathbf{H})$.

Assume that the maximum Doppler frequency ω_n, the vector $\mathbf{\Phi}^H(n)$ and the covariance matrix $\mathbf{R}_e$(n) are all known. Then the most accurate linear estimator of $\mathbf{X}(n)$ and thus of θ^o(n) based on data up to time $n-1$ is given by the Kalman predictor:

$$\widehat{\mathbf{X}}(n+1) = \mathcal{F}(\omega_m)\widehat{\mathbf{X}}(n) + \mathbf{K}(n)\epsilon(n) \tag{6.7}$$

$$\epsilon(n) = y(n) - \mathbf{\Phi}^H(n)\widehat{\mathbf{X}}(n) \tag{6.8}$$

$$\widehat{\theta}(n) = \mathcal{H}\widehat{\mathbf{X}}(n) \tag{6.9}$$

$$\mathbf{K}(n) = \frac{\mathcal{F}(\omega_n)\mathbf{P}(n)\mathbf{\Phi}(n)}{\sigma_v^2 + \mathbf{\Phi}^H(n)\mathbf{P}(n)\mathbf{\Phi}(n)} \tag{6.10}$$

Here $\epsilon(n)$ is the prediction error and $\mathbf{K}(n)$ is the Kalman Gain. This gain depends on the estimate of the state covariance matrix $\mathbf{P}(n)$, which is updated every symbol by the Riccati Difference Equation:

$$\mathbf{P}(n+1) = \mathcal{F}(\omega_n)\{\mathbf{P}(n) - \mathbf{P}(n)\mathbf{Q}(n)\mathbf{P}(n)\}\mathcal{F}^T(\omega_n) + \mathbf{R}_1(n) \tag{6.11}$$

where:

$$\mathbf{Q}(n) = \frac{\mathbf{\Phi}(n)\mathbf{\Phi}^H(n)}{\sigma_v^2 + \mathbf{\Phi}^H(n)\mathbf{P}(n)\mathbf{\Phi}(n)} \tag{6.12}$$

$$\mathbf{R}_1(n) = E[\mathcal{G}(\omega_m)e(n)e^H(n)\mathcal{G}^T(n)] = \mathcal{G}(\omega_m)\mathbf{R}_e(n)\mathcal{G}^T(\omega_m) \tag{6.13}$$

Here $\mathbf{\Phi}(n)$ consists of the training sequence symbols in the training mode and consists of the decisioned symbols in the decision directed mode. In order to reduce the effect of wrong decision and reduce the risk of the tracking going off-track completely (which is especially likely to happen when the tracking algorithm just enters the decision directed mode from the training mode because of the especially high adaptation gain at that time) a limiter function is used to operate on the prediction error $\epsilon(n)$ before it is multiplied by the Kalman Gain. It is understood that the data vector $\theta(n)$ consists of training sequence symbols during the period that the algorithm is working in the training mode and it represents decisioned symbols in the period when the algorithm is in decision directed mode. It is also understood that the Kalman estimator would be working in conjunction with a symbol detector (in our study, it is the MLSE Viterbi detector applied to TETRA data frames). Note that incorrect decisioned symbols may throw the tracking off track since the Riccati update requires the decisioned symbols $\mathbf{\Phi}(n)$. Hence we have no proof that the optimal Kalman Channel Tracker would be optimal in the BER sense. From a practical point of view, the optimal Kalman estimator is not of much use because of two reasons:

1. The Algorithm requires very precise estimation of the maximum Doppler frequency.
2. The computational complexity of the algorithm is very high owing to the high computational load of the online update of the $\mathbf{P}$(n) matrix in the Riccati Equation.

An ideal estimator would be one that is able to provide a performance that approaches the performance of the optimal Kalman estimator while working at a much lower computational load. Such an algorithm can be derived. It is called the Kalman Least Mean Squares (KLMS) algorithm and can be used to track channel variations caused by fast Rayleigh fading. The KLMS algorithm provides a performance almost equal to the optimal Kalman estimator and works at LMS computational load. Its backbone lies in the fact that the online update of the $\mathbf{P}(n)$ matrix through the Riccati equation is eliminated by use of a constant averaged $\mathbf{P}$ matrix that provides acceptable results in the overall working of the algorithm. The derivation of the KLMS algorithm from the optimal Kalman estimator is explained in the following section.

6.2 Simplifications made to derive the simplified channel tracker: The KLMS algorithm

We have seen that the optimal version of the Kalman estimator is complex and computationally intensive from an implementation point of view. A simplified version of the Kalman estimator is derived based on the fact that we can use an averaged time-invariant $\mathbf{P}$ matrix in place of a $\mathbf{P}(n)$ matrix that must be updated every symbol using the Riccati Equation. This way the dominant computational load of the Kalman estimator is avoided. Further simplifications are made based on certain assumptions that must hold, finally leading to a channel tracking algorithm that can provide very good performance at almost LMS computational load. A rather long series of approximation are made to the Optimal Kalman estimator in a step-by-step manner to derive the form for the practically implementable (sub-optimal) version of the estimator. This series of steps is described below:

Step 1

As a first step the matrix $\mathbf{F}(\omega_m)$ and $\mathbf{G}(\omega_m)$ are replaced by other matrices of smaller order and these new matrices are now treated as design parameters rather than functions of the instantaneous maximum Doppler frequency $\omega_m(n)$. We assume that the channel co-efficients can be reasonably well described by:

$$x_k(n+1) = \mathbf{F}_d x_k(n) + \mathbf{G}_d e_k(n)$$

$$h_k(n) = \mathbf{H} x_k(n) \tag{6.14}$$

where $\mathbf{F}_d$ is $n_a|n_a$ and $\mathbf{G}_d$ is $n_a|1$. The matrices $\mathbf{F}_d$ and $\mathbf{G}_d$ are pre-determined and are of lower order than $\mathbf{F}(\omega_m)$ and $\mathbf{G}(\omega_m)$. In particular, the Riccati equation is now given by:

$$\mathbf{Q}(n) = \frac{\mathbf{\Phi}(n)\mathbf{\Phi}^H(n)}{\sigma_v^2 + \mathbf{\Phi}^H(n)\mathbf{P}(n)\mathbf{\Phi}(n)} \quad \mathbf{R}_1 = \mathbf{G}_d\mathbf{R}_e\mathbf{G}_d^T$$

$$\mathbf{P}(n+1) = \mathcal{F}_d\mathbf{P}(n) - \mathbf{P}(n)\mathbf{Q}(n)\mathbf{P}(n)\mathcal{F}_d^T + \mathbf{R}_1 \tag{6.15}$$

where we have assumed $E|v(n)|^2 = \sigma_v^2$ and $E|e(n)e^H(n)| = \mathbf{R}_e$ to be constant. This is a reasonable assumption over one data burst. We can now express the state estimate as:

$$\widehat{\mathbf{X}}(n+1) = \mathcal{F}_d\widehat{\mathbf{X}}(n) + \mathbf{K}(n)\epsilon(n) \tag{6.16}$$

$$\widetilde{\mathbf{X}}(n) = \frac{\mathbf{P}(n)\mathbf{\Phi}(n)\epsilon(n)}{\sigma_v^2 + \mathbf{\Phi}^H(n)\mathbf{P}(n)\mathbf{\Phi}(n)} \tag{6.17}$$

$$\widehat{\mathbf{X}}(n+1) = \mathcal{F}_d(\widehat{\mathbf{X}}(n) + \widetilde{\mathbf{X}}(n)) \tag{6.18}$$

where $\widetilde{\mathbf{X}}(n)$ can be regarded as a correction term for updating the state space estimate. Notice that $\widetilde{\mathbf{X}}(n)$ is proportional to $\mathbf{P}(n)\mathbf{\Phi}(n)$ which means that $\mathbf{P}$(n) can be regarded as a matrix that constantly provides the correct direction in which to update the state estimate. However, we could, by means of averaging, find a time-invariant $\mathbf{P}$ matrix that overall makes a reasonable correction to the state estimate vector.

Step 2

By empirical means it is seen that $\mathbf{P}(n)$ consists of elements that display a certain kind of behavior. Note that $\mathbf{P}(n)$ is a square matrix of dimension $(m+1)n_a$, where $m+1 =$ the number of channel coefficients being modeled ($m = 1$ is a common value. For the TETRA standard, it is adequate to consider a two ray FIR channel model to cover most practical cases of ISI mainly because of te long symbol duration ($55.5\mu s$)) and n_a is the order of the filter used for the generation of channel coefficients (ref. the use of a-priori information) and is also the order of the square matrix $\mathbf{F}_d$ and of the vector $\mathbf{G}_d$. Except for initial transients, the time variation of $\mathbf{P}(n)$ is caused by the time variation of $\mathbf{\Phi}^H(n)$. By regarding the elemnts of $\mathbf{\Phi}^H(n)$ as stationary stochastic variables, and averaging with respect to them, approximations of mean value sequences of $\mathbf{P}(n)$ can be calculated off-line and utilized in a simplified estimator. Before we proceed, let us first illustrate the idea by a simple example:

Consider the following model specification:

$$\mathbf{F_d} = \begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix} \quad \mathbf{G_d} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \mathbf{R_e} = \begin{pmatrix} 10^{-6} & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{H} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \mathbf{\Phi^H(n)} = \begin{pmatrix} d(n)\mathbf{H} & d(n-1)\mathbf{H} \end{pmatrix}$$

since $m = 1$, the matrix $\mathbf{P}(n)$ has dimensions $4|4$

$$\mathbf{P(n)} = \begin{pmatrix} p_{1,1}(n) & \dots & p_{1,4}(n) \\ \vdots & \ddots & \vdots \\ p_{4,1}(n) & \dots & p_{4,4}(n) \end{pmatrix}$$

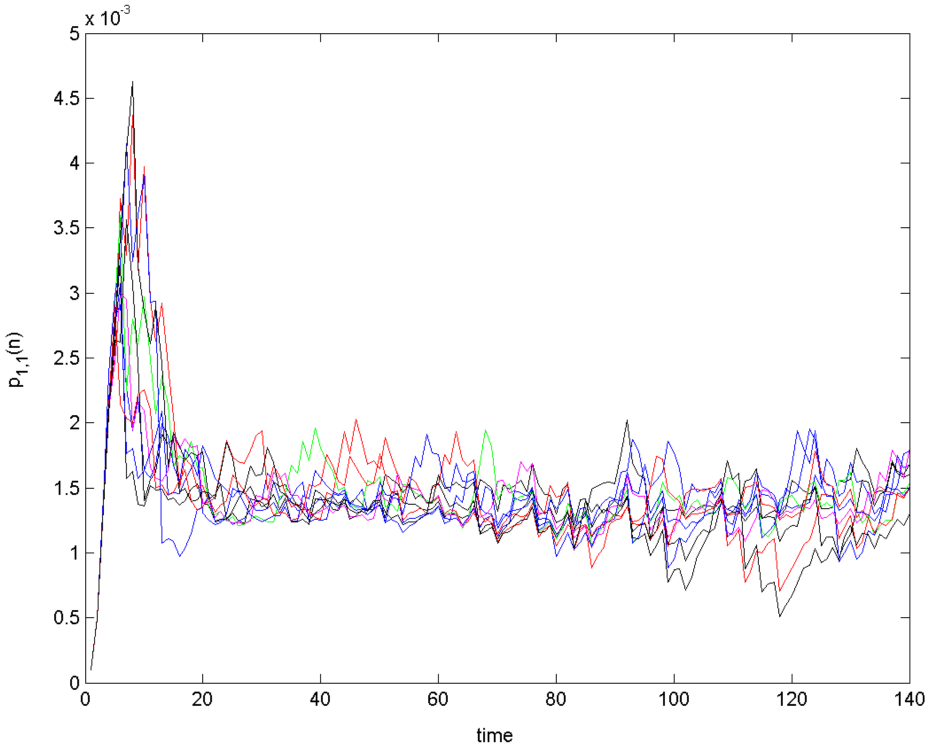

Figure 6.1: The trajectories of the element $p_{1,1}$ for ten independent realizations of the input sequence $\{d(n)\}$

The input sequence $\{d(n)\}$ is white with zero mean, where $d(n)$ have possible values of $e^{j\psi}$ where $\psi = \pm\pi/4, \pm 3\pi/4, \pm\pi/2, 0, \pi$. This implies that the diagonal elements in $\mathbf{\Phi}(n)\mathbf{\Phi}^H(n)$ are constants and that $E|d(n)|^2 = \sigma_d^2 = 1$. We initialize the Riccati equation with $\mathbf{P}(0) = 10^{-4}\mathbf{I}$, where $\mathbf{I}$ is he identity matrix. The element $p_{1,1}(n)$ for ten realizations was plotted and it was seen that the trajectories of $p_{1,1}(n)$ in different data bursts vary around a different ensemble mean value for each time instant. The sequenceof estimated mean values for 50 independent realizations was plotted and it was seen that after an initial transient phase, the estimated ensemble means do not vary much with time and could possibly be replaced with a time - invariant value.

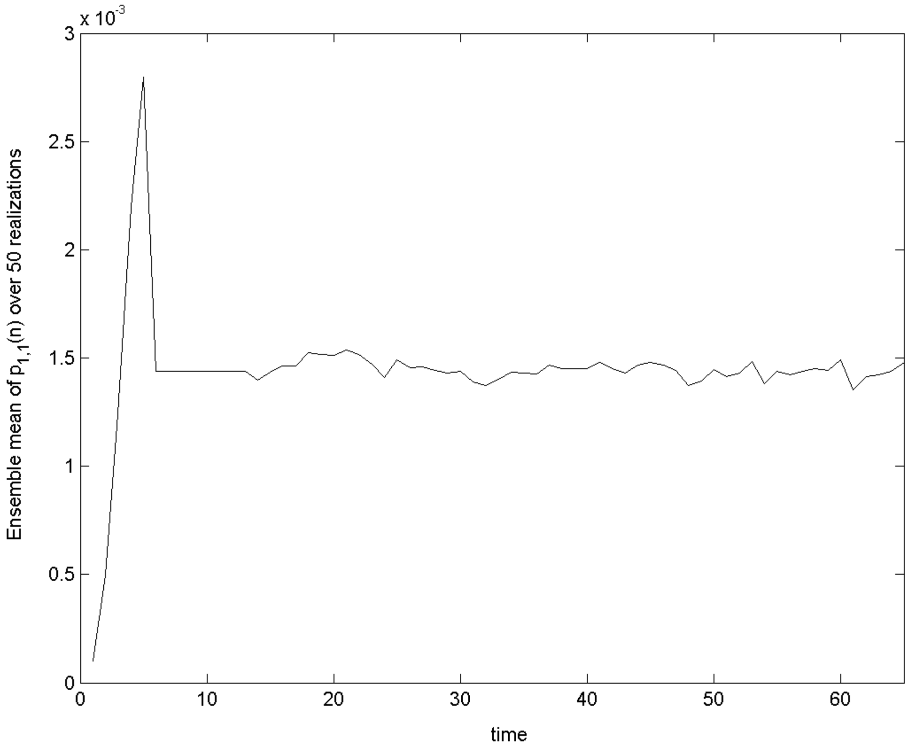

Figure 6.2: Estimated ensemble means obtained from 50 independent realizations

Each element of $\mathbf{P}(n)$ if observed over a large set of values of n, displays a characteristic behavior: at first passing through a transient period and then hovering around more or less a constant value. If independent runs are made and the same element of the $\mathbf{P}(n)$ matrix observed over the same range of values of n, very similar behavior is noticed. An ensemble mean can therefore be derived from a large number of such runs using data symbols from the set of symbols that are actually used by the standard that we are implementing the channel tracker for, and by calculating each iteration of the $\mathbf{P}(n)$ matrix using the Riccati equation discussed earlier. After a certain value of n, all elements of the $\mathbf{P}(n)$ matrix settle down or hover around a more or less constant value suggesting that the $\mathbf{P}(n)$ matrix can now be replaced by the time invariant $\mathbf{P}$-matrix. A simplification would then be to replace the sequence of ensemble means by a mean value over time. Thus, after the transient, it is possible to view $p_{1,1}(n)$ as a stationary stochastic process. Based on the intuitive reasoning above, let us regard $\{\mathbf{P}(n)\}$ as a matrix valued stochastic process and the $\mathbf{P}(n)$-sequences in each burst as an independent realization. Assume the ensemble mean $E[\mathbf{P}(n)]$ to exist. It can be expressed by taking expectation expectation on both sides of (6.15):

$$E[\mathbf{P}(n+1)] = \mathbf{F}_d E[\mathbf{P}(n) - E[\mathbf{P}(n)\mathbf{Q}(n)\mathbf{P}(n)]\mathbf{F}_d^T + \mathbf{R}_1 \tag{6.19}$$

$$\bar{\mathbf{P}}_n = E[\mathbf{P}(n)] \tag{6.20}$$

$$\Delta\mathbf{P}_n = \mathbf{P}(n) - \bar{\mathbf{P}}_n \tag{6.21}$$

$$\bar{\mathbf{P}}_{n+1} = \mathbf{F}_d\{\bar{\mathbf{P}}_n - E[(\bar{\mathbf{P}}_n + \Delta\mathbf{P}_n)\mathbf{Q}(n)(\bar{\mathbf{P}}_n + \Delta\mathbf{P}_n)]\} + \mathbf{F}_d^T + \mathbf{R}_1 \tag{6.22}$$

$$= \mathbf{F}_d\{\bar{\mathbf{P}}_n - E[\bar{\mathbf{P}}_n\mathbf{Q}(n)\bar{\mathbf{P}}_n + \Delta\mathbf{P}_n\mathbf{Q}(n)\bar{\mathbf{P}}_n + \bar{\mathbf{P}}_n\mathbf{Q}(n)\Delta\mathbf{P}_n + \Delta\mathbf{P}_n\mathbf{Q}(n)\Delta\mathbf{P}_n]\}\mathbf{F}_d^T + \mathbf{R}_1 \tag{6.23}$$

$$\bar{\mathbf{P}}_{n+1} = \mathbf{F}_d\{\bar{\mathbf{P}}_n - \underbrace{\bar{\mathbf{P}}_n\bar{\mathbf{Q}}_n\bar{\mathbf{P}}_n}_{(1)} - \widetilde{\mathbf{P}}_n\}\mathbf{F}_d^T + \mathbf{R}_1 \tag{6.24}$$

where:

$$\bar{\mathbf{Q}}_n = E[\mathbf{Q}(n)] = E\left[\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^H(n)}{\sigma_v^2 + \mathbf{\Phi}^H(n)[\bar{\mathbf{P}}_n + \Delta\mathbf{P}_n]\mathbf{\Phi}n}\right] \tag{6.25}$$

$$\widetilde{\mathbf{P}}_n = \underbrace{E[\Delta\mathbf{P}_n\mathbf{Q}(n)]}_{(2)}\bar{\mathbf{P}}_n + \bar{\mathbf{P}}_n\underbrace{E[\mathbf{Q}(n)\Delta\mathbf{P}_n]}_{(3)} + \underbrace{E[\Delta\mathbf{P}_n\mathbf{Q}(n)\Delta\mathbf{P}_n]}_{(4)} \tag{6.26}$$

Assume that $E[||\Delta\mathbf{P}_n||] \ll ||\bar{\mathbf{P}_n}||$, where $||.||$ is the spectral norm function. The term(4) in $\widetilde{\mathbf{P}}_n$ will then be small compared to (1). Furthermore $\bar{\mathbf{P}}_n$ will dominate over $\Delta\mathbf{P}_n$. Thus $\Delta\mathbf{P}_n$ becomes weakly correlated with $\mathbf{Q}(n)$ and the terms (2) and (3) in $\widetilde{\mathbf{P}}_n$ would also be small compared to (1). a approximation of the mean value sequence $\{\bar{\mathbf{P}}_n\}$ could then be calculated from a Ricatti-like equation:

$$\bar{\mathbf{Q}}_n = E\left[\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^H(n)}{\sigma_v^2 + \mathbf{\Phi}^H(n)\bar{\mathbf{P}}_n\mathbf{\Phi}n}\right] \tag{6.27}$$

$$\bar{\mathbf{P}}_{n+1} = \mathbf{F}_d[\bar{\mathbf{P}}_n - \bar{\mathbf{P}}_n\bar{\mathbf{Q}}_n\bar{\mathbf{P}}_n]\mathbf{F}_d^T + \mathbf{R}_1 \tag{6.28}$$

The second equation above is Riccati-like in form. As discussed earlier, after a certain value of n, the elements of the $\mathbf{P}(n)$ matrix settle down to a more or less constant mean value and so we may replace $\bar{\mathbf{P}}_{n+1}$ and $\bar{\mathbf{P}}_n$ on the two sides of the above equation by $\bar{\mathbf{P}}$.

$$\bar{\mathbf{Q}} = E\left[\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^H(n)}{\sigma_v^2 + \mathbf{\Phi}^H(n)\bar{\mathbf{P}}\mathbf{\Phi}n}\right] \tag{6.29}$$

$$\bar{\mathbf{P}} = \mathbf{F}_d[\bar{\mathbf{P}} - \bar{\mathbf{P}}\bar{\mathbf{Q}}\bar{\mathbf{P}}]\mathbf{F}_d^T + \mathbf{R}_1 \tag{6.30}$$

The above equations are coupled. To solve the latter (6.30), we must solve the former (6.29). However note that $\mathbf{Q}(n)$ is a random matrix with a finite number of possible values since $d(n)$ takes on a finite set of possible values. For M-ary signaling and a channel with $m+1$ coefficients, there are M^{m+1} possible values. Hence it is in principle possible to evaluate the expectation. Further, in the case of slowly time varying coefficients, the product term in the denominator of the expression for $\mathbf{Q}$ may be disregarded. However, in our application, it is hard to motivate this approximation and we will instead use another approximation which will drastically simplify the filter design.

Step 3

A particular solution to the coupled equation above is one in which $\mathbf{P}$ is a block diagonal matrix (and consequently so is $\mathbf{Q}$, if we assume that the data symbols are of constant modulus and comprise a white sequence with zero mean). This can lead to the decoupling of the two equations as follows: Under the assumption that the sequence $\{d(n)\}$ is white and $d(n)$ have constant modulus, the matrix $\mathbf{Q}$ is given by (With constant modulus and a block diagonal $\mathbf{P}$matrix, the denominator of the expression for $\mathbf{Q}$ becomes constant making it easy to evaluate the expectation):

$$\bar{\mathbf{Q}} = \frac{E[\mathbf{\Phi}(n)\mathbf{\Phi}^H(n)]}{\sigma_v^2 + \sigma_d^2 \sum_{i=0}^{m} \mathbf{H}\overline{\mathbf{P}}_{ii}\mathbf{H}^T} = \frac{\sigma_d^2 \mathbf{H}^T\mathbf{H}}{\sigma_v^2 + \sigma_d^2 \sum_{i=0}^{m} \mathbf{H}\overline{\mathbf{P}}_{ii}\mathbf{H}^T} \tag{6.31}$$

$$= \frac{\sigma_d^2}{\sigma_v^2 + \sigma_d^2 \sum_{i=0}^{m} \mathbf{H}\overline{\mathbf{P}}_{ii}\mathbf{H}^T} diag(\mathbf{H}^T\mathbf{H}.......\mathbf{H}^T\mathbf{H}) \tag{6.32}$$

where $\mathbf{H}$ has dimensions $1|n_a$. Under such a simplification, the equation for the $\mathbf{P}$-matrix can be written as a set of coupled algebraic equations:

$$\overline{\mathbf{P}}_{kk} = \mathbf{F}_d \left\{ \overline{\mathbf{P}}_{kk} - \overline{\mathbf{P}}_{kk} \frac{\mathbf{H}^T\mathbf{H}}{\varrho_k + \mathbf{H}\overline{\mathbf{P}}_{kk}\mathbf{H}^T} \overline{\mathbf{P}}_{kk} \right\} \mathbf{F}_d^T + \sigma_{e_k}^2 \mathbf{G}_d \mathbf{G}_d^T \tag{6.33}$$

$$\varrho_k = \frac{\sigma_v^2}{\sigma_d^2} + \sum_{i \neq k}^{m} \mathbf{H}\overline{\mathbf{P}}_{ii}\mathbf{H}^T \qquad k = 0, ...m. \tag{6.34}$$

Step 4

The terms $\sum_{i\neq k}^{m} \mathbf{H}^T\overline{\mathbf{P}}_{ii}\mathbf{H}$ in the denominators of the above $(m+1)$ equations cause the equations to be coupled. This a consequence of the fact that we are trying to estimate $(m+1)$ channel coefficients from one output signal. This coupling will be low when $\frac{\sigma_v^2}{\sigma_d^2} \ll \sum_{i\neq k}^{m} \mathbf{H}^T\overline{\mathbf{P}}_{ii}\mathbf{H}$. If we ignore the summation term in the denominators, the equations then become decoupled and approximate solutions to $\overline{\mathbf{P}}_{kk}$ can be calculated separately using the simple algebraic Riccati equations. In practice the ϱ_k are set to values slightly higher than $\frac{\sigma_v^2}{\sigma_d^2}$ to account for the unknown positive contributions of $\sum_{i\neq k}^{m} \mathbf{H}^T\overline{\mathbf{P}}_{ii}\mathbf{H}$. The error introduced because of this approximation is insignificant to the algorithms performance. Thus the calculation of an approximate averaged Kalman gain has been reduced to solving the $m+1$ separate algebraic Riccati equations.

The simplified Kalman Channel Tracking Algorithm is in summary as follows:

$$\mathbf{K}(n) = \frac{\mathbf{F}_d\overline{\mathbf{P}}\mathbf{\Phi}(n)}{\sigma_v^2(n) + \mathbf{\Phi}^H(n)\overline{\mathbf{P}}\mathbf{\Phi}n} = \frac{diag(\mathbf{F}_d\overline{\mathbf{P}}_{00}......\mathbf{F}_d\overline{\mathbf{P}}_{mm})(d(n)\mathbf{H}.......d(n-m)\mathbf{H})^H}{\sigma_v^2 + \sigma_d^2 \sum_{i=0}^{m} \mathbf{H}\overline{\mathbf{P}}_{ii}\mathbf{H}^T} \tag{6.35}$$

$$= \begin{pmatrix} L_0 d^*(n) \\ L_1 d^*(n-1) \\ \vdots \\ L_m d^*(n-m) \end{pmatrix}$$

where

$$L_k = \frac{\mathbf{F}_d \overline{\mathbf{P}}_{kk} \mathbf{H}^T}{\sigma_d^2 (\varrho_k \mathbf{H} \overline{\mathbf{P}}_{kk} \mathbf{H}^T)} \qquad k = 0, 1 \dots m. \tag{6.36}$$

Let:

$$\mathbf{P}_{kk} = \frac{\overline{\mathbf{P}}_{kk}}{\varrho_k} \tag{6.37}$$

$$\gamma_k = \frac{\sigma_{e_k}^2}{\varrho_k} \tag{6.38}$$

The design variables are $\mathbf{F}_d$, $\mathbf{G}_d$ and γ_k. The online computations required are:

$$\hat{x}(n+1) = \mathbf{F}_d \hat{x}(n) + \mu(n) L_k d^*(n-k) sat_\alpha[\epsilon(n)] \tag{6.39}$$

$$\hat{h}_k(n) = \mathbf{H}\hat{x}(n) \qquad k = 0, 1 \dots m. \tag{6.40}$$

where

$$\epsilon_n = y(n) - \mathbf{\Phi}^H(n)\widehat{\theta}(n) \tag{6.41}$$

The *sat* function is used to limit the forward prediction error to prevent the algorithm from going off track. The gains $L_k, k = 0, 1 \dots m.$ can be calculated from the solutions to:

$$\mathbf{P}_{kk} = \mathbf{F}_d \left[\mathbf{P}_{kk} - \frac{\mathbf{P}_{kk}\mathbf{H}^T\mathbf{H}\mathbf{P}_{kk}}{1 + \mathbf{H}\mathbf{P}_{kk}\mathbf{H}^T} \right] \mathbf{F}_d^T + \gamma_k \mathbf{G}_d \mathbf{G}_d^T \tag{6.42}$$

$$L_k = \frac{\mathbf{F}_d \mathbf{P}_{kk} \mathbf{H}^T}{\sigma_d^2 (1 + \mathbf{H}\mathbf{P}_{kk}\mathbf{H}^T)} \tag{6.43}$$

The time varying positive scalar has been introduced in order to speed up the convergence during the transient phase and by using for example, the recursion:

$$\mu(n) = \beta\mu(n-1) + (\beta - 1) \tag{6.44}$$

What remains now is to choose suitable values for our design variables $\mathbf{F}_d$ and $\mathbf{G}_d$. This is described next.

6.3 The second order autoregressive ($\mathbf{AR_2}$) stochastic model

The KLMS algorithm uses second order models to model the variation of channel coefficients (this is the internal stochastic modeling that represents the *a priori* information being utilized by the algorithm and also the choice of our design variables). We know that the channel co-efficients in a Rayleigh fading environment behave as narrow band noise with a spectral peak [Lin01R]. The simplest model that describes such behaviour of the channel coefficients, particularly suited to the Jakes' fading model is the second order autoregressive ($\mathbf{AR_2}$) model [[Lin06R]] with real valued coefficients:

$$h_k(n) = \frac{1}{1 + a_1 q^{-1} + a_2 q^{-2}} e_k(n) \tag{6.45}$$

$$a_1 = -2 r_d cos(\omega_d) \tag{6.46}$$

$$a_2 = r_d^2 \tag{6.47}$$

Or in observable canonical state space:

$$\mathbf{F_d} = \begin{pmatrix} -a_1 & 1 \\ -a_2 & 0 \end{pmatrix}$$

$$\mathbf{G_d} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\mathbf{X} = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

The pole locations are $r_d e^{\pm j\omega_d}$. The pole radius r_d reflects the damping and ω_d is the dominating frequency of the coefficient variations. The design variables are then r_d and ω_d . The variance of $h_k(n)$ is given by:

$$\sigma_{h_k}^2 = \left(\frac{1 + a_2}{1 - a_1}\right) \frac{\sigma_{e_k}^2}{(1 + a_2)^2 - a_1^2} \tag{6.48}$$

The variance of the noise sequence $\sigma_{e_k}^2$ is adjusted to as to make the variance of the channel coefficients $\sigma_{h_k}^2$ equal to the desired value. With the second order Autoregressive model, simple analytical expressions for the gains in the simplified Kalman

Algorithm exist:

$$L_k = \frac{1}{\sigma_d^2(1+p_{1,1})} \begin{pmatrix} -a_1 p_{1,1} + p_{1,2} \\ -a_2 p_{1,1} \end{pmatrix} \tag{6.49}$$

where $p_{1,1}$ and $p_{1,2}$ are the elements of the **P**-matrix and simple analytical expressions exist for them too:

$$\zeta = \frac{\alpha_0 - 2a_2 + \sqrt{(\alpha_0 + 2a_2)^2 - 4a_1^2(1+a_2)^2}}{2} \tag{6.50}$$

$$\alpha_0 = 1 + a_1^2 + a_2^2 + \gamma_k \tag{6.51}$$

$$p_{1,1} = \frac{\zeta + \sqrt{\zeta^2 - 4a_2^2}}{2} - 1 \qquad p_{1,2} = \frac{a_1 a_2}{1 + a_2 + p_{1,1}} p_{1,1} \tag{6.52}$$

Typical values of r_r used in the model above are between 0.990 and 0.999. ω_d is calculated based on the value of r_d chosen and the estimate of the Doppler frequency. In the following chapter we describe a Doppler estimation algorithm that can be used in conjunction with our channel tracker.In chapter 8, we present simulations results for the performance evaluation of the KLMS algorithm on the TETRA mobile radio standard.

CHAPTER 7

Doppler Estimation

7.1 Introduction

Doppler shift is the frequency shift experienced by the radio signal when either the transmitter or the receiver is in motion, and Doppler spread is a measure of the spread broadening caused by the temporal rate of change of the mobile radio channel. At a carrier frequency of 450 Mhz (the carrier frequency of the TETRA mobile radio system), a typical mobile terminal experiences a Doppler spread in the range of 0-250 Hz depending on the vehicle speed. For eg. a vehicle travelling a speed of $v = 100$ Kmph would experience a maximum Doppler spread of $f_d = v/\lambda = 83$ Hz, where λ is the wavelength corresponding to the carrier frequency.

In an adaptive receiver, Doppler information can be used to improve performance or reduce complexity. Receiver parameters, such as channel tracker step size can be optimized based on Doppler spread information. Similarly, Doppler information could be used to control the transmitter or receiver adaptively for different mobile speeds, like variable coding and interleaving schemes. Also, radio network control algorithms, such as handoff and channel allocation in cellular systems can utilize the Doppler information. In our work, use of a Doppler estimation algorithm is motivated by the fact that parameters of the stochastic hypermodels on which the Kalman Least Mean Squares tracking algorithm is based, are a function of the Doppler frequency.

7.2 Related work

Doppler spread information has been studied for various applications in the past. Methods based on correlation and variation of channel estimates have been used for Doppler Estimation. Lindbom describes a simple method which uses differentials of the complex channel estimates [Lin01R]. The differentials of the channel estimates are noisy, requiring low-pass filtering , the bandwidth of which is also a function of the Doppler estimate. In [Mor01R], a Doppler estimation scheme based on the autocorrelation of the channel estimates is described.

In [Kra01R], an efficient Doppler Estimation algorithm has been discussed for mobile radio systems based on Maximum Likelihood (ML) estimation theory. The algorithm relies on periodinc channel estimation through periodiic pilot channels present in the slot structre of the mobile radio slot structure.

7.3 Our work

Here, we adapt the ML algorithm in [Kra01R], to the TETRA uplink channel wherein no pilot channels are available in the data slot apart from the training sequence which occurs as a midamble (ref. section 2.1). By coupling the ML Doppler Estimation algorithm with Normalized Least Mean Square Channel tracking in conjunction with a Viterbi detector, we are able to use the channel coefficient estimates from the chosen output trellis of the Viterbi detector as a 'peudo-pilot channel'. These channel coefficient estimates are then used to compute an instantaneous channel correlation matrix over the slot. Using this channel correlation matrix and an estimate of the variance as an input the algorithm computes the ML estimate for the Doppler frequency. While the algorithm requires knowledge of the noise variance (which in turn would require an estimate of the SNR at the receiver) and a separate algorithm for the estimation of SNR may be used (the path metric of the best path in the Viterbi algorithm can be used to get an estimate of the noise variance), it can be shown that the ML Doppler estimation algorithm is fairly robust to the mismatch between an assumed SNR and the actual value of SNR at the receiver [Kra01R]. Typical values of SNR may range between 10 and 25 dB. We have used an SNR of 17 dB in our simulations of the ML Doppler estimation algorithm (see section 8.3).

7.4 The ML Doppler estimation algorithm

The band-limited signal s(t), which is transmitted through a fading mobile radio channel is received by the receiver antenna. The received signal is modeled as:

$$r(t) = h(t).s(t) + w(t) \tag{7.1}$$

where the channel is described by the complex Gaussian process $h_k(t)$, ($k = 0, 1$ for our model of a two-tap dispersive Rayleigh fading channel) with coreallation fuction $K_h(t_1, t_2; f_d)$, which dedepends upon the Doppler spread, f_d, and $w(t)$ is zero-mean,

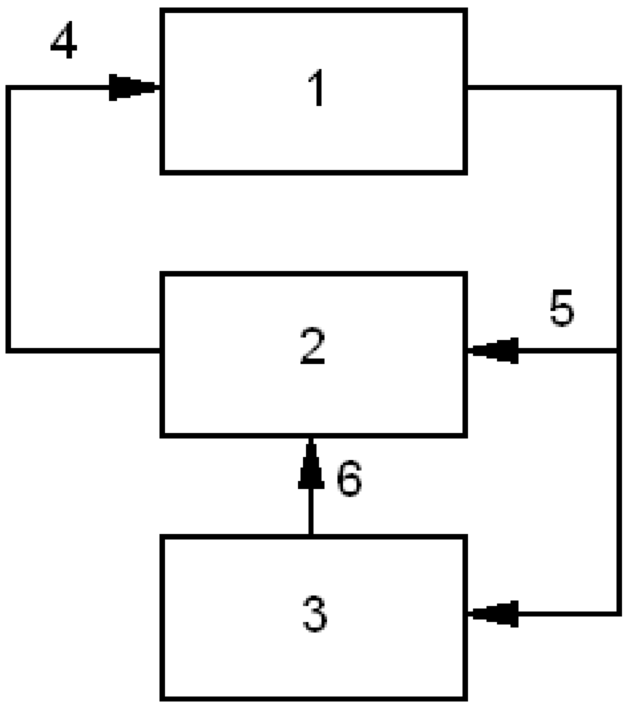

Figure 7.1: Conceptual bock diagram showing how the Doppler estimator is used in conjuntion with the channel tracker. 1: Channel tracking device working on a stochastic hypermodel, 2: Doppler estimator, 3: MLSE Viterbi decoder, 4: Doppler frequency estimate fed to the channel tracking stochastic hypermodel, 5: Channel co-efficient estimates from the training sequence fed to the Doppler estimator, 6: Channel coefficients from the best path in the Viterbi trellis to serve as 'pseudo-pilot channels' fed to the Doppler estimotor.

complex AWGN with variance σ_w^2. The TETRA uplink slot format (using $\pi/4$-shifted DQPSK modulation) shown in Fig (2.2) is assumed. The entire slot is 231 symbols long and the symbol rate is 18 kbaud. In our adaptation of the ML algorithm to the TETRA uplink channel, we estimate the channel during the training sequence using the least squares (Ref. section 8.1.3) and subsequent channel coefficient estimates during the slot are picked up from the sequence of channel estimates corresponding to the maximum likelihood sequence that is the output of the Viterbi detector. Henceforth we shall refer to these channel coefficient estimates, picked at periodic intervals through the data frame, as 'Pseudo-pilot channel estimates'.

We assume that the channel estimates $\hat{h}_1(i), \ldots, \hat{h}_Q(i)$ are available for Q slots, where $\hat{h}_q(i)$ are the channel estimates for a given channel tap at the ith 'pseudo-pilot location' for the qth slot, $i \in [1, N]$, and N is the number of channel estimates for each slot including the estimate obtained from the training sequence. In our implementationo of the algorithm we will use only one of the two taps in our dispersive channel model. Alternately, the estimation algorithm may be run on channel coefficient estimates on both the taps separately and the Doppler frequency be expresses as a men of the estimates obtained from the two taps. Three requirements are introduced for a synthesis of the ML estimate:
Requirement 1: The channel estimates $\hat{h}_q(i)$ are Gaussian distributed, because they are a linear function of $r(t)$.
Requirement 2: The errors of the channel estimates $\hat{h}_{q1}(i)$ and $\hat{h}_{q2}(i)$ for different slots $(q_1 \neq q_2)$ are statistically independent.
Requirement 3: The Doppler spread f_d does not change during Q slots.

Taking these requirements into account, it can be shown that ML Doppler spread estimate minimizes the goal function:

$$F_{opt}(f_d) = S(f_d) + \sum_{i=1}^{N}\sum_{l=1}^{N} \widehat{K}(i,l).K^{-1}(i,l;f_d) \tag{7.2}$$

where $S(f_d) = \ln[\det(\mathbf{K(f_d)})]$, $\det(\mathbf{K(f_d)})$is the determinant of the correlation matrix $\mathbf{K(f_d)}$ with elements

$$K(i,l;f_d) = K_h(i,l;f_d) + \sigma_w^2/N_p.\delta_{il} \tag{7.3}$$

corresponding to the Doppler spread hypothesis f_d, δ_{il} is the Kronecker symbol, N_p is the number of pilot symbols for each slot.

$$\widehat{K}(i,l) = \frac{1}{Q}\sum_{q=1}^{Q} \widehat{h}_q(i)\widehat{h}_q^*(l) \tag{7.4}$$

is an estimate of the channel correlation matrix, $K^{-1}(i,l;f_d)$ denotes the elemnts of the matrix $\mathbf{K^{-1}(f_d)}$, which is an inverse of the matrix $\mathbf{K(f_d)}$.

The algorithm above describes a multichannel system which produces the metrocs (2). Each channel of this system combines the estimates $\widehat{K}(i,l)$ using the weights $K^{-1}(i,l;f_d)$. The weights for the mth channel correspond to some alternative value

$f_{d,m}$ of the Doppler spread. Then the metrics $F_{opt}(f_{d,1}), \ldots, F_{opt}(f_{d,M})$ are compared with each other. On the basis of the comparisions between M Doppler spread alternatives $f_{d,1}, \ldots, f_{d,M}$, the optimal estimator generates the Doppler estimate $\hat{f}_d = f_{d,m}$ if the mth channel has the minimum output value. One can see that the knowledge of the channel correlation matrix $\mathbf{K}(\mathbf{f_d})$ has significant implications for the design of a pactical system. The problem of mismatch arises when the true characteristics of the channel coreallation matrix differ from the assumed ones. ONe of the sources of the error which occurs in practical systems is the mismatch between true ans assumed signal-to-noise ratio. However the ML Doppler estimation algorithm proves to be quite robust to this type of mismatch.

7.5 Simulation study

In our simulation study we use the optimal Doppler estimation algorithm along with the NLMS (Normalized Least mean Squares) channel tracking algorithm in conjunction with a per survivor viterbi detector. The 'pseudo-pilot channel' estimates are obtained by channel coefficients at fixed intervals within the slot from the maximum likelihood trellis chosen by the Viterbi detector. The Rayleigh fading TETRA uplink channel is modelled by a two tap system at symbol space with the taps experiencing independant Rayleigh fading. Rayleigh fading waveforms are generated using the Jakes fading model $K_h(\tau; f_d) = \sigma_h^2 J_0(2\pi f_h \tau)$ [3.3]. A signal - to - noise ratio of 17 dB is assumed for all simulations. Results for these simulations will be presenter in chapter 8.

CHAPTER 8

TETRA case study and simulation results

In this chapter performace investigations of adaptive channel tracking algorithms discussed in chapter [6] and of the adaptive Viterbi scheme discussed in section [5.2] are carried out for a Rayleigh fding channel model. All BER simulations are based on 1000 data bursts ($\approx$ 108000 unknown bits) at signal-to-noise ratios between 15 and 35 dB. The considered transmission system is presented in section [2.1] "The TETRA starndard" and reproduced below for convenience.

Before presenting simulation results for the adaptive channel tracking and Doppler estimation schemes discussed in this thesis we present results from a set of tests that were performed to verify the various modules in our simulation of the TETRA transmitter and receiver system. We present results from these tests below for the transmitter, channel and receiver used in our simulation study, along with the relevant parameters specified in the TETRA mobile radio standard.

8.1 Simulation system

8.1.1 Transmitter

A detailed description of the TETRA transmission system was presented in section [2.1]. The following specifications from the TETRA mobile radio standard were adhered to while designing the simulator for the radio transmitter:

- Carrier Frequency, $f_c = 450$ Mhz
- Symbol rate, $R_s = 18$ Kbaud.
- Burst size = 227 symbols, including 11 training sequence symbols in the middle
- Modulation: $\pi/4$ - shifted Differentially Encoded Quadrature Phase Shift Keying (DQPSK)

• SRRC filter roll-off factor, $\alpha = 0.35$

The SRRC pulse shaper was designed for simulation purposes to contain 64 samples with a spread of 4 lobes on either side of the central lobe, a single lobe having 8 samples. This is depicted in figure [8.1].

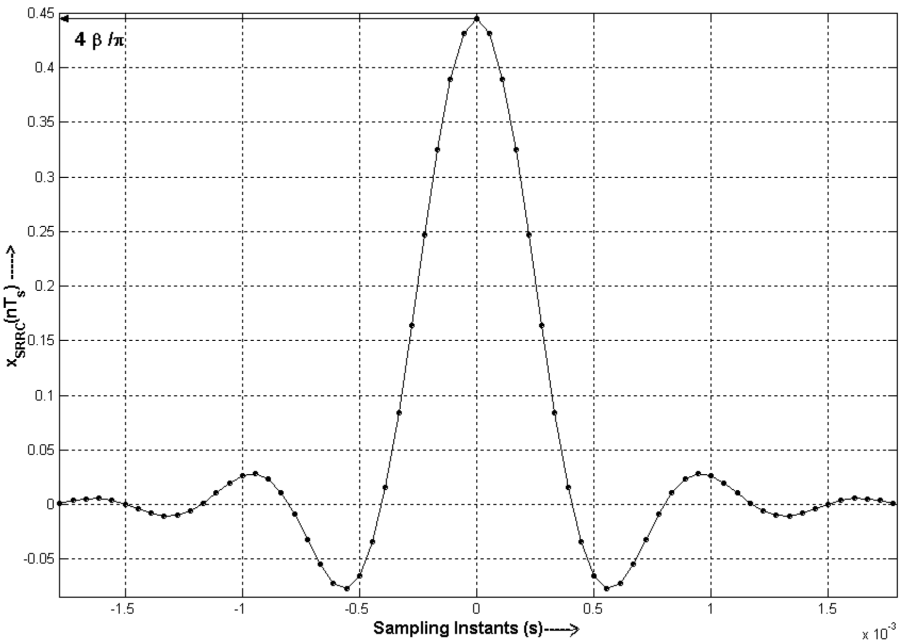

Figure 8.1: Generated SRRC pulse used for pulse shaping at transmitter

Figure [8.2] shows a simulation of our $\pi/4$ - shifted DQPSK modulator through its constellation diagram. The figure shows transitions between modulated symbols after they have been shped by the SRRC filter above. Note the the diagram is in the shape of a donut with a 'hole', indicating that symbol transitions tend to remain away from the origin thus avoiding large changes in the enveloe of the transmitted signal.

8.1.2 Channel Modelling

The performance of the suggested channel tracking and Doppler estimation methods has been evaluated on the channel model:

$$y(n) = h_0(n)d(n) + h_1 d(n-1) + v(n) \tag{8.1}$$

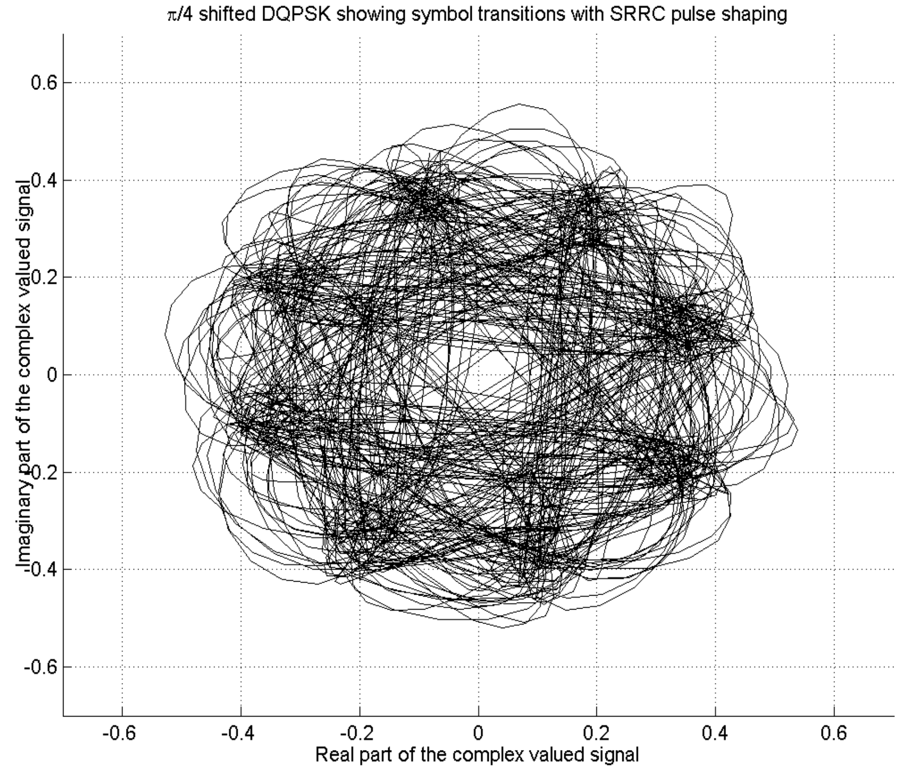

Figure 8.2: Constellation diagram of the transmitted symbols showing transitions between symbols

Here h_0 and h_1 are co-efficients of a two-ray FIR channel model spaced at symbol duration and subjected to Rayleigh fading such that $E|h_0|^2 = E|h_1|^2 = 1$. $v(n)$ is a white noise process. This model adequately describes channels encountered in the TErrestrial Trunked RAdio standard mainly because of the relatively long symbol time ($55.5\mu s$). It is also assumed that the influence of the receiver filter is small and can be neglected. Thus $h_0(n)$ and $h_1(n)$ are mutually uncorrelated. Thus $Re\{h_k(n)\}$ and $Im\{h_k(n)\}$, $k = 0, 1$ are mutually independent Gaussian processes with normalised auto-covariance functions

$$R(l) = J_0(2\pi \frac{f_m}{f_s} l) \quad l = 0, \pm 1, \ldots \tag{8.2}$$

where $J_0(.)$ is the Bessel function of the first kind and zero order, f_m is the maximum Doppler frequency in Hz and f_s is the sampling rate. The simulation inf a Rayleigh fading environment was done using the Jakes fading model (ref. Appendix C). For the purpose of performance evaluation simulations have been carried out on the described channel model at doppler frequencies of 42, 83 and 100 Hz at value of E_b/N_o ranging from 15 dB to 35 dB (where E_b is the energy per bit and N_0 is the noise spectral density). The frequencies 40, 83 and 100 Hz correspond to vehicle speeds of approximately 100, 200 and 240 Kmph.

Figure [8.3] shows a simulation of a Rayleigh fading variable using the Jakes' model, over a large number of samples. Figure [8.4] shows the same Rayleigh fade variable over the duration of one half of a TETRA frame. (Recall that the training sequence in the TETRA slot structure is place in the middle. Therefore, the amount of fading that is to be dealt with in one slot duration is effectively only that which occurs over half the slot as each half is separately viterbi decoded using the training sequence midamble)

The Jakes model parameters used were:

- $N_0 = 15$
- $N = 4N_0 + 2 = 62$
- $\beta_i = i * \pi/15$

The designed Jakes model was tested against mathematically simulated autocorrelation curves and the following figures depict the results :

8.1.3 Receiver

The receive filter used was matched to the transmitter SRRC filter with a roll-off factor of 0.35. A per survivor Viterbi implementation (Ref. section [5.2]) was used in conjunction with the channel tracking algorithm. The channel tracking schemes implemented and tested on the TETRA uplink channel are: the Normalized Least Mean Squares (NLMS) algorithm and the Kalman Least Mean Squares (KLMS) algorithm based on styochastic hypermodelling.

The Parameters used in the NLMS channel tracking implementation were

- $\lambda_{NLMS} = 0.5$
- $\lambda_{smoothing} = 0.9$

The implementation of the KLMS algorithm used in our simulations utilizes the

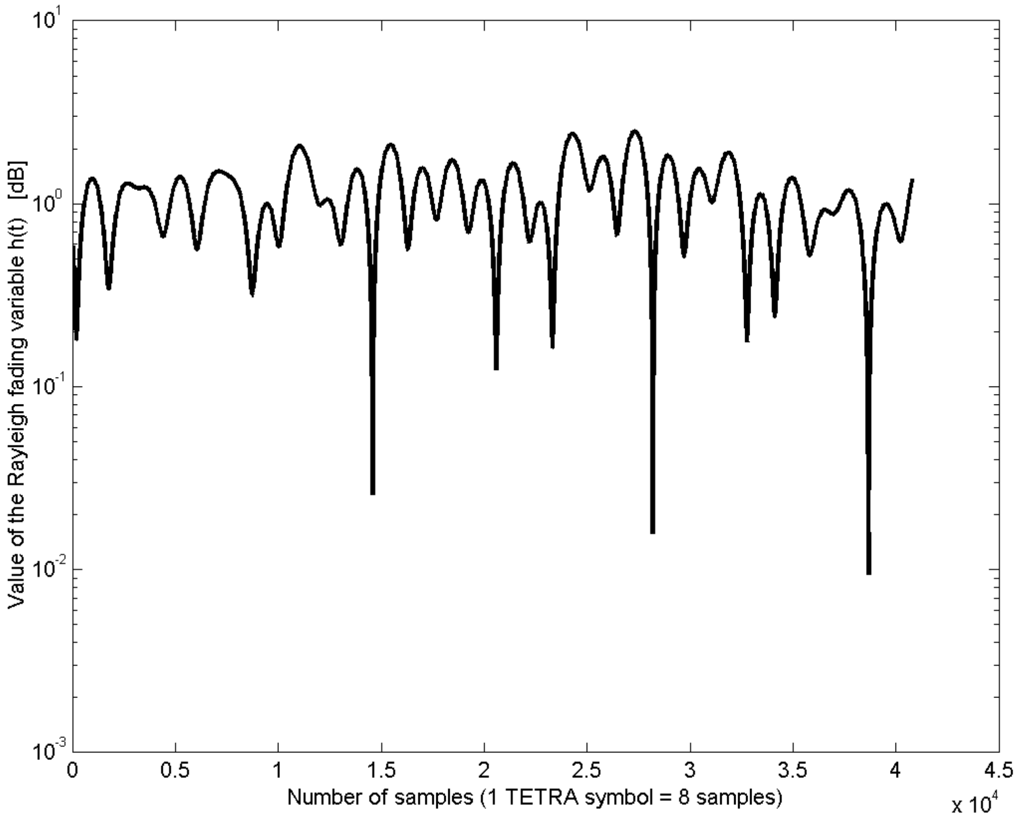

Figure 8.3: Fading characteristics of a single channel co-efficient in a FIR channel model, generated by the Jakes fading model.

second order autoregressive ($\mathbf{AR_2}$) hypermodel described in section [6.3]. The design parameters used in the hypermodel were:

- $r_d = 0.998$
- $\omega_d = 2\pi\frac{f_d}{f_s}$

where f_d is the estimated maximum Doppler spread and f_s is the symbol rate.

The NLMS and KLMS channel tracking algorithms must be initialised when they enter the decisio - directed modefrom the learning mode. This is done in the following steps:

1. The mean levels of the channel co-efficients are first estimated in the duration of the training sequence using the Least Squares method. (This implies use

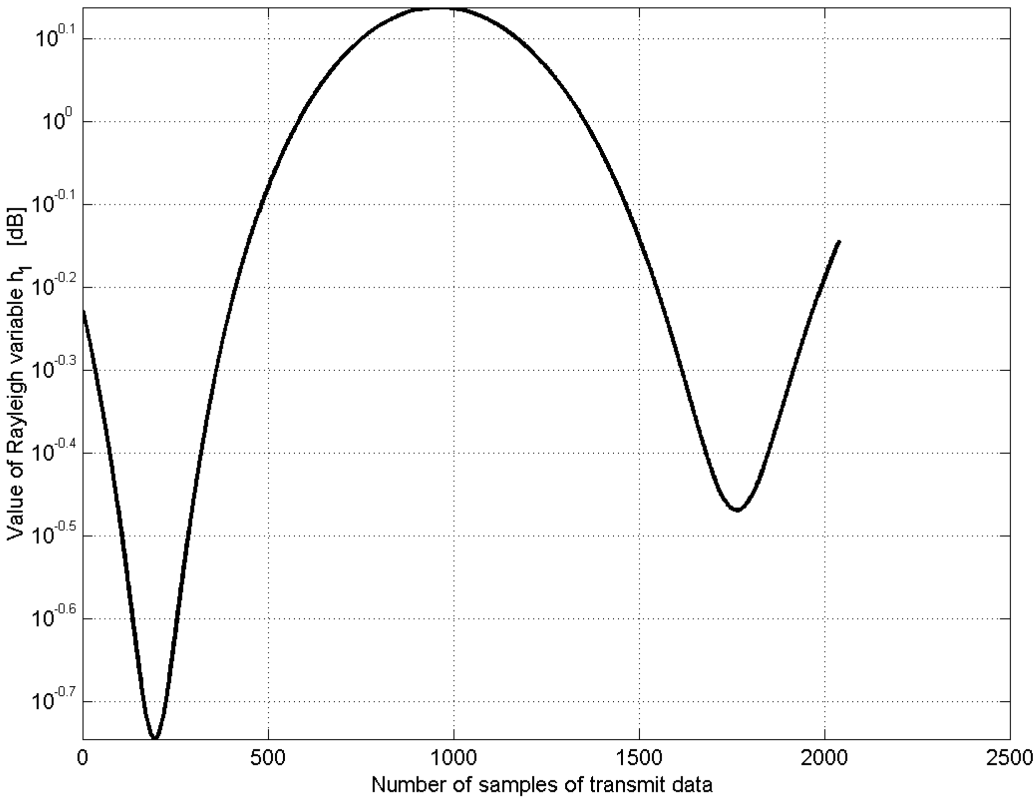

Figure 8.4: Variation of a channel co-efficient during a single slot (255 symbols * 8 samples/symbol = 2040 samples

of the minimum - mean square error criterion to estimate an average channel co-efficient vector over the duration of the training sequence alone).

$$y(n) = \phi^H \theta + v(n)$$

$$\phi^H(n) = (d(n) \dots d(n-m)) \quad \theta^T = (h_0 \dots h_m)$$

The Least Squares solution to $\sum_{s=1}^{N_{tr}} |y(s) - \phi^H(s)\theta|^2$ (the minimum mean square error) is given by:

$$\hat{\theta} = \left[\sum_{s=1}^{N_{tr}} \phi(s)\phi^H(s) \right]^{-1} . \sum_{s=1}^{N_{tr}} \phi(s)y(s). \tag{8.3}$$

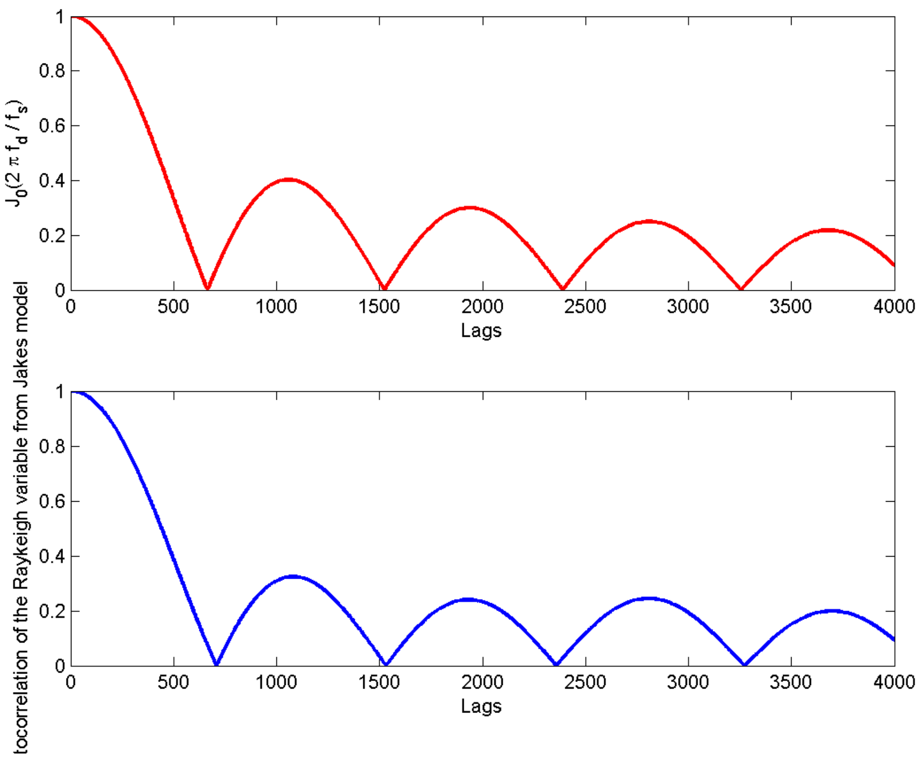

Figure 8.5: Jakes model - A comparison of theoretical and implemented versions based on autocorrelation of generated waveform

2. Having obtained a mean estimate of the channel co-efficients by the least squares method, the obtained channel estimates are used as a starting estimate and the NLMS algorithm is rum over the trainging sequence in conjunction with the per survivor Viterbi detector. This gives a good estimate of the channel co-efficients at the point in the slot where the training sequence ends and the data begins. For NLMS tracking, the channel co-efficient estimates thus obtained are used as a starting estimate and the NLMS algorithm is run over the data burst. For KLMS tracking, the state space vector is initialized using the estimates obtained from the least squares estimation thus:

$$\hat{x}(N_{tr}) = -\begin{pmatrix} a_1 & \dots & a_{n_a} \\ \vdots & \ddots & \\ a_{n_a} & & 0 \end{pmatrix} \cdot \begin{pmatrix} \hat{h}_k(N_{tr}-1) \\ \vdots \\ \hat{h}_k(N_{tr}-n_a) \end{pmatrix} \tag{8.4}$$

where $\hat{h}_k(N_{tr}-1) = \ldots = \hat{h}_k(N_{tr}-n_a) = \hat{h}_k$ = mean level of the kth channel co-efficient obtained from the least squares run over the training sequence.

For KLMS tracking, the variable $\varrho_k = \sigma_v^2/\sigma_d^2 + \sum_{i\neq k} H\overline{P}_{ii}H^T$ was required to calculate the adaptation gains L_k. In our simulations, this variable is approximately extimated from the forward prediction error as follows:

$$\hat{\varrho}_k \approx \hat{\sigma}_\epsilon^2/\sigma_d^2 \tag{8.5}$$

where $\hat{\sigma}_\epsilon^2$ is an estimate of the varianve of the prediction errors. We estimate the variance of the prediction error within a burst, by using symbols from the training sequence only. For rapidly time varying channels, it would not be possible to estimate the variance from one training sequence alone, however under the assumptions of stationary measurement noise and SNR, an estimate of the variance $\hat{\sigma}_\epsilon^2$ could be computed from en ensemble of of "local" estimates of the variance, obtained from previous training sequences. Thus in our simulations, an estimate of the variance of the prediction error (and thus an estimate of the variable $\hat{\varrho}_k$) is obtained as:

$$\hat{\sigma}_\epsilon^2(i) = \hat{\sigma}_\epsilon^2(i-1) + \frac{1}{i}(\hat{\sigma}_{\epsilon tr}^2(i) - \hat{\sigma}_\epsilon^2(i-1)) \quad i = 1, 2, \ldots \tag{8.6}$$

8.2 Simulation results: Adaptive channel tracking using the KLMS and NLMS algorithms

We now presents results from our performance evaluation study of the studied channel tracking algorithms and adaptive Viterbi scheme. Figure [8.6] and [8.7] show the variation of the simulated Rayleigh fading variable over one half of a TETRA data slot (108 symbols) along with the tracking profile of the NLMS and the KLMS algorithms implemented separately in the receiver.

Figures [8.8] and [8.9] demonstrate the effect of smoothing of channel co-efficient estimates obtained by the NLMS channel tracking algorithm at a high Doppler (100 Hz) and a low Doppler frequency (10 Hz) respectively. It can be clearly seen that at a low Doppler frequency, smoothing is highly advantageous while at high Doppler there is not much differece between the smoothed and un-smoothed channel estimates.

Figures [8.10], [8.11] and [8.12] show the simulation results for the BER performance evaluation of the KLMS and the NLMS channel tracking algorithms in conjunction with a per surivor Viterbi MLSE decoder.

The second order autoregressive hypermodel chosen for the implementation of the Kalman Least Mean Squares (KLMS) algorithm requires an estimate of the maximum doppler spread. In figure [8.13] we demonstrate the sensitivity of the KLMS channel tracker to this estimate. It can be seen that a mismatch between the actual maximum

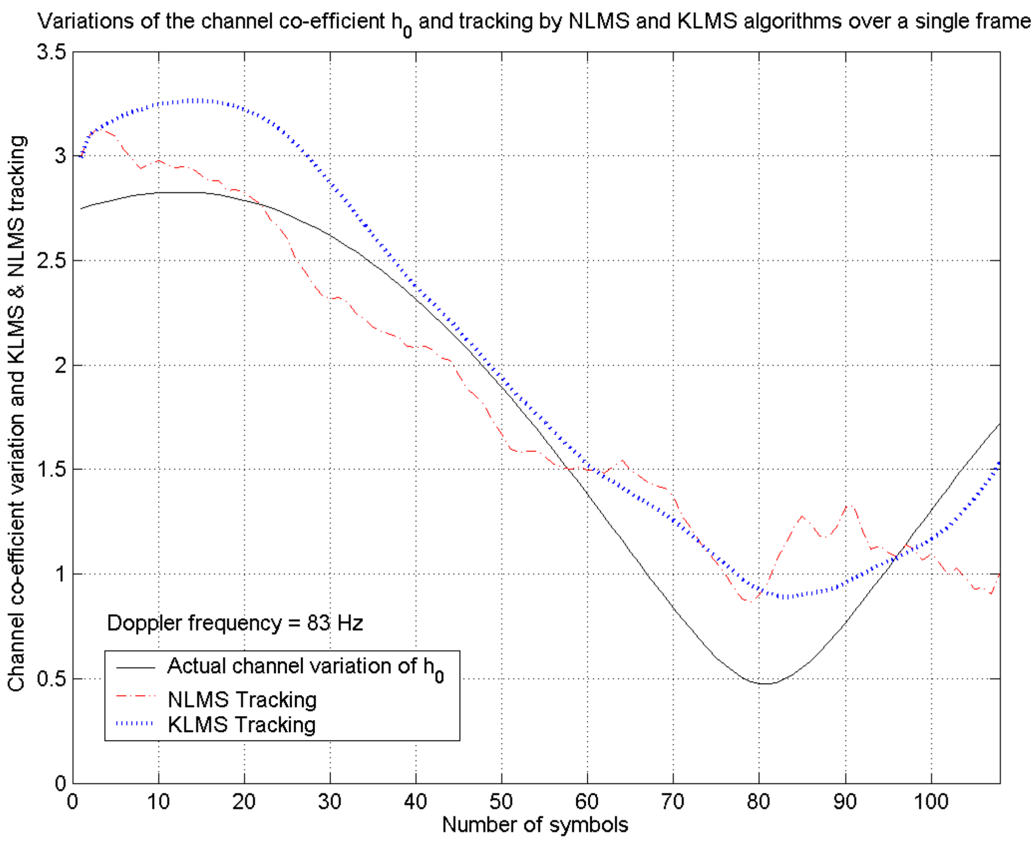

Figure 8.6: Tracking of channel coefficient $h_0(n)$ by the KLMS and NLMS channel trackers

Doppler frequency and that assumed by the $\mathbf{AR_2}$ hypermodel results in an inferior BER performance. This calls for a Dopper estimation algorithm that can provide as input to the KLMS channel tracker, an instantaneous estimate of the maximum Doppler frequency.

8.3 Simulation results: Doppler estimation

Figure [8.14] shows an inverse goal function $F_{opt}^{-1}(f_d)$ as a function of frequency. Analysis of the curves in fugure [8.15] shows that the optimal algorithm estimates the Doppler frequency with high accuracy and does not deviate too much if there is a mismatch between the real and the *a priori* assumed signal-to-noise ratio. Finally figure [8.16] shows that rate of convergence of the ML algorithm. It can be seen that

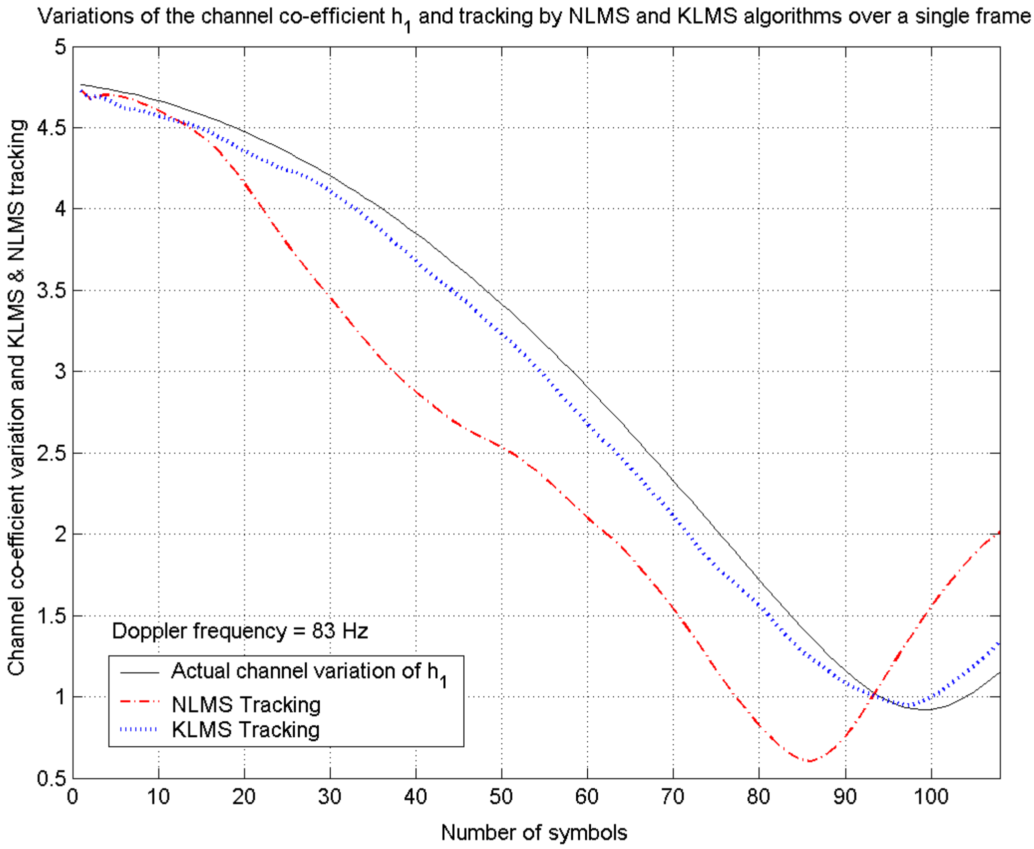

Figure 8.7: Tracking of channel coefficient $h_1(n)$ by the KLMS and NLMS channel trackers

a fairly close estimate can be obtained by running the algorithm over 10 to 20 slots.

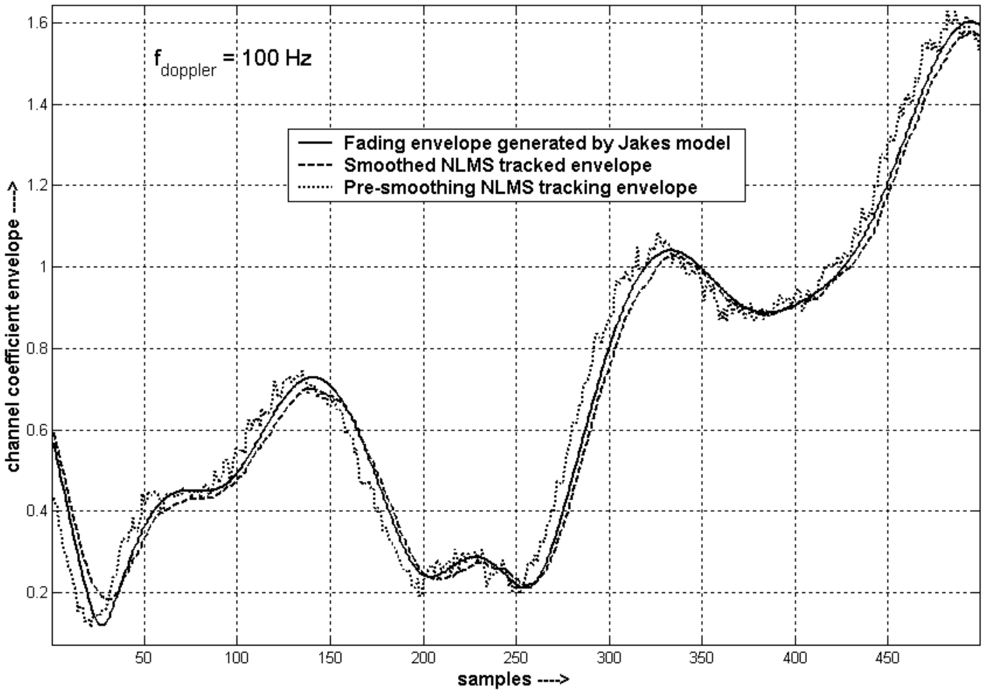

Figure 8.8: Smoothing of channel co-efficient estimates obtained by NLMS tracking at a high Doppler frequency

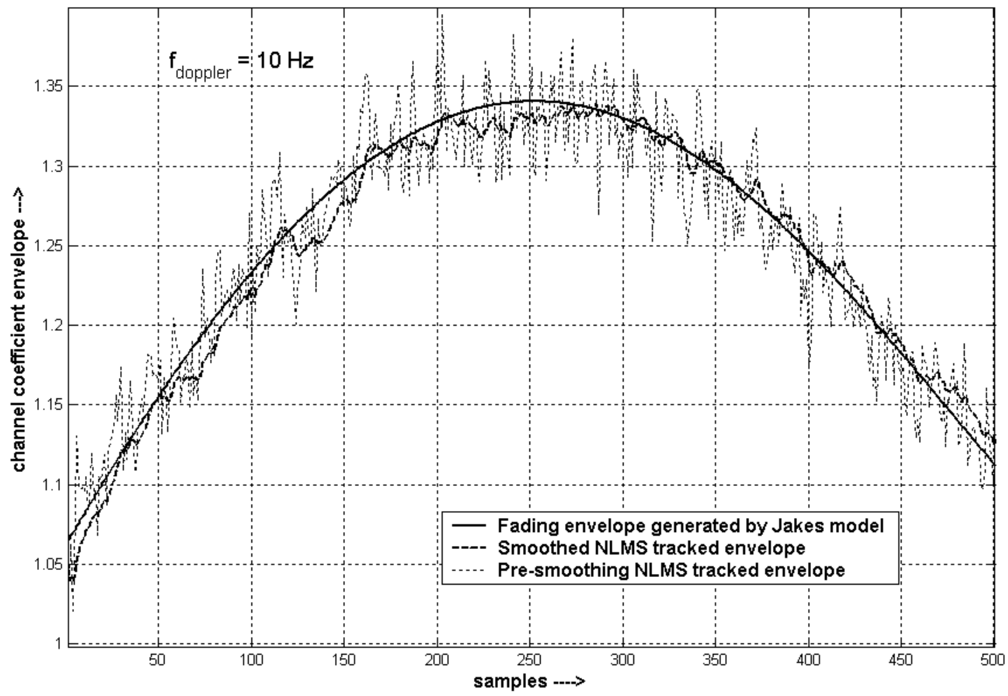

Figure 8.9: Smoothing of channel co-efficient estimates obtained by NLMS tracking at a high Doppler frequency

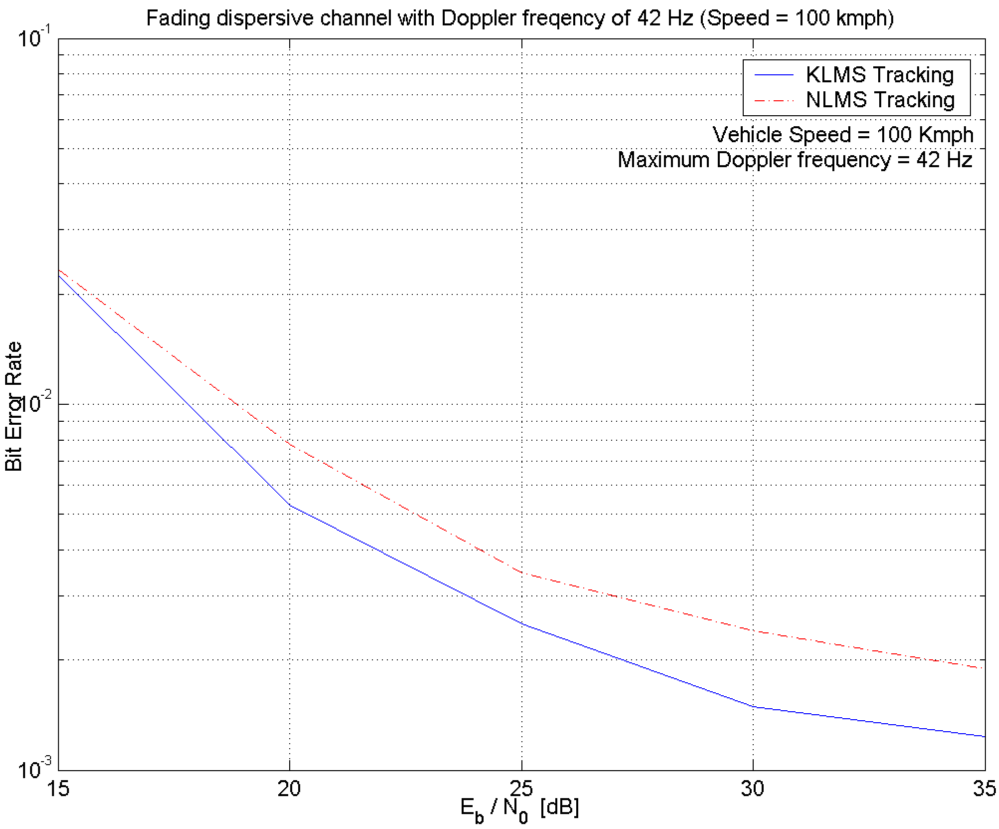

Figure 8.10: BER performance of KLMS against NLMS at $f_d = 42Hz$

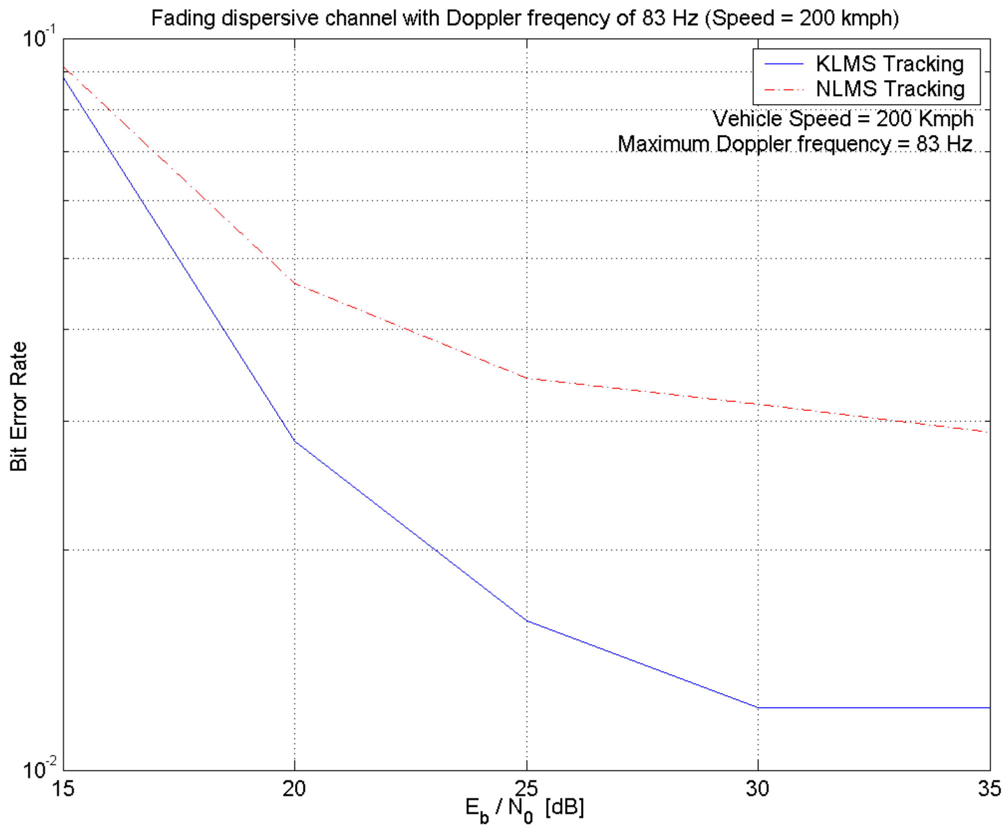

Figure 8.11: BER performance of KLMS against NLMS at $f_d = 83Hz$

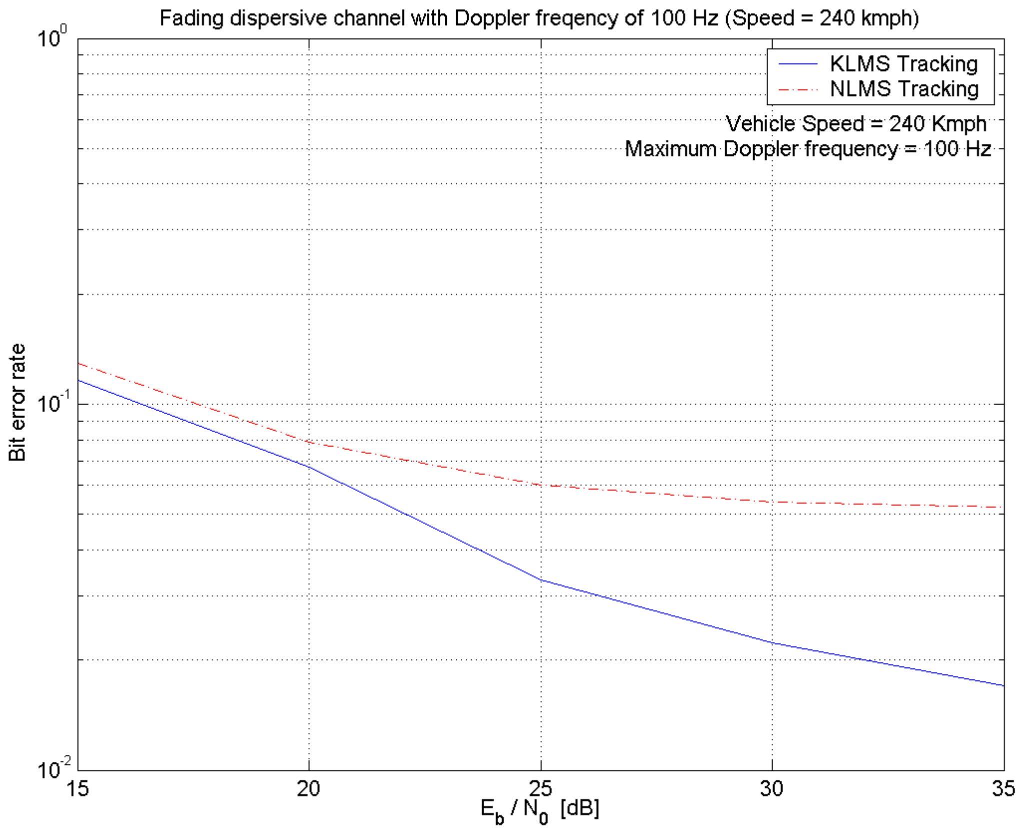

Figure 8.12: BER performance of KLMS against NLMS at $f_d = 100Hz$

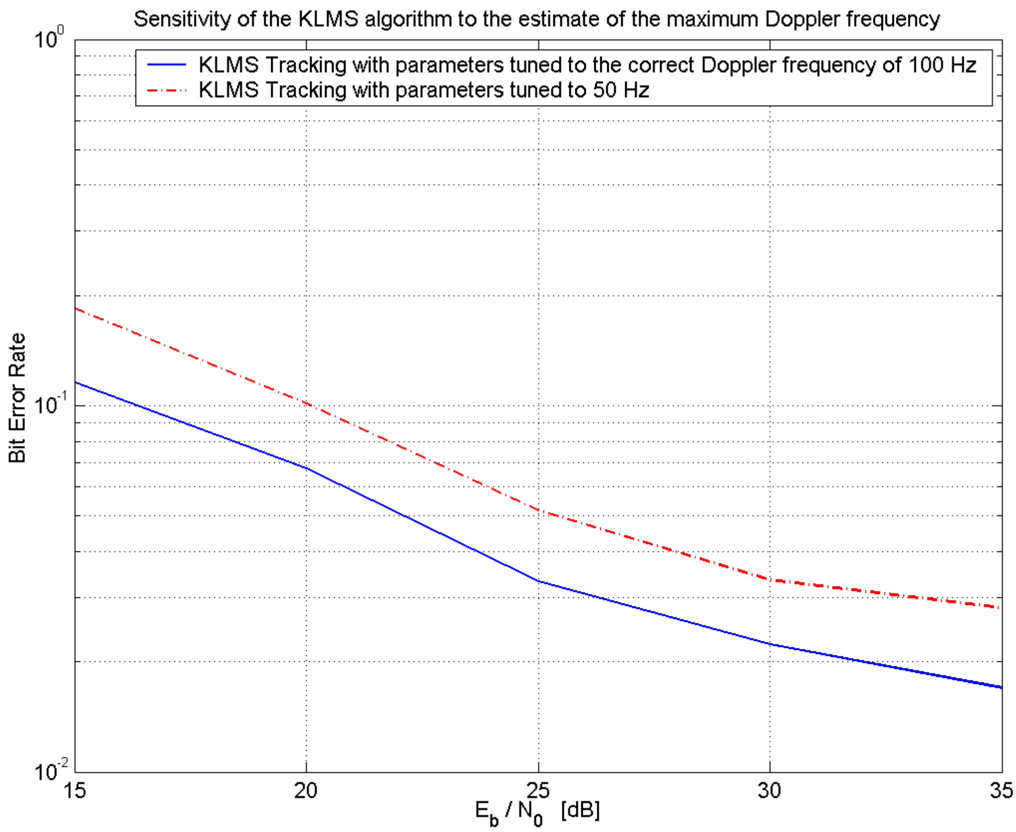

Figure 8.13: Sensitivity of the KLMS channel trackeing algorithm to Doppler frequency estimate

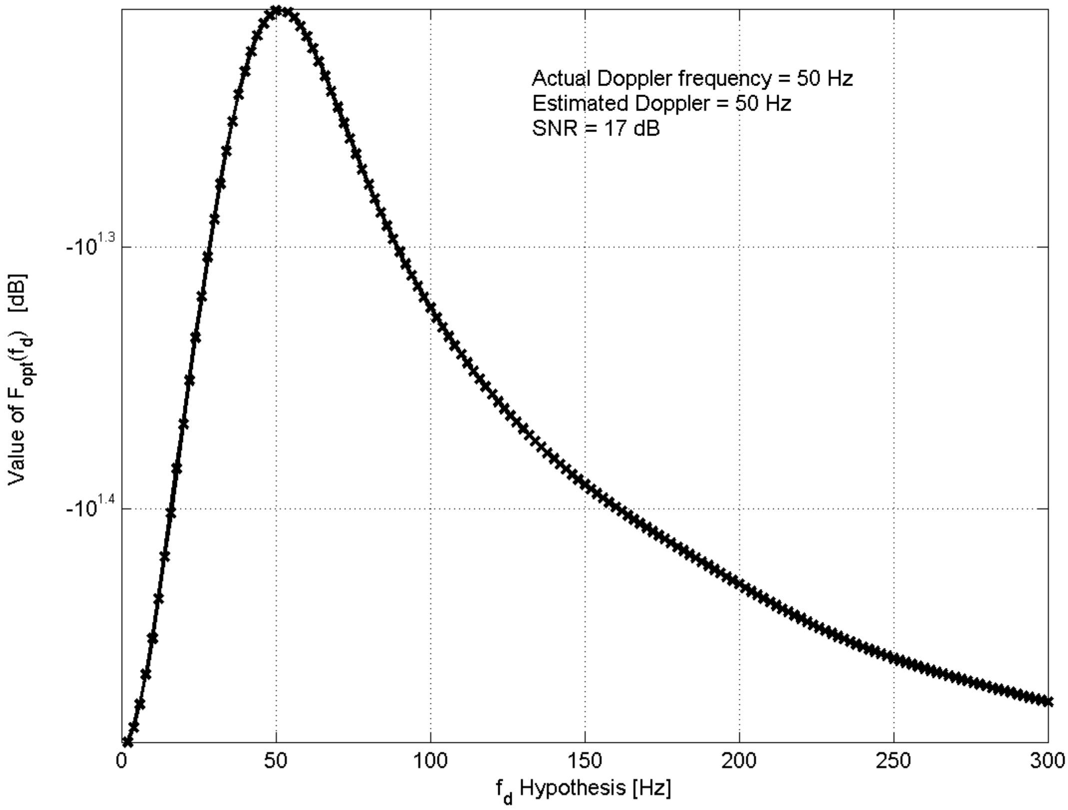

Figure 8.14: Performance of the ML algorithm

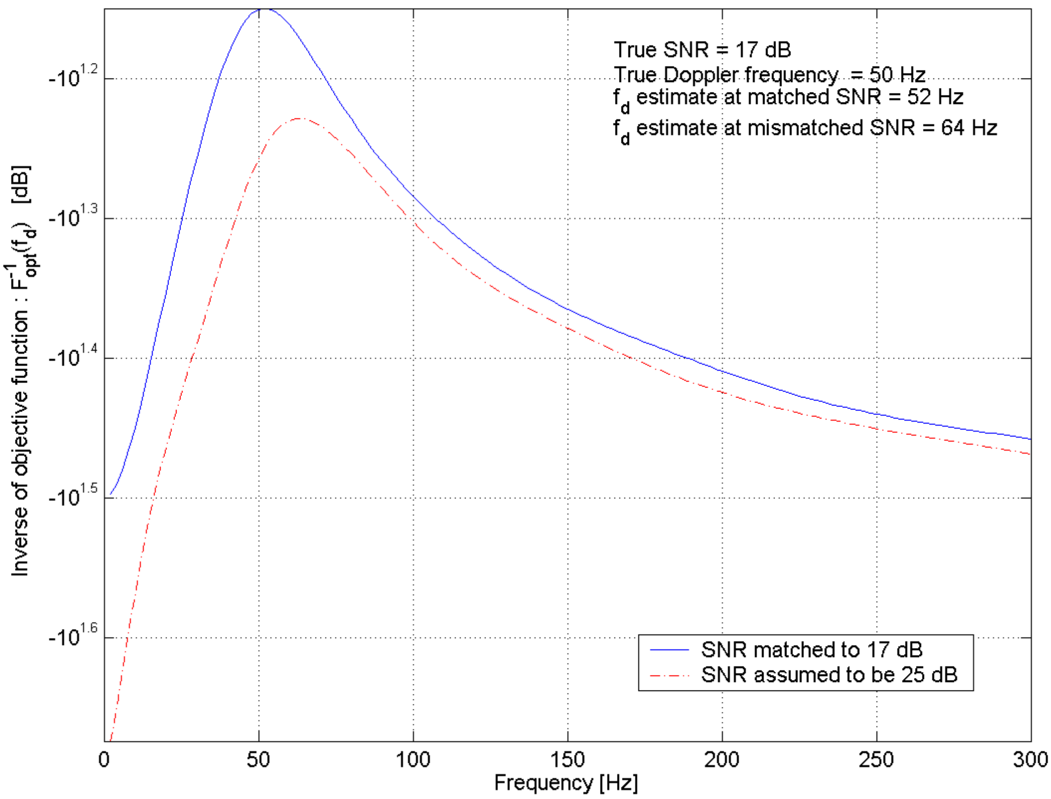

Figure 8.15: ML estimate of Doppler Spread at a mismatched signal-to-noise ratio

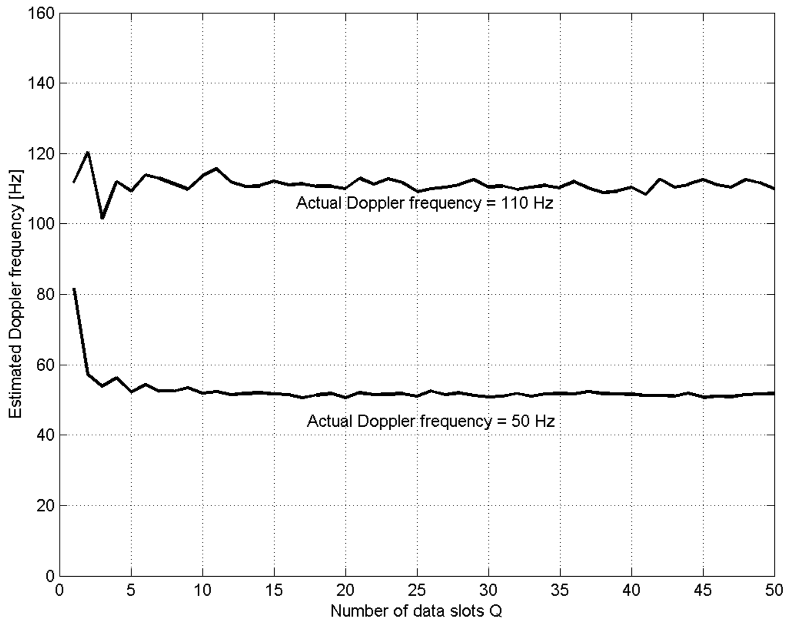

Figure 8.16: Convergence of the ML Doppler estimation algorithm

CHAPTER 9

Conclusions and Future Work

9.1 Conclusions and summary

In this thesis indirect methods for adaptive channel equalization were studied. The problem in focus was the type of Rayleigh fading mobile radio channels encountered in the proposed TETRA digital mobile radio standard. Our major concern has been to estimate the channels under various conditions. The channel estimators must be combined with some equalizers or Viterbi schemes. The combination of a rapidly time - varying channel and decision directed adaptation results in a rather demanding tracking problem. A guiding principle here has been the the use of prior knowledge to tailor estimators to specific needs.

It has been shown that channel estimators can be easily and drastically improved, compared to conventional RLS/LMS tracking, by modeling the time-varying parameters of the channel with simple methods. When the symbol sequence is white, low complexity estimators with high performance can be obtained. A low complexity channel estimator based on simplified internal stochastic modeling of the time varying channel co-efficients was applied to the TETRA uplink channel to demonstrate its superior performance at low complexity. This algorithm combines the computational simplicity of a pure gradient based algorithm and the performance of a Kalman predictor. The adaptation gains can be easily tailored to specific fading rates and noise variances.

Having demonstrated the dependence of the above channel tracking algorithm on the estimate of the maximum instantaneous Doppler spread, a maximum likelihood Doppler estimator was applied on the TETRA uplink. Simulations demonstrate that this Doppler estimator performs well on the given channel conditions, providing fairly accurate estimates of the instantaneous Doppler spread as input information to the channel tracking algorithm.

9.2 Future Work

The improvement of all estimators are limited by properties of the system. In the type of flat fading environments often encountered in mobile radio channels, the possibility of incorrect decisions is high if flat fading occurs frequently. This means incorrect decision must be frequently used in the estimation. In such a situation , the difference between different kinds of estimators would diminish and further performance enhancement would only be possible with both, channel diversity systems and a priori-based channel adaptation at the cost of increased complexity.

Bibliography

[Lin01R] Lindbom, Lars : *Adaptive Equalization for Fading Mobile Radio Channel*, Licentiate dissertation, Technology Dept. Uppsala University, Uppsala Sweden, 1992.

[Lin02R] Lindbom, Lars : *A Wiener Filtering Approach to the Design of Filtering Algorithms with Applications in Mobile radio Communications*; Technology Dept. Uppsala University, Uppsala, Sweden, 1995.

[Lin03R] Lindbom, Lars : *Simplified Kalman estimation of fading mobile radio channels: High performance at LMS computational load*, Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on ,Volume: 3 , 27-30 April 1993

[Lin04R] Lindbom, Lars : *Channel tracking with WLMS algorithms: high performance at LMS computational load*, Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st ,Volume: 1 , 15-18 May 2000.

[Lin05R] Lindbom, Lars : *Tracking of time-varying mobile radio channels - Part I: The Wiener LMS algorithm*, Communications, IEEE Transactions on ,Volume: 49 , Issue: 12 , Dec. 2001.

[Lin06R] Lindbom, Lars : *Tracking of time-varying mobile radio channels. II. A case study*, Communications, IEEE Transactions on ,Volume: 50 , Issue: 1 , Jan. 2002.

[Kra01R] Krasny, Leonid : *Doppler Spread Estimation in Mobile Radio Systems*, IEEE Communications Letters, Vol. 5, No. 5, May 2001.

[Mor01R] Moreli. M., Mengali U. and Vitteta G. M. : *Further results in carrier frequency estimation for transmission over flat fading channels*, IEEE Communications Lett. Vol 2, Dec 1998

[Hay01B] Haykin, S. : *Communication Systems*, John Wiley and Sons, 1998.

[Kre01B] Kreyzig, E. : *Advanced Engineering Mathematics*, John Wiley and Sons, 1996.

[Pro01B] Proakis, J. G. : *Digital Communications*, Mc-Graw Hill, 1989.

[Rap01B] Rappaport, T. S. : *Wireless Communications: Principles and Practice*, Prentice Hall, 1991.

[Jak01B] Jakes, W. C. :*Microwave Mobile Communications*, John Wiley and Sons, 1974.

[Che01R] Chennakeshu, S., Saulnier, G., "Differential Detection of $\pi/4$ shifted DQPSK for Cellular Radio" *IEEE Transactions on Vehicular Technology*, Vol. 42, No. 3, pg 46-57, February 1993.

[Den01R] Dent, P., Bottomley, G. E., Croft, T., "Jakes Fading Model Revisited",*Electronic Letters*, Vol. 29, No. 13, pg 1162-1163, June 1993.

[Jam01R] Jamal, K., Brismark, G., Gudmundson, B., "Adaptive MLSE Performance on the D-AMPS 1900 channel" *IEEE Transactions on Vehicular Technology*, Vol. 46, No. 3, pg 634-641, August 1997.

[Skl01R] Sklar, B., "Rayleigh Fading Channels in Mobile Digital Communication Systems, Part I : Characterisation", *IEEE Communications Magazine*, pg 91-100, July 1997.

[Skl02R] Sklar, B., "Rayleigh Fading Channels in Mobile Digital Communication Systems, Part II : Mitigation", *IEEE Communications Magazine*, pg 101-109, July 1997.

[ETSI01R] TETRA working group : *Terrestrial Trunked Radio (TETRA): Voice + Data; Part II: Air Interface*, ETSI EN 300 392-2 V2.42 (2004-02) European Standard (Telecommunication Series), February 2004

[IEEExplore] *http://ieeexplore.ieee.org*

[TETRA] *http://www.tetramou.org*

[ETSI] *http://www.etsi.org/services_products/freestandard/*

Appendix A

Some important algorithms

Consider the basic channel tracking framework:

$$y(n) = \phi^H(n)\theta(n) + v(n) \quad n = 1, \ldots, N \tag{A.1}$$
$$\phi^H(n) = (d(n) \ldots d(n-m))$$
$$\theta^T(n) = (h_0(n) \ldots h_m(n))$$

The goal of the time-varying parameter estimation problem is to determine a sequence of estimates $\{\hat{\theta}(n)\}_1^N\}$ such that the cumulative squared prediction errors are minimized.

$$\{\hat{\theta}(n)\} = arg\left[Min_{\theta_n}\left(\sum_{n=1}^{N}|\epsilon(n)|^2\right)\right] \tag{A.2}$$

In order to minimize this citerion, parameters of time-varying systems are often eestimated by recursive algorithms, having the structure

$$\hat{\theta}(n) = \hat{\theta}(n-1) + K(n)\epsilon(n) \tag{A.3}$$

$$\epsilon(n) = y(n) - \phi^H(n)\hat{\theta}(n-1) \tag{A.4}$$

The choice of the gain vector sequence differs in different algorithms.

A.1 LMS Algorithm

This recursive algorith corresponds to the following choice of the gain vector:

$$K(n) = P(n)\phi(n) \tag{A.5}$$

$$P(n) = diag(\mu_0(n) \ldots \mu_m(n)).$$

Thus $P(n)$ is a diagonal matrix characterized by the scaling factors $\mu_k(n)$, $k = 0, \ldots, m$. The scalar design variable $\mu_k(n)$ are often chosen as constants having the same magnitude. The matrix $P(n)$ can then be replaced by a scalar μ. The algorithm is then referred to as the *Least mean squares* algorithm. It is sometimes formulated in normalized form with $K(n) = \mu\phi(n)/(1+\mu|\phi(n)|^2)$. However in the particular case when the elements in the regressos consist of symbols having constant modulus, the normalization has no effect since $|\phi(n)|^2$ is then constant. Furthermore, the step-size becomes equal in all directions of the parameter space. With symbols having constant modulus, the LMS algorithm is equivalent to a time-invariant filter.

A.2 Optimal Kalman Estimation

Assume that ω_n, $H(n)$ and the covariance matrix R_e(n) are all known. Then the most accurate linear estimator of $X(n)$ and thus of θ^o(n) based on data up to time $n-1$ is given bythe Kalman predictor:

$$\widehat{X}(n+1) = F(\omega_m)\widehat{X}(n) + K(n)\epsilon(n) \tag{A.6}$$

$$\epsilon(n) = y(n) - \phi^H(n)\widehat{X}(n) \tag{A.7}$$

$$\widehat{\theta}(n) = H\widehat{X}(n) \tag{A.8}$$

$$K(n) = \frac{F(\omega_n)P(n)\phi(n)}{\sigma_v^2 + \phi^H(n)P(n)\phi(n)} \tag{A.9}$$

Here $\epsilon(n)$ is the prediction error and $K(n)$ is the Kalman Gain. This gain depends on the estimate of the state covariance matrix $P(n)$, which is updated every symbol by the Riccati Difference Equation:

$$P(n+1) = F(\omega_n)\{P(n) - P(n)Q(n)P(n)\}F^T(\omega_n) + R_1(n) \tag{A.10}$$

where:

$$Q(n) = \frac{\phi(n)\phi^H(n)}{\sigma_v^2 + \phi^H(n)P(n)\phi(n)} \tag{A.11}$$

$$R_1(n) = E[G(\omega_m)e(n)e^H(n)G^T(n)] = G(\omega_m)R_e(n)G^T(\omega_m) \tag{A.12}$$

sectionThe KLMS Algorithm The Kalman Least Mean Squares algorithm may be summarised in the following equations which represent the on-line computations required by the simplified channel estimator:

$$\epsilon(n) = y(n) - (d(n), \ldots, d(n-m)) \begin{pmatrix} \hat{h}_0(n) \\ \vdots \\ \hat{h}_m(n) \end{pmatrix} \tag{A.13}$$

$$\hat{x}_k(n+1) = \begin{pmatrix} -a_1 & 1 \\ -a_2 & 0 \end{pmatrix} + \frac{\mu(n)}{\sigma_d^2(1+p_{1,1}^{(kk)})} \begin{pmatrix} -a_1 p_{1,1}^{(kk)} + p_{1,2}^{(kk)} \\ -a_2 p_{1,1}^{(kk)} \end{pmatrix} d^*(n-k) sat_\alpha[\epsilon(n)] \tag{A.14}$$

$$\hat{h}_k(n) = \begin{pmatrix} \hat{h}_0(n) & 0 \end{pmatrix} \hat{x}_k(n) \tag{A.15}$$

$$\mu(n) = \beta\mu(n-1) + (1-\beta) \quad 0 \le \beta \le 1 \quad \mu(0) > 1 \tag{A.16}$$

where $p_{1,1}^{(kk)}$ and $p_{1,2}^{(kk)}$ are pre-computed and are given by:

$$p_{1,1} = \frac{\zeta + \sqrt{\zeta^2 - 4a_2^2}}{2} - 1 \qquad p_{1,2} = \frac{a_1 a_2}{1 + a_2 + p_{1,1}} p_{1,1} \tag{A.17}$$

$$\zeta = \frac{\alpha_0 - 2a_2 + \sqrt{(\alpha_0 + 2a_2)^2 - 4a_1^2(1 + a_2)^2}}{2} \tag{A.18}$$

$$\alpha_0 = 1 + a_1^2 + a_2^2 + \gamma_k \tag{A.19}$$

We thus estimate two complex-valued parameters (elements of $\hat{x}_k(n)$) per channel co-efficient $h_k(n)$

A.3 Least squares method: minimum mean square error criterion

$$y(n) = \phi^H \theta + v(n)$$
$$\phi^H(n) = (d(n) \dots d(n-m)) \quad \theta^T = (h_0 \dots h_m)$$

The Least Squares solution to $\sum_{s=1}^{N_{tr}} |y(s) - \phi^H(s)\theta|^2$(the minimum mean square error) is given by:

$$\hat{\theta} = \left[\sum_{s=1}^{N_{tr}} \phi(s)\phi^H(s) \right]^{-1} . \sum_{s=1}^{N_{tr}} \phi(s)y(s). \tag{A.20}$$

Appendix B

Stochastic hypermodels

To describe the channel co-efficients corresponding the FIR Channel model, h_k, we use simplified fading models in the form of marginally stable autoregressive models of order n_D with equal dynamics for all channel taps.

$$\overline{h}_t = \frac{1}{D(q^{-1})} I e_t = \frac{1}{1 + d_1 q^{-1} + \ldots + d_{n_D} q^{-n_D}} . I e_t \tag{B.21}$$

The notation $\overline{h}_t$ is used to denote that [B.21] will not be a perfect description of h_t. Here q_{-1} denotes the backward shift operator and e_t is a white zero - mean random vector sequence with covariance matrix R_e.

An exact representation of fading statistics would require the use of an autoregressive fading model [B.21] of infinite order. We will here describe the special cases of [B.21]

1. **RW**, Random walk modeling, $D(q^{-1}) = 1 - q^{-1}$, results in an LMS algorithm. It is a common first choice when no prior information is available.
2. **FRW** (Filtered Random Walk):

   $$D(q^{-1}) = (1 - q^{-1})(1 - a q^{-1}) \tag{B.22}$$

   with $|a| < 1$, is a useful model in many situations. The special case of an integrated random walk (**IRW** , $a = 1$) will be appropriate if the short term behaviour is well approximated by linear trends.
3. $\mathbf{AR_2}$, (Autoregressive second order model):

   $$D(q^{-1}) = 1 - 2\rho \cos(\omega_0 q^{-1}) + \rho^2 q^{-2} \tag{B.23}$$

   Here, ρ and ω_0 determine the degree of damping and the dominating frequency respectively. For Jakes fading models, a reasonable bandwidth is obtained with $\omega_0 = \Omega_D^o / \sqrt{2}$ where Ω_D^o is a nominal or estimated Doppler frequency.
4. $\mathbf{AR_2 I}$, (Autoregressive and integrating model):

   $$D(q^{-1}) = (1 - 2\rho \cos(\frac{\Omega_D^o}{\sqrt{2}}) q^{-1} + \rho^2 q^{-2})(1 - q^{-1}) \tag{B.24}$$

   This model is useful when some parameters are varying while others are slowly oscillating or are constant. The integrating term $(1 - q^{-1})$ will guarantee an unbiased estimate of the constant parameters when $\mathbf{R}$ is non - singular. The AR2I model is also of use for long range prediction of oscillations around a non-zero mean, which occurs in Rician fading; without an integrating model factor, prediction estimates would be biased towards zero.

5. $\mathbf{AR_{n_D}}$, autoregressive models of order $n_D = 3$ or higher are appropriate when important properties of the covariance function of the time-variation are difficult to match with a few parameters.

Appendix C

The Jakes fading model

The Jakes fading model is a deterministic method for simulating time-correlated Rayleigh fading waveforms and is still widely used today. The model assumes that N equal-strength rays arrive at a moving receiver with uniformly distributed arrival angles α_n, such that ray n experiences a Doppler shift $\omega_n = \omega_M \cos(\alpha_n)$, where $\omega_M = 2\pi f \frac{v}{c}$ is the maximum Doppler shift, v is the vehicle speed, f is the carrier frequency and c, the speed of light.

Using $\alpha_n = 2\pi \frac{n}{N}$, there is a quadrantal symmetry in the magnitude of the Doppler shift, except for angles 0 and π. As a result, the fading waveform can be modelled with $N_0 + 1$ complex oscillators, where $N_0 = \frac{(\frac{N}{2}-1)}{2}$. This gives the fading waveform as

$$T(t) = K\left\{ \frac{1}{\sqrt{(2)}}[\cos(\alpha)+i\sin(\alpha)]\cos(\omega_M t+\theta_n)+\sum_{n=1}^{N_0}[\cos(\beta_n)+i\sin(\beta_n)]\cos(\omega_n t+\theta_n) \right\} \tag{C.25}$$

where K is a normalization constant, α (not the same as α_n) and β_n are phases, and θ_n are initial phases, usually set to zero. Setting $\alpha = 0$ and $\beta = \frac{\pi n}{N_0+1}$ gives zero cross correlation between the real and imaginary parts of $T(t)$.

To generate multiple uncorrelated waveforms, Jakes suggests using phases $\theta_{n,j} = \beta_n + 2\pi \frac{(j-1)}{(N_0+1)}$, where $j = 1 \ldots N_0$ is the waveform index. However this gives almost uncorrelated waveforms j and k only when $\theta_{n,j} - \theta_{n,k} = i\pi + \frac{\pi}{2}$, for some integer i. Otherwise, the correlation between certain waveform pairs can be significant. A remedy for this problem is achieved by reformulating the Jakes model in terms of slightly different arrival angles.

To provide quadrantal symmetry for all Doppler shifts, which leads to equal power oscillators, the following arrival angles are used: $\alpha_n = 2\pi \frac{(n-0.5)}{N}$. This leads to the model:

$$T(t) = \sqrt{\frac{2}{N_0}}\left\{ \sum_{n=1}^{N_0}[\cos(\beta_n) + i\sin(\beta_n)]\cos(\omega_n t + \theta_n) \right\} \tag{C.26}$$

where $N_0 = \frac{N}{4}$ and the normalization factor $\sqrt{\frac{2}{N_0}}$ gives $E(T(t)T^*(t)) = 1$.