NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

iii-iv

This reproduction is the best copy available.

UMI®

Feedforward Symbol Synchronization Techniques for Digital Receiver

Yupeng Yan

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

November 2003

Your file Votre référence
ISBN: 0-612-93537-X
Our file Notre référence
ISBN: 0-612-93537-X

ACKNOWLEDGEMENTS

I would first like to thank my thesis supervisors, Dr. Wei-Ping Zhu and Dr. M. N. S. Swamy, for their guidance and support throughout the course of this study. Without their direction and advice both in terms of research and in the writing of the thesis, the completion of this work would not have been possible. I would like to thank Dr. M. O. Ahmad for helpful discussions during our research meetings. I am also grateful to NSERC and Micronet, a National Network of Centers of Excellence, Canada, for the financial support given to my supervisors to carry out this project. Last but not least, I would like to thank my mother and my father and all my family, for their consistent support and encouragement.

LIST OF FIGURES

LIST OF SYMBOLS

*	Convolution operator
α	Roll-off factor
a_k	Transmitted k-th symbol
A_n	Averaged real values
B_n	Averaged image values
$\hat{A}_n$	Averaged real values after Kalman filter
$\hat{B}_n$	Averaged image values after Kalman filter
$C(n)$	Measurement matrix
$e(n)$	State error vector
E/No	Signal to noise ratio
$F(n)$	Transition matrix
$g_R(t), G_R(\omega)$	Impulse and frequency response of the receiver matched filter
$g_T(t), G_T(\omega)$	Impulse and frequency response of the transmitter matched filter
$g(t), G(\omega)$	Impulse and frequency response of the overall channel
$H(z)$	The response of lowpass filter
$K(n)$	Kalman gain of time n
$Q(n)$	Correlation matrix of process noise vector $w_1(n)$

$R(n)$	Correlation matrix of process noise vector $w_2(n)$
$P(n)$	Correlation matrix of $e(n)$
$r(t)$	received complex baseband signal after matched filter
$r_I(t)$	received Inphase baseband signal after matched filter
$r_Q(t)$	received Quadrarure baseband signal after matched filter
$r'(t)$	received complex baseband signal before matched filter
$n'(t)$	Additive white gaussian noise
$n(t)$	Filtered AWGN noise
$s'(t)$	Transmitted signal
T	Symbol duration
T_s	Sample duration
$\Lambda_L(\tau)$	Log Maximum Likelihood function
τ_{ML}	estimated optimum timing
x_k	Squared received signal r_k
X_m	Fourier transform of x_k
$u_I(t)$	Inphase real output of the low-pass filter
$u_Q(t)$	Quadrature real output of the low-pass filter
$v_I(t)$	Inphase image output of the low-pass filter
$v_Q(t)$	Quadrature image output of the low-pass filter
$w_1(n)$	Process noise

$w_2(n)$	Process noise
$x(n)$	State vector
$y(n)$	Observation vector
$y_I(t)$	Inphase complex output of the low-pass filter
$y_Q(t)$	Quadrature complex output of the low-pass filter
$\tilde{\phi}_n$	Kalman-filtered timing estimation
$\hat{\phi}_n$	timing estimation from basic algorithm

LIST OF ABBREVIATIONS

AWGN	Additive White Gaussian Noise
BER	Bit Error Zrate
BPF	BandPass Filter
CR	Carrier Recovery
DA	Data-Aided
DD	Decision Directed
DFT	Discrete Fourier Transform
DPLL	Digital Phase Locked Loop
DSP	Digital Signal Processing
FB	Feedback
FF	Feedforward
FH	Frequency Hopping
IF	Intermediate Frequency
LPF	Low-Pass Filter
ML	Maximum Likelihood
MPSK	M-ary Phase Shift Keying
NCO	Numerical Controlled Oscillator
NDA	Non Data-Aided
PLL	Phase Locked Loop

QAM	Quadrature Amplitude Modulation
STR	Symbol Timing Recovery
SNR	Signal to Noise Ratio
TDMA	Time Division Multiple Access
VCO	Voltage Controlled Oscillator

Chapter 1

Introduction

In the last few decades, digital communication has been undergoing rapid development. This development is principally attributed to the progress made in microelectronics, which allows most of the tasks in a communication receiver to be implemented in a digital manner. The present VLSI and ASIC technology have made it possible for a receiver to be integrated on a small chip. The cost of a digital receiver thus depends heavily on the rate of the transmitted signal as well as the local sampling rate. It is of crucial importance to reduce the sampling rate as much as possible in order to decrease the receiver cost. Symbol synchronization is one of the core tasks at the front end of a receiver. The received continuous-time signal is sampled in the receiver and the sampling clock should be synchronized with the symbol clock in the transmitter such that the transmitted information symbols can be correctly detected. If the local sampler is not synchronized with the in-coming signal, symbol synchronization or symbol timing recovery is required. It is well known that the mismatch between the sampling clock in the transmitter and that in the receiver would cause inter-symbol interference (ISI), which may deteriorate dramatically the bit-error-rate (BER) performance of the communication system. Therefore, it is imperative to develop efficient symbol synchronization

techniques, especially those that are very suitable for a digital implementation.

1.1. A Brief Review

Many symbol synchronization techniques have been proposed so far to combat the inter-symbol interference [1-17]. In conventional receivers, the symbol synchronization is performed using a feedback loop that controls the phase of the sampling clock. In 1960's , feedback symbol synchronization techniques were implemented by analog circuits in which the phase locked loop technique is utilized. In this technique, the received signal is sent to a nonlinear device, where a signal representing the timing error can be generated. This error signal is then feed into an analog voltage controlled oscillator (VCO) to control the phase of the sampling clock. Later on, with the development of the VLSI technology, it became possible to implement symbol synchronization by digital devices such as digital signal processor (DSP), where the traditional VCO is replaced with a numerically controlled oscillator (NCO) on the phase-locked loop. The digital realization of the synchronization task has exhibited many advantages such as the reduction of the receiver size, and the high precision and reliability of the timing recovery which is, unlike the analog realization, less sensitive to the temperature variation. In recent years, therefore, the investigation of timing recovery has been focusing mainly on the implementation of a fully-digital receiver in which an intermediate frequency (IF) input signal is sampled first with a fixed system clock, and then processed in a digital format [4] -[17].

In addition to the feedback recovery technique, another broad class of symbol synchronization methods is the so-called feedforward synchronization, which is known to be ideally suited for implementation by DSP or ASIC chips. In the feedforward synchronization, the symbol timing offset is first estimated and then exploited for interpolation among the signal samples in order to reconstruct the original symbols transmitted by the transmitter. The feedforward scheme is also found very suitable for burst-mode communication systems because of its short acquisition time in capturing the symbol timing of the received signal. Compared to continuous communication, a burst communication system only transmits tens to hundreds of symbols without interruption and therefore, it needs a rapid synchronization algorithm. The transmission of the signal in the burst mode very often appears in wireless communications. The time division multiple access (TDMA) system, which enables multiple users to communicate with each other through different time slots over a common transmission channel, is one of such example. The TDMA system has been adopted by a number of communication standards such as GSM, DAMPS and IEEE 802.11b. The frequency hopping communication, which is widely used in military systems, is another example of burst-mode communication. As such, there has been a growing research interest in feedforward timing recovery techniques [10-17].

The digital square timing recovery algorithm proposed by Oerder and Meyr [10] is probably one of the earliest and most efficient feedforward recovery algorithms. It extracts the timing information from the squared signal by computing the spectral

component at the symbol rate, yielding an unbiased estimate of the timing phase. The Oerder and Meyr algorithm enables a VLSI realization of the digital receivers, which operate on a sampled input signal without any feedback to the sampling device. In [11], the Oerder and Meyr algorithm has been used for symbol synchronization in 16-QAM (Quadrature Amplitude Modulation) TDMA receivers, showing a satisfactory synchronization performance in both the AWGN (Additive White Gaussian Noise) and slow Rayleigh fading environments, when a sampling rate of 4 samples/symbol was used. More recently, a feedforward synchronization method has been proposed for the simultaneous symbol timing and carrier phase estimation [12]. In this method, the auto-correlation of the preamble in the received signal is exploited to compute the sampling time mismatch, leading to a fast estimation of the symbol timing error. This method requires only two samples per symbol period and is very suitable for the DSP/ASIC implementation. As this method needs a preamble inserted in each data frame, it belongs to the category of the data/preamble aided feedforward symbol synchronization. Some of the other feedforward synchronization techniques can be found from the recent literature [13] – [17].

1.2. Motivation and Objective

In order to reduce the implementation complexity of the timing estimation and interpolation, the lowest possible sampling rate would be preferred provided that the

targeted signal-to-noise ratio can be met. Although many new symbol synchronization techniques have been proposed in the recent literature, there is little work published that considers non-data-aided (NDA) feedforward symbol synchronization at a very low oversampling rate for burst communications. Although the Oerder and Meyr algorithm makes it possible to compute the timing estimate directly without using a feedback loop, and has reduced the implementation complexity for the receiver, yet it requires at least four samples/symbol sampling rate. This may be critical for some applications, where the cost of receiver depends heavily on the local sampling rate such as in a mobile cellular telephone. The auto-correlation algorithm [12] is able to perform a fast timing estimation at 2 samples per symbol, but it needs a preamble in the transmitted signal to help estimate the timing mismatch. Driven by the need to develop simple, cheap and efficient timing offset detectors, the study of the feedforward synchronization techniques is highly motivated. The objective of this thesis is to develop a few efficient feedforward timing recovery schemes that use two samples/symbol sampling rate, with emphasis on a fast and accurate estimation of the timing offset. As a part of the research goal, the implementation issues related to the proposed synchronization techniques are also investigated.

1.3. Organization of the Thesis

The rest of the thesis is organized as follows.

Chapter 2 provides an overview of some of the existing symbol synchronization methods. Firstly, the synchronization issues in a typical digital communication system, including carrier and symbol synchronizations, are addressed with emphasis on the symbol synchronization technique and its categorization. Secondly, two broad classes of symbol synchronization methods, namely, the feedback and feedforward schemes, are described, showing both their advantages and weakness, as well as the importance of the feedforward method in burst mode communication systems. Thirdly, a more detailed review of both the feedback and feedforward synchronization techniques is given, indicating the research focus of the last decade in the area.

In Chapter 3, an in-depth study of a class of feedforward symbol timing recovery method for digital receivers is carried out. The research focus is on the development of a symbol timing estimation algorithm that uses two samples/symbol sampling rate and on the performance analysis. The proposed algorithm consisting of modulation, filtering, squaring and averaging operations on the sampled baseband signals is analyzed in terms of the mean and variance of the estimated symbol timing offset. A simulation study of the proposed algorithm is also carried out with various simulation results for different parameters such as the rolloff factor, the length of the estimation interval and the order of the lowpass filter used in the algorithm. Some of the implementation issues of the proposed method are also addressed in this chapter.

Chapter 4 presents a few improved algorithms for the estimation of the timing offset. First, a scheme using both the in-phase and the quadrature baseband signals is proposed in order to reduce the variance of the timing estimate. Then, a Kalman filtering-based post-processing strategy is considered for further smoothing the estimated timing, for which a few simple yet efficient smoothing schemes are proposed and simulated, resulting in a significantly reduced variance of the timing estimate in comparison to the basic timing estimation algorithm presented in Chapter 3. The computational complexity of the improved timing estimation algorithms is also discussed.

Chapter 5 concludes the thesis by summarizing some of the research results and providing some suggestions for future study.

Chapter 2

Overview of Symbol Synchronization Techniques

In this chapter, a typical digital communication system that comprises some major function blocks in both the transmitter and the receiver is first described, with which the synchronization issue is addressed. Then, some of the principal symbol synchronization techniques existing in literature are reviewed, showing their strengths and drawbacks. The emphasis of this chapter is placed on the category of the synchronization algorithms that can be implemented digitally, based on which some efficient feedforward synchronization techniques are developed in the succeeding chapters.

2.1. Synchronization Issues in Communication Systems

2.1.1. A Typical Synchronous Communication System

Fig 2.1 shows a typical digital communication system that contains major constituent blocks in the transmitter and the receiver. The source encoder in the transmitter is to perform signal/data compression to create properly a binary sequence that is suitable for further processing and transmission. The channel encoder is to add some redundancy bits

into the binary sequence for the purpose of correcting bit errors due to transmission. With channel encoding which is also called forward error control (FEC) coding, the BER performance of a communication system can be significantly enhanced. The modulator does the baseband modulation, which typically includes the bit-to-symbol mapping, the in-phase (I) and quadrature (Q) signal modulation and pulse shaping. The radio frequency (RF) modulation is usually the last step in the transmitter, which converts the analog baseband signal to the high frequency passband signal for wired or wireless transmission.

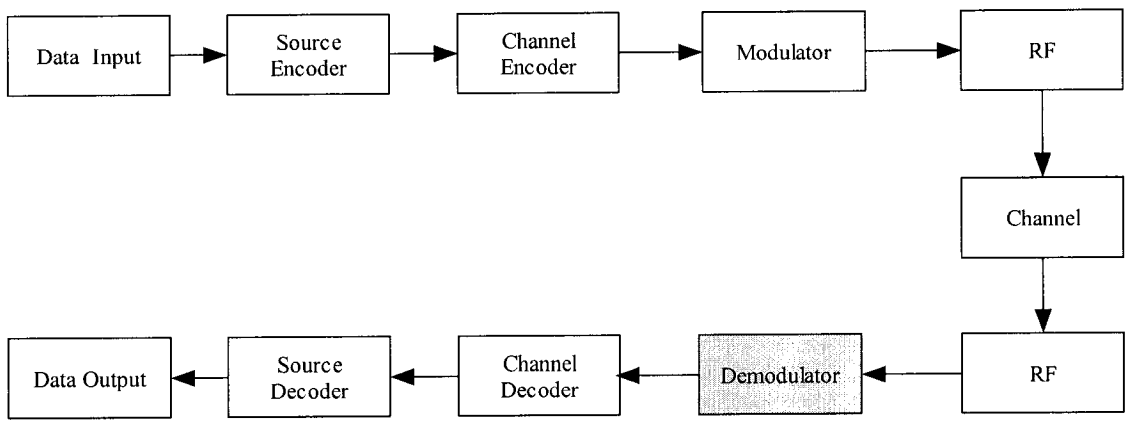


Fig 2.1. Basic structure of a digital communication system

In the receiver side, an inverse process of coding and modulation carried in the transmitter is performed, namely, the received analog signal is first demodulated by radio frequency in the analog domain and then converted to a digital baseband signal in the demodulator for synchronization, I and Q demodulation and symbol detection, etc. The demodulated baseband signal is then mapped back to a binary stream for channel decoding and source decoding.

As there might be a small frequency error between the carrier frequency in the transmitter

and that in the receiver, causing the received I and Q signals to be rotated at a constant rate, a carrier synchronization is usually required in a coherent receiver in addition to the symbol synchronization. The carrier frequency synchronization is to estimate the frequency offset and instantaneous carrier phase, and then use them for the correction of the signal constellation. In this study, we deal with the symbol synchronization only.

2.1.2. The Classification of Symbol Synchronization

Many symbol synchronization algorithms have been developed for a digital receiver. They can be divided into two broad categories, the feedback synchronization and feedforward synchronization. Under each category, the synchronization methods can be further classified as independent symbol synchronization, and joint carrier recovery (CR) and symbol synchronization. Moreover, irrespective of whether it is the independent feedforward or the independent feedback synchronization, the symbol synchronization algorithm belongs to one of the three categories, namely, data-aided (DA) method, non-data-aided (NDA) method and decision-directed (DD) method.

Data-aided (DA) method uses a predefined training sequence called the preamble to aid the timing recovery. This method has an advantage of reducing the synchronization time. But using a preamble would occupy extra bandwidth, leading to less efficient utilization of the channel resource.

Non-data-aided (NDA) method estimates the timing information from the transmitted

signal itself without using a training sequence/preamble. Therefore, the channel resource is fully exploited. However, the performance of the NDA symbol synchronization techniques is generally lower than that of the data-aided techniques.

Decision-directed (DD) method makes use of the output of the receiver's decision device to aid the symbol synchronization. This technique also has a short acquisition time while rendering the receiver to be more complicated due to the involvement of a decision device.

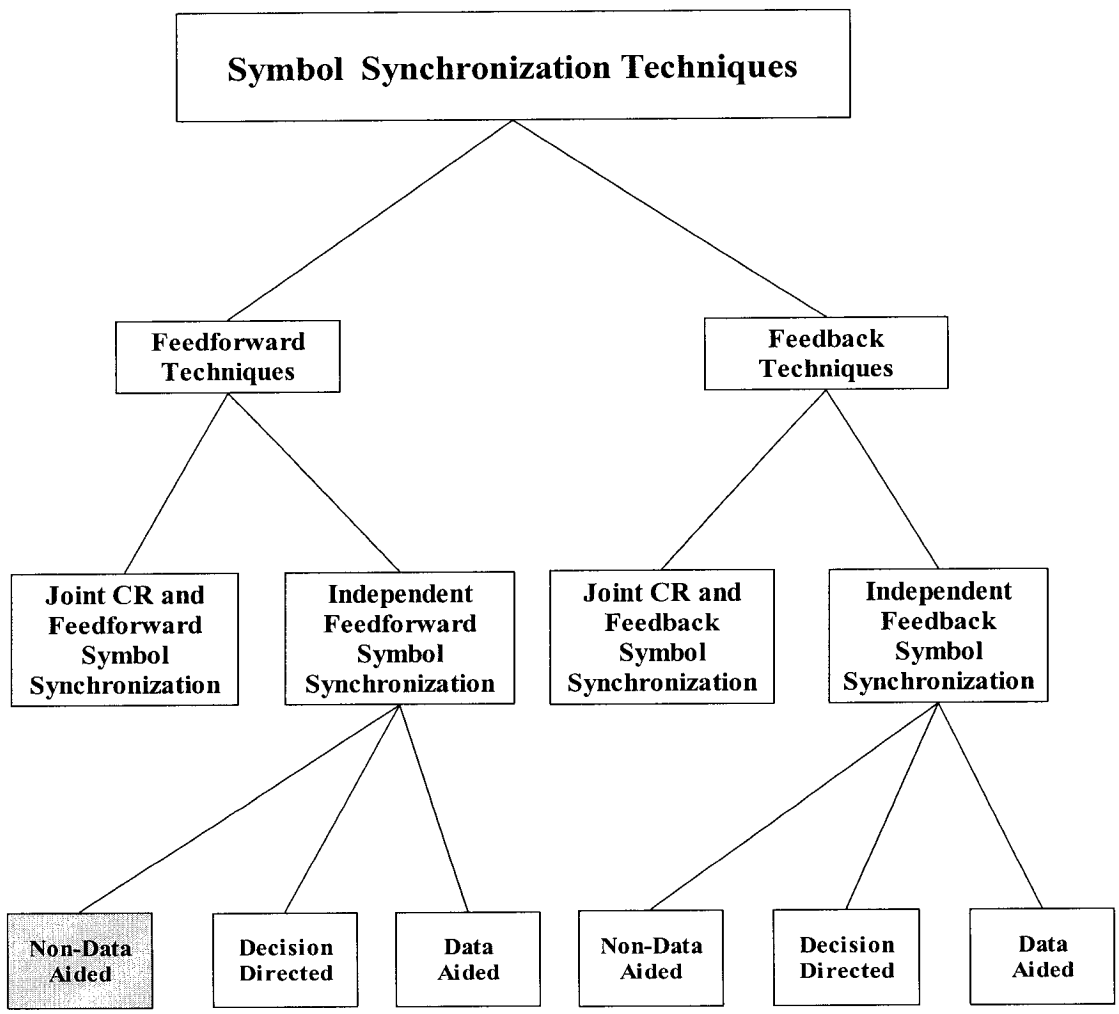


Fig 2.2. Classification of symbol synchronization

Fig.2.2 illustrates a tree diagram for the classification of the symbol synchronization techniques. The major difference between the feedback and feedforward methods lies in that the former uses a closed-loop system or a phase-locked loop to control the phase of the sampler and the latter is an open loop system that uses a free-running sampler without requiring a feedback loop. It is known that the feedforward scheme is well suited for a digital implementation with DSP or ASIC chips. Therefore, we focus on the category of feedforward and non-data-aided synchronization technique in this study.

2.1.3. Feedforward Synchronization and Burst Mode Communication

Typical applications of burst communication include time division multiple access (TDMA) digital celluar radio systems such as the GSM, DAMPS and IEEE 802.11, and frequency hopping (FH) communication systems. In TDMA systems, each user transmits information in the allocated time slot. All users can receive the transmitted information from each user and should be able to determine whether the information was sent to them. In FH communication systems, the carrier frequency for each user varies in a pseudo-random way. Both the TDMA and FH systems belong to burst-mode communication system. For example, in FH systems, user's information is broken into bursts and transmitted at different carrier frequencies. These bursts are, in general, packets of tens to hundreds of symbols. Similarly, in TDMA systems, each user's data are packed and distributed to different time slots, forming a burst mode communication

system.

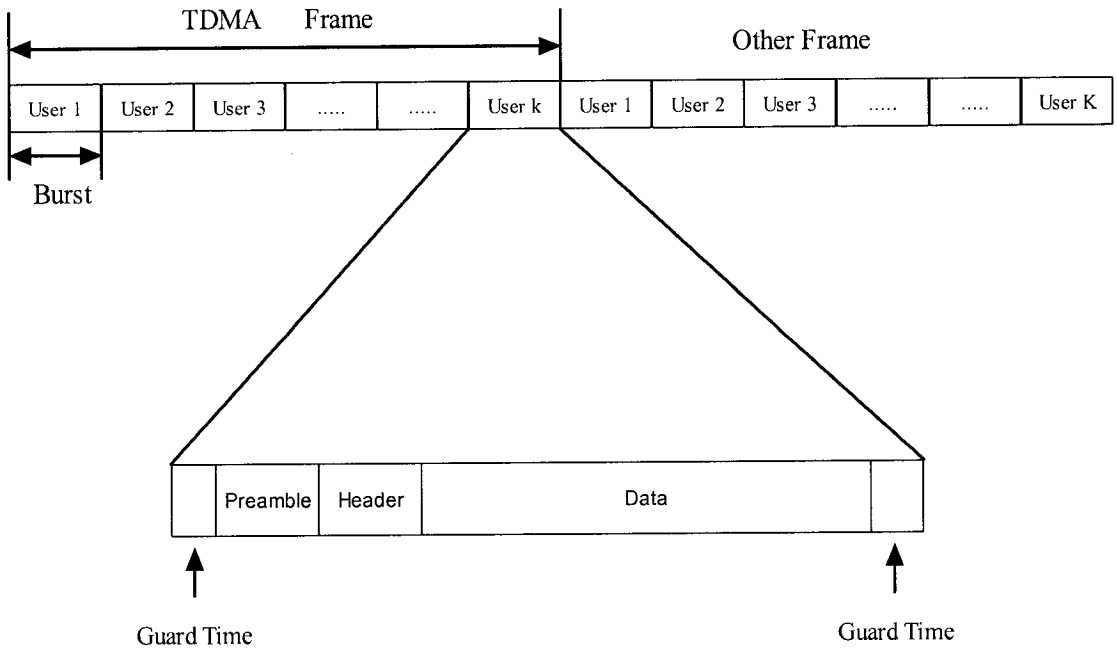


Figure 2.3. TDMA time slot architecture

Figure 2.3 shows the time slot architecture of a TDMA system. Note that guard times are usually required at the beginning and the end of each slot to avoid the overlap between the adjacent bursts. The use of burst mode transmission will increase the complexity of the demodulator compared with the continuous mode system.

Generally, each burst contains a preamble or a predefined sequence, header and data, where either the preamble or data can be used for synchronization, i.e., symbol timing and carrier recovery, depending on whether data-aided or non-data-aided synchronization method is used. Regardless of the type of synchronization scheme employed, the

synchronization should be completed within the burst time. Therefore, a fast acquisition of symbol timing information is highly desirable. In comparison to the feedback scheme, a feedforward algorithm usually provides a much faster timing estimation. As such, feedforward synchronization is favorably considered in burst mode communication systems. A more detailed review of feedback and feedforward algorithms will be provided in the following sections.

2.2. Feedback Timing Recovery Techniques

2.2.1. A Example of Traditional Feedback Synchronization

Fig 2.4 shows a typical PSK (phase shift keying) demodulator with feedback symbol and carrier synchronization. The symbol synchronization block is detailed in Fig. 2.5, where the phase-locked loop is used for the symbol-timing tracking. The PLL-based error tracking synchronization is a traditional synchronization technique. The received and referenced signals are input to a nonlinear processing block, which can calculate the timing error signal. The error signal is passed through a loop filter to get a lower variance and then to drive a voltage controlled oscillator (VCO) to generate the symbol timing clock. This clock is used to sample the received signal taken at the instant of maximum eye opening point. The timing error detector (TED) in Fig. 2.5 is employed to determine if the current sampling clock is advanced or retarded to the right symbol timing phase. In

the following subsections, we will investigate a few typical feedback algorithms.

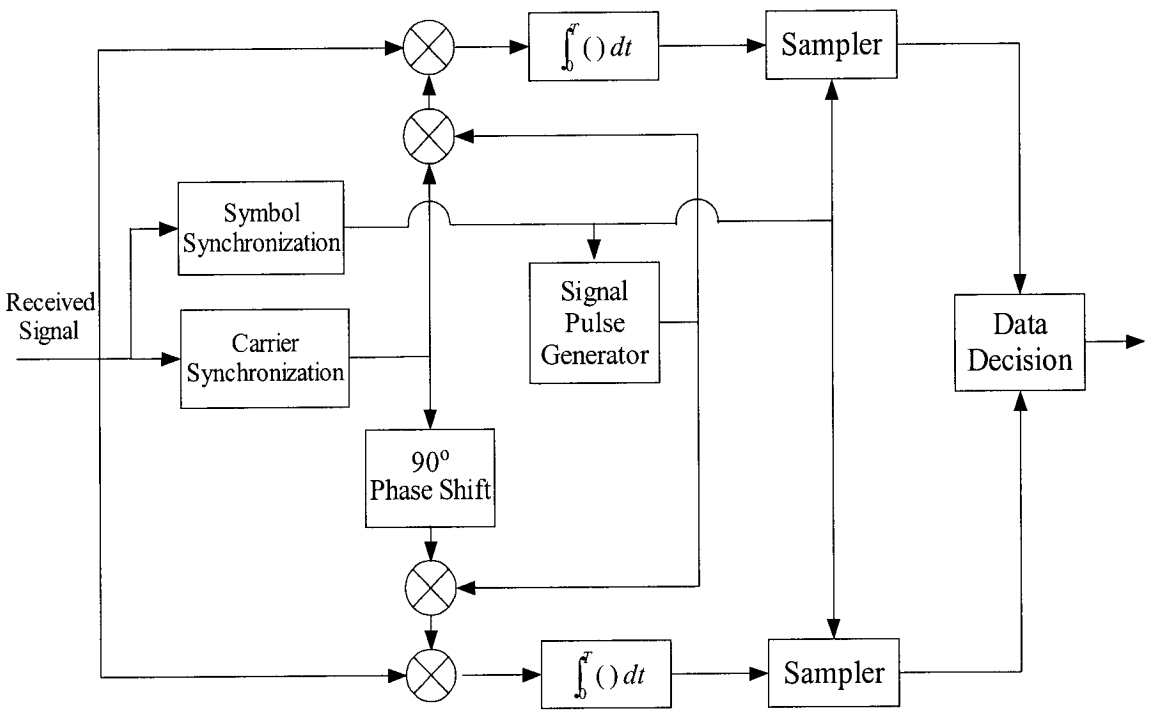


Fig 2.4. The structure of traditional PSK demodulator with synchronization

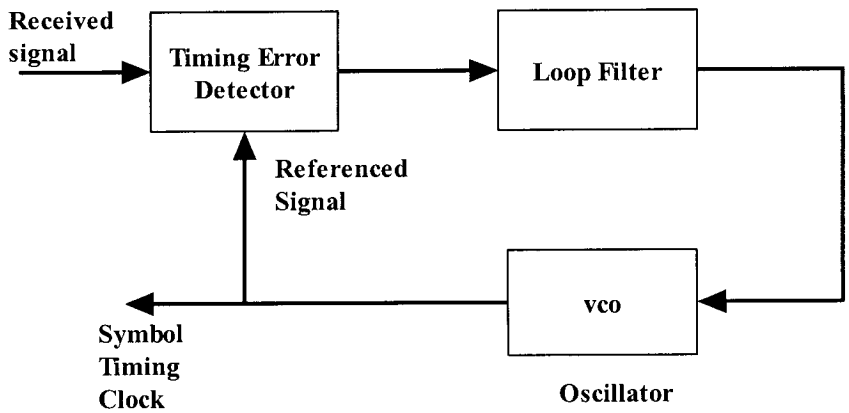


Fig 2.5. Basic PLL based Error Tracking Synchronization

2.2.2. Early-Late Gate Symbol Synchronization

Early-late Gate synchronization exploits the symmetry character of the received signal. Because of the noise, it is difficult to sample at exactly the peak point of the pulse. However, one can sample the signal at an early point (sampling time t = peak point - δ), and a late point (t = peak point + δ) as shown in Fig. 2.7. Obviously, the absolute values of the samples at the early and late points are smaller than the samples at the peak point.

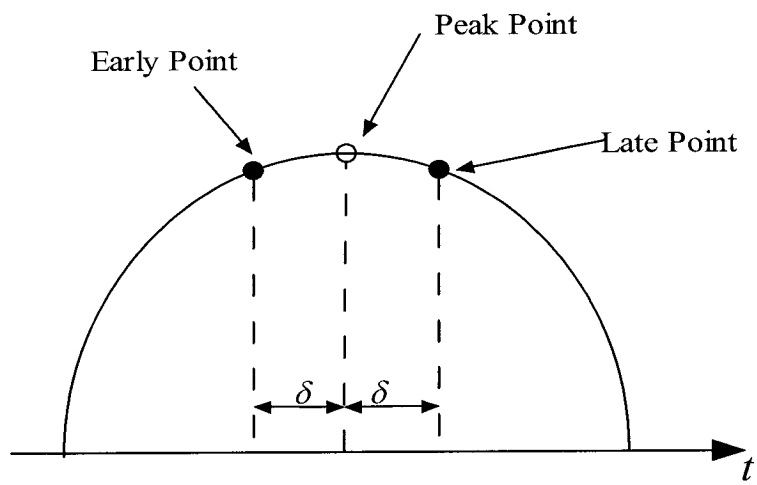


Fig 2.6. Peak point sample and early & late point samples

Fig. 2.7 shows the architecture of the early-late gate symbol synchronizer. The received signal is multiplied with the generated symbol waveform and integrated over the symbol interval T. The signal is sampled ahead of the estimated optimum timing in the top branch

and it is sampled behind the optimum sampling instant in the bottom branch. The early and late sampled signals are squared in the square law device and compared to form an error signal. The error signal is then sent to the loop filter in order to reduce its variance. The smoothed error signal drives VCO to create the desired optimum sampling clock. Using the optimum sampling clock, a symbol waveform can be generated, which is fed back to the received signal to compute the new correlation function for further tracking the sampling phase.

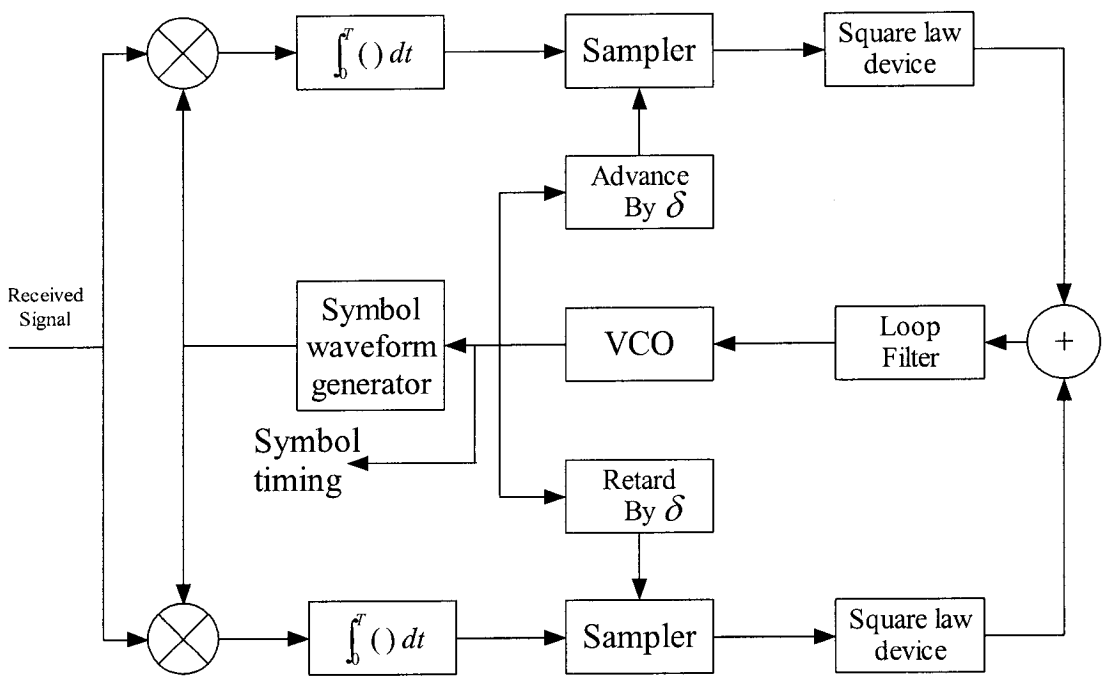


Fig 2.7. Early-late symbol synchronization

2.2.3. Discrete-Time Feedback Symbol Synchronization

Over the last decade, the investigation of feedback symbol synchronization has focused on the implementation of a fully-digital receiver with a digital feedback timing recovery algorithm. As shown in Fig. 2.8, the input signal is first sampled with a free-running fixed clock. The timing error detector (TED) is employed to estimate the timing phase error, which will be smoothed by the loop filter. The smoothed timing error is then used to control an interpolator. The interpolator is to reconstruct the right symbol from the signal samples. As the feedback algorithm needs a longer period of time, usually hundreds of symbols, to get the initial phase, it is not suited for burst mode communication systems especially with short bursts.

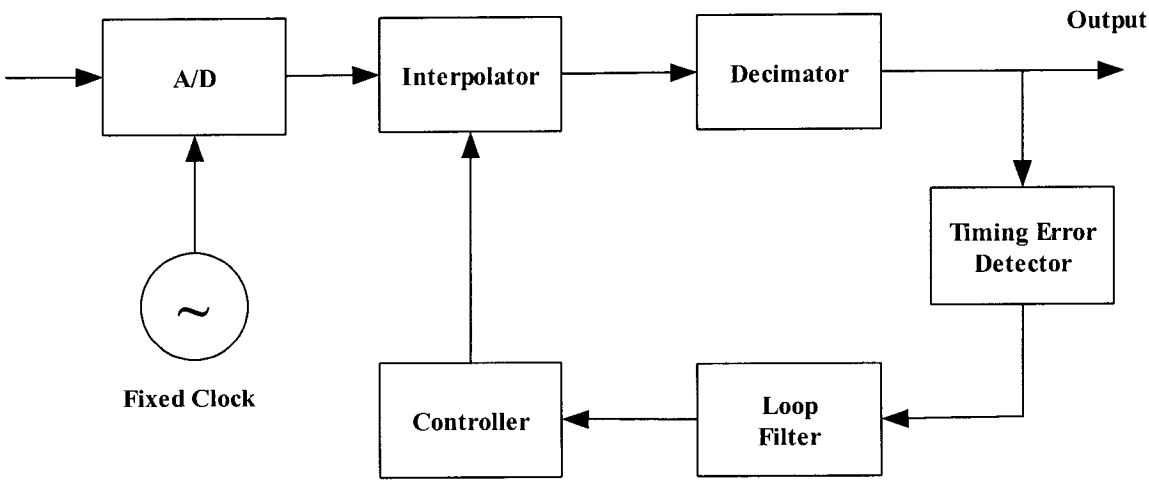


Fig 2.8. Fully digital feedback symbol synchronization

2.3. Feedforward Timing Recovery Techniques

There has been a growing interest in feedforward recovery techniques. Since a feedforward scheme is able to achieve a fast estimation of the timing without requiring a feedback control loop, it is very suitable for burst-mode communication systems.

2.3.1. General Structure of Feedforward Synchronization

Fig. 2.9 shows a fully-digital implementation of NDA feedforward timing recovery. First, the input analog baseband signal is sampled by a free-running oscillator. The sampled signal is stored in data buffer to wait for interpolation. The TED is to estimate the symbol timing phase from the sampled signal. This timing phase, measured as a fraction of the symbol duration, is then used to determine the interpolator coefficients. The interpolator, usually implemented as an FIR filter, reconstructs the optimal signal sample near the maximum eye-open instant. The interpolator coefficients should be updated on a symbol-by-symbol basis. The precision of the interpolation depends on the timing estimate and the number of samples used for the interpolation.

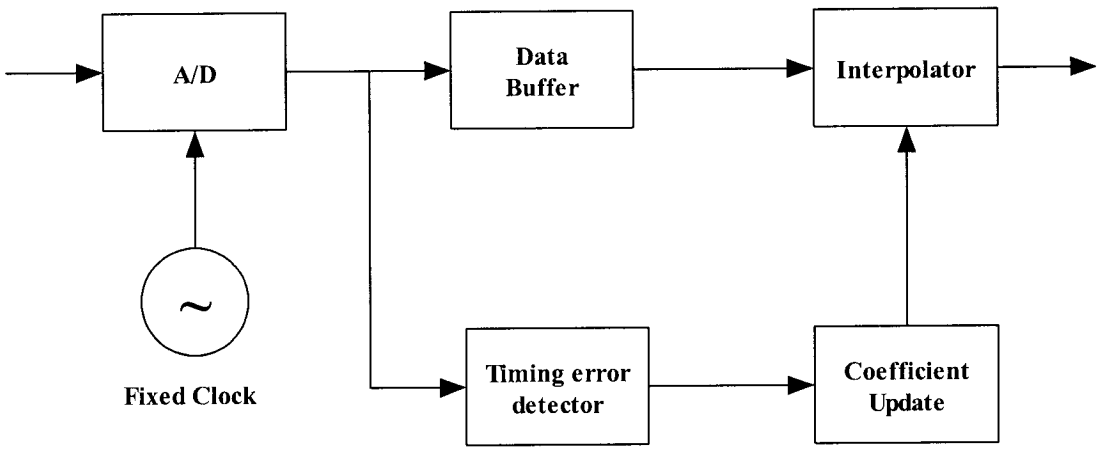


Fig 2.9. Digital Feedforward Timing Recovery Scheme

The use of a free-running oscillator has avoided the VCO in the analog front end and hence reduced the complexity of the receiver. It also allows for a fast acquisition of the symbol timing phase and therefore, it is well suited for high bit-rate communication receiver.

2.3.2. Maximum Likelihood (ML) Feedforword Symbol Synchronization

Fig. 2.10 shows a simplified baseband transmission model which contains the transmit filter $G_T(\omega)$, the AWGN channel and the receive filter $G_R(\omega)$. The input symbols a_k are sent to the transmit filter for pulse shaping to create a baseband analog signal. The

receive filter serves as the matched filter, which is to pre-process the received signal plus noise in the front end of the receiver. The overall frequency response for the channel can be written as

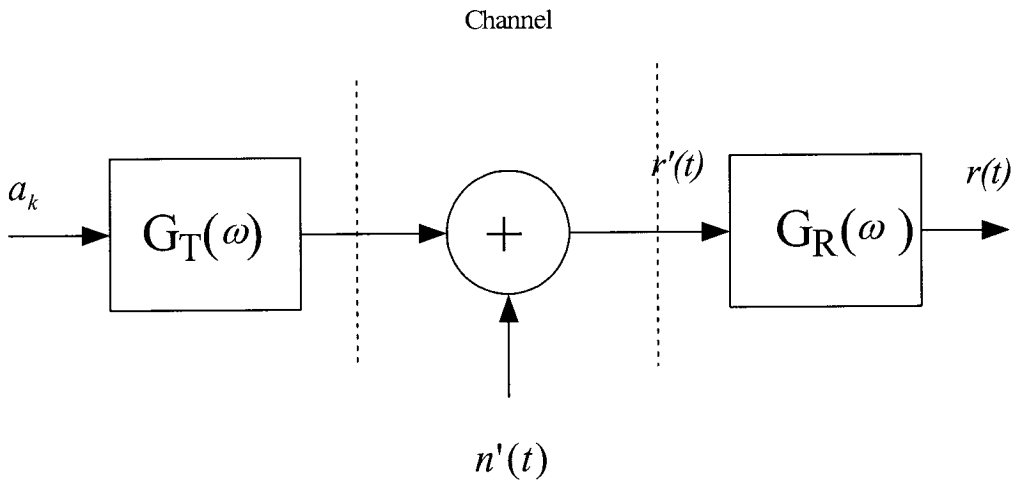


Fig 2.10. Equivalent Baseband Model of digital transmission

$$G(\omega) = G_T(\omega)G_R(\omega) \tag{2-1}$$

whose time-domain impulse response can be expressed as the following convolution

$$g(t) = g_T(t) * g_R(t). \tag{2-2}$$

Very often, $G_T(\omega)$ as well as $G_R(\omega)$ are designed to be a root raised cosine filter such that the overall frequency response $G(\omega)$ is of the raised cosine characteristic, which is well known to be very efficient for signal transmission.

Clearly, the signal arriving at the receive filter can be written as

$$r'(t) = s'(t) + n'(t) \tag{2-3}$$

where

$$s'(t) = \sum_k a_k g_T(t - \varepsilon - kT) \tag{2-4}$$

represents the transmitted baseband signal. In (2-4), ϵ is the symbol timing offset, and T the symbol duration used in the transmitter. The output of the receive/matched filter can be expressed as

$$r(t) = \sum_k a_k g(t - \varepsilon - kT) + n(t) \tag{2-5}$$

where

$$n(t) = n'(t) * g_R(t). \tag{2-6}$$

The signal model assumed above will also be used in Section 2.3.4 and the succeeding chapters.

In maximum likelihood estimation, the received signal is sampled at the symbol rate and then a log-likelihood function is defined over N symbols. It has been shown that the log-likelihood function can eventually be expressed as [34]

$$\Lambda_L(\tau) = \sum_{i=1}^{N} \left| r(iT, \tau) \right| \tag{2-7}$$

with

$$r(iT, \tau) = \sum_{k=-\infty}^{+\infty} a_k g\ (iT - \tau T - kT) + n(iT) \tag{2-8}$$

where τ represents the normalized timing offset measured as a fraction of the symbol duration, which can be determined by maximizing the log-likelihood function $\Lambda_L(\tau)$.

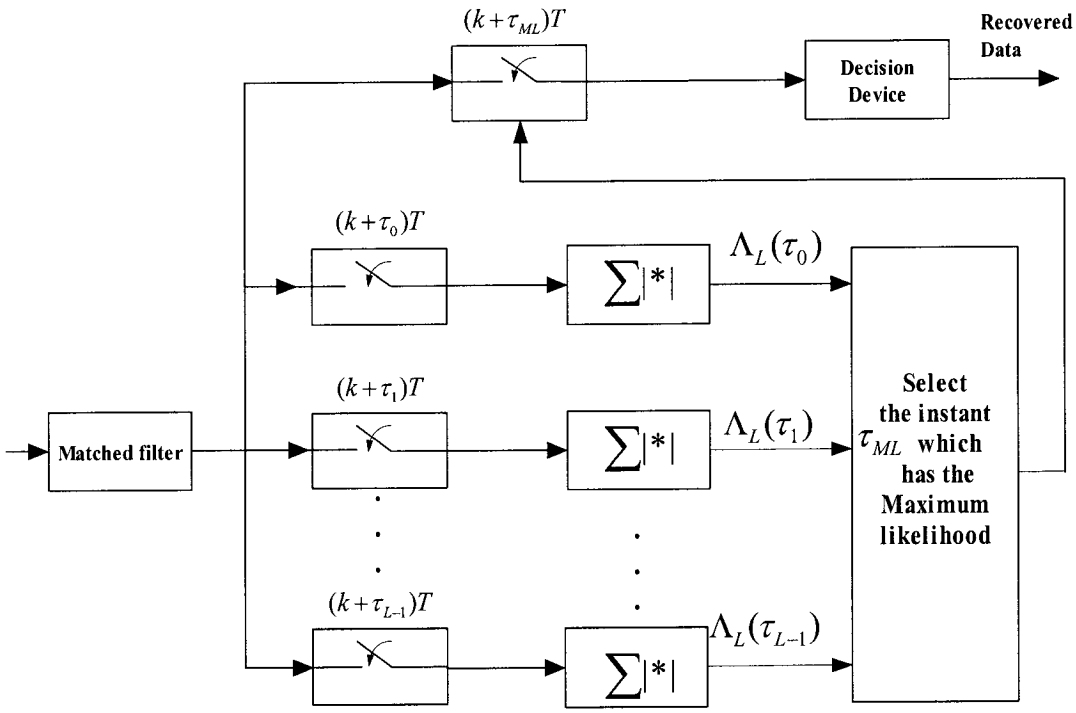


Fig. 2.11. Maximum likelihood feedforward symbol synchronization Scheme I

Fig. 2.11 illustrates a digital implementation of the feedforward maximum likelihood symbol synchronizer in which L branches have been used to compute and compare the likelihood functions obtained from the sampled signals with different delays. The L branches give the minimum discernable timing error of T/L. Fig. 2.12 depicts another implementation of the feedforward ML symbol synchronizer. Instead of using L samplers, this implementation employs one sampler that oversamples the received signal by a factor of L. The high-rate sequence is then decimated to L symbol-rate sequences, each of

which has a different sampling phase. The sequence that gives the maximum likelihood function represents the desired symbols or synchronized data.

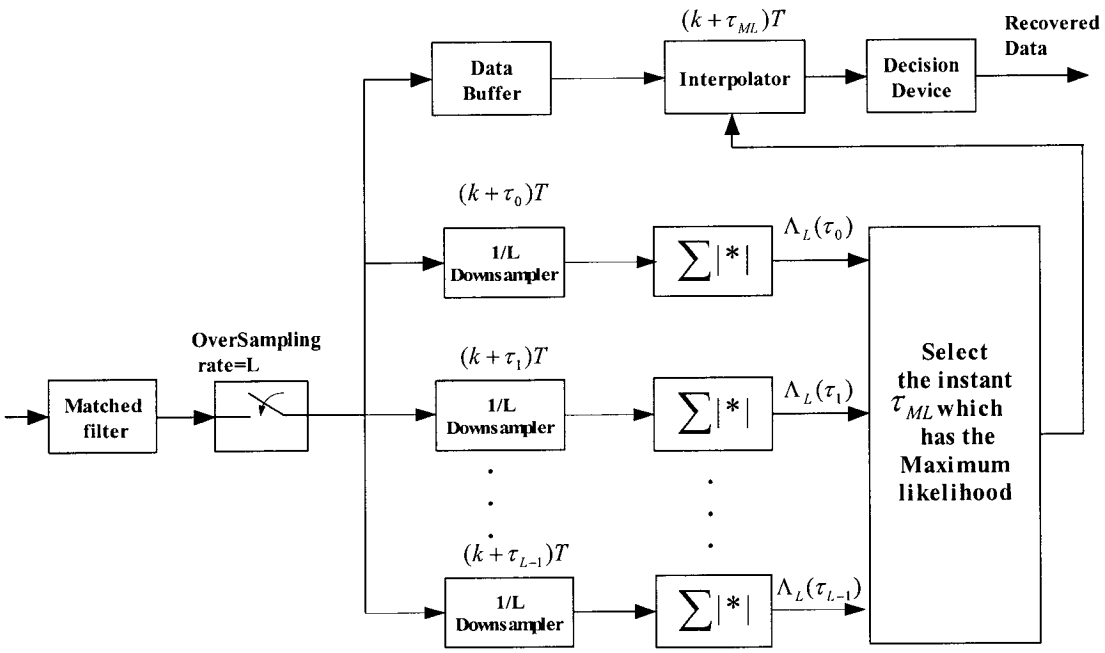


Fig 2.12. Maximum likelihood feedforward symbol synchronization Scheme II

The above Scheme II is more suitable for hardware implementation, since it requires only one A/D converter and the delay of the sampled signal can be very easily implemented with a digital device. On the other hand, Scheme I can be applied to very-high symbol-rate receivers, where L symbol-rate samplers serve as a high speed (N times symbol rate) A/D converter.

The maximum likelihood synchronizer enables a very fast acquisition of the symbol

timing. With 10 ~ 20 symbols, the maximum value of the likelihood function can usually be identified.

2.3.3. Spectral Line Generating Symbol Synchronization

The spectral line generating synchronizer shown in Fig. 2.13 represents another important subcategory of symbol synchronization. The received signal passes through a nonlinear device to generate a spectral line at the symbol rate 1/T, which contains the symbol timing information. The spectral line method can be used for both the feedback and feedforward synchronizations. When it is used in a feedback scheme, the symbol-rate spectral component is extracted by using a bandpass filter or a PLL, and compared with a local clock to create the timing phase, which can then be used to track the local sampling clock. When it is employed in a feedforward scheme, the timing phase will be used to determine the coefficients for an interpolator. It is worth mentioning that the bandpass filter or PLL can be replaced with the discrete Fourier transform (DFT), which computes directly the symbol timing phase. One of the examples of applying the DFT to the spectral line method for the estimation of the symbol timing is the Oerder and Meyr method, which will be outlined in the next section.

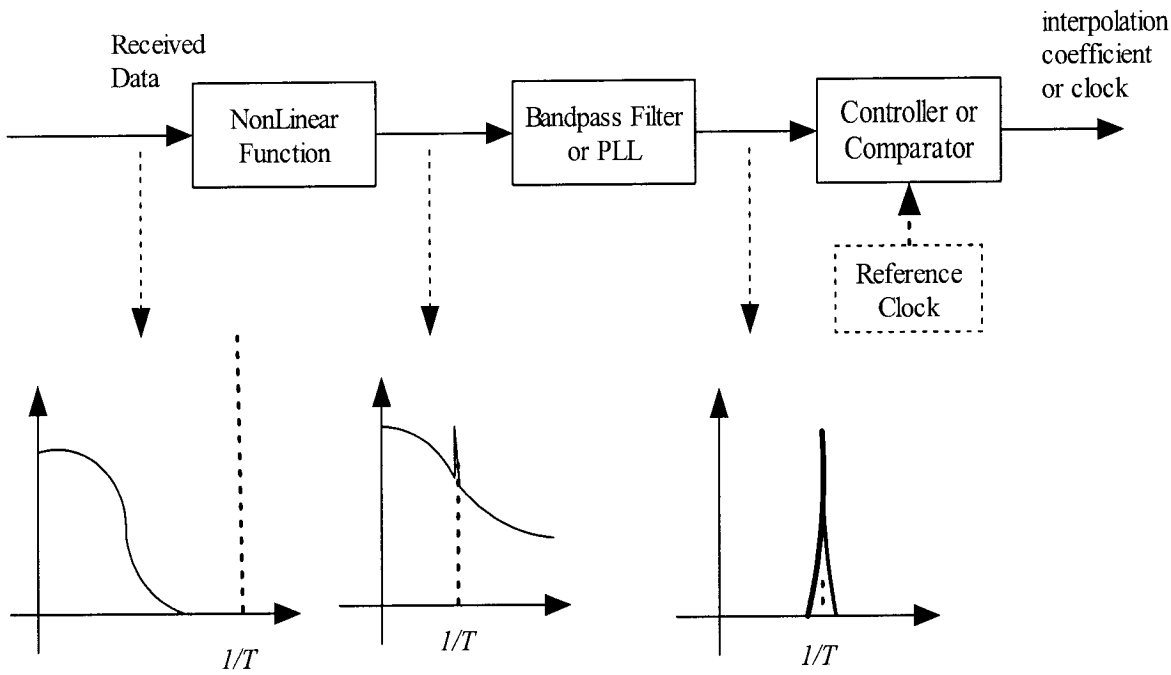


Fig. 2.13. Spectral line generating synchronizer and signal spectrums

2.3.4. The Oerder & Meyr Timing Error Detector

The Oerder and Meyr (O&M) algorithm [9] extracts the timing information from the squared signal and yields an unbiased estimate of the timing phase. Figure 2.14 shows the major functional blocks required by the O&M algorithm. The signal input to the sampler is given by (2-5). After sampling, it can be written as

$$r_n = r(n\tfrac{T}{N}) = \sum_k a_k g(n\tfrac{T}{N} - \varepsilon - kT) + n(n\tfrac{T}{N}) \qquad (2\text{-}9)$$

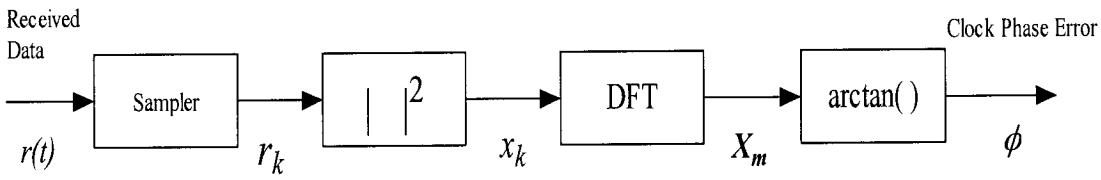


Fig. 2.14. The Oerder & Meyr square timing error detector

The O&M algorithm needs to compute a DFT on the squared sampled signal. Let the output of the square block be x_n. The DFT block produces

$$X_m = \sum_{n=mLN}^{(m+1)LN-1} x_n e^{-j2\pi n/N} = \sum_{n=mLN}^{(m+1)LN-1} |r_n|^2 e^{-j2\pi n/N} \quad . \tag{2-10}$$

Then, the instantaneous estimate for the timing phase is given by

$$\phi_m = -\frac{1}{2\pi} \arg(X_m) \, . \tag{2-11}$$

The advantage of the O&M algorithm is that it allows for a direct calculation of the symbol timing offset from the sampled signal. Moreover, the required square and DFT operations can easily be implemented with DSP and ASIC chips. However, the O&M algorithm needs at least 4 samples/symbol sampling rate. In the next chapter, a new symbol timing estimation algorithm that uses only 2 samples per symbol will be presented.

2.4. Conclusion

In this chapter, the carrier and symbol synchronization issues in communication receivers have been addressed with emphasis on the discussion of symbol synchronization methods. Two broad classes of symbol timing recovery techniques, namely the feedback and feedforward synchronization methods, have been reviewed. It has been pointed out that the feedback methods usually need a longer period of time to acquire the initial timing offset and then are able to track well the timing phase by a feedback loop. Therefore, feedback methods are suitable for continuous mode communication. Feedforward schemes allow the use of a free-running oscillator which samples the received analog signal independently. The symbol timing phase can then be estimated from the sample data. Feedforward algorithms enable a fast acquisition of the symbol timing phase and enjoy a straightforward computation and therefore, they are well suited for burst mode communication and digital implementation. Under the category of feedforward synchronization, three techniques, the maximum likelihood (ML) estimation, the spectral line method, and the Oeder and Meyr algorithm, have been discussed. The ML method yields a very fast estimation of the symbol timing and requires only some ten symbols to get the timing offset. But it needs a high-speed sampling device equivalent to at least eight times symbol rate, or a number of symbol-rate sampling devices. The O&M

algorithm as an implementation scheme of the spectral line generating synchronization enjoys a small computational complexity. However, it still requires at least 4 samples/symbol sampling rate, which may be critical for high-rate communication systems where the cost of the receiver heavily depends on the sampling rate.

Chapter 3

A Symbol Timing Recovery Algorithm using Two Samples per Symbol

As shown in Chapter 2, most of the exiting feedforward timing recovery methods use four samples per symbol or a higher sampling rate. From the point of view of the cost of a communication receiver, however, the lowest possible sampling rate is preferred since the complexity of a receiver is dictated to a great extent by the sampling rate. In this chapter, a feedforward timing recovery scheme that employs two samples/symbol sampling rate is presented with an emphasis on the timing estimation algorithm and the evaluation of its performance. Some issues pertaining to the implementation of the proposed timing recovery scheme are also addressed.

3.1. Timing Estimation

The key to a feedforward recovery technique is an accurate and fast estimation of the timing offset. In this section, we will present a timing estimation algorithm that is based on the low-pass filtering and squaring operations on the baseband signal.

The received complex baseband signal, after the matched filter, can be expressed as

$$r(t) = r_I(t) + jr_Q(t) \tag{3-1}$$

where $r_I(t)$ and $r_Q(t)$ represent the in-phase (I) and the quadrature (Q) signals, respectively, as given below

$$r_I(t) = \sum_k a_{Ik} g(t - \varepsilon - kT) + n_I(t), \qquad (t = nT_s) \tag{3-2}$$

$$r_Q(t) = \sum_k a_{Qk} g(t - \varepsilon - kT) + n_Q(t). \qquad (t = nT_s) \tag{3-3}$$

In the above two equations, a_{Ik}'s and a_{Qk}'s are, respectively, the in-phase and quadrature components of the transmitted complex symbols, T the symbol duration, ϵ the timing offset, and $g(t)$ the overall baseband impulse response that can usually be modeled as

$$g(t) = g_T(t) * g_R(t) \tag{3-4}$$

where $g_T(t)$ and $g_R(t)$ represent, respectively, the impulse response of the pulse-shaping filter in the transmitter and that of the matched-filter in the receiver. As in many communication systems, $g_T(t)$ and $g_R(t)$ can each be chosen to be a square-root raised-cosine filter and thus, the overall baseband impulse $g(t)$ is of the raised-cosine characteristic as given by

$$g(t) = \frac{T \sin(\pi t / T) \cos(\alpha \pi t / T)}{\pi t[1 - (2\alpha t / T)^2]} \tag{3-5}$$

where α is the roll-off factor. The noise terms $n_I(t)$ in (3-2) and $n_Q(t)$ in (3-3) are filtered zero-mean additive white Guassian noise (AWGN) by the receiving filter $g_R(t)$.

Note that $n_I(t)$ and $n_Q(t)$ still have a zero mean though they are no longer white after filtering. In this chapter, only the in-phase signal is used for the estimation of the timing offset. A similar estimation result can be obtained if only the quadrature signal is used. In the next chapter, it will be shown that the estimation precision can be improved if both I and Q signals are applied.

Fig 3.1 shows the block diagram for the proposed timing estimator. The I (or Q) samples are first multiplied by a half-symbol rate quadrature sinusoidal and low-pass filtered to construct the real and imaginary parts of a complex sequence $y_I(nT_S)$. The resulting complex sequence is then squared and averaged over an interval of L symbols. The averaged real and imaginary parts are sent to the *arctan* function to calculate the timing estimate. In what follows, we will provide a theoretical justification.

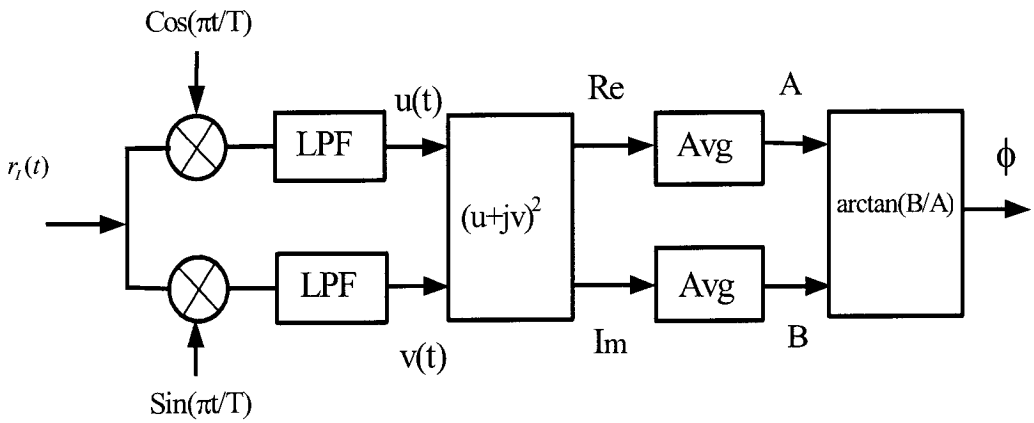


Fig. 3.1. Block diagram for the proposed timing estimator

The proposed algorithm starts with a complex modulation of the received in-phase

baseband signal with a half-symbol rate complex sinusoidal $e^{j\frac{\pi t}{T}}$, followed by a low-pass filtering with the impulse response $h(t)$. Therefore, the output of the low-pass filter is a complex-valued signal that can be written as

$$y_I(t) = u(t) + jv(t) = [r_I(t)e^{j\frac{\pi t}{T}}] * h(t) \tag{3-6}$$

where $u(t)$ and $v(t)$ are, respectively, the real and imaginary parts of $y(t)$. We assume that $h(t)$ is an ideal low-pass filter with a cutoff frequency $\alpha/2T$. This assumption is for the simplicity of the justification of the algorithm. As a matter of fact, this filter does not have to be ideal. As will be shown in the simulation part, one can even use a first-order IIR filter to achieve an accurate timing estimate. The received in-phase signal can be written as

$$r_I(t) = s_I(t) + n_I(t), \tag{3-7}$$

where

$$s_I(t) = \sum_k a_{Ik} g(t - \varepsilon - kT) \tag{3-8}$$

represents the transmitted "clean" in-phase signal and $n_I(t)$ is the AWGN noise. By substituting (3-7) into (3-6) and using the frequency responses of the raised-cosine filter and the low-pass filter $h(t)$, we have

$$y_I(t) = [s_I(t)e^{j\frac{\pi t}{T}}] * h(t) + [n_I(t)e^{j\frac{\pi t}{T}}] * h(t). \tag{3-9}$$

We assume that the low-pass filter in Fig 3.1 is an ideal filter with a cutoff frequency $\frac{\alpha}{2T}$. The Fourier transform of $s_I(t)$ can be expressed as

$$S_I(\omega) = F[s_I(t)] = e^{-j\varepsilon\omega} \sum_k a_{Ik}\, G(\omega)\, e^{-jkT\omega}, \tag{3-10}$$

where $G(\omega)$ is given by

$$G(\omega) = \begin{cases} T & 0 < |\omega| < (1-\alpha)\frac{1}{T} \\ \frac{T}{2}\left\{1 - \sin\left[\frac{\pi(|\omega| - \frac{1}{T})}{2\alpha\frac{1}{T}}\right]\right\} & (1-\alpha)\frac{1}{T} \le |\omega| \le (1+\alpha)\frac{1}{T} \\ 0 & |\omega| > (1+\alpha)\frac{1}{T} \end{cases}$$

The modulation of $s_I(t)$ with $e^{j\frac{\pi t}{T}}$ yields the shifted version of the spectrum $S_I(\omega)$, that is,

$$F\left[s_I(t)\, e^{j\frac{\pi t}{T}}\right] = S_I\left(\omega - \frac{\pi}{T}\right). \tag{3-11}$$

The low-pass filtering of $s_I(t)\, e^{j\frac{\pi t}{T}}$ gives the following spectral component,

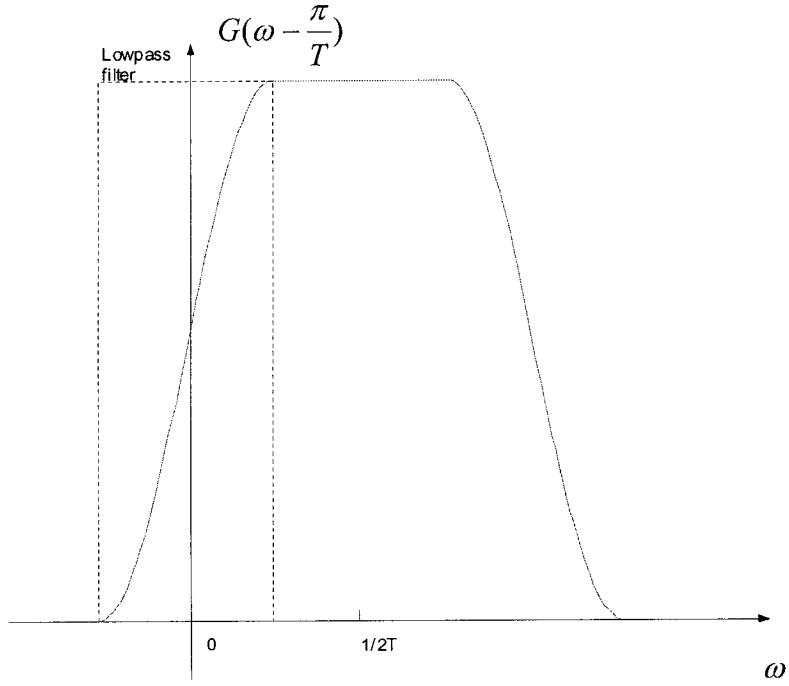


Fig. 3.2. Spectrum of $G(\omega - \frac{\pi}{T})$

$$S_I\left(\omega-\frac{\pi}{T}\right)_{LPF}=\frac{T}{2}e^{j\frac{\pi\varepsilon}{T}}e^{-j\varepsilon\omega}e^{-jkT\omega}e^{jk\pi}\sum a_{Ik}[1+\sin\frac{\pi T\omega}{\alpha}]. \quad (3\text{-}12)$$

That is, the output signal of the low-pass filter can be written as

$$y_I(t)=F^{-1}[S_I\left(\omega-\frac{\pi}{T}\right)_{LPF}]+[n_I(t)e^{j\frac{\pi t}{T}}]*h(t). \quad (3\text{-}13)$$

Fig. 3.2 shows the shifted baseband spectrum along with the frequency response of the low-pass filter. Assume that $f(t)$ is the inverse Fourier transform of the low-pass filtered version of the shifted baseband spectrum. We then have

$$f(t)=F^{-1}[1+\sin(\frac{\pi T\omega}{\alpha})]=f_1(t)+jf_2(t) \quad (3\text{-}14)$$

with

$$f_1(t)=\frac{\sin(\frac{\alpha t}{2T})}{\pi t}, \quad (3\text{-}15)$$

$$f_2(t)=\frac{1}{2}[f_1(t-\frac{T}{2\alpha})-f_1(t+\frac{T}{2\alpha})]. \quad (3\text{-}16)$$

From (3-12) to (3-16), we obtain

$$y_I(t)=\frac{T}{2}e^{j\frac{\pi\varepsilon}{T}}\sum_k a_{Ik}[f_1(t-\varepsilon-kT)+jf_2(t-\varepsilon-kT)]+n_{I1}(t) \quad (3\text{-}17)$$

The noise component $n_{I1}(t)$ in (3-17) is the modulated and filtered version of $n_I(t)$, as given by

$$n_{I1}(t)=[n_I(t)e^{j\frac{\pi t}{T}}]*h(t). \quad (3\text{-}18)$$

As the transmitted symbols are independent of each other and independent of the noise $n_{I1}(t)$ as well, we have

$$E(a_{Ik}a_{Il}) = \begin{cases} E(a_{Ik}^2) = E_S & k = l \\ 0 & k \neq l \end{cases} \tag{3-19}$$

$$E[a_{Ik}n_{I1}(t)] = E(a_{Ik})E[n_{I1}(t)] = 0\,. \tag{3-20}$$

In (3-19), E_S represents the symbol energy. By using (3-19) and (3-20), one can find the expectation of the squared complex signal $y_I(t)$ as given below

$$E[y_I^{\,2}(t)] = \frac{T^2E_S}{4}e^{j\frac{2\pi\varepsilon}{T}}\sum_k f^2(t-\varepsilon-kT) + E[n_{I1}^2(t)]\,. \tag{3-21}$$

Note that

$$\begin{aligned} E[n_{I1}^2(t)] &= E\{[(n_I(t)e^{j\frac{\pi t}{T}})*h(t)]^2\} \\ &= E[\int n_I(t-\tau)e^{j\frac{\pi(t-\tau)}{T}}h(\tau)d\tau \int n_I(t-s)e^{j\frac{\pi(t-s)}{T}}h(s)ds] \\ &= e^{j\frac{2\pi t}{T}}\int\int E[n_I(t-\tau)n_I(t-s)]h(\tau)h(s)e^{-j\frac{\pi(\tau+s)}{T}}dsd\tau \\ &= e^{j\frac{2\pi t}{T}}\int\int R_n(\tau-s)h(\tau)h(s)e^{-j\frac{\pi(\tau+s)}{T}}dsd\tau \end{aligned} \tag{3-22}$$

where $R_n(\tau-s)$ represents the auto-correlation function of $n_I(t)$. Here, we have assumed that $n_I(t)$ is a wide-sense stationary process. It is of interest to note that $E[n_{I1}^2(t)]$ is a complex sinusoidal with a frequency that is equal to the symbol rate. Using (3-22), (3-21) can be rewritten as

$$E[y_I{}^2(t)] = \frac{T^2 E_S}{4} e^{j\frac{2\pi\varepsilon}{T}} \sum_k f^2(t-\varepsilon-kT) + Ke^{j\frac{2\pi t}{T}} \tag{3-23}$$

where K is a constant given by

$$K = \int_\tau \int_s R_n(\tau - s)h(\tau)h(s)e^{-j\frac{\pi(\tau+s)}{T}} ds\, d\tau \,. \tag{3-24}$$

We now integrate $E[y_I^2(t)]$ over a number of symbol durations. Since the received analog baseband signal is already sampled by a free-running oscillator, the integration of $E[y_I^2(t)]$ can be written as the following summation form,

$$\sum_n E[y_I{}^2(nT_S)] = \frac{T^2 E_S}{4} e^{j\frac{2\pi\varepsilon}{T}} \sum_n \sum_k f^2(nT_S - \varepsilon - kT) + K\sum_n e^{j\frac{2\pi nT_S}{T}} \tag{3-25}$$

where T_S represents the sampling duration. It is evident that the second term on the right-hand side of (3-25) vanishes as long as the sampling frequency is chosen to be an integral multiple of the symbol rate, i.e., $T = NT_S$, where N is an integer representing the oversampling factor and $N \geq 2$ in the proposed scheme. In what follows, we will show that the double summation in (3-25) leads to a real value.

Using (3-14), we obtain

$$\begin{aligned}\sum_n \sum_k f^2(nT_S - \varepsilon - kT) = &\sum_n \sum_k [f_1^2(nT_S - \varepsilon - kT) - f_2^2(nT_S - \varepsilon - kT)] \\ &+ j2\sum_n \sum_k [f_1(nT_S - \varepsilon - kT) f_2(nT_S - \varepsilon - kT)]\end{aligned} \tag{3-26}$$

From (3-16), the imaginary part on the right-hand side of (3-26) can be written as

$$
\begin{aligned}
&2\sum_n\sum_k[f_1\,(nT_S-\varepsilon-kT)f_2\,(nT_S-\varepsilon-kT)] \\
&\quad=\sum_k[\sum_n[f_1(nT_S-\varepsilon-kT)f_1(nT_S-\varepsilon-kT-\tfrac{T}{2\alpha})]- \\
&\quad\quad\sum_k[\sum_n[f_1(nT_S-\varepsilon-kT)f_1(nT_S-\varepsilon-kT+\tfrac{T}{2\alpha})] \\
&\quad=\sum_k[R_{f_1,k}(-\tfrac{T}{2\alpha})-R_{f_1,k}(\tfrac{T}{2\alpha})]
\end{aligned}
\tag{3-27}
$$

where $R_{f_1,k}(\cdot)$ represents the time-domain auto-correlation function of $f_1(nT_S-\varepsilon-kT)$. Noting that the auto-correlation function is an even function as long as the summation interval for which the correlation is computed is large enough, (3-27) yields a zero value and (3-26) is reduced to a real number. As a result, the timing offset ε can be calculated from (3-25) as

$$
\begin{aligned}
\phi=\varepsilon/T&=\frac{1}{2\pi}\arg\{\sum_n E[y_I{}^2(nT_S)] \\
&=\frac{1}{2\pi}\arg\{\sum_n E[u(nT_S)+jv(nT_S)]^2\} \\
&=\frac{1}{2\pi}\arctan\left(\frac{\sum_n 2E[u(nT_S)v(nT_S)]}{\sum_n E[u^2(nT_S)-v^2(nT_S)]}\right).
\end{aligned}
\tag{3-28}
$$

Equation (3-28) gives a theoretical result for the normalized timing offset relative to the symbol duration. To use (3-28), one needs to compute first a few expectations. In practice, however, these expectations are replaced with time-domain averaging operations, which can readily be combined with the required summation. Therefore, the timing estimate can be computed as

$$\hat{\phi} = \hat{\varepsilon} / T = \frac{1}{2\pi} \arg\left(\sum_{n=0}^{LN-1} y^2(nT_S) \right) = \frac{1}{2\pi} \arctan\left(\frac{\sum_{n=0}^{LN-1} [u(nT_S)v(nT_S)]}{\sum_{n=0}^{LN-1} [u^2(nT_S) - v^2(nT_S)]} \right) \quad (3\text{-}29)$$

where L represents the number of symbols used to estimate the timing offset.

Thus, we have justified the estimation algorithm shown by the diagram in Fig. 3.1. Note that the average operation, instead of the accumulation/summation as required by (3-29), has been used in the diagram. Obviously, it does not make any difference in terms of the phase calculation, yet in the implementation it may avoid an overflow due to the continuous summation of samples when a large number of symbols are used for the estimation.

3.2. The Mean and Variance of the Timing Estimate

In this section, the performance of the proposed timing estimation algorithm is examined in terms of the mean and variance of the resulting timing estimate.

3.2.1. The Mean

The mean of the timing estimate can be written as

$$E(\hat{\phi}) = \frac{1}{2\pi} E[\arg(Z)] \quad (3\text{-}30)$$

where

$$Z = \sum_{n=0}^{LN-1} y_I^{\,2}(nT_S)\,. \tag{3-31}$$

For small variance of an estimate, we can linearize the arg-operation [9], that is,

$$\begin{aligned} E(\hat{\phi}) &\approx \frac{1}{2\pi}\arg\left[E(Z)\right] \\ &= \frac{1}{2\pi}\arg\left[E\left(\sum_{n=0}^{LN-1} y_I^{\,2}(nT_S)\right)\right] = \frac{1}{2\pi}\arg\left[\sum_{n=0}^{LN-1} E\left(y_I^{\,2}(nT_S)\right)\right] \end{aligned} \tag{3-32}$$

By comparing (3-32) with (3-28), we obtain $E(\hat{\phi}) = \phi$ when $L \to \infty$, implying that $\hat{\phi}$ is an unbiased estimate of ϕ as long as L is large enough.

3.2.2. The Variance

In order to simplify the computation of the variance of $\hat{\phi}$, we assume that $\varepsilon = 0$ and therefore, $\phi = \varepsilon / T = 0$ and $E(\hat{\phi}) = \phi = 0$. (It can be shown that the results to be obtained are valid for arbitrary ε [9]). We then have

$$\begin{aligned} Var\ (\hat{\phi}) &= E\,(\hat{\phi}^{\,2}) \\ &= \frac{1}{4\pi^{\,2}} E\left(\left[\arg(\ Z\ )\right]^2\right) \\ &\approx \frac{1}{4\pi^{\,2}} \frac{E\left[(\mathrm{Im}\ \ Z\ )^2\right]}{\left[E\ (\mathrm{Re}\ \ Z\ )\right]^2} \end{aligned} \tag{3-33}$$

where Z is given by (3-31). The approximation in (3-33) is valid since the imaginary part of Z has a zero mean, as seen from the discussion in the previous subsection, and the

variances of both the real and imaginary parts are small compared to the squared real mean [9]. Using (3-17), (3-25) and (3-31), and noting that $\varepsilon = 0$, $E(\mathrm{Re}\, Z)$ can be calculated as follows.

$$\begin{aligned} E[\mathrm{Re}(Z)] &= \mathrm{Re}[E(Z)] \\ &= \mathrm{Re}\left\{ \sum_{n=0}^{LN-1} E\left[\left(\frac{T}{2} \sum_k a_{Ik} f(nT_S - \varepsilon - kT) e^{jk\pi} + n_{I1}(nT_S) \right)^2 \right] \right\} \\ &= \mathrm{Re}\left[\frac{T^2 E_S}{4} \sum_{n=0}^{LN-1} \sum_k f^2(nT_S - \varepsilon - kT) + K \sum_n e^{j\frac{2\pi n T_S}{T}} \right] \\ &= \frac{T^2 E_S}{4} \mathrm{Re}\left[\sum_n \sum_k f^2(nT_S - \varepsilon - kT) \right] \end{aligned} \tag{3-34}$$

From (3-34), it is seen that the denominator in (3-33), $[E(\mathrm{Re}\, Z)]^2$, is a constant that depends only on the signal. Thus, the key to obtaining a closed-form expression for (3-33) is the computation of the numerator, $E[(\mathrm{Im}\, Z)^2]$. To this end, we rewrite Z as

$$\begin{aligned} Z &= \sum_{n=0}^{LN-1} \left[\frac{T}{2} \sum_k a_{Ik} f(nT_S - \varepsilon - kT) e^{jk\pi} + n_{I1}(nT_S) \right]^2 \\ &= Z_1 + Z_2 + Z_3 \end{aligned} \tag{3-35}$$

where

$$Z_1 = \frac{T^2}{4} \sum_{n=0}^{LN-1} \sum_k \sum_l a_{Ik}\, a_{Il}\, (-1)^{k+l} f(nT_S - \varepsilon - kT) f(nT_S - \varepsilon - lT) \tag{3-36}$$

$$Z_2 = T \sum_{n=0}^{LN-1} \sum_k a_{Ik}\ (-1)^k f(nT_S - \varepsilon - kT)\, n_{I1}(nT_S) \tag{3-37}$$

$$Z_3 = \sum_{n=0}^{LN-1} n_{I1}^2(nT_S) \tag{3-38}$$

Then, the numerator in (3-33) can be written as

$$E[(\text{Im } Z)^2] = E(I_1^2) + E(I_2^2) + E(I_3^2) + 2[E(I_1 I_2) + E(I_2 I_3) + E(I_1 I_3)] \tag{3-39}$$

where I_i (i=1,2,3) each represent the imaginary part of Z_i, i.e.,

$$I_1 = \text{Im}\left[\frac{T^2}{4}\sum_{n=0}^{LN-1}\sum_{k}\sum_{l} a_{Ik}\, a_{Il}(-1)^{k+l} f(nT_S - \varepsilon - kT) f(nT_S - \varepsilon - lT)\right] \tag{3-40}$$

$$I_2 = \text{Im}\left[T\sum_{n=0}^{LN-1}\sum_{k} a_{Ik}\,(-1)^k f(nT_S - \varepsilon - kT) n_{I1}(nT_S)\right], \tag{3-41}$$

$$I_3 = \text{Im}\left[\sum_{n=0}^{LN-1} n_{I1}^2(nT_S)\right]. \tag{3-42}$$

We now show that all the expectations associated with the cross-terms in (3-39) would vanish, that is, the three components are orthogonal to each other.

First, we may write Z_2 as

$$Z_2 = \sum_{n}^{LN-1} s(nT_S) n_{I1}(nT_S) \tag{3-43}$$

where $s(nT_S)$ represents the signal part in (3-36), as given by

$$s(nT_S) = T\sum_{k} a_{Ik}\,(-1)^k f(nT_S - \varepsilon - kT)\,. \tag{3-44}$$

Then, the imaginary part of Z_2 can be expressed as

$$I_2 = \sum_{n}^{LN-1}[s^{(r)}(nT_S)n_{I1}^{(i)}(nT_S) + s^{(i)}(nT_S)n_{I1}^{(r)}(nT_S)] \tag{3-45}$$

where the superscripts "*r*" and "*i*" are used to indicate the relevant real and imaginary parts, respectively. By using (3-45) and noting that $n_{I1}(nT_S)$ is independent of the transmitted symbols, one can express $E(I_1I_2)$ as a linear combination of $E[n_{I1}^{(r)}(nT_S)]$ and $E[n_{I1}^{(i)}(nT_S)]$. Since the noise has zero mean, i.e., $E[n_{I1}^{(r)}(nT_S)] = E[n_{I1}^{(i)}(nT_S)] = 0$, we obtain $E(I_1I_2) = 0$.

According to the fact that a_k has a zero mean and is independent of $n_1(nT_S)$, it can be shown that $E(I_2I_3)$ is linearly proportional to $E(a_k)$ and therefore, equals zero. As for $E(I_1I_3)$, we have

$$\begin{aligned} E(I_1I_3) &= E(I_1)E(I_3) \\ &= E(I_1)E[\mathrm{Im}(Z_3)] \\ &= E(I_1)\,\mathrm{Im}[E(Z_3)] \\ &= E(I_1)\,\mathrm{Im}\left(\sum_{n}^{LN-1}E[n_{I1}^2(nT_S)]\right) \\ &= 0 \end{aligned} \tag{3-46}$$

In obtaining (3-46), we have used the discrete-time version of the expression for $E[n_1^2(t)]$ as given by (3-22). As a result, (3-37) is simplified as

$$E[(\mathrm{Im}\ Z)^2] = E(I_1^2) + E(I_2^2) + E(I_3^2) \tag{3-47}$$

where $E(I_1^2)$, $E(I_2^2)$ and $E(I_3^2)$ are associated with the signal, the product of signal and noise and the noise component, respectively. Using (3-47), the variance given by (3-33) can be simplified as

$$Var\ (\hat{\phi}) = \frac{1}{4\pi^2}\frac{1}{[E(\mathrm{Re}\ Z)]^2}[E(I_1^2) + E(I_2^2) + E(I_3^2)] \qquad (3\text{-}48)$$
$$= \sigma_{sxs}^2 + \sigma_{sxn}^2 + \sigma_{nxn}^2$$

where the three variance components are for the signal, the noise and the product of signal and noise.

3.3. Simulation Results

This section presents some simulation results for the estimation method proposed in Section 3.1, showing the estimation performance for different choices of estimation interval length, roll-off factor, and the low-pass IIR filter.

3.3.1. Eye Diagram and Signal Constellation

For computer simulation, we apply the proposed timing recovery method to a QPSK modulation system. The transmitter generates a baseband QPSK signal that is intended to pass through an AWGN channel. In the receiver, the received baseband signal is first sampled at two samples per symbol sampling rate. The baseband samples corresponding to either the in-phase or quadrature component are used for the timing offset estimation. Then, the baseband samples are interpolated using the estimated timing information to reconstruct the original I and Q symbols. Fig. 3.3 shows the eye-diagrams of the

reconstructed QPSK signal for a noise-free channel. When the AWGN channel is set to have SNR=10 db, the reconstructed in-phase and quadrature signals have the eye diagrams shown in Fig.3.4. The plot of the scattered signal constellation is shown in Fig.3.5 for the noise-free case and in Fig.3.6 for SNR=10db. It is seen from the noise-free case plots that the proposed estimation algorithm has a very good precision and the scattering effect due to the estimation and the second-order interpolation is negligible. Although the signal is scattered when SNR=10 db, a correct decision can still be guaranteed according to the eye and constellation diagrams.

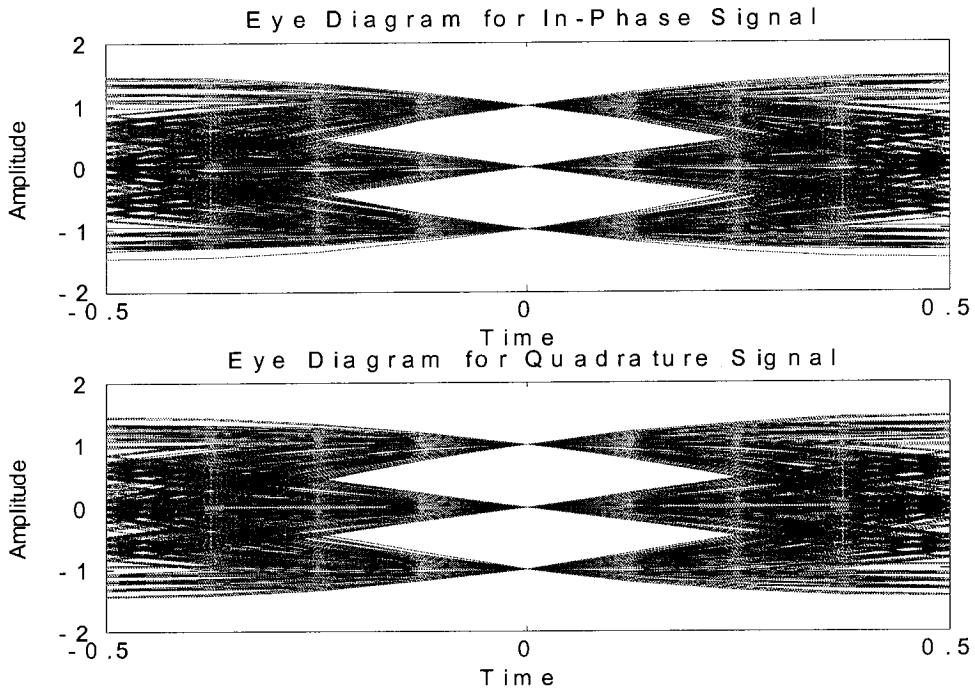


Fig. 3.3. Eye diagrams for QPSK signal in no noise-free channel:

(a) In-phase signal (b) Quadrature signal

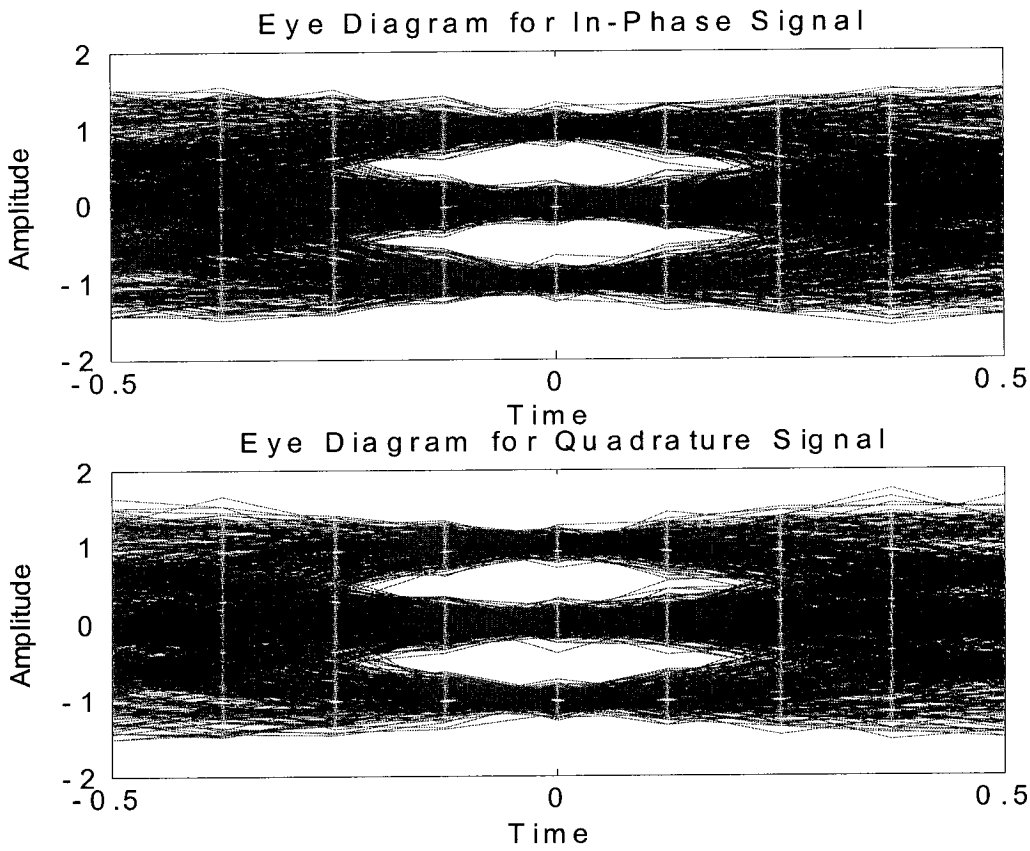


Fig. 3.4. Eye diagrams for QPSK with SNR=10 db

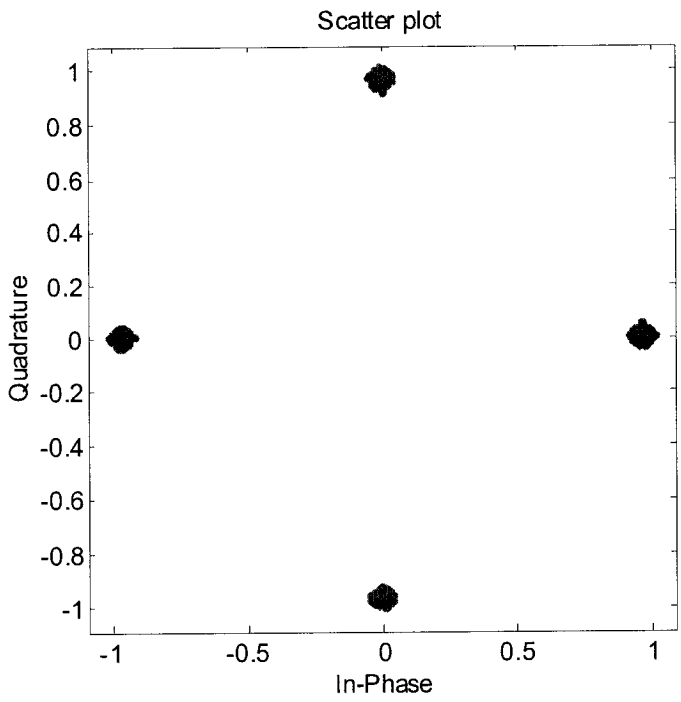


Fig. 3.5. Signal scatter-plot after interpolation without noise

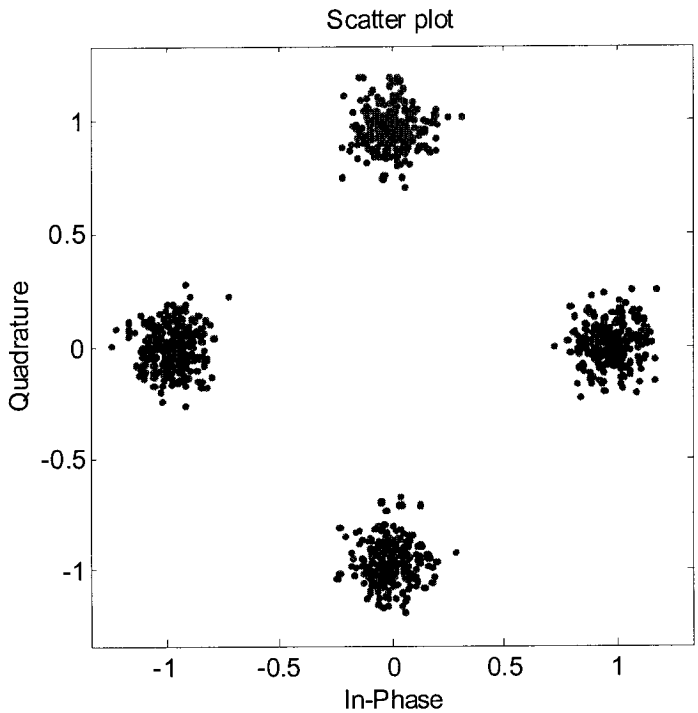


Fig. 3.6. Signal scatter-plot after interpolation with SNR=10db

3.3.2. The Length of Estimation Interval

This subsection shows the impact of the estimation length on the variance of the timing estimate.

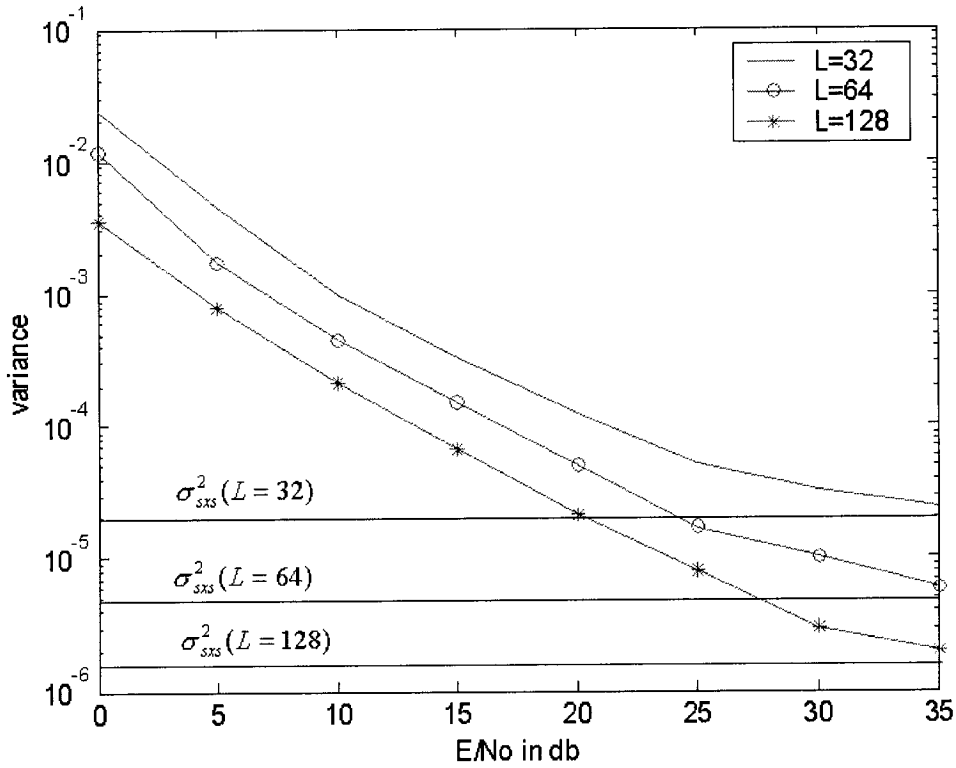


Fig. 3.7. Variance of the timing estimate

Fig 3.7 shows the variances of the timing estimate with different estimation interval lengths. Clearly, the variance is reduced as the estimation length is increased. But increasing the estimation length would increase the acquisition time of timing phase estimation in addition to causing more operations in the average unit. Also, the improvement of the timing error is not linearly proportional to the increase in the estimation length, implying that one has to increase the interval significantly in order to

decrease the variance by a certain amount. Extensive simulation work has shown that L=64~128 appears to be a good trade-off between the estimation performance and the computational complexity.

3.3.3. The Influence of Low-Pass Filter

In order to see the influence of the low-pass filter on the estimation performance, we employ different IIR (infinite-duration impulse response) digital filters in the simulation. The simplest one is given by the first-order transfer function

$$H(z) = \frac{0.2}{1 - 0.8z^{-1}}$$

which has the lowest implementation complexity. Other filters attempted in the simulation are higher order Butterworth filters with different cutoff frequencies.

Fig.8 depicts the variance versus E/N_0 plot for the four IIR filters, when the roll-off factor is chosen as 0.5. The cut-off frequencies for the three fourth-order filters are $0.25F_s$, $0.35F_s$ and $0.45F_s$, where Fs denotes the sampling frequency of the designed filters. It is seen that the higher-order filters perform slightly better than the first-order filter only if the SNR level is low (< 15 db). The first-order filter is even superior when there is almost no noise (SNR ≥25 db). Fig.9 shows the simulation results for the same set of filters as used in Fig.8, when the roll-off factor is chosen to be 0.7. Clearly, at the same SNR level, a larger roll-off factor yields a slightly smaller variance.

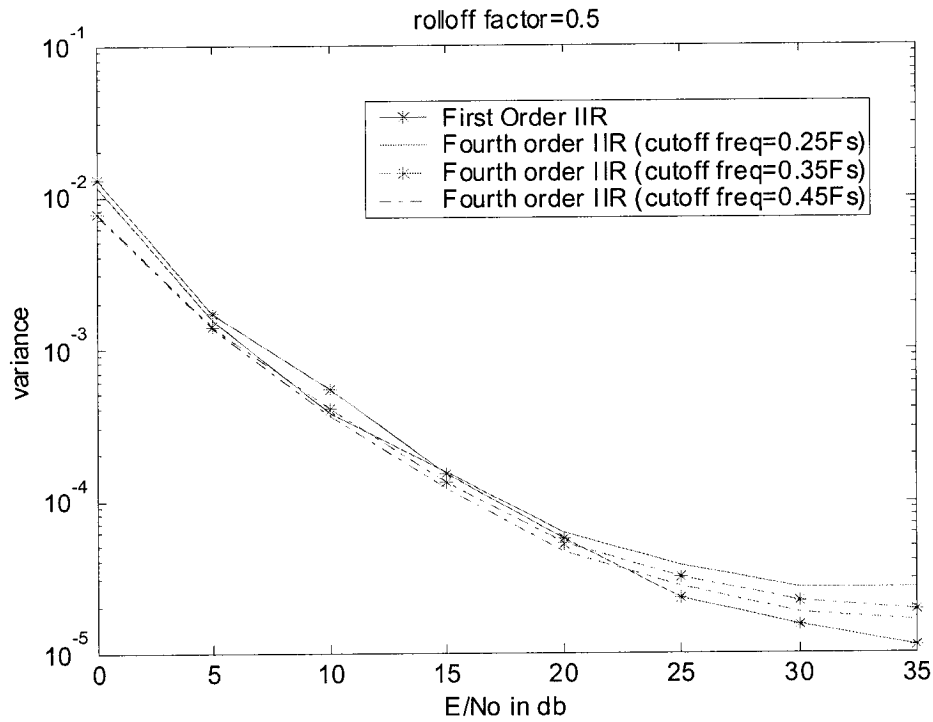


Fig. 3.8. Variance of the timing estimate with different lowpass filters (α=0.5)

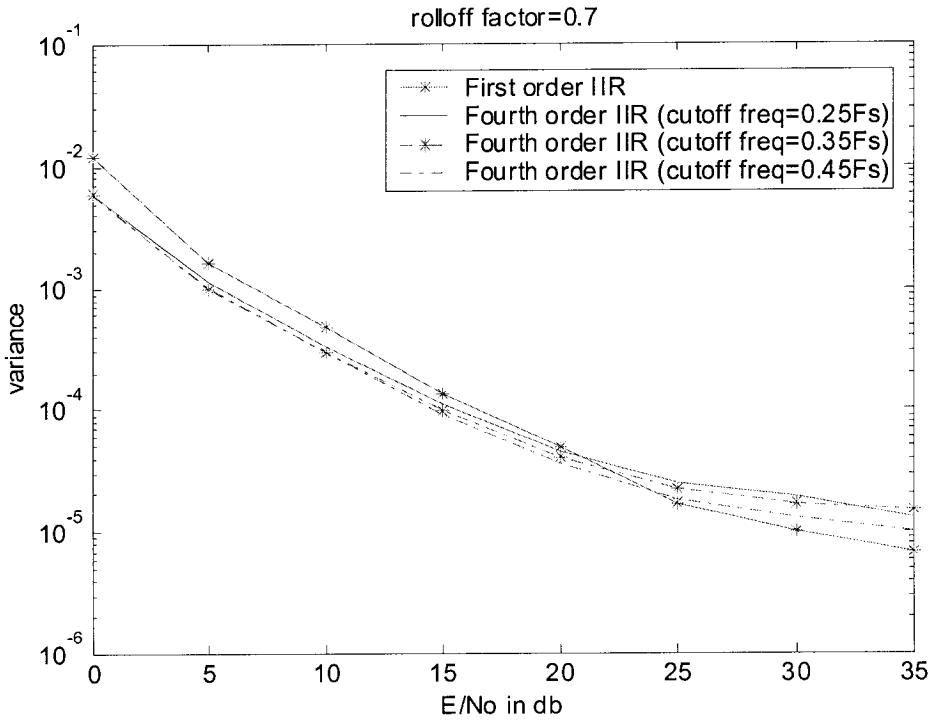


Fig. 3.9. Variance of the timing estimate with different lowpass filters (α=0.7)

3.3.4. The Effect of Roll-Off Factor

The roll-off factor α is a key parameter that influences the timing estimation performance. Fig. 3.10 depicts the variance curves for different values of α, where the estimation length is set to be equal to 64 symbols. It is seen that the algorithm works better for large roll-off factors. For applications where a small excess bandwidth has to be used, one can improve the timing variance by using a large value of L.

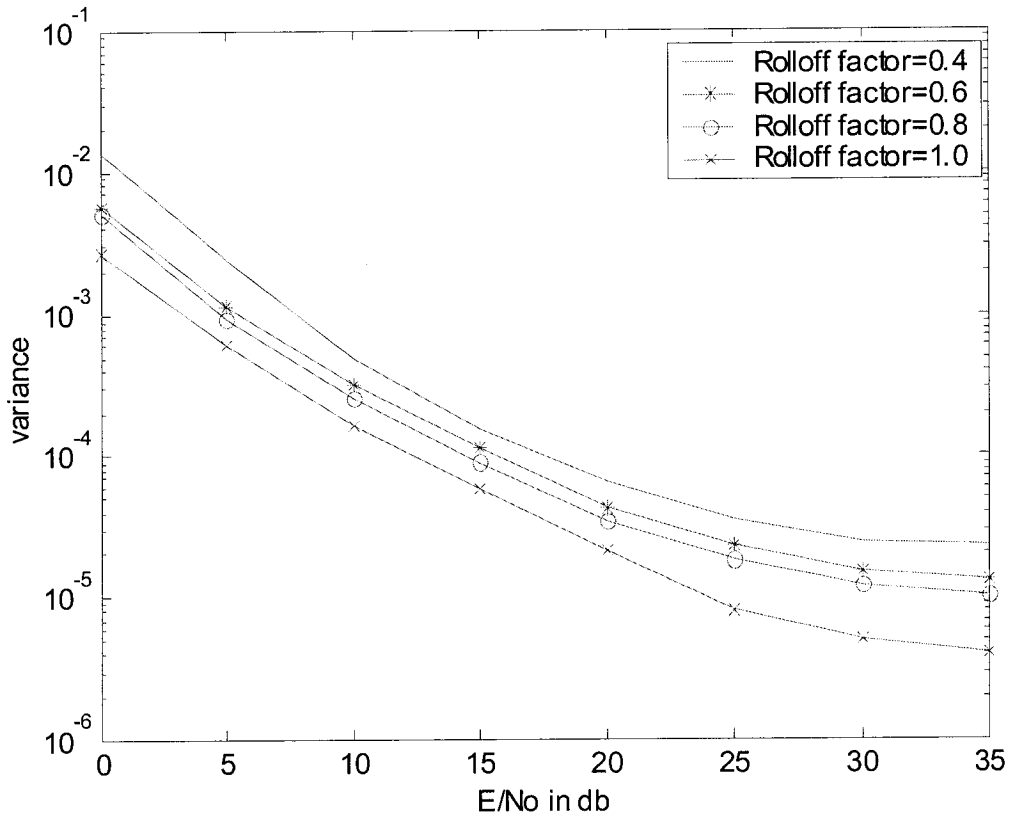


Fig. 3.10. Variance of the timing estimate with different roll-off factors

3.4. Implementation Issues

In many communication systems, the symbol synchronization algorithm will eventually be implemented in an integrated circuit chip. The computational and implementational complexity will be directly relevant to the cost of the receiver. In this section, we will discuss some of the implementation issues for the proposed symbol synchronization technique.

3.4.1. Implementation of Arctan(x)

A common task in feedforward estimation methods is to implement arctan(x). This is usually done by a look-up table that stores the principle phase values in [0,π/4] which corresponds to 0≤x≤1. By fitting the arctan(x) function in the interval [0,1], we find that the following second-order two-coefficient polynomial

$$f(x) = 1.062x - 0.276x^2 \tag{3-49}$$

is able to provide a very good approximation within $0 \le x \le 1$. As required by (3-29) and Fig. 3.1, one division B/A is involved in arctan function, where A and B are signed values. In order to avoid the division by a very small number, we do the following:

(i) If |A|≥|B|, calculate x=|B|/|A| and then use (3-49) to obtain the principle value of ϕ in 0~π/4;

(ii) If |A|<|B|, use arctan(|A/B|)=π/2-arctan(|B/A|) along with the approximation formula (3-49).

Clearly, the true quadrant of ϕ can be determined from the signs of A and B. The above polynomial approximation scheme is very suitable for real-time DSP implementation. If the look-up table is used, one would need to do many comparisons to find a proper entry for the table.

3.4.2. Interpolation

Interpolation is another important module in the implementation of feedfroward symbol synchronization. As seen from Section 2.3, the received signal is first sampled by a fixed clock. The timing phase is estimated from these samples and then used for interpolation among the signal samples to find the correct symbols. Essentially, the interpolator is an FIR filter whose coefficients can be determined by Lagrange interpolation formula.

A. Four-point Lagrange interpolator

A four-point Lagrange interpolator is very useful in symbol synchronization. Fig. 3.11 shows a scenario where four samples y_{-1}, y_0, y_1 and y_2, which are located at x_{-1}, x_0, x_1 and x_2, respectively, are known, and the right sample is deviated from x_0 by μ. Therefore, μ indicates the timing offset which is measured as a fraction of 2π

according to the arctan() function. Usually, μ is normalized for interpolation. For example, it can be rescaled to the range of [0,2], where $\mu = 0$ implies a zero clock phase and $\mu = 2$ means the maximum timing phase equal to 2π.

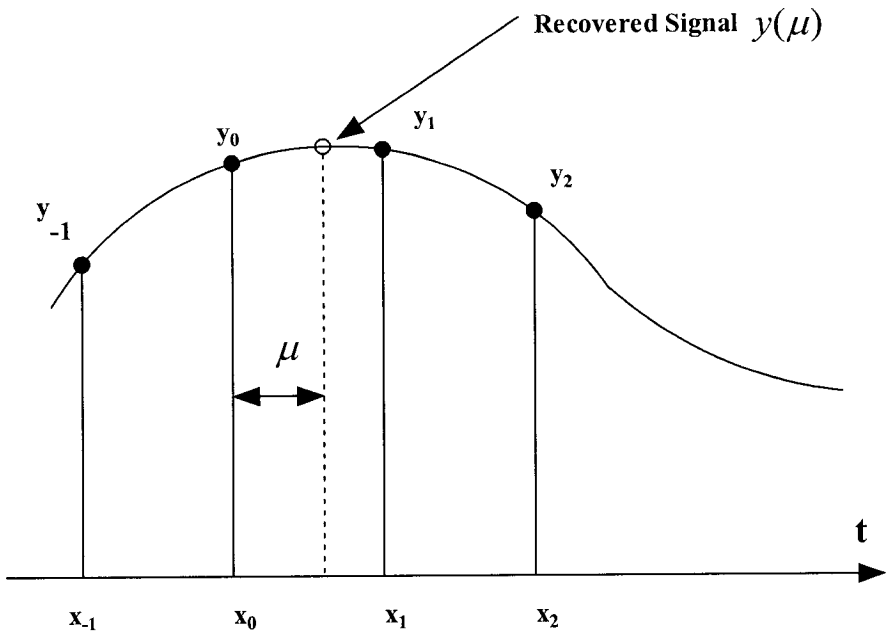


Fig. 3.11 Four-point Lagrange interpolation

Once the timing offset μ is computed using the proposed estimation method, it can be shown that the desired sample or the recovered symbol located $x_0 + \mu$ is given by [28]

$$y(\mu) = (-\frac{1}{6}\mu^3 + \frac{1}{2}\mu^2 - \frac{1}{3}\mu)y_{-1} + (\frac{1}{2}\mu^3 - \mu^2 - \frac{1}{2}\mu + 1)y_0 + (-\frac{1}{2}\mu^3 + \frac{1}{2}\mu^2 + \mu)y_1 + (\frac{1}{6}\mu^3 - \frac{1}{6}\mu)y_2. \quad (3\text{-}50)$$

It is known that the Farrow structure [32] can be used for fast implementation of the

Lagrange interpolation. Using the Farrow algorithm, the desired output sample for the four-point interpolation can be computed in a matrix form as follows.

$$y(\mu)=\begin{pmatrix} -\frac{1}{6} & \frac{1}{2} & -\frac{1}{3} & 0 \\ \frac{1}{2} & -1 & -\frac{1}{2} & 1 \\ -\frac{1}{2} & \frac{1}{2} & 1 & 0 \\ \frac{1}{6} & 0 & -\frac{1}{6} & 0 \end{pmatrix} \begin{pmatrix} \mu^3 \\ \mu^2 \\ \mu \\ 1 \end{pmatrix} \begin{pmatrix} y_{-1} & y_0 & y_1 & y_2 \end{pmatrix} \tag{3-51}$$

The Farrow implementation structure corresponding to (3-51) is shown in Fig. 3.12. A number of FIR filters are contained in the structure and the outputs of these constant-coefficient FIR filters are properly weighted and combined to produce the interpolated sample.

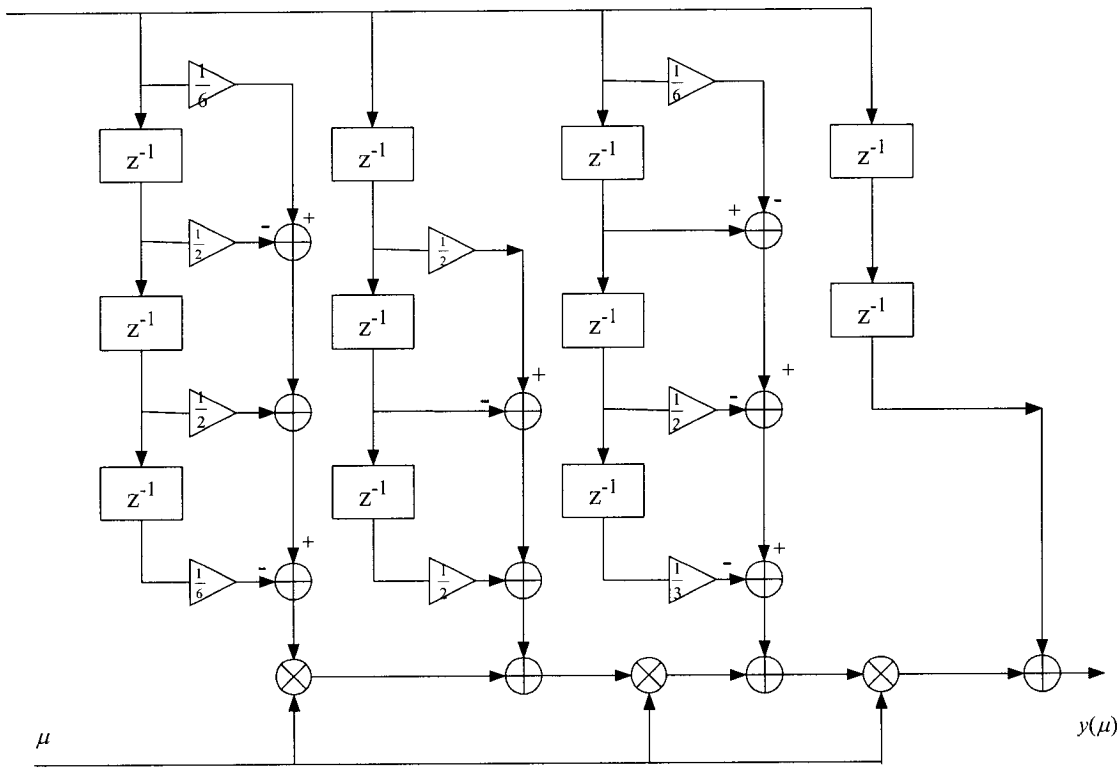


Fig. 3.12. Farrow structure of four-point interpolation

B. Three-point Lagrange interpolator

Similar to Four-point Lagrange interpolation, one can use three signal samples, say y_{-1}, y_0, and y_1, for interpolation. This three-point second-order interpolation is particularly useful when two samples per symbol sampling rate is employed. Fig. 3.13 shows the situation where three samples are available and the desired sample is off x_0 by μ. In this case, the correct sample can be calculated by the second-order Lagrange interpolation formula [28]

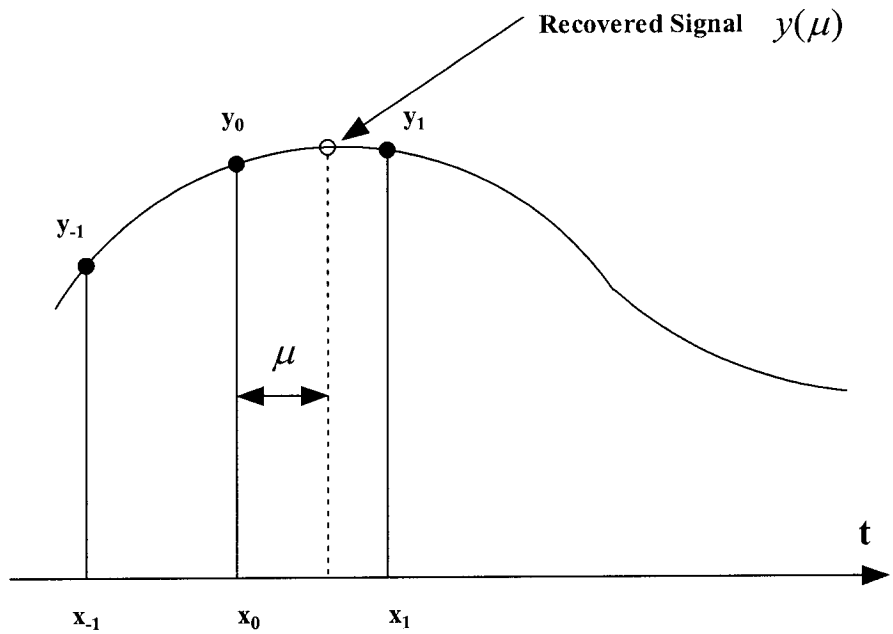


Fig. 3.13. Three-point Lagrange interpolation

$$y(\mu) = (\frac{1}{2}\mu^2 - \frac{1}{2}\mu)y_{-1} + (1-\mu^2)y_0 + (\frac{1}{2}\mu^2 + \frac{1}{2}\mu)y_1 \qquad (3\text{-}52)$$

which can be written in matrix form as

$$y(\mu)=\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & 0 \\ -1 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}\begin{pmatrix} \mu^2 \\ \mu \\ 1 \end{pmatrix}\begin{pmatrix} y_{-1} & y_0 & y_1 \end{pmatrix} \tag{3-53}$$

The Farrow implementation structure for the three-point interpolation is shown in Fig. 3.14, where the FIR filter has been reduced to three taps and furthermore, the number of non-zero coefficients is decreased significantly.

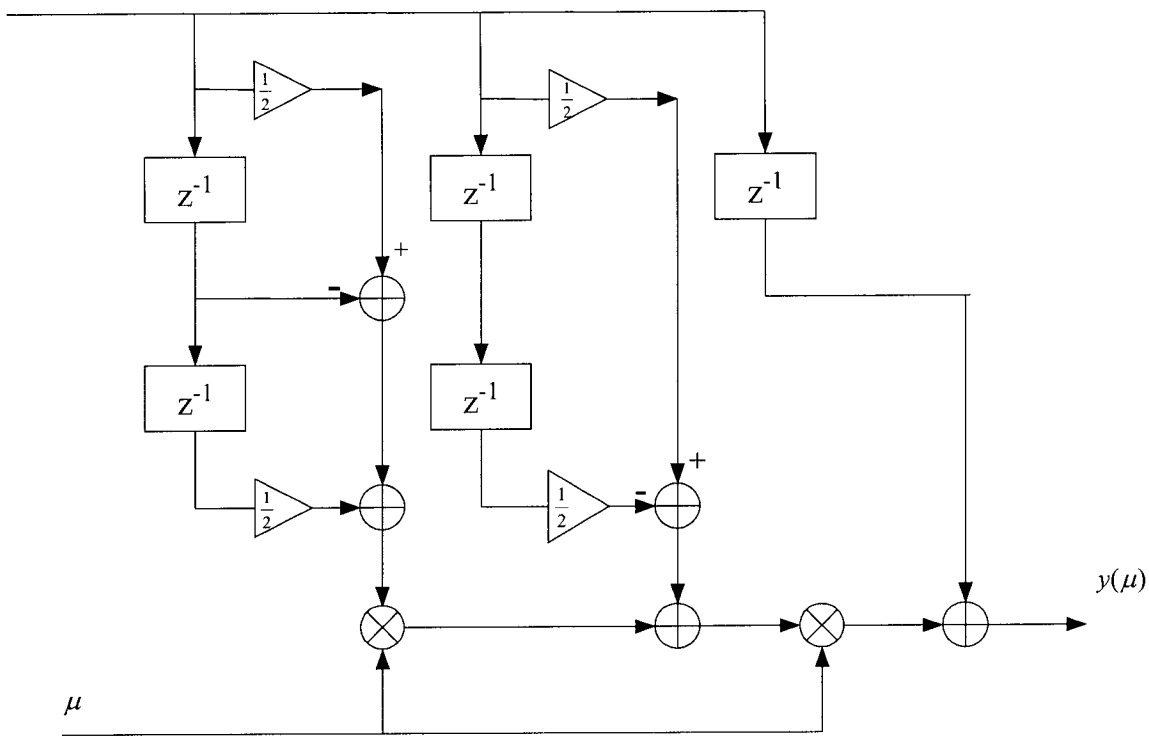


Fig. 3.14. Farrow structure for three-point interpolation

The four-point interpolation is a bit superior to its three-point counterpart in termns of the interpolation precision. However, its implementation complexity is much higher. As such, the second-order three-point interpolation is preferred from the application point of view, especially when the sampling rate of the receiver is two samples per symbol.

C. Linear interpolator

Generally speaking, the interpolation precision depends on the number of samples in addition to the estimation algorithm. The second-order polynomial interpolation that is basically used for two samples/symbol sampling rate is able to give a very satisfactory precision for QPSK modulation, even if the SNR is very low such as less than 3dB. For higher-order modulation such as 8PSK or 16PSK, the second-order interpolation may not be enough if a certain BER performance needs to be guaranteed at a low SNR value. Then we may need four samples/symbol sampling rate for which the proposed timing estimation algorithm is directly applicable. In this case, a fourth-order polynomial that is analogous to the previous four-point and three-point interpolation can readily be used for the approximation of the baseband signal waveform. Nevertheless, in order to reduce the implementation complexity, a simple linear interpolation can be used with little sacrifice in precision. The linear interpolation can be expressed as

$$y_{out} = x_{k-1} + \mu(x_k - x_{k-1}) \tag{3-54}$$

where y_{out} is the output of the interpolator, x_k and x_{k-1} are the two input samples which are closest to the centre point, and μ represents the interpolator parameter which can be calculated from the estimated timing offset ϕ,

$$\mu = \begin{cases} 4\phi & 0 \le \phi < \frac{1}{4} \\ 4\phi - 1 & \frac{1}{4} \le \phi < \frac{1}{2} \\ 4\phi - 2 & \frac{1}{2} \le \phi < \frac{3}{4} \\ 4\phi - 3 & \phi \ge \frac{3}{4} \end{cases}. \tag{3-55}$$

In (3-55), we have assumed that ϕ is normalized to the interval [0,1].

The linear interpolation needs only one multiplication and two additions for each output sample. It is worth mentioning that the linear interpolation can also be applied for two samples/symbol sampling rate in BPSK or QPSK modulation, where the SNR is not a major concern.

3.6. Summary

In this chapter, a new symbol timing recovery scheme has been developed. A timing estimation algorithm has been proposed for two samples per symbol or higher sampling rate. The variance of the timing estimate has been analyzed to show the performance of

the proposed estimation algorithm. The proposed method has been simulated, giving the variance plots of the timing estimate for various parameters such as the estimation interval, low-pass filter and the roll-off factor. Some of the implementation issues have also been discussed. Based on the analysis and computer simulation results, the following remarks/claims can be made.

1. The variance of the timing estimate can be reduced by increasing the length of the estimation interval. However, increasing the estimation length would require more computations in the averaging unit and make the tracking of the timing slower.
2. Higher-order IIR filters give smaller variances when the SNR is low, and do not improve the variance when there is no or little AWGN noise.
3. The proposed estimation algorithm performs better for large roll-off factors ($\alpha >$ 0.5).
4. The second-order three-point interpolation gives a satisfactory interpolation for the QPSK modulation. The linear interpolation can be used for high SNR environments, where the cost of the receiver instead of the SNR is a major issue.

Chapter 4

Improved Timing Estimation Algorithms

In Chapter 3, we have developed a basic timing estimation algorithm that uses the in-phase (I) or quadrature (Q) signal only. In this chapter, we present a few methods to improve the estimation performance by taking both the I and Q signals into account and by employing a post-processing technique.

4.1. Timing Estimation Utilizing Both I and Q Signals

In this section, we will demonstrate that the timing estimation performance can be improved significantly if both the I and Q signals are simultaneously applied. Fig. 4.1 shows a modified scheme that utilizes both the I and Q signals to compute the timing estimate, where $r_I(t)$ and $r_Q(t)$ are the real and imaginary parts of the received complex baseband signal, i.e.,

$$r(t) = r_I(t) + jr_Q(t). \tag{4-1}$$

The I and Q signals are each multiplied by a 1/2 symbol rate quadrature sine sequence and then low-pass filtered and squared to construct two complex pairs, $u_I(t) + jv_I(t)$,

and $u_Q(t) + jv_Q(t)$,

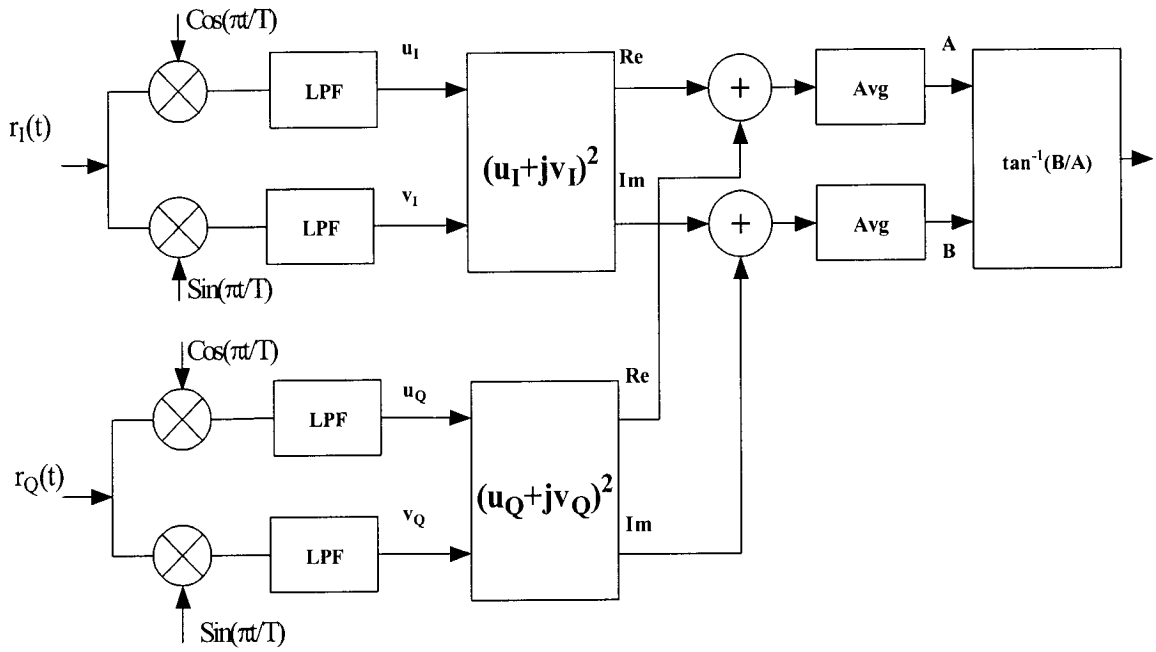


Fig.4.1. Modified estimation algorithm

$$y_I(t) = u_I(t) + jv_I(t) = [r_I(t)e^{j\frac{\pi t}{T}}] * h(t), \tag{4-2}$$

$$y_Q(t) = u_Q(t) + jv_Q(t) = [r_Q(t)e^{j\frac{\pi t}{T}}] * h(t). \tag{4-3}$$

The real parts as well as the imaginary of $y_I(t)$ and $y_Q(t)$ are added and averaged, to calculate the timing estimate. It is evident that the estimate can be written as

$$\hat{\phi} = \frac{1}{2\pi}\arg(Z_I + Z_Q) \tag{4-4}$$

where Z_I and Z_Q represent the complex numbers obtained from the I and Q signals, respectively, are given by

$$Z_I = \sum_{n=0}^{LN-1} y_I^2(nT_S) \tag{4-5}$$

$$Z_Q = \sum_{n=0}^{LN-1} y_Q^2(nT_S) \tag{4-6}$$

Using the approximation formula in (3-32), the mean value of $\hat{\phi}$ can be written as

$$E(\hat{\phi}) = \frac{1}{2\pi}\arg[E(Z_I + Z_Q)] = \frac{1}{2\pi}\tan^{-1}\frac{E[\mathrm{Im}(Z_I + Z_Q)]}{E[\mathrm{Re}(Z_I + Z_Q)]} \tag{4-7}$$

According to the discussion in the previous chapter, each of Z_I and Z_Q gives an unbiased estimate for ϕ, which implies that $\arg[E(Z_I)] = \arg[E(Z_Q)] = \phi$. Using this result, it is easy to verify that $E(\hat{\phi}) = \phi$ in (4-7), implying that the estimate given by (4-4) is also unbiased.

Although it is possible to derive a closed-form expression for the variance of the above timing estimate, yet our objective is to demonstrate that the estimate given by (4-4) has a smaller variance compared to the estimate obtained by the basic estimation algorithm. For this purpose, we substitute $\hat{\phi}$ from (4-4) into (3-33), yielding

$$Var(\hat{\phi}) = \frac{1}{4\pi^2}\frac{E\left([\mathrm{Im}(Z_I) + \mathrm{Im}(Z_Q)]^2\right)}{\left(E[\mathrm{Re}(Z_I)] + E[\mathrm{Re}(Z_Q)]\right)^2} \tag{4-8}$$

Since the I and Q signals have the same symbol energy, we have

$$E[\mathrm{Re}(Z_I)] = E[\mathrm{Re}(Z_Q)] \tag{4-9}$$

Thus, the denominator of (4-8) is quadrupled compared to the case of using the I or Q signal only. The numerator can be denoted as $E[(X_1 + X_2)^2]$ for notational convenience, where X_1 and X_2 represent, the imaginary part of Z_I and that of Z_Q, respectively. From the discussion in Section 3.2, we have $E(X_1) = E(X_2) = 0$ since we assume that

$\varepsilon = 0$ as in the basic algorithm. As such, it is easy to verify that

$$\left|E(X_1 X_2)\right| \le \sqrt{E(X_1^{\,2})E(X_2^{\,2})} = E(X_1^2) \tag{4-10}$$

which in turn leads to

$$E\left[(X_1 + X_2)^2\right] = E(X_1^2) + E(X_2^2) + 2E(X_1 X_2) \le 4E(X_1^2) \tag{4-11}$$

The above inequality holds only if X_1 and X_2, i.e., $\mathrm{Im}(Z_I)$ and $\mathrm{Im}(Z_Q)$, are linearly correlated, which, however, is not possible even though they may be correlated somehow because of the relationship between the transmitted I and Q symbols. The above observation has shown that the numerator in (4-8) does not increase as much as the denominator does. Therefore, the estimate given by the modified scheme always has a smaller variance.

The simulation results convincingly show a significant reduction in the variance of the estimate resulting from the above modified algorithm, even for very low SNR channels. Fig. 4.2 depicts the variance curves obtained from the basic and the modified algorithms for comparison. It is seen that the variance for the modified algorithm is roughly equivalent to that for the basic version with a double estimation length.

It should be mentioned that the above modified algorithm can also be applied to other feedforward symbol synchronization techniques such as the O&M and maximum likelihood estimation techniques reviewed in Chapter 2.

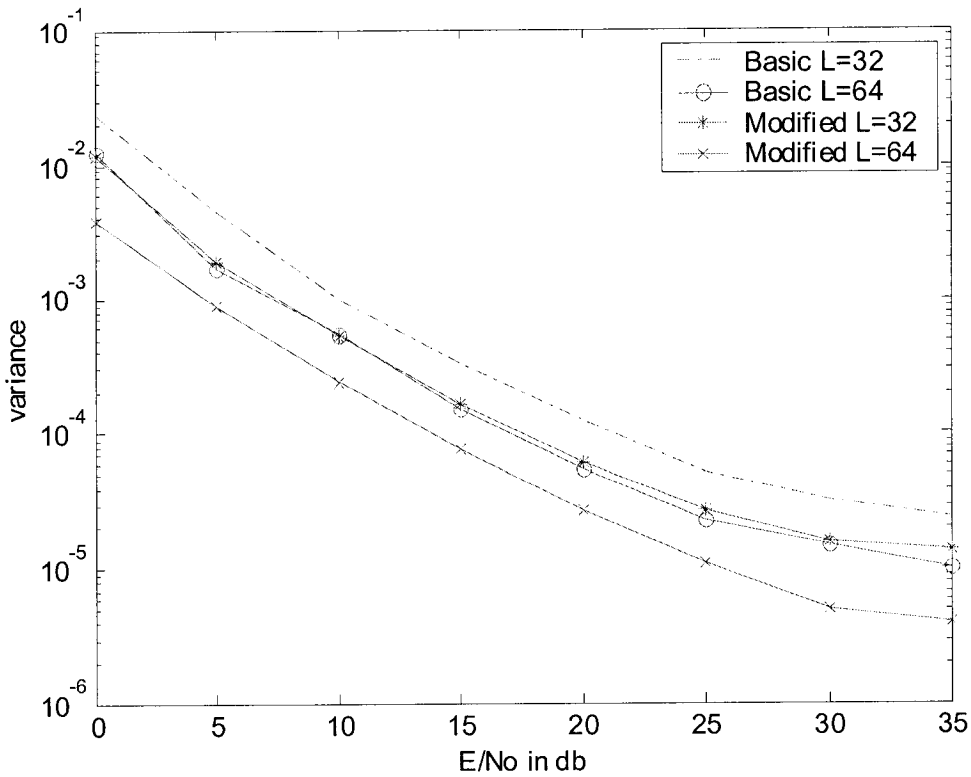


Fig. 4.2. Variance of the timing estimate

4.2. Smoothing of Timing Estimate using Post-Processing Technique

The timing estimate can be viewed as a constant for the duration of a number of symbols. Therefore, we can apply a post-processing technique such as Kalman filtering to smooth the timing estimate. In the following, we propose a few Kalman filtering-based post-processing methods to reduce the timing estimate obtained in the previous chapter.

The Kalman filtering problem can be formulated as [18] [19]

$$\mathbf{x}(n+1) = \mathbf{F}(n)\mathbf{x}(n) + \mathbf{w}_1(n) \tag{4-12}$$

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{w}_2(n) \tag{4-13}$$

where $\mathbf{x}(n)$ is called the state vector at time n, $\mathbf{y}(n)$ the measurement vector, $\mathbf{w}_1(n)$ and $\mathbf{w}_2(n)$ the noise vectors representing the process noise and the measurement noise, respectively, $\mathbf{F}(n)$ the transition matrix, and $\mathbf{C}(n)$ the measurement matrix. It is normally assumed that $\mathbf{w}_1(n)$ and $\mathbf{w}_2(n)$ are zero-mean white noise process with the following correlation matrices,

$$E[\mathbf{w}_1(n)\mathbf{w}_1(k)] = \begin{cases} \mathbf{Q}(n) & (n = k) \\ \mathbf{0} & (n \neq k) \end{cases} \tag{4-14}$$

$$E[\mathbf{w}_2(n)\mathbf{w}_2(k)] = \begin{cases} \mathbf{R}(n) & (n = k) \\ \mathbf{0} & (n \neq k) \end{cases} \tag{4-15}$$

where $\mathbf{Q}(n)$ and $\mathbf{R}(n)$ are diagonal matrices. It is also assumed that $\mathbf{w}_1(n)$ and $\mathbf{w}_2(n)$ are statistically independent and both are independent of observed signal/data. Fig. 4.3 shows a typical Kalman filter structure, where the observed data $\mathbf{y}(n)$ serve as the original timing estimate and the estimate of $\hat{\mathbf{x}}(n+1)$, the state vector at time $n+1$, serves as the smoothed/filtered version of the estimated timing offset. As seen from the

diagram, the estimate of the state vector can be written as

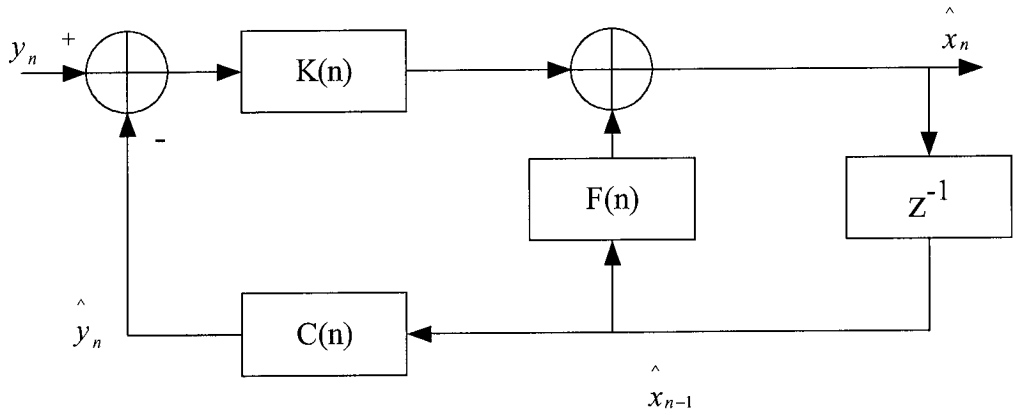


Fig 4.3 Kalman filter architecture

$$\hat{\mathbf{x}}(n+1) = \mathbf{F}(n)\hat{\mathbf{x}}(n) + \mathbf{K}(n)[\mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n)] \tag{4-16}$$

where $\hat{\mathbf{x}}(n)$ is the estimate of the state vector at time n, and $\mathbf{K}(n)$ is an unknown matrix, referred to as the Kalman gain, that can be determined by minimizing the correlation matrix of the state error vector. The correlation matrix is formulated as

$$\mathbf{P}(n) = E[\mathbf{e}(n)\mathbf{e}^T(n)] \tag{4-17}$$

where

$$\mathbf{e}(n) = \mathbf{x}(n) - \hat{\mathbf{x}}(n) \tag{4-18}$$

Minimization of $\mathbf{P}(n)$ leads to a few updating formulas for the computation of $\mathbf{K}(n)$, i.e.,

$$\mathbf{K}(n) = \mathbf{F}(n)\mathbf{P}(n)\mathbf{C}^T(n)[\mathbf{C}(n)\mathbf{P}(n)\mathbf{C}^T(n) + \mathbf{R}(n)]^{-1} \tag{4-19}$$

$$\mathbf{P}'(n) = \mathbf{P}(n) + \mathbf{F}^{-1}(n)\mathbf{Q}(n-1)[\mathbf{F}^{-1}(n)]^T \mathbf{P}(n) \tag{4-20}$$

$$\mathbf{P}(n+1) = \mathbf{F}(n)\mathbf{P}'(n)[\mathbf{F}(n)]^T + \mathbf{Q}(n) \tag{4-21}$$

In this study, we use the scalar-form Kalman filter to improve the timing estimation performance. Thus, all the matrices in (4-19)-(4-21) are reduced to scalars. We also assume that $\mathbf{F}(n)$ and $\mathbf{C}(n)$ are the identity matrix, i.e., the unity in the scalar case. Then, the updating formulas for the evaluation of the Kalman gain are simplified as

$$K(n) = P(n)[P(n) + R(n)]^{-1} \tag{4-22}$$

$$P'(n) = P(n) + Q(n-1)P(n) \tag{4-23}$$

$$P(n+1) = P'(n) + Q(n). \tag{4-24}$$

4.2.1 Kalman Filtering of the Timing Estimate — Scheme 1

As shown in Fig. 4.4, an obvious approach of post-processing is to apply a Kalman filter to the timing estimate $\hat{\phi}$ obtained from the basic algorithm in Chapter 3, yielding a smoothed version of $\hat{\phi}$ as given by

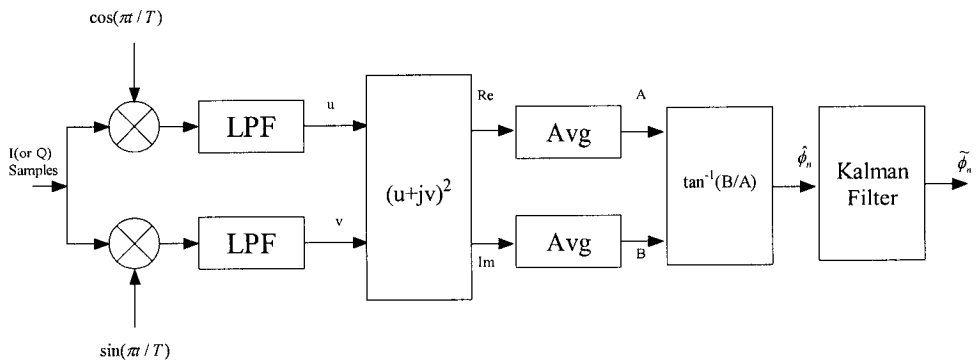


Fig. 4.4. Post-processing using Kalman filtering — Scheme 1

$$y_n = \hat{\phi}_n$$

$$\hat{x}_n = \tilde{\phi}_n$$

$$\tilde{\phi}_n = \tilde{\phi}_{n-1} + K(n)[\hat{\phi}_n - \tilde{\phi}_{n-1}] \qquad (4\text{-}25)$$

where $\hat{\phi}_n$ represents the timing offset estimated at time n (after n symbols have passed), and $\tilde{\phi}_n$ is the smoothed version of $\hat{\phi}_n$. To calculate the Kalman gain $K(n)$, we need to determine the initial values of $Q(n)$, $R(n)$ and $P(n)$ as required by (4-22) through (4-24).

In our simulation, $Q(n)$ and $R(n)$ are treated as constant. We have found that the smoothing performance depends to a considerable degree on the ratio of $R(n)$ to $Q(n)$. As for the initial value of $P(n)$, we choose $P(0) = 0$, since the initial input estimate $\hat{\phi}_0$ can be used as $E(\hat{\phi}_0)$ and thus $P(0) = E\left([\hat{\phi}_0 - E(\hat{\phi}_0)]^2\right) = 0$. Fig. 4.5 shows the result using Kalman filtering with $Q(n) = 1$ and $R(n) = 0.05$. It is clear that the amplitude of the timing estimation error has been suppressed significantly. Fig. 4.6 shows the variance curves for both the Kalman-filtered and non-Kalman-filtered timing estimates with different estimation lengths. It is seen that the variance has been reduced using the Kalman filter regardless of the length of the estimation interval.

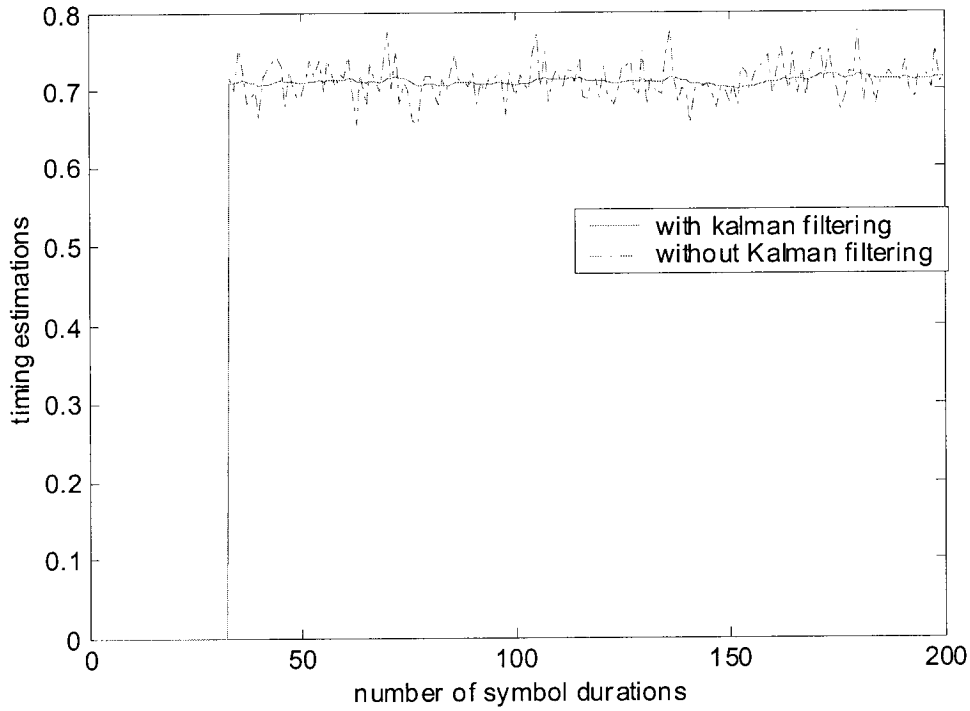


Fig. 4.5. Kalman filtering of the timing estimate

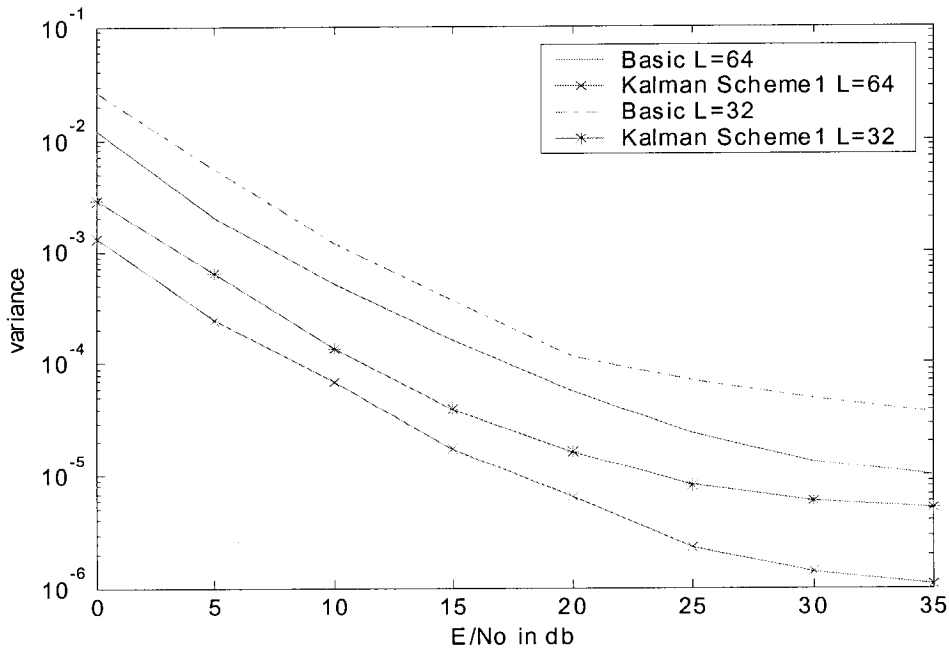


Fig. 4.6. Variance of the timing estimate with Kalman filtering Schme 1

4.2.2. Kalman Filtering of A and B Values — Scheme 2

Another approach of smoothing the timing estimate is to apply two Kalman filters to the averaged values A and B in Fig. 3.1., which can be shown in Fig. 4.7.

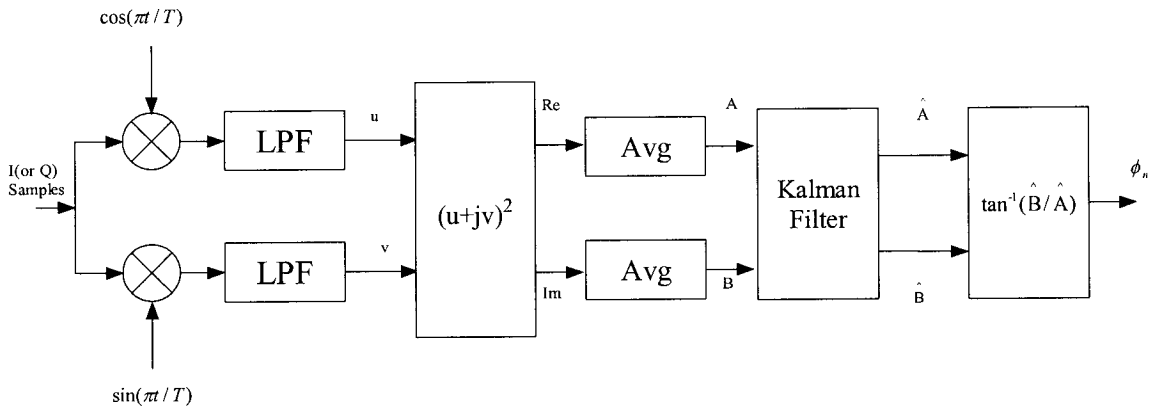


Fig. 4.7. Post-processing using Kalman filtering Scheme 2

The A and B values from the averaging module at time n can be expressed as the following vector form

$$y_n = \begin{bmatrix} A_n \\ B_n \end{bmatrix}.$$

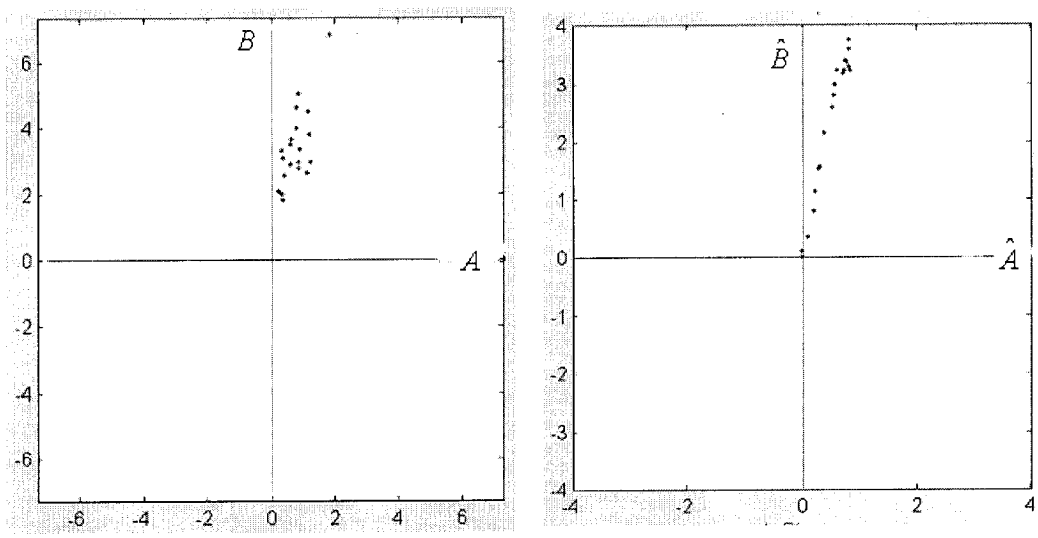
After Kalman filtering, A and B are smoothed, and denoted as

$$\hat{x}_n = \begin{bmatrix} \hat{A}_n \\ \hat{B}_n \end{bmatrix}.$$

From (4-19) to (4-21), we obtain an update formula for $\hat{A}_n$ and $\hat{B}_n$ as

$$\begin{bmatrix} \hat{A}_n \\ \hat{B}_n \end{bmatrix} = \begin{bmatrix} \hat{A}_{n-1} \\ \hat{B}_{n-1} \end{bmatrix} + K(n)\left(\begin{bmatrix} A_n \\ B_n \end{bmatrix} - \begin{bmatrix} \hat{A}_{n-1} \\ \hat{B}_{n-1} \end{bmatrix} \right) \tag{4-26}$$

Fig. 4.8 shows the A and B values as complex pairs plotted in the two-dimensional plane, where the horizontal and vertical axes represent the value of A and that of B, respectively. In this simulation, we have chosen $Q(n) = 1$ and $R(n) = 0.05$ as discussed in Section 4.2.1.



(a) (b)

Fig.4.8. Plot of A and B values: (a) Before Kalman filtering (b) After Kalman filtering

Fig. 4.9 shows the variance curves of the timing estimate obtained from the Kalman filtered A and B values. As expected, similar to Scheme 1, this scheme also provides an

improved estimation performance. It is worth mentioning that although Scheme 2 needs two Kalman filters, which doubles the computational complexity of the post-processing unit in contrast to Scheme 1, it has another advantage in addition to reducing the variance of the timing estimate.

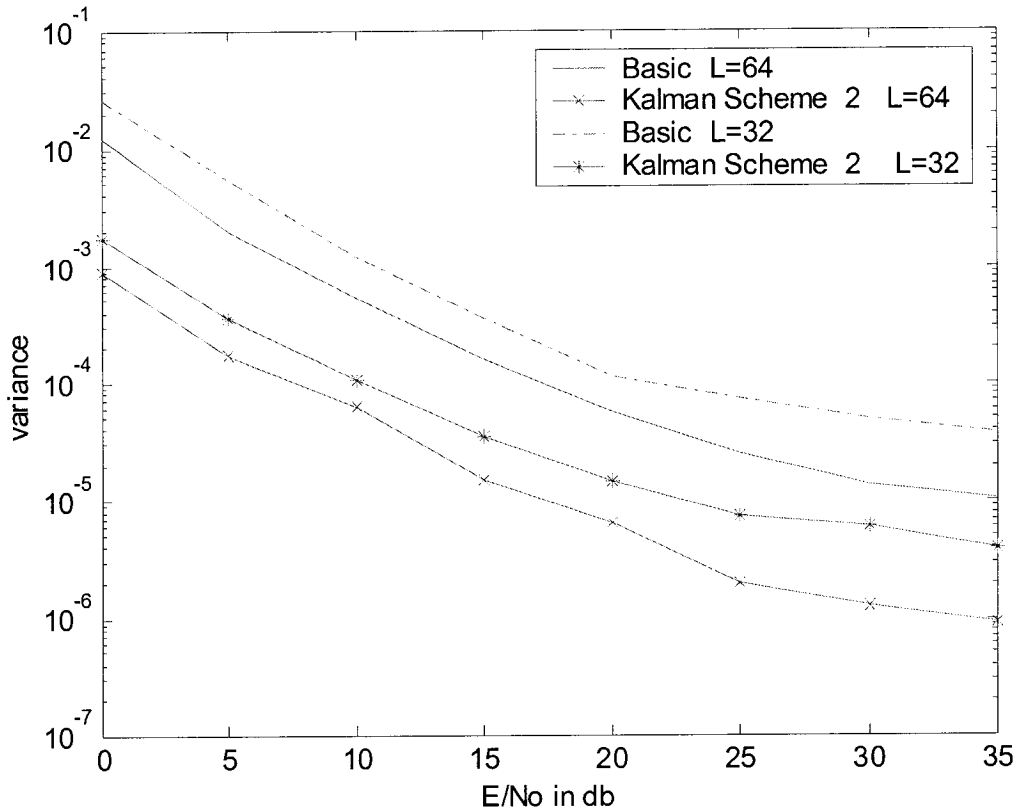


Fig. 4.9. Variance of timing estimate with Kalman filtering — Scheme 2

In feedforward timing estimation, timing phase slip will possibly occur due to the phase ambiguity of arctan(B/A) between 0 and 2π. When $A>0$ and $B\approx 0$, arctan(B/A) may result in 0 or 2π depending on the sign of B. Due to the noise and the deviation of the timing

estimate, the value of B may vary around 0 from a small negative value to a small positive one. Thus, the estimated timing phase could slip between 0 and 2π, when the clock phase of the local sampler is nearly synchronized with the symbol timing. The phase slip does not create inter-symbol interference, but it causes the interpolator either to skip one right sample or to repeat one and therefore, produces a symbol error.

By using Kalman filtering Scheme 2, the possibility of phase slipping can be decreased greatly, since, after Kalman filtering, the values of A and B tend to be more convergent around the straight line of a slop being equal to the true timing offset. Fig. 4.10 shows that phase slip has occurred, where the estimated phase jumps between γ and $2\pi - \gamma$. Fig. 4.11 illustrates that the phase slipping did not happen since the Kalman filtering Scheme 2 has been used and the values of A and B have been smoothed.

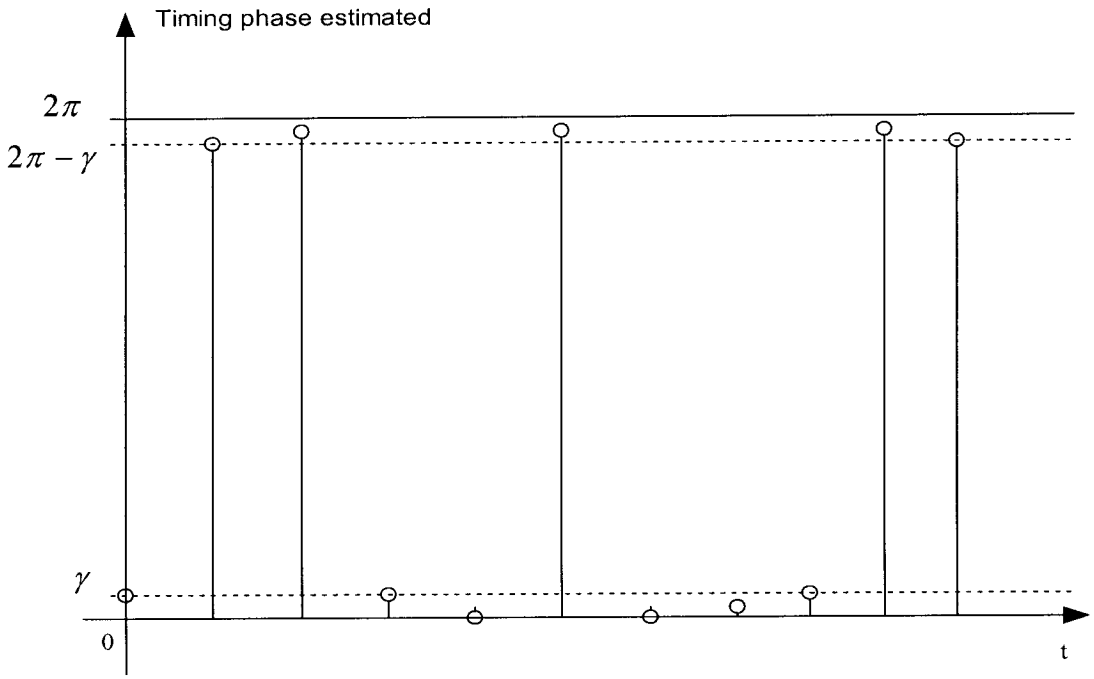


Fig. 4.10. Phase slipping before Kalman filtering Scheme 2

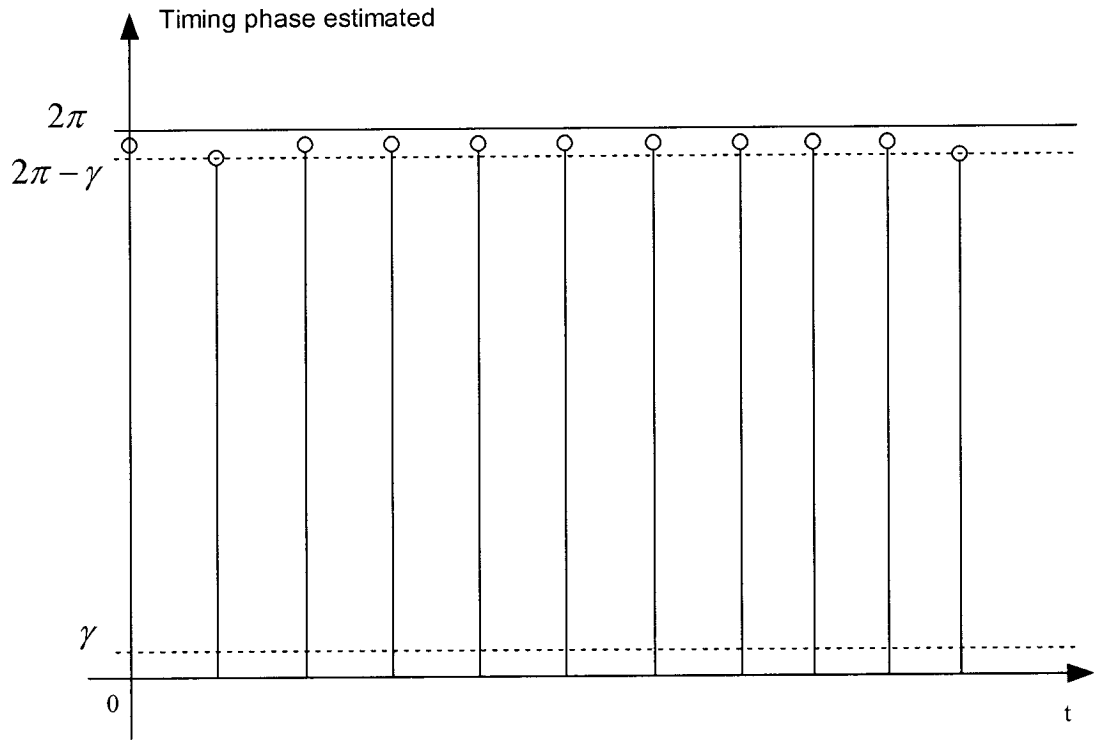


Fig. 4.11. No phase slipping after Kalman filtering Scheme 2

4.2.3. Simplified Smoothing Scheme — Scheme 3

The above Kalman filter is equivalent to a first-order recursive filter with a time-varying coefficient $K(n)$. If the Kalman gain $K(n)$ remains a constant in each iteration, then the Kalman filter is reduced to a regular IIR filter with the following transfer function

$$H(z) = \frac{k}{1-(1-k)z^{-1}}. \tag{4-27}$$

We may use (4-27) for the post-processing of the timing estimate $\hat{\phi}$ as shown in Fig. 4.12. Although (4-27) offers the smallest computational complexity, it does not work well for high SNR channels. Fig 4.13 shows the variance curves for the simplified smoothing scheme along with the basic estimation method without smoothing, when the value of k in (4-27) is chosen as 0.2. It is seen that the simplified scheme performs well if the SNR is smaller than 15 db, but it does not reduce the variance when the SNR is larger than 25 db compared to the basic algorithm. In any case, the larger the length of the estimation interval is, the better the estimation result can be obtained.

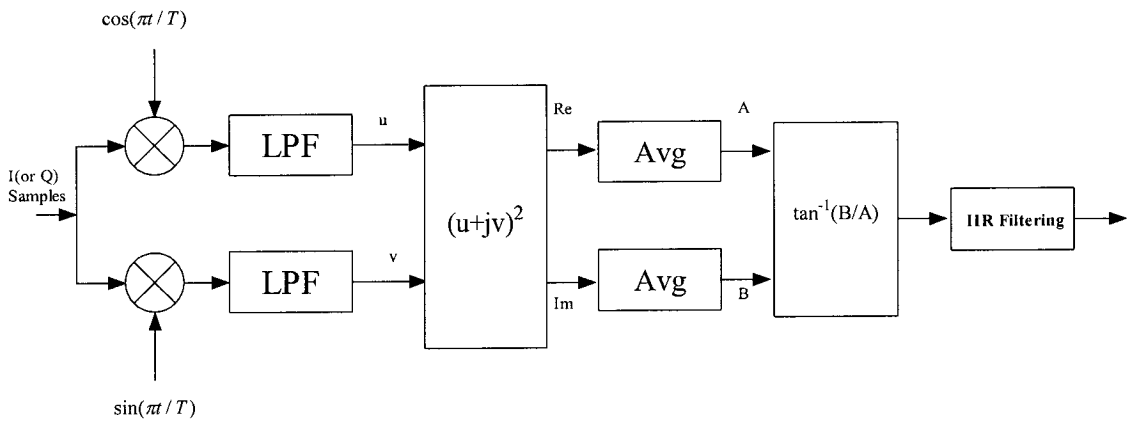


Fig. 4.12. Basic timing estimation followed by the 1st-order IIR filtering

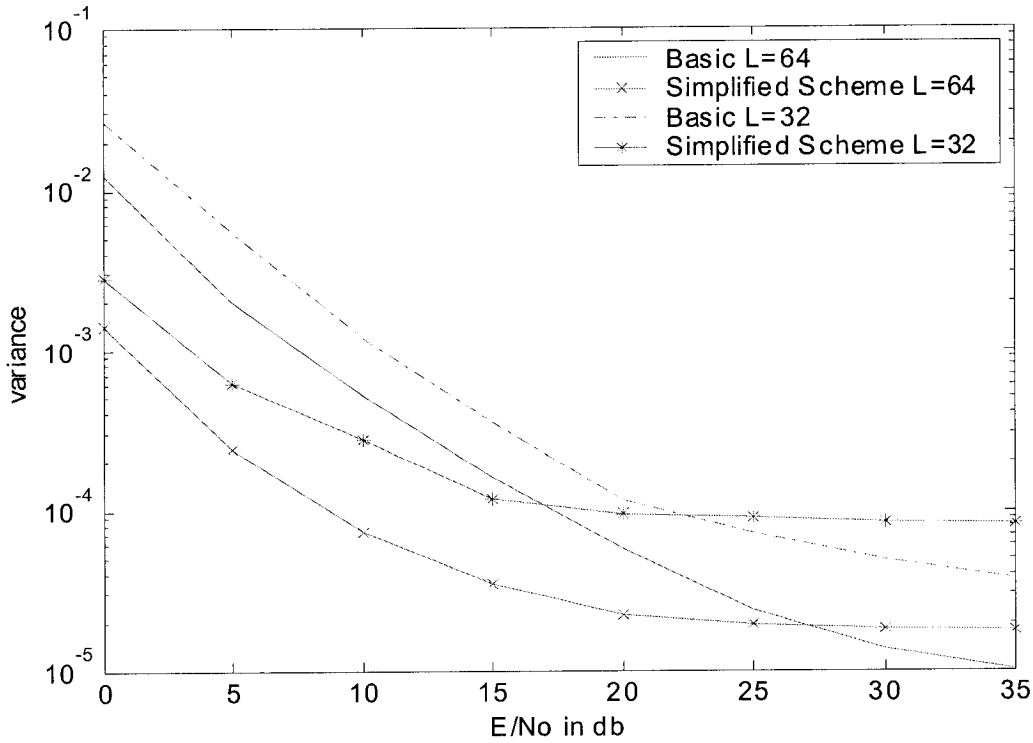


Fig 4.13 Variance of the timing estimate with the simplified scheme

4.2.4. Comparison of Three Smoothing Schemes

Fig.4.14 shows the variance plots for the three proposed smoothing schemes together with that of the basic algorithm for the purpose comparison. The estimation length is set to 32 symbols in this simulation. A similar comparison of the three schemes for L=64 is also made in Fig. 4.15. It is of interest to note that Schemes 1 and 2 result in a very similar performance, whereas Scheme 3 performs very well only if the SNR is under 15 db. As for the implementation complexity, Scheme 3 is the simplest one and therefore, is preferred in communication systems where there exists heavy noise. Scheme 2 has the

largest complexity , but offers the advantage of eliminating the timing phase slipping. Generally speaking, Scheme 1 provides a good compromise between the complexity and performance.

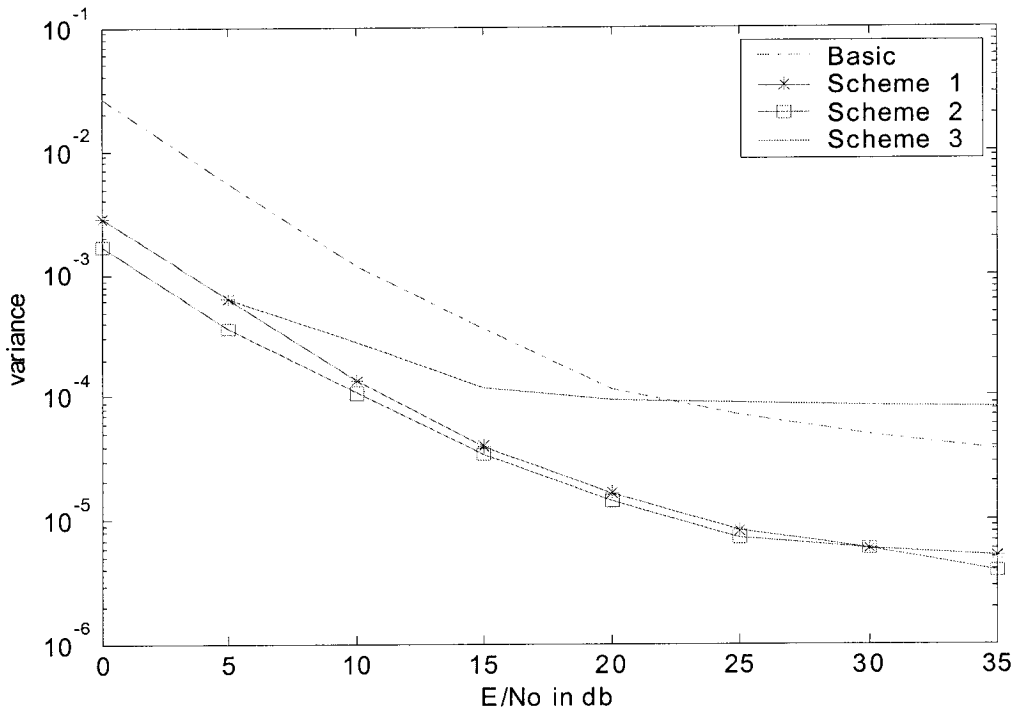


Fig.4.14. Comparison of three smoothing schemes with L=32

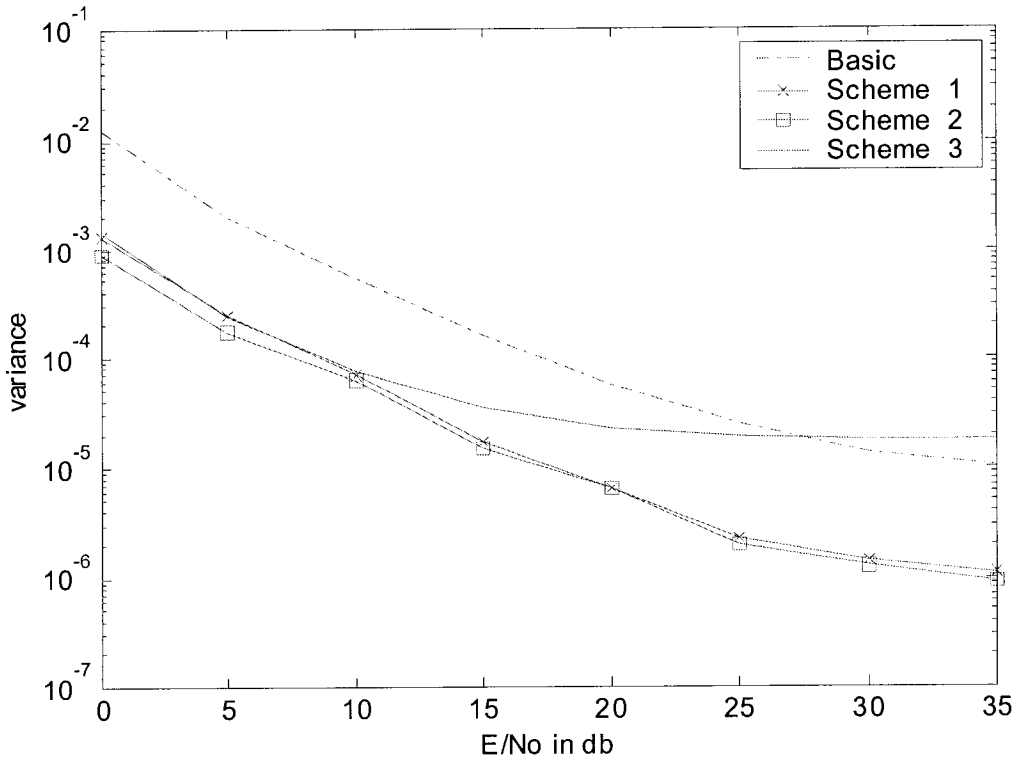


Fig.4.15. Comparison of three smoothing schemes with L=64

4.3. Summary

In this chapter, a few improved estimation algorithms have been presented to reduce the variance of the timing estimate. Firstly, the basic estimation algorithm developed in Chapter 3 has been modified by employing both the I and Q signals. Theoretical analysis as well as computer simulation has shown that the modified scheme gives a smaller variance. Similar to the basic algorithm, the modified scheme using both the I and Q

signals enjoy a short acquisition time for the timing offset, making it possible to track the symbol timing in burst-mode communication systems. Secondly, the Kalman filter has been exploited to smooth the timing estimate. For this purpose, three schemes, the direct Kalman filtering of the timing estimate, the Kalman filtering of the A and B values, and the simplified smoothing scheme using only a first-order IIR filter, have been proposed. The three schemes have been simulated and compared with the basic algorithm without post-processing. Different scenarios for which each of the three schemes is suited have also been identified. The first scheme works very well on an overall basis and appears to be a good solution counting performance and complexity as well, the second one is able to prevent the timing phase from slipping between 0 and 2π, and the third one is a very good alternative for low SNR channels.

Chapter 5

Conclusion

5.1. Summary of Research Results

In this thesis, an in-depth study of a broad class of feedforward symbol synchronization techniques has been conducted. Some of the existing feedback and feedforward synchronization methods have been reviewed, showing their advantages and drawbacks. Motivated by the advantage of the feedforward synchronization in its fast acquisition of symbol timing and straightforward digital implementation, a few feedforward timing estimation schemes have been proposed for burst-mode communication systems.

A basic timing estimation scheme that uses either the in-phase or the quadrature signal has been proposed. The mean and variance of the timing estimate has been analyzed to show the estimation performance of the proposed algorithm. The proposed method, consisting of modulation, low-pass filtering, squaring and averaging operations, has been simulated, and the variance plots of the timing estimate given for different parameters such as the estimation interval, low-pass filter and the roll-off factor. It has been shown that the variance of the timing estimate can be reduced by increasing the length of the estimation interval. However, increasing the estimation length would require more

computations in the averaging unit and this make the tracking of the timing slower. Higher-order IIR filters yield a smaller variance when the SNR is low, and do not improve the variance when there is no or little AWGN noise. As expected, the proposed estimation algorithm performs better for large roll-off factors ($\alpha > 0.5$). Some of the implementation issues have also been discussed. The proposed synchronization method needs an interpolation among signal samples, once the symbol timing is estimated. It has been shown that a second-order three-point interpolation gives a satisfactory precision for the QPSK modulation. The linear interpolation can be used for high SNR environments, where the cost of the receiver rather than the SNR is a major concern.

A few improved estimation algorithms have been presented to reduce the variance of the timing estimate. Firstly, the basic timing estimation algorithm has been modified by exploiting both the in-phase and quadrature signals for the estimation of the symbol timing. Both the theoretical analysis and computer simulation have shown that the modified scheme gives a significantly reduced variance of the timing estimate. Secondly, the Kalman filter has been exploited to smooth the timing estimate. For this purpose, three schemes, the direct Kalman filtering of the timing estimate, the Kalman filtering of the real and imaginary parts of a complex value for the computation of the timing phase, and the simplified smoothing scheme using only a first-order IIR filter, have been proposed. The three schemes have been simulated and compared with the basic algorithm without post-processing. Different scenarios for which each of the three schemes is suited have also been identified. The first scheme, working very well in general, appears to be a

good solution taking both performance and complexity into consideration; the second one is able to prevent the timing phase from slipping between 0 and 2π; and the third one is a very good alternative for low SNR channels.

5.2. Suggestions for Future Study

The simulation study performed in this thesis is basically for QPSK-modulated signals in a AWGN channel. For QAM modulation, such as 16 QAM and 64QAM, the proposed algorithms should be directly applicable. However, their performance needs to be examined through simulation. Application of the proposed synchronization methods to multipath fading channel might be possible. Although a theoretical justification of the proposed algorithms for use in multipath fading channel may be difficult, it is worth-while to carry out a simulation study in this aspect. Therefore, the simulation of the transmission of QPSK and QAM signals in multipath fading channel using the proposed synchronization method is recommended.

The Kalman filtering-based post-processing scheme has been presented to reduce the variance of the estimated symbol timing. One may attempt adaptive algorithms such as the LMS (least mean square) and RLS (recursive least square) algorithms to implement a fast smoothing scheme for the timing estimate, since a fast estimation of the timing offset is very important for burst-mode communications.

References

1 Yupeng Yan, W.-P. Zhu, M.N.S. Swamy and M. O. Ahmad, "A feedforward symbol timing recovery technique for burst-mode communication systems" *Micronet Annual Workshop 2003*, Toronto, Canada, May 2003

2 W.-P. Zhu, Yupeng Yan, M.O. Ahmad and M.N.S. Swamy, "A feedforward timing recovery scheme using two samples per symbol: algorithm, performance and implementation issues," in *Proc. IEEE ISCAS2003*, Bangkok, Thailand, May 2003.

3 W.-P. Zhu, M.O. Ahmad and M.N.S. Swamy, "A fully digital timing recovery scheme using two samples per symbol," in *Proc. IEEE ISCAS2001,* pp. 421-424 Sydney, Australia, May 2001.

4 F. M. Gardner, "A BPSK/QPSK timing-error detector for sampled data receivers," *IEEE Trans. Commun.*, vol. 34, pp. 423-429, May 1986.

5 K. H. Lee, Y. H. Kim, and H. J. Choi, "A new symbol timing recovery for all-digital high speed symbol synchronization," *IEICE Trans. Commun.* vol. E80-B, no.9, pp.1290-1299, Sep. 1997.

6 U. Lambrette, K. Langhammer and H. Meyr, "Variable sample rate digital feedback NDA timing synchronization," in *Proc. of IEEE Globecom '96*, vol. 2, pp. 1348–1352, Nov. 1996.

7 F. M. Gardner, "Interpolation in digital modems — part 1: fundamentals," *IEEE Trans. Commun.*, vol. 41, no. 3, pp. 501-507, March 1993.

8 L. Erup, F. M. Gardner and R. A. Harrris, "Interpolation in digital modems—part 2: implementation and performance," *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 501-507, June 1993.

9 D. Kim, M. J. Narasimha and D.C. Cox, "Unbiased timing-error estimation in the presence of nonideal interpolation," *IEEE Trans. Commun.*, vol. COM-45, pp. 647-650, June 1997.

10 M. Oerder and H. Meyr, "Digital filter and square timing recovery", *IEEE Trans. Commun.*, vol. 36, no. 6, pp. 605-612, May 1988.

11 D. Efstathiou and A. H. Aghvami, "Preamle-less nondecicsion-aided feedforward synchronization techniques for 16-QAM TDMA demodulators," *IEEE Trans. Vehicular Technology,* vol. 47, no.2, pp. 673-685, May 1998.

12 D. Fu and A. N. Willson, Jr., "A fast synchronizer for burst modems with simultaneous symbol timing and carrier phase estimations," in *Proc. IEEE ISCAS2000*, pp. 379 –382, Geneva, Switzerland, May 2000.

13 J. H. Grimm, R. N. Kumar, J. Kusuma and J. V. Krogmeier, "Real time implementation of a symbol timing recovery algorithm for narrow-band wireless modem,"in *Proc. of IEEE ICASSP'98*, vol. 6, pp. 3253-3256, May 1998.

14 J. Vesma, M. Refors, and J. Rinne, "Comparison of efficient interpolation techniques for symbol timing recovery," in *Proc. IEEE Globecom'96,* London, UK, pp.953-957, Nov. 1996.

15 M. B. Breinholt and M. D. Zoltowski, "An open-loop cyclostationarity-based timing

recovery algorithm for accelerated timing acquisition in frequency-selective channels," in *Proc. of IEEE ICASSP'2002*, vol. 3, pp. 2293-2296, April 2002.

16 H. Meyr, M. Moeneclaey and S.A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation and Signal Processing*, John Wiley & Sons, Inc., 1998.

17 L.P.Sable and W.G.Cowley, "Block processing feedforward symbol timing estimator for digital modems", *Electronics Letters*, vol.30, pp1273-1274, Aug. 1994.

18 B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-hall, Inc., 1979.

19 Simon Haykin, *Adaptive Filter Theory,* 3rd Edition, Prentice-Hall, Inc., 1996.

20 W.G. Cowley and L.P. Sabel, ``The performance of two symbol timing recovery algorithms for PSK demodulators," *IEEE Trans. Commun.*, vol. COM-42, pp. 2345-2355, June 1994.

21 A. Haoui, H.H. Lu and D. Hedberg, ``An all digital timing recovery scheme for voiceband data modem," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp.1911-1914, April 1987.

22 H.K. Yang and M.Snelgrove, ``Symbol timing recovery using oversampling techniques," in *Proc. IEEE ICC'96*, pp.1296-1300, June 1996.

23 N.R. Sollenberger and J.-I. Chuang, ``Low-overhead symbol timing and carrier recovery for TDMA portable radio systems," *IEEE Trans. Commun.*, vol. COM-38, pp. 1886-1892, Oct. 1990.

24 G. De Jonhe and M. Moeneclaey, "Cycle slip analysis of the NDA FF carrier synchronizer based on Viterbi&Viterbi algorithm" *in Proc. ICC'94,* pp.880-884, May 1994

25 G. De Jonhe and M. Moeneclaey, "The effect of the averaging filter on cycle slipping of NDA feedforward carrier synchronizer for MPSK", in *Proc. ICC'92*, Chicago, pp.365-369, June 1992.

26 J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, 1995.

27 S. Haykin, *Digital Communications*, Wiley, New York, 1988.

28 A. V. Oppenheim, R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, New Jersey, 1989.

29 M. Abramowitz and I.A. Stegun, Eds., *Handbook of mathematical Functions.* Nat. Bur. Stds., Appl. Math. Series, vol. 55, p.878, June 1964.

30 D. Fu and A. N. Wilson, Jr., "A high speed processor for retangular-to-polar conversion with applications in digital communications," in *Proc. Globecom 1999*, vol. 4, pp.2172-2176, Dec.1999.

31 A. Chen, et al., "Modified Cordic demodulator implementation for digital IF-sampled receiver," in *Proc. Globecom 1995*, vol2, pp.1450-1454, Nov. 1995.

32 J. Vesma, V. Tuukkanen, M. Renfors, "Maximum likelihood feedforward symbol timing recovery based on efficient digital interpolation techniques," in *Proc .IEEE Nordic Signal Processing Symp.*, pp.183-186, Espoo, Finland, Sep. 1996.

33 C. W. Farrow, "A continuously variable digital delay element", in *Proc ISCAS 1988.* ,

pp.2614-2645, Espoo, Finland, June 1988.

34 Robert Morawski, Low-Complexity Structures for Digital Symbol Timing Recovery, M. A. Sc. thesis, Concordia University, 2000.

35 W.-P. Zhu, Yupeng Yan, M. O. Ahmed and M.N.S. Swamy, " A feedforward symbol timing recovery technique using two samples per symbol", submitted to *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications.*