Coherent Peak Detection Algorithms for UTRA First Stage Code Acquisition

Yazan Chakfeh

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
of the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

July 2008

Your file Votre référence
ISBN: 978-0-494-45279-0
Our file Notre référence
ISBN: 978-0-494-45279-0

ABSTRACT

Coherent Peak Detection Algorithms for UTRA First Stage Code Acquisition

Yazan Chakfeh

The first stage of UTRA code acquisition is the most crucial stage of the cell search process because it has to deal with the largest amount of uncertainty. We assume that the cell search starts when the mobile station is turned on with no prior information about the base station(s) that may be in the vicinity of the mobile station. The mobile station must acquire slot boundaries in order to obtain timing information of the detected base station. Therefore, this large amount of uncertainty gives longer mean acquisition time.

In this thesis, we have devised four new peak detection algorithms for first stage code acquisition. We utilize the available base stations with few multipath signals along with the standard oversampling and pipelining utilization. We compare different coherent and non-coherent combining techniques and run simulations for different carrier frequency errors. We have developed simulation software using MATLAB, to simulate the performance of these algorithms. Our simulation results show that coherent combining for some of the new algorithms provide results close if not better than its non-coherent counterpart even at moderate carrier frequency errors especially at low signal to interference ratios.

ACKNOWLEDGEMENTS

I must truly thank my thesis supervisor, Dr. Ahmad K. Elhakeem, for his support, advice and continuous assistance throughout my thesis work. His vast knowledge and attention to small details gave me the inspiration to finish a refined complete research. Thanks to him, I have grown to work with constructive criticism and actually strive on it. He allowed me to redefine myself and evolve into a person capable of thinking outside the limits; developing my skills as an engineer and as a researcher.

I would like to express my gratitude to my graduate colleagues of the Electrical and Computer Engineering department at Concordia University; especially Rishi Singh. He kindly offered to proof read the final version of my work, in order to submit an improved straightforward research.

Also, I want to acknowledge the faculty members, staff, system administrators and the fellow graduate students of the department of Electrical and Computer Engineering for their cooperation.

Furthermore, I would like to acknowledge my parents, my sisters and my love Nadya whom lived the stress with me during the long working nights. Their constant support was incessant making my two years of thesis work a pleasant journey.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

1G	First Generation
2G	Second Generation
3G	Third Generation
AI	Acquisition Indicator
AICH	Acquisition Indication Channel
AMPS	Advanced Mobile Phone System
AWGN	Additive White Gaussian Noise
BCH	Broadcast Channel
BS	Base Station
CDMA	Code Division Multiple Access
CPCH	Common Packet Channel
CPICH	Common Pilot Channel
CS	Circuit Switched
DCH	Dedicated Channel
DPCCH	Dedicated Physical Control Channel
DPDCH	Dedicated Physical Data Channel
DS	Direct Sequencing
DS-CDMA	Direct Sequencing- Code Division Multiple Access
DSCH	Downlink Shared Channel
DS/SS	Direct Sequencing Spread Spectrum

ETSI	European Telecommunication Standard Institute
FACH	Forward Access Channel
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FH-CDMA	Frequency Hopping-Code Division Multiple Access
FH/SS	Frequency Hopping Spread Spectrum
FSK	Frequency Shift Keying
GF	Galoa Field
GPRS	General Packet Radio Serves
GPS	Global Positioning System
GSM	Global system for Mobile
HSDPA	High-Speed Down-link Packet Access
IMT	International Mobile Telecommunications
IS-95	Interim Standard 95
ITU	International Telecommunication Union
LFSRs	Fibonacci linear Feedback Shift Registers
MS	Mobile Station
OVSF	Orthogonal Variable Spreading Factor
P-CCPCH	Primary Common Control Physical Channel
PCH	Paging Channel
PCPCH	Physical Common Packet Channel
P-CPICH	Primary CPICH
$\mathbf{P_d}$	Probability of Detection

P_{fa}	Probability of False Alarm
PICH	Page Indicator Channel
PN	Pseudo-noise
PRACH	Physical Random Access Channel
PS	Packet Switched
PSC	Primary Synchronization Code
PSK	Phase Shift Keying
P-SCH	Primary Synchronization Channel
QPSK	Quadrature Phase Shift Keying
RACH	Random Access Channel
RARASE	Recursion Added RASE
RASE	Rapid Acquisition by Sequential Estimation
RS	Reed-Solomon
S-CCPCH	Secondary Common Control Physical Channel
SCH	Synchronization Channel
S-CPICH	Secondary CPICH
SFs	Spreading Factors
SMS	Short Message Service
SNR	Signal to Noise Ratio
SS	Spread Spectrum
SSC	Secondary Synchronization Codes
S-SCH	Secondary Synchronization Channel
T_{acq}	Total Acquisition Time

TDD	Time Division Duplex
TH	Threshold
T_i	Integration Time
T_S	Synchronization Time
UMTS	Universal Mobile Telecommunications System
UTRA	UTMS Terrestrial Radio Access

Chapter 1

Introduction

Mobile communication systems have been a major area of interest in wireless communication systems due to their defined goal of providing certain services to anyone, anywhere and anytime. According to the employed technology and provided service(s), the mobile communication systems are divided into three generations: the first generation (1G), the second generation (2G) and the third generation (3G). The 1G cellular systems were based on analog technologies with Frequency Division Multiple Access (FDMA) [1] and designed to support voice application traffic, e.g. the Advanced Mobile Phone System (AMPS) [2]. Their digital counterparts are well known as 2G systems e.g. Interim Standard 95 (IS-95) [3] and Global system for Mobile (GSM) [4]. The second generation was designed to support voice traffic application with major advantages such as more network capacity, efficient use of the available spectrum and larger coverage area in addition to supporting short message service (SMS) and access to data networks through the General Packet Radio Serves (GPRS) [4]. However, due to the increasing demand for more network capacity and the necessity for better and faster data communications that could not be supported by 2G cellular systems; it was necessary to initiate the standardization process of new cellular systems with more network capacity and new applications, i.e. the third generation (3G) cellular systems. The 3G cellular systems are optimized to support multimedia communications at rates higher than those provided by 1G, 2G systems.

1.1 Third Generation (3G) Cellular Systems

Third generation cellular systems have been designed to support new defined consistent services such as mobile internet browsing, high speed data transfer, multimedia, video telephony, video on demand and email. The primary requirements for the next generation cellular systems are [5]:

- Support of large number of users with high quality voice traffic.
- Support both circuit switched (CS) and packet switched (PS) data services.
- Support of high speed data rate transfer for the various mobile environments as shown in Table 1.1.
- Efficient usage of the available spectrum.
- Standardized and compatible world-wide.
- Flexibility and compatibility with pre-existing networks.
- Services must be independent from radio access technology and are not limited by the network infrastructure.

Table 1.1: 3G Required Data Rate

Mobility environment	Minimum data rate
Indoor or stationary	2.048 Mbps
Urban outdoor and pedestrian	384 kbps
Wide area vehicular operation	114 kbps

Almost all of the 3G cellular systems use spread spectrum techniques as their basic access scheme. Such schemes provide the basis of code division multiple access (CDMA) technology.

1.2 Spread Spectrum Technology

Spread Spectrum (SS) is defined as spreading the transmitted signal power over a bandwidth larger than the signal information bandwidth in a pre-defined method. This technique is utilized at the transmitter and the receiver where spreading and despreading take place respectively. Signal mixing at the transmitter by a pseudo-noise (PN) binary-valued sequence results in spreading the signal over larger bandwidth and demodulation of the signal at the receiver with a PN replica results in despreading and restoring the original signal. The SS technique varies according to the method used in spreading the information signal.

Direct Sequencing Spread Spectrum (DS/SS) is based on phase modulation: Phase Shift Keying (PSK), where the phase of the carrier is shifted pseudo randomly according to the PN code sequence. The information signal is modulated by the PN code resulting in spreading the signal power over the available larger bandwidth. The sequence of the PN code is known to both the transmitter and the receiver.

Frequency Hopping Spread Spectrum (FH/SS) is based on frequency modulation: Frequency Shift Keying (FSK), where the available bandwidth is divided into a large number of frequency slots. The information signal may be transmitted over one or more of the available frequency slots. The PN code does not directly modulate the signal but it is employed to select the frequency slot(s) in which the information signal is transmitted.

The PN code sequence defines the pattern of frequency hopping and is known to both the transmitter and the receiver.

By utilization of SS techniques in digital communication systems, an improvement in performance is obtained [6] allowing the possibility of multiple-users accessing the same channel simultaneously, where each user has its own individual PN sequence. This type of digital communication is called Code Division Multiple Access (CDMA).

1.3 CDMA Systems

CDMA technology has become the most important multiple access technology for 2G and 3G wireless communication systems, where most of the 3G standards are engineered based on CDMA. One of the most important characteristics of CDMA systems is that it allows all users to simultaneously transmit their information using the same frequency band within the same time duration by using different codes. A large number of users can be supported by the CDMA systems because each one uses a different shift of the PN spreading sequence. Thus, the number of users can be increased or decreased relatively easier than other systems. The signal from other users appears as an additive interference, where the level of interference varies depending on the number of the users at any instant. The system performance is severely affected if the PN code sequences are not properly chosen; orthogonality should be maintained between different users with different PN sequence shifts. The following points are the major features of CDMA technologies [7]:

- Multiple access capability
- Protection against multiple interference

- Anti-jamming capability
- Privacy, interference rejection
- Low probability of interception
- Possibility to overlay with existing radio systems
- Low transition power emission (which is important to reduce health risks)

CDMA systems can be implemented by various methods depending on the SS technique used. The most important and commonly used systems are Direct Sequencing – Code Division Multiple Access (DS – CDMA) and Frequency Hopping – Code Division Multiple Access (FH – CDMA).

1.3.1 Direct Sequence – CDMA

Direct Sequence – CDMA is the most common multiple access interface on which 3G mobile cellular systems are based. Such a scheme is used for its relatively low average power emission, dynamic allocation of the bandwidth and its high spectrum efficiency. DS – CDMA as the name implies is based on DS/SS. The coherence between the transmitted and received signals can be maintained over a time interval that is relatively long compared to the reciprocal of the transmitted signal bandwidth, even with wide area vehicular coverage. The DS – CDMA compared to other digital wireless communication systems, i.e. FDMA, has relatively simplified hardware, no need for frequency synthesizer [8] and yet the enhancement in performance is comparable.

1.3.2 Frequency Hopping – CDMA

In CDMA systems that are based on Frequency Hopping – SS (FH – CDMA), each user has its pseudo random FH pattern (based on the PN sequence generator). The FH – CDMA system provide large processing gain and relatively high system capacity where the transmitted signal is spreaded over a bandwidth significantly larger than that currently possible for a DS spread spectrum system. Several factors prevent the FH – CDMA from being used in real applications. The first issue is the need for a very accurate reference clock in the whole wireless system. This accurate wide network reference clock is very expensive to implement with available technology. The second issue is the difficulty and complexity in designing a fast-settling frequency synthesizer with a large number of carrier frequencies. Therefore, the main use of FH – CDMA is in some wireless systems which require a relatively small coverage area and relatively low power transmission e.g. Bluetooth wireless technology [9].

Chapter 2

UTRA Air Interface

2.1 Introduction

UTRA is one of the 3G mobile cellular standards proposed by the International Telecommunication Union (ITU) and the European Telecommunication Standard Institute (ETSI). The 3G systems within the ITU are called the International Mobile Telecommunications 2000 (IMT-2000). One of the most promising 3G technologies defined in the IMT-2000 is the air interface based on DS-CDMA, more commonly known as the Universal Mobile Telecommunications System (UMTS). The UMTS Terrestrial Radio Access (UTRA) [10] with Frequency Division Duplex (FDD) is the WCDMA radio interface of the UMTS. The information has been spreaded over a bandwidth of 5 MHz, thus the wideband CDMA name has been given for such air interface. UTRA-FDD is an asynchronous cellular system, i.e. the Base Station (BS) has no common time reference supported by the Global Positioning System (GPS) [11]. UTRA standards support two modes of operation [12]:

- Frequency Division Duplex (FDD)
- Time Division Duplex (TDD)

In the FDD operation mode, the up-link and the down-link traffic are carried over two different radio channels (multiplexing in frequency). In the TDD operation mode, one radio channel is time-shared between the up-link and the down-link (multiplexing in time). The capability of UTRA to operate in either FDD or TDD mode allows efficient utilization of the available spectrum in situations where the available spectrum is very limited. The following table shows the technical specifications of the UTRA radio interface [13], [14]:

Table 2.1: UTRA Technical Specifications:

Multiple Access Method	DS-CDMA
Duplex Method	FDD / TDD
Multi-Rate / Variable Rate concept	Variable spreading factor and multi-code
Bandwidth	FDD: 2 × 5 MHz paired (Up-link / Down-link)
	TDD: 1 × 5 MHz unpaired
Carrier Spacing	4.4 MHz – 5.2 MHz
Chip rate	3.84 Mcps
Frame Length	10 ms (15-slots frame, 0.667 ms slot)
Modulation	QPSK (Quadrature Phase-Shift Keying)
Channel Coding (FDD only)	Convolutional Code (rate 1/2 & rate 1/3)
	Turbo Code (rate 1/3)
Detection	Coherent using pilot symbol or common pilot
Inter Base station Synchronization	FDD: No accurate synchronization required
	TDD: Accurate synchronization required
Base Station synchronization	Asynchronous operation

The standard 3.84 Mcps chip rate can be extended to two or three times allowing the accommodation for data rates higher than 2 Mbps. The support of higher data transfer speeds and capacity was achieved with the introduction of High-Speed Down-link Packet

Access (HSDPA) [15]. Each frame of 38400 chips (10 ms) is divided to 15 slots where each slot is 2560 chips (0.67 ms). The use of 200 kHz channel raster [16] has been chosen to facilitate inter operability with GSM.

2.2 UTRA-FDD Codes

2.2.1 Channelization Code

In UTRA-FDD, orthogonal variable spreading factor (OVSF) codes [17] are used as Channelization codes to separate the physical channels. The OVSF codes are Walsh codes of different lengths that maintain orthogonality between physical channels; even though they operate at different data rates with various spreading factors (SFs). The OVSF tree is shown in Figure 2.1. The notation $C_{SF,\ \text{code number}}$ denotes the SF which ranges from 4 to 512 and the code number ranges from 1 to SF level. Increasing SF value is done according to Figure 2.2 providing no code in the path from the specific code to the root of the tree or in sub-tree below the specific code is assigned. Higher SF value means lower data rates.

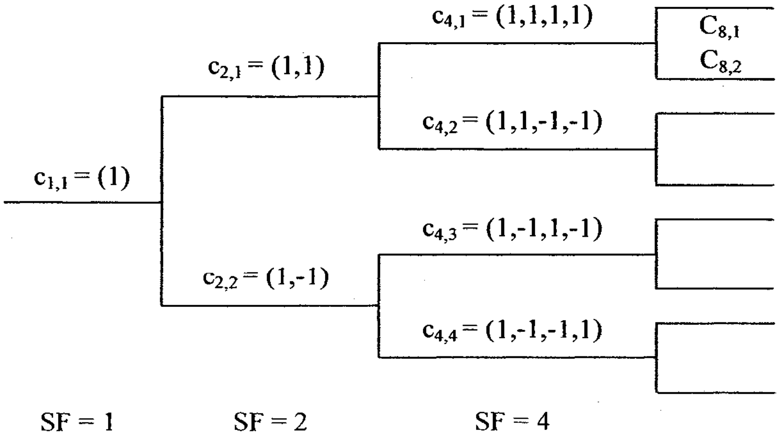

Figure 2.1: OVSF code tree

A proper code allocation algorithm is used to prevent the BS from running out of codes, and to utilize the system resource effectively according to the available codes of the BS and the capability of the Mobile Station (MS). However, Walsh codes have poor autocorrelation functions and, in the presence of fading they have non-zero cross-correlation. This renders them unsuitable for multiple access codes. Since the channelization codes are Walsh codes of different lengths, it is essential that additional utilization of complex scrambling codes have good auto-correlation and cross-correlation properties.

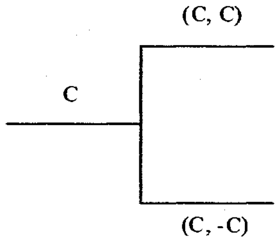

Figure 2.2: Branching through the OVSF tree

2.2.2 Down-link Scrambling Code

The scrambling process is used in the UTRA systems transmitter, in addition to signal spreading by OVSF codes, to separate transmissions from different sources. Since scrambling is used along with spreading as shown in Figure 2.3, implying that the scrambling process does not change the bandwidth of the signal but only makes the signals from different user more separable. Scrambling codes can be long or short codes. In the up-link, long and short codes are used depending on the BS receiver, (long codes for Rake receiver and short codes for multi-user detectors).

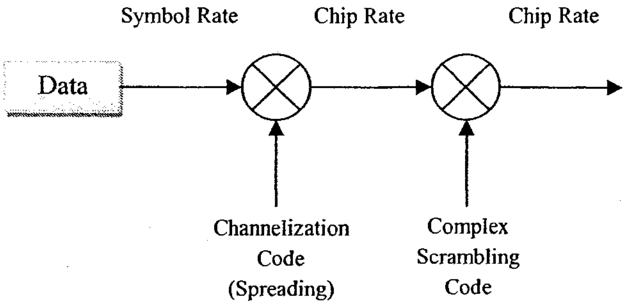

Figure 2.3: Relation between Spreading and Scrambling

In the WCDMA down-link, the scrambling code employed in UTRA-FDD is a truncated 38400-chip segment of $2^{18}-1=262143$ available length Gold code. However not all of the scrambling codes are used. Large sets of Gold codes have low cross-correlation properties therefore many potential users can access the same channel with minimum mutual interference. Also, scrambling codes are used in pair in the Inphase and Quadrature channels, i.e. complex scrambling codes are used to provide constant envelope. The scrambling codes have Inphase and Quadrature components and their total number is 8192. The code duration is 10 ms where the chip rate is 3.84 Mchips/s and the truncated code length is 38400 chips, i.e. the duration of the scrambling code lasts for one 15-slot frame. The 8192 codes are divided into 512 sets, representing 512 down-link primary scrambling codes, allowing unique cell identification. The 512 sets are divided into 64 code groups defined in [18], [19] with 8 codes sets in each group. Each set has 16 scrambling codes, a primary scrambling code and 15 secondary scrambling codes, where the 15 secondary scrambling codes are cyclic shifts of the primary scrambling code. The scrambling code hierarchy is shown in Figure 2.4.

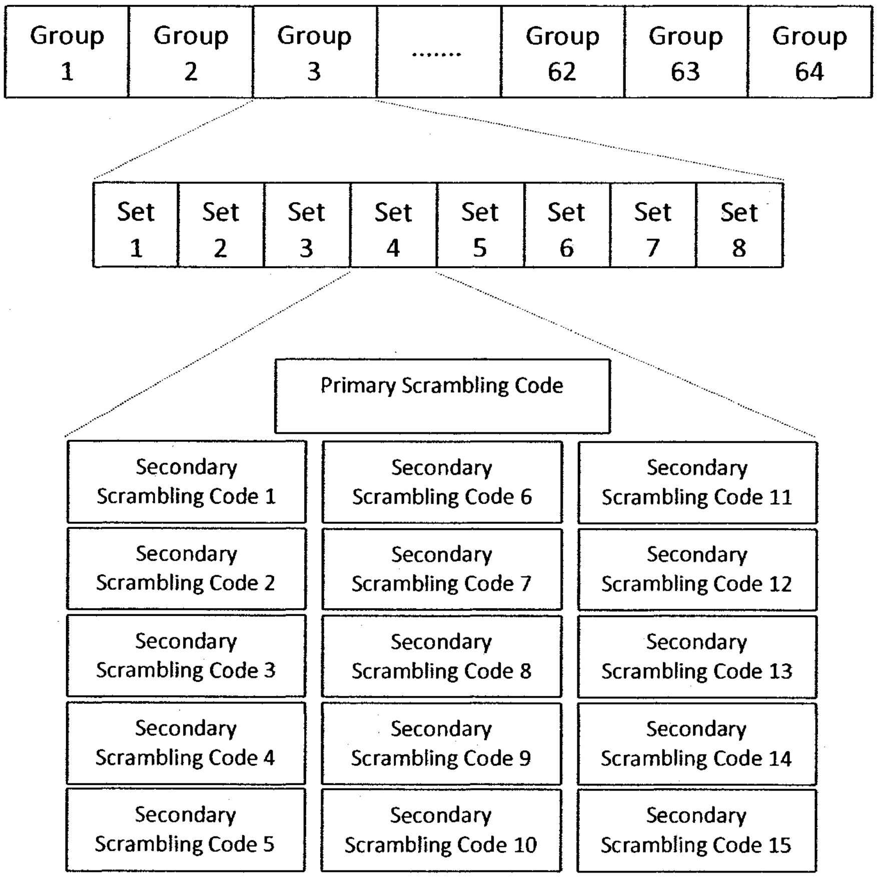
Group 1
Group 2
Group 3
....
Group 62
Group 63
Group 64
Set 1
Set 2
Set 3
Set 4
Set 5
Set 6
Set 7
Set 8
Primary Scrambling Code
Secondary Scrambling Code 1
Secondary Scrambling Code 6
Secondary Scrambling Code 11
Secondary Scrambling Code 2
Secondary Scrambling Code 7
Secondary Scrambling Code 12
Secondary Scrambling Code 3
Secondary Scrambling Code 8
Secondary Scrambling Code 13
Secondary Scrambling Code 4
Secondary Scrambling Code 9
Secondary Scrambling Code 14
Secondary Scrambling Code 5
Secondary Scrambling Code 10
Secondary Scrambling Code 15

Figure 2.4: Scrambling code hierarchy

The primary scrambling code in the down-link consists of scrambling codes $p = 16j$ where j = 0, 1... , 511 and the j^{th} set of secondary scrambling codes consist of scrambling codes $n = 16j + k$ where k = 1... , 15. The long scrambling code sequences are generated by combining two real sequences into a complex sequence. Each of the real sequences is constructed as the position wise modulo 2 sums of 38400 chip segments of two binary m-sequences. The two binary m-sequences are constructed using a primitive polynomial of degree 18 over GF (2), i.e. the two binary m-sequences are generated by two q-stage (q=18) Fibonacci linear feedback shift registers (LFSRs) with appropriate initial conditions as illustrated in Figure 2.5.

The polynomials that define the two real sequences are:

$$G_X(x)=X^{18}+X^{7}+1 \tag{2.1}$$

$$G_Y(x)=X^{18}+X^{10}+X^{7}+X^{5}+1 \tag{2.2}$$

The n^{th} Gold code sequence at the output of the scrambling code generator is defined as:

$$z_n(i)=x_n(i+n)\oplus y(i) \tag{2.3}$$

where the n^{th} code corresponds to advancing n-position(s) of the X sequence while preserving the same Y sequence as shown in Figure 2.5. Advancing the X sequence is possible by loading the appropriate initial conditions where each initial state corresponds to n^{th} code. The initial states can be pre-calculated and stored in a memory unit. The binary sequences generated by (2.3) are mapped to real value sequences:

$$\mathrm{Z}_n(\mathrm{i})=\begin{cases}+1 & \text{if } \mathrm{z}_n(\mathrm{i})=0\\ -1 & \text{if } \mathrm{z}_n(\mathrm{i})=1\end{cases} \tag{2.4}$$

In order to construct the complex scrambling code, the sequence $Z_n(i)=I_n(i)$ is delayed by $\tau=2^{q-1}$ in order to obtain $Z'_n(i)=Z_n(i+\tau)=Q(i)$ and the complex down-link scrambling code can be written as:

$$S_n(i)=I_n(i)+jQ_n(i) \tag{2.5}$$

A detailed derivation of the down-link complex scrambling code generation is explained in [18].

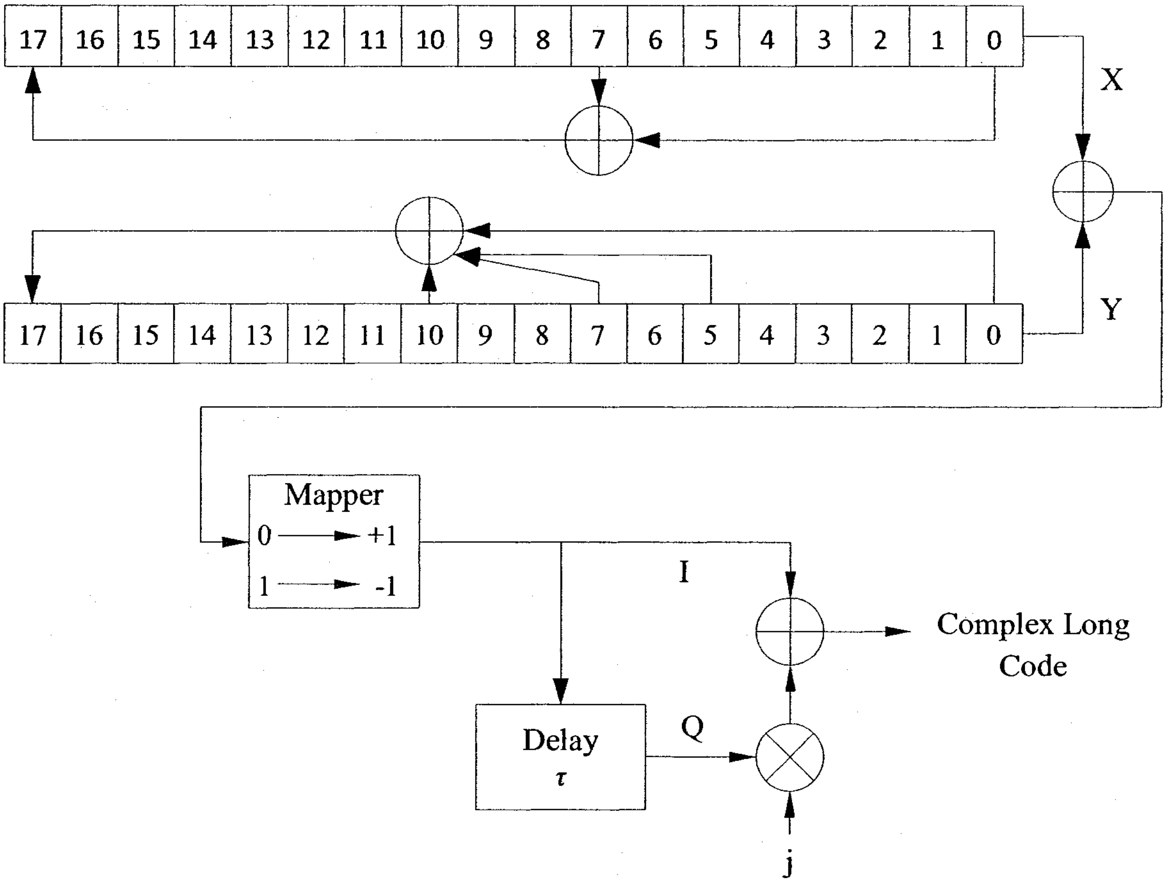

Figure 2.5: Down-link complex scrambling code generator

2.2.3 Synchronization Codes

Synchronization codes are used by the MS to detect the presence of the BS and to synchronize with the BS. The synchronization codes are divided into two codes: the Primary synchronization Code (PSC) and Secondary Synchronization Codes (SSC). The PSC is 256 concatenated Golay code sequences that provide good auto correlation properties [18], [19]. The PSC is mapped to 256-chips. The same PSC is used by all BSs and transmitted at the beginning of every slot. Therefore, the MS utilizes the PSC initially to detect the presence of the nearby BS to identify the slot boundaries, (the position of the SSC). In the generation of PSC, a 16-chip code is modulated by another 16-chip code resulting in a 256-chip code sequence. The PSC is defined for the Inphase and Quadrature channels as follows:

$$C_{PSC} = \{a, a, a, -a, -a, a, -a, -a, a, a, a, -a, a, -a, a, a\} \qquad (2.6)$$

Where 'a' is a vector containing 16 elements of value 1 or -1 as shown below:

$$a = \{1, 1, 1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1, -1, -1, 1\} \qquad (2.7)$$

The SSC is used by the MS to find the code group to which the primary scrambling code of the selected BS belongs (The BS whose slot boundaries were found in the first stage) as well as the frame boundaries. Unlike the PSC, there are 16 different SSCs where each is a short Gold code (256-chips), which are mapped to 16 symbols labeled from 1 to 16. The 15 code-symbols of a certain code group that BS belongs to are sequentially transmitted over each frame. The concatenation of the 15 SSCs constitutes a Reed-Solomon (RS) code word, and each RS code word represents a code group. The utilization of the Comma-Free Reed-Solomon code was introduced in order to maximize the minimum distance between different cyclic shifts of a code ward or between any cyclic shifts of different code wards [20]. A RS (15, 3) using GF (2^4) is used, where the minimum distance of the code is 13, and the error detection and correction capabilities are 12 and 6 respectively [21]. The same RS code word is repeated over every frame in a cell. The 64 groups of the 512 primary scrambling codes are identified by 64 different RS code words with unique cyclic-shifts. Therefore, by identifying the RS code word transmitted in every frame, the frame boundaries and the scrambling code group can be determined after trying all possible cyclic shifts.

2.3 UTRA-FDD Transport and Physical Channels

The transport channels of UTRA-FDD system are seven different channels. Those channels can be divided into one dedicated transport channels and six common transport channels.

- The Dedicated Channel (DCH) is the transport channel for the up-link and down-link. The DCH carries information coming from layers above the physical layer, such as data for the actual service provided to the user as well as higher layer control information.

The common transport channels with their functions are [13]:

- Broadcast Channel (BCH) is a down-link transport channel that transmits related information to the network or the given cell. Decoding the BCH is mandatory for registering the terminal to the cell.

- Forward Access Channel (FACH) is a down-link transport channel which carries control information as well as the possibility of transmitting packet data to terminals located in the given cell
- Paging Channel (PCH) is a down-link transport channel. The PCH carries data related to the paging procedure that allows the cell to initiate communication with the terminal.
- Random Access Channel (RACH) is an up-link transport channel. The RACH is designed to carry control information from the mobile terminal, e.g. connection setup requests. Also a small amount of packet data can be transmitted from the terminal to the network.
- Common Packet Channel (CPCH) is an up-link transport channel designed as an extension to RACH where packet data can be transmitted over.
- Downlink Shared Channel (DSCH) is a down-link transport channel that carries dedicated user data and may carry control information.

The physical channels of UTRA-FDD system can be divided into two categories: dedicated channels and common channels. The dedicated channels are up-link/down-link and used by the MS during a call or data exchange. The common channels are used to carry information to all MSs within a cell and used by the MSs to access the network. The two types of dedicated physical channels are:

- The dedicated physical control channel (DPCCH) that is responsible for carrying the control information of the physical layer.

- The dedicated physical data channel (DPDCH) which carries user traffic, i.e. data exchange, and control information from higher layers. Both the DPCCH and DPDCH are mapped to the DCH.

The up-link common physical channels are:

- The Physical Random Access Channel (PRACH) where main use is signaling functions and is mapped to the RACH. The signaling functions of the RACH are initiated in order to register powered-on mobile terminal to the network, to carryout location update in case of mobility or to initiate a call.
- The Physical Common Packet Channel (PCPCH) supports the CPCH access procedure.

The synchronization process is performed on the down-link of synchronous systems such as UTRA-FDD. The common physical channels on the down-link are [22]:

- The common pilot channel (CPICH) carries the down-link common pilot symbols to enable coherent demodulation at the MS and has two sub-channels.
- The primary common control physical channel (P-CCPCH) carries synchronization and broadcast information for MSs, i.e. carries the Broadcast Channel (BCH), and is transmitted over the entire cell. The P-CCPCH is not transmitted during the first 256 chips of each slot. There is only one P-CCPCH within a cell. The SF of the P-CCPCH is 256 (fixed rate at 30 kbps).
- The secondary common control physical channel (S-CCPCH) transmits paging messages and packet data over the entire cell. Two different common channels are

transmitted over the S-CCPCH; the FACH and the PCH. The SF of the S-CCPCH may vary from 4 to 256 (variable rates between 30kbps and 1920 kbps).

- The synchronization channel (SCH) is used by the MS in the cell search. There are two sub-channels of the SCH, the Primary Synchronization Channel (P-SCH) and the Secondary Synchronization Channel (S-SCH). The P-SCH and S-SCH are transmitted during the first 256 chips of each slot while the remainder of each slot carries the P-CCPCH.
- The acquisition indication channel (AICH) controls the random access procedure on the up-link, (MS to BS) where the Acquisition Indicator (AI) is carried on this channel. The Primary CPICH provides phase reference to the AICH, and the SF of the AICH is 256 (fixed rate).
- The page indicator channel (PICH) is used by the BS to initiate communication with the MS and provide the MS with efficient sleep mode operation. The SF of the PICH is 256 (fixed rate).

Three of the down-link common physical channels are used in the cell search in UTRA-FDD systems, the P-SCH, S-SCH and CPICH [19]. Figure 2.6 illustrates one 15-slot frame of these channels. The CPICH carries pre-defined bit sequences that run at fixed rate of 30 kb/s and SF of 256. Within each CPICH slot, there are 10 pilot symbols, each one spread by 256 chips. There are two types of common pilot channels: Primary CPICH (P-CPICH) and Secondary CPICH (S-CPICH). The P-CPICH uses the same channelization code, $C_{ch\ 256,\ 0}$ which is all logical 1's, and scrambled by the primary scrambling code of the BS. There is one P-CPICH per cell that broadcasted over the entire cell. The S-CPICH uses any channelization code of SF equal to 256 and could be scrambled by the primary or a secondary scrambling code of the BS.

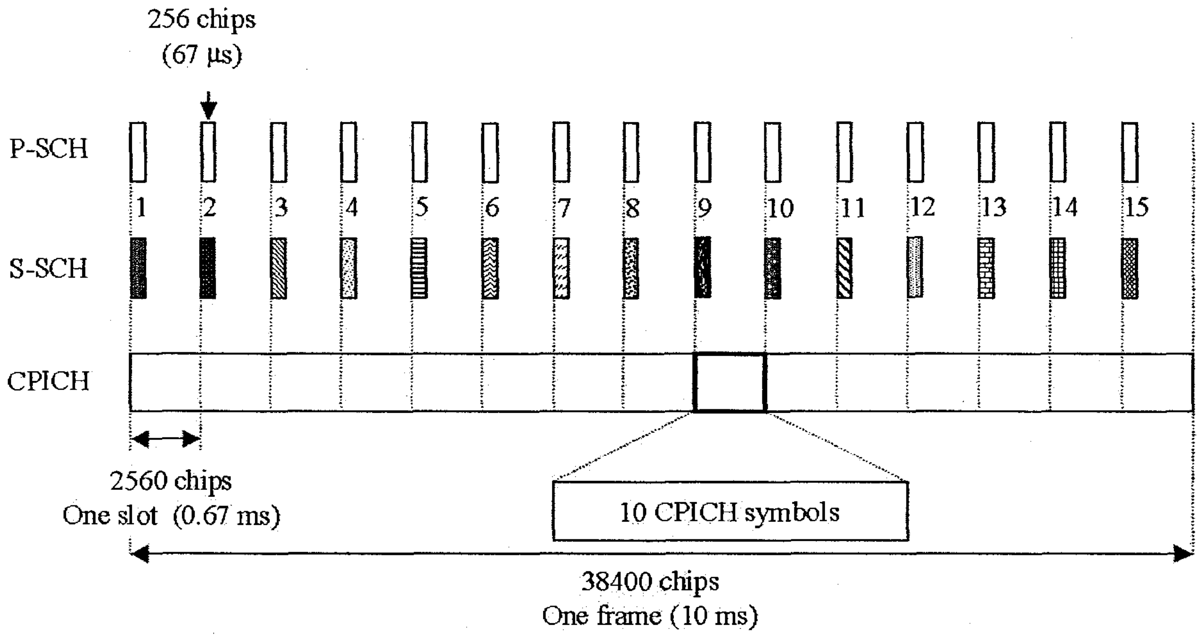

Figure 2.6: Frame and slot structures for CPICH, P-SCH and S-SCH

The number of S-CPICH per cell may vary from zero to several channels that may be transmitted over the entire cell or only over a part of the cell. Unlike the CPICH, neither P-SCH nor S-SCH is scrambled by the primary scrambling code. The primary synchronization code (PSC) and the secondary synchronization code (SSC) are transmitted over the P-SCH and the S-SCH respectively. Both PSC and SSC are transmitted during the first 256 chips of each slot, where the P-CCPCH is transmitted during the reaming time of a slot, i.e. 2560 – 256 = 2304 chips.

Chapter 3

Spread Spectrum and UTRA-FDD Code Acquisition Schemes

3.1 Introduction

In spread-spectrum communication systems, code synchronization is carried out by generating a replica of the spreading sequence at the receiver, which is synchronized with the received sequence in both code phase and frequency offset. Initial code synchronization is a very crucial task in spread-spectrum systems, where the overall system performance is significantly affected by the performance of the synchronization stage. Also, precision clocks are utilized at the transmitter and the receiver in order to limit timing uncertainty. The most straightforward synchronization scheme is to sweep the generated spreading sequence phase until proper phase alignment with the received sequence is attained. When the phase is aligned, the input signal spectrum collapses and energy appears at the output of a narrow-band filter. According to spread-spectrum techniques, the spreading sequence either modulates the signal or selects the pattern in which the frequency carrier hops, i.e. code synchronization is either sequence or pattern synchronization.

Initial code synchronization process has been facilitated by dividing synchronization into two sub-procedures: code acquisition and code tracking. Code acquisition is defined as the process of bringing the code phase of the generated sequence within a fraction of a chip of the code phase of the received sequence. Code tracking is defined as the process of maintaining code generator synchronized with the received code. The main spread-spectrum scheme used in digital communications is direct sequencing (DS), where the spread spectrum receiver despreads the received pseudo noise (PN) code with a locally generated one. The received signal is subject to interference; therefore an acquisition system with the capability of operating under the anticipated level of interference is required. The most effective techniques that meet this condition are:

- Serial-search acquisition
- Matched-filter acquisition

3.2 Serial Search Acquisition

A commonly used technique for initial code synchronization is serial-search, i.e. all possible generated code phases are searched sequentially until the local spreading sequence is synchronized with the received spreading sequence. Each code phase/frequency is tested by trying to despread the received signal where both sequences are applied to a non-coherent correlator. If the sampled correlator output exceeds a certain threshold, it means that the estimated code phase and frequency are correct and despreading takes place. The search is stopped and the two sequences are now running in parallel. If the sampled correlator output does not exceed the threshold, it means that the estimated code phase or frequency is incorrect, therefore despreading will not occur. The cell (code phase) under test is rejected and the local sequence phase is stepped to a new code phase/frequency for revaluation.

The step size in which the search system advances through the uncertainty region has a considerable impact on the system performance because the received spreading sequence may have a phase error of ± ½ chip. Therefore, the step size is selected to be ½ chip in order to overcome phase drifts. Decreasing the step size will result in increasing both the average detected energy during acquisition and the number of cells to be searched.

In serial-search acquisition schemes, two probabilities are defined in order to calculate the mean synchronization time (T_S), the probability of detection and probability of false alarm. The probability of detection (P_d) is the probability of the investigated cell being the correct one where the probability of false alarm (P_{fa}) is the probability of the investigated cell being an incorrect one. Both of the probabilities are a function of the integration time (T_i) and signal to noise ratio (SNR).

To simplify the mean acquisition time calculation, both assumptions of no carrier frequency error and uniform distribution of the correct code phase over the region ΔT seconds are considered. A step size (Δt) seconds is chosen according to the spreading sequence autocorrelation function. The total number of uncertain cells (code phases) that a spreading sequence has is defined as integer $C = \Delta T/\Delta t$. The serial search can start from any cell since all cells have same probability of being the correct phase. If acquisition stage evaluates C cells and synchronization is not achieved, the search will start over again.

Considering all possible events that lead to a correct code synchronization will enable the calculation of the mean acquisition time in an uncomplicated way [23]. The total synchronization time for an event defined by (n, j, k) is:

$$T(n,j,k) = nT_i + jCT_i + kT_{fa} \tag{3.1}$$

where n is a location for the correct cell, j is a number of missed detections of the correct cell, and k is a number of false alarms of all evaluated incorrect cells. The time to evaluate each cell is the integration time T_i and the time to reject an incorrect cell is penalty time T_{fa}. The penalty time T_{fa} is undesirable since it is an additional time and it will not lead to a correct cell. Based on the serial-search scheme that is used. The penalty time T_{fa} may be larger than integration time, T_i. The probability of the event (n, j, k) [24] is:

$$P(n,j,k) = \frac{1}{C} P_d (1-P_d)^j \binom{K}{k} P_{fa}^k (1-P_{fa})^{K-k} \tag{3.2}$$

where 1/C is the probability of the nth cell being the desired one and $P_d(1-P_d)^j$ is the probability of j misses followed by a correct detection. The orderings of k false alarms out of K incorrect cells are $\binom{K}{k}$ and $P_{fa}^k(1-P_{fa})^{K-k}$ is the probability of the orderings. The mean synchronization time can be defined as follows:

$$T_S = \sum_{n,j,k} T(n,j,k)P(n,j,k) \qquad (3.3)$$

Substituting (3.1) and (3.2) in (3.3) and simplifying yields:

$$T_s = \frac{1}{C}\sum_{n=1}^{C}\sum_{j=0}^{\infty}[(n+jC)T_i + KT_{fa}P_{fa}]P_d(1-P_d)^j \qquad (3.4)$$

where K is the total number of incorrect cells. K is defined as $((n+jC)-(j+1))$: $(n+jC)$ is the total number of evaluated cells and $(j+1)$ is the total number of correct cells. Further simplifications result in:

$$T_S = (C-1)T_{da}\left(\frac{2-P_d}{2P_d}\right) + \frac{T_i}{P_d} \qquad (3.5)$$

where T_{da} is the average wasted time at an incorrect cell:

$$T_{da} = T_i + T_{fa}P_{fa} \qquad (3.6)$$

A major issue in spread-spectrum based systems is to minimize the mean synchronization time. The mean synchronization time in (3.5) is a function of different parameters, i.e. C, P_d, P_{fa}, T_i and T_{fa}. The length of the PN sequence is fixed by system standards, i.e. the phase uncertainty region ΔT is fixed. However, according to the definition of the total number of uncertain cells C, increasing or decreasing the step size Δt will result in changing C in order to enhance system performance. The other variables are dependent one on another, e.g. higher P_d value and lower P_{fa} value implies larger T_i. The needed time to recover from a false alarm is T_{da}, (3.6). The equation consists of T_i added to T_{fa} time multiplied with P_{fa}. The integration time, T_i, is cell evaluation time. The false alarm time, T_{fa}, is the needed time to declare that the current cell is incorrect and P_{fa} is the probability of occurrence of such event.

3.2.1 Single-dwell Serial PN Acquisition

A direct-sequence spread-spectrum single dwell PN code acquisition system is shown in Figure 3.1. The received signal plus noise is actively correlated with a locally generated PN code, and then passed through a band-pass filter. The output of the filter is passed through a square law envelop detector then passed through a low-pass filter to eliminate the high frequency terms. The filter output is integrated for a fixed integration time T_i, and then it is compared to a predefined threshold. If the comparison results in threshold crossing, then detection is declared and the system enters tracking mode. Otherwise a new cell is chosen and the search process starts again. Since there is only one correct cell, the system attempts to examine each cell within the entire uncertainty region until the correct cell is acquired. The mean synchronization time in (3.5) is for a single-dwell serial-search system.

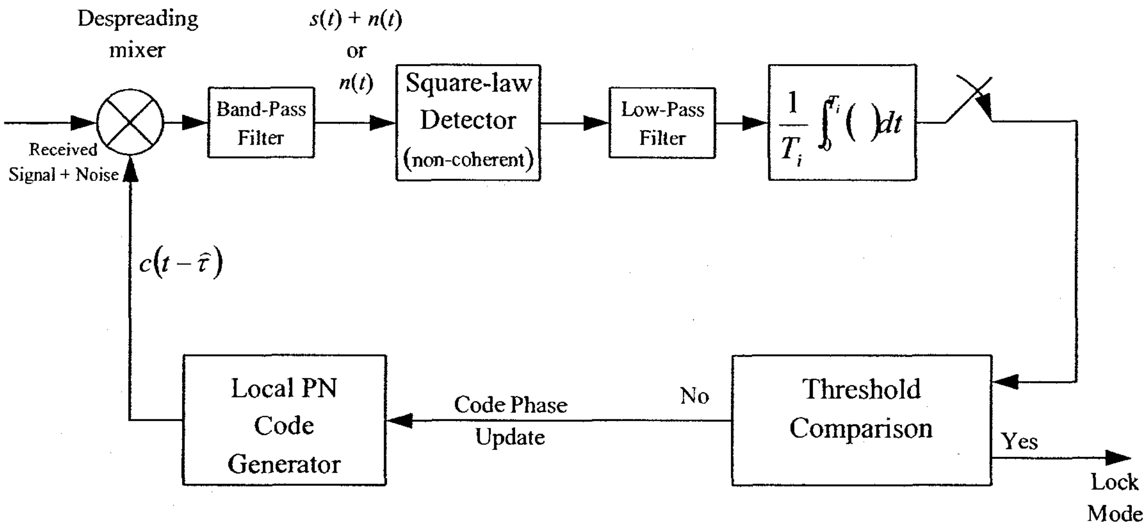

Figure 3.1: Block diagram of single dwell PN acquisition system

3.2.2 Multiple-dwell Serial PN Acquisition

Multiple-dwell search scheme consists of three search modes: search mode, verification mode and tracking mode. The search mode consists of one step where the system selects a cell (or code phase) for evaluation and evaluates this cell. The evaluation time (integration time) in this step is very short in order to result in instantaneous rejection of incorrect cells. This short integration time however leads to a high false-alarm probability, P_{fa}. The system will enter the verification mode if the first step results in detection or false-alarm. The second evaluation of the same cell starts using longer integration time leading to reject most of the false alarms of the first evaluation step. A third evaluation step may follow since the verification mode may consist of one or more steps according to a particular performance level. Once the evaluated cell passes all N tests of the search and verification modes, acquisition of the correct cell is declared and the system enters tracking mode. In tracking mode, the system will continue to evaluate the acquired cell and compare the output of the integrator to a threshold. Once the output of the integrator fails to pass the threshold, a second tracking stage with a larger integration time and different threshold may follow since the tracking mode may consist of one or more steps. All multiple-dwell systems' parameters such as integration-times, thresholds and the number of required stages used to implement the multiple-dwell scheme are chosen to result in minimum synchronization time.

A direct-sequence spread-spectrum multiple dwell PN code acquisition system is shown in Figures 3.2 and (3.3 or 3.4). The received signal plus noise is actively correlated with a locally generated PN code and then passed through a band-pass filter. The output of the filter is passed through a square law envelop detector then passed through a low-pass filter to eliminate the high frequency terms. The filter output is integrated for the first time over short fixed integration time, T_{i1}, and then compared to first stage predefined threshold. If the integrator output crosses the first stage threshold, the system leaves the search mode and enters the verification mode. The verification mode may consist of one or more stages where each stage has a fixed integration time and a stage related predefined threshold. The same concept of integrating the signal and comparing it against a threshold is repeated in each stage of the verification mode where each stage has predefined parameters. For each successfully evaluated stage, the system moves on and starts evaluating the next stage. Once all of the verification stage(s) are successful, detection is declared and the system enters tracking mode. If the integrator output fails to cross the threshold in one of the stages, the system either returns to the previous stage and revaluates the cell or chooses a new cell where the search process starts again, according to the search strategy.

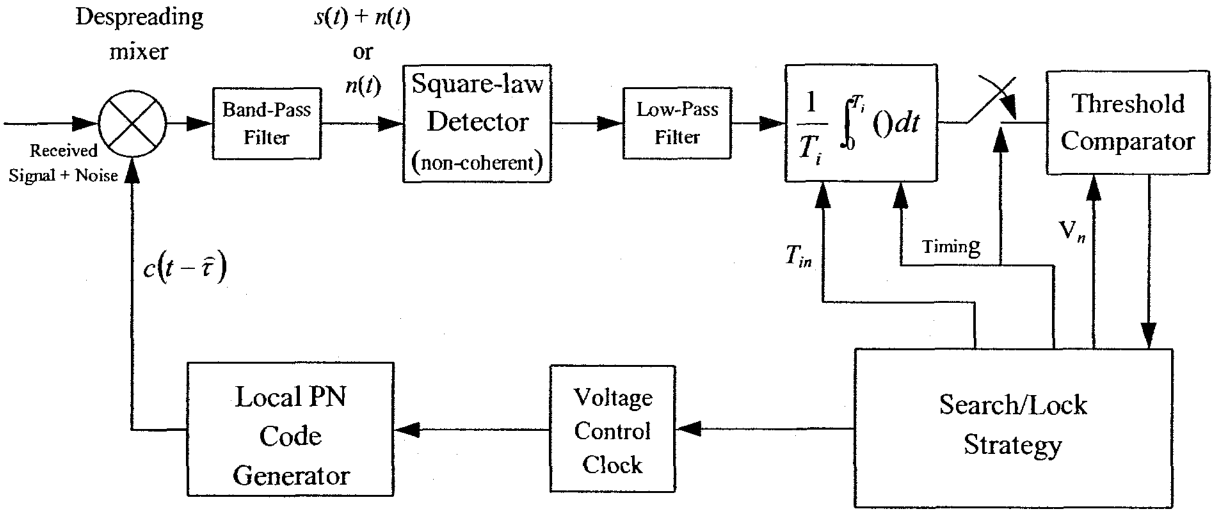

Figure 3.2: Block diagram of multiple dwell PN acquisition system

Multiple-dwell systems may use consecutive-count strategy or up-down strategy. Figure 3.3 shows a flow diagram for a consecutive-count strategy where a failed stage causes a cell to be rejected immediately and reacquisition begins in the search mode with a new cell. Figure 3.4 shows a flow diagram for an up-down strategy where a failed stage causes a cell to be revaluated by a previous stage until the tested cell fails the search mode stage followed by reacquisition beginning in the search mode with a new cell. The two strategies are used to define the search and acquisition modes in multiple-dwell systems [25].

Multiple-dwell serial search systems are a generalized model of the single-dwell system where extra evaluation stages are presented with different integration times and threshold comparisons for each stage. However, these schemes are considered as a fixed integration time PN acquisition system because the variation in integration time is achieved by using a series of fixed short time integrators. In general, each stage

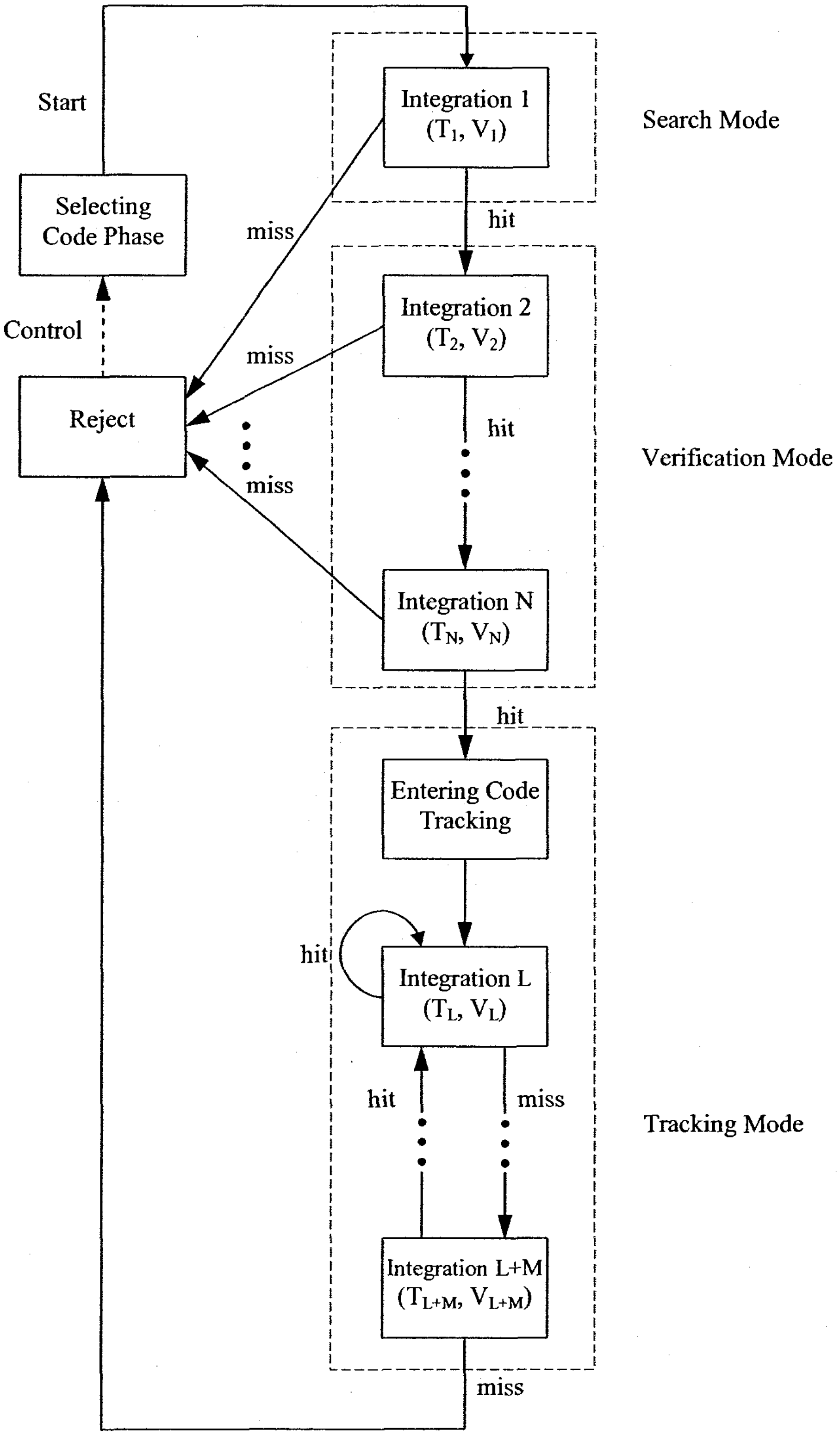

Figure 3.3: Flow graph of multiple-dwell system with consecutive-count strategy

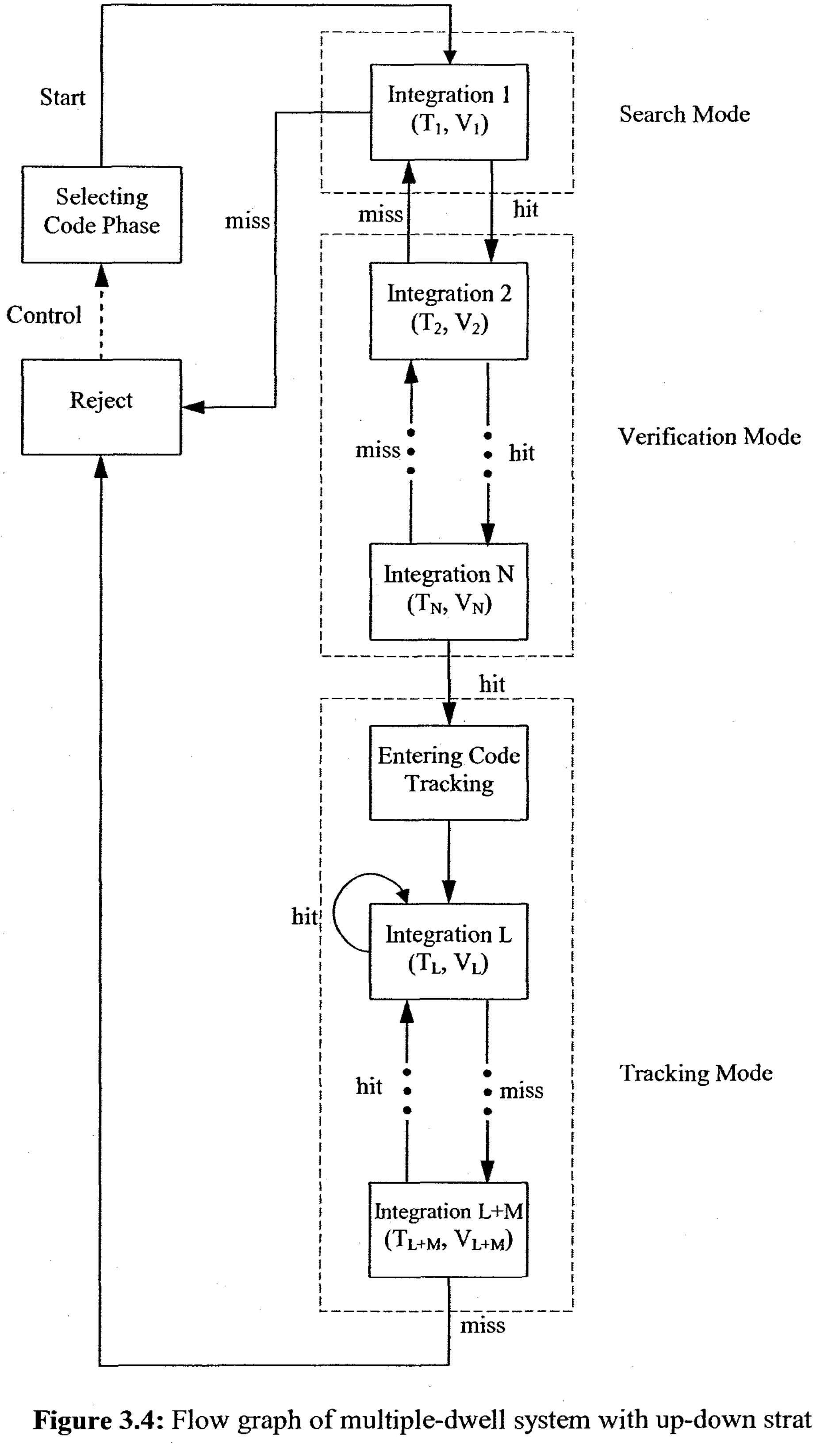

Figure 3.4: Flow graph of multiple-dwell system with up-down strategy

Integration time is longer than the precursor; the longer integration time is the more confidence in acquiring the correct cell. Integration time of each cell examination is increased to its maximum value in discrete steps which permits the dismissal of an incorrect cell earlier than single-dwell system which is forced to integrate over the full examination period. In the case of long spreading sequences, a considerable reduction in acquisition time is obtained by a quick elimination of the incorrect cells (multiple-dwell serial search systems).

In multiple-dwell search systems, the received signal plus noise is multiplied by a replica of the spreading sequence (PN code) after which the output is applied to each of the N non-coherent detectors. Each detector is characterized by integration time T_{in} and threshold V_n where n=1, 2 ...N. In general, detector integration times are designed such that $T_{i1} \leq T_{i2} \ldots \leq T_{iN}$ in order to achieve the early elimination property. The major advantage of the N dwell system over the single dwell system is the fact that the minimum integration time is T_{i1} and maximum integration time is T_{iN}, consequently most of the cells can be discard after T_{in}, $n \leq N$. In contrast, the single dwell system examines every cell for a time corresponding to T_{iN}.

3.3 Matched-Filter Acquisition

An effective code acquisition technique is a matched-filter that is typically matched to part or all of the period of the spreading PN sequence. The matched-filter outputs a pulse when received code symbols are matched to a locally stored code. The envelop detector output will have an auto-correlation spike in case of code detection that is compared with a threshold. The occurrence of threshold exceedance will provide timing information that is used by the local code generator and then the code synchronization process is completed. In most spread-spectrum communication systems, the spreading code period is long. Therefore, matched-filters with extremely time-bandwidth products are required. As a solution, an easily programmable matched-filter is programmed for a particular code phase that is based on propagation delay estimation. The programmability of the matched-filter is attained by employing high-speed digital correlators and programmable convolvers. Based on the timing information obtained, the code generator (shift registers) will be loaded with an appropriate value to result in demodulating the received stream. Since code acquisition is done at reduced signal-to-noise ratio, the loaded value may not be correct and further shift register loads will have to be loaded and evaluated. As in serial search, each loaded value is evaluated by trying to despread the received signal. This method is called rapid acquisition by sequential estimation (RASE) [26]. A modified version of the RASE matched-filter is called recursion added RASE (RARASE) [27], where it performs an initial evaluation of the shift register loaded value by comparing the output with some of the received symbols before starting the full despreading.

A direct-sequencing spread spectrum receiver that uses a matched-filter synchronizer is shown in Figure 3.5. The received signal plus noise is fed to the spreading code tracking loop, to the on-time despreader for data detection and to a band-pass filter that is matched to a segment of the direct-sequence spreading code. If the received signal is matched to the segment of the code that filter is matched to, the matched-filter will generate an output pulse. The receiver uses an envelope detector to sense the generated pulse followed by a threshold comparison that compares the peak to a predefined threshold. The local code generator starts code generating at the correct code phase when the envelope detector peak results in threshold crossing. The active code generator starting phase is a function of the matched-filter code segment and the delays of the detection and the matched-filter.

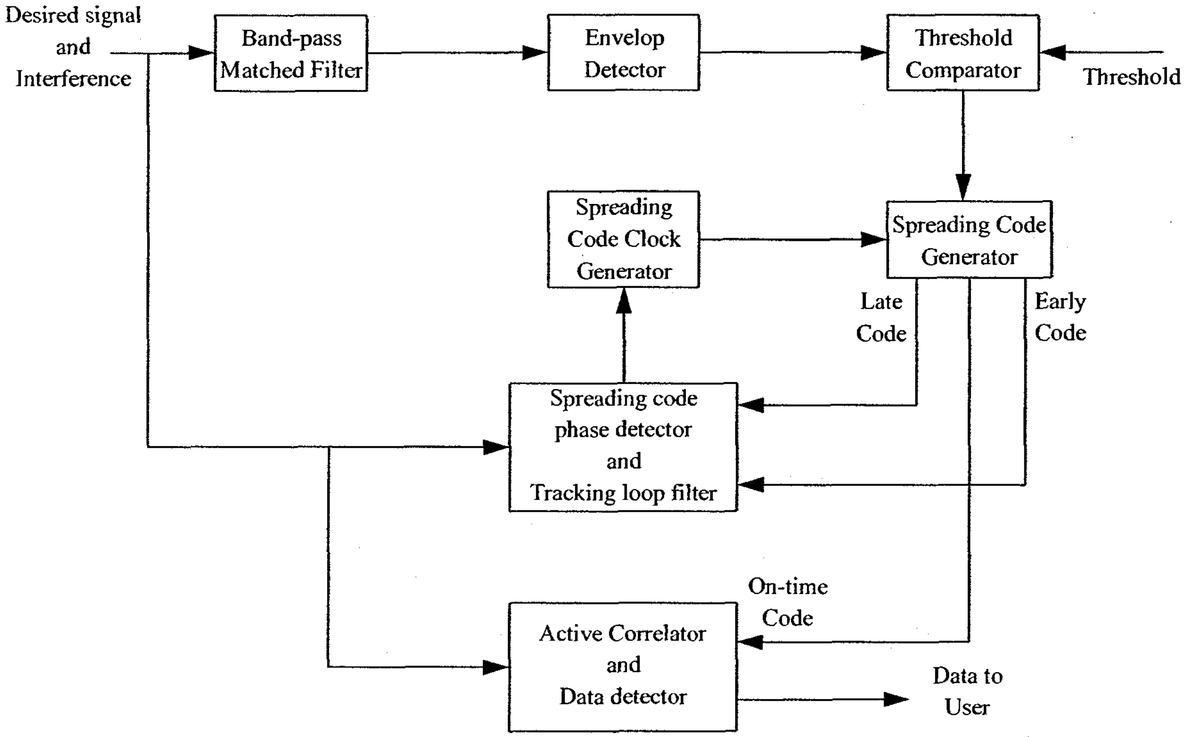
Desired signal and Interference
Band-pass Matched Filter
Envelop Detector
Threshold Comparator
Threshold
Spreading Code Clock Generator
Spreading Code Generator
Late Code
Early Code
Spreading code phase detector and Tracking loop filter
On-time Code
Active Correlator and Data detector
Data to User

Figure 3.5: Direct-Sequence spread-spectrum receiver using Matched-filter code acquisition

The mean synchronization time of the matched-filter is significantly reduced by using a parallel set of matched-filters [28]. The correct code phase is divided into several parts and each filter of the parallel set is matched to one of these phases. All of the parallel matched-filters work separately and independently while the output of each filter is compared against other outputs. The largest output is compared against a predefined threshold to make sure that the acquired code phase is correct. Therefore, the enhancement in system performance depends on the number of the parallel matched-filters.

3.4 UTRA-FDD Code Acquisition

The serial search acquisition and matched-filter acquisition techniques described earlier are used in spread spectrum systems that use the same scrambling code in all BSs where each BS is defined by unique code phase shift. This arrangement is possible when the spread spectrum system relies on synchronized BSs. Consequently, time uncertainty between BSs is solved, i.e. all BSs have common time reference where the most common time reference is supported by GPS. On the other hand, UTRA-FDD systems are asynchronous communication systems, so using different phases of the same scrambling code to identify BSs is not sufficient because resolving the code uncertainty is carried out prior to resolving the time uncertainty. Thus, each BS is identified by a unique code.

The initial cell search is performed when the MS is powered on. The MS starts searching for the strongest cell in its vicinity. In order to achieve initial code synchronization with the strongest cell, the MS should resolve both code and timing uncertainty.

As stated before, a total of 512 primary down-link complex scrambling codes, each of 34800 chips long are used but searching through all of the code with no prior timing information is ambiguous. In order to reduce the difficulty of searching through the 512 codes, UTRA-FDD code acquisition (initial cell search) is typically carried out in three stages: 1) slot synchronization, 2) frame synchronization and code-group identification, and 3) scrambling-code identification. The main goal of the initial cell search is to decode the cell identity of the acquired signal. Thus, two extra stages are needed: 4) frequency acquisition and 5) detecting cell identity. However, identifying and synchronizing to the down-link scrambling code is sufficient to identify any given cell of interest, since the mapping of the down-link scrambling codes and cell identity is unique (one scrambling code per cell). Therefore, the initial cell search is conducted in a three stage procedure [29].

The first stage of the cell search carries slot synchronization, where the MS uses the primary synchronization code to acquire slot boundaries of a cell. Usually, this can be done by using a single matched-filter matched to the primary synchronization code, since this code is common in all slots for all cells. However, in the presence of large frequency error, using classic matched-filter may result in a large amount of coherence loss. Several schemes are proposed to deal with this problem. The first scheme uses partial correlation where the PSC is divided into short sequences. Therefore, the PSC classic matched-filter is divided into number of shorter filters, each is matched to a short sequence of the total PSC. The outputs of the matched-filters are combined non-coherently [30]. The timing information of slot boundaries of the cell can be obtained by using an envelope detector to detect peaks in the matched-filter output and comparing to a cretin threshold. The second scheme uses an inner-slot differential combining that utilizes partial correlation of the PSC where the PSC is also divided into shorter sequences. This scheme uses two successive partial correlation values to extract the differential product, and then the timing information of the slot boundaries of the cell is obtained by using the maximum likelihood method [31].

The second stage of the cell search procedure carries frame synchronization and code-group identification, where the MS uses the secondary synchronization code to find frame boundaries and identify the code group of the BS that is found in the first step. This is done by correlating the received signal of SCH with all 16 possible secondary synchronization code sequences using a bank of 16 correlators. Then, the outputs of the correlators are accumulated over a predefined interval according to the 64 RS code word, and by finding the maximum correlation value, the second acquisition stage is finished. Since the cyclic shifts of the sequences are chosen to be unique, the code group and the frame boundaries are determined. In the second stage coherent or non-coherent detections may be used [30].

The third stage of the cell search procedure carries scrambling-code identification, where the MS determines the exact primary scrambling code used by the cell. Typically, the primary scrambling code is identified through symbol-by-symbol correlation over the CPICH within all codes within the identified code group in the second stage. One method is to correlate each received CPICH symbol with a predefined number of hypothesis, where the correlations are carried out over 256 or 64 chips duration over one frame. For each correlated symbol, the long scrambling code with the largest value is selected. After collecting 150 symbols, 10 symbols per slot, the scrambling code with the highest score is selected and compared against predetermined threshold [30]. Another suggested method is to implement the serial active code acquisition system using 16 parallel functions [32]. After identifying the primary scrambling code, the P-CCPCH can be detected. Since the primary scrambling code scrambles the cell specific broadcast channel

(BCH), this information can now be recovered. Both PSC and SSC are not subjected to multiplication by either channelization code or scrambling codes.

Whenever the MS has received information about which scrambling codes to search for, the second and third stages can be simplified. The fourth stage carries frequency acquisition based on CPICH in order to reduce initial frequency error so that the MS can decode the BCH, and the fifth stage carries detecting cell identity by reading the BCH information.

Chapter 4

Simulation for Slot Acquisition in UTRA-FDD Systems

4.1 Pipeline Approach to UTRA-FDD Code Acquisition

In UTRA-FDD systems, the three code acquisition stages can be arranged in very efficient pipelining architecture where each stage's acquisition results are continuously fed to the next stage every in pipelining cycle. The pipelining design of verification circuits inherently reduces the penalty time due to false alarm or missing correct code. The pipeline cycle is typically one 15-slot frame (slow processing) but it could be shorter, e.g. one-slot cycle (fast processing). All of the acquisition and verification circuits are always in use, e.g. while the third stage is processing the previous results of the second stage, the second stage is running in parallel trying to find a new code group for the primary scrambling code and frame boundaries to be fed again to the third stage. To facilitate the efficient pipeline architecture, we assumed that the processing time of the three code acquisition stages are equal, i.e. n_1=n_2=n_3 slot(s). Figure 4.1 shows the fast processing design where the processing cycle equals one slot, i.e. 670 μs, and each one of the verification circuits uses four processing cycles. Assuming the first three verification trials failed and the fourth succeeded, the total acquisition time in the parallel pipelined case is $(6n_1) + (n_1n_4) = 10$ slots where n_1n_4 is the verification time and $6n_1$ is the wasted time. This time is much less than the serial search with no pipelining nor parallelism, which is given in this instance by 4 trials multiplied by the search time for each trial, i.e. $4(3n_1 + n_1n_4) = 12n_1 + 4n_1n_4 = 28$ slots. The reason is simple since most verification losses due to code misses or false alarms are absorbed in the parallel pipelined architecture. In Figure 4.1, the verification circuits are reused, for example after $7n_1$ slots, all 4 verification circuits are in use. Fortunately, the first verification circuit finishes its work at this time and can be reassigned to verify the results of third stage. At any time (except for the few slots after starting the acquisition operation), the three acquisition stages and the four verification circuits are in use all the time. Such combined use of parallelism and pipelining will reduce the mean acquisition time as we will see shortly.

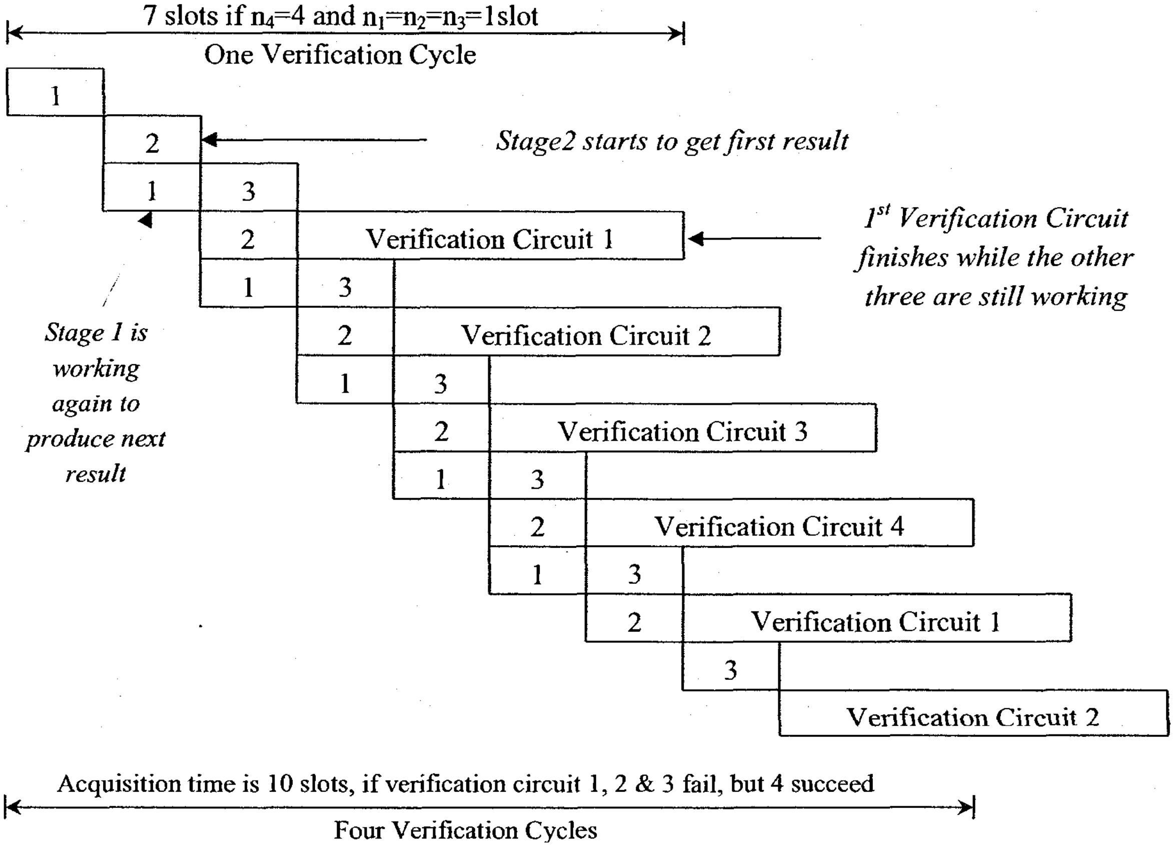
7 slots if n_4=4 and n_1=n_2=n_3=1slot
One Verification Cycle
1
2
Stage2 starts to get first result
1
3
2
Verification Circuit 1
1st Verification Circuit finishes while the other three are still working
1
3
Stage 1 is working again to produce next result
2
Verification Circuit 2
1
3
2
Verification Circuit 3
1
3
2
Verification Circuit 4
1
3
2
Verification Circuit 1
3
Verification Circuit 2
Acquisition time is 10 slots, if verification circuit 1, 2 & 3 fail, but 4 succeed
Four Verification Cycles

Figure 4.1: UTRA-FDD parallel pipelined operation with 4 simultaneous verification circuits (fast processing case) [33].

Besides fast processing design, slow processing design was also investigated, where n_1=15 slots and n_4=1. This implies that code verification time is one frame = $n_1 \times n_4$ =15 slots. Also, each acquisition stage costs one frame of 15-slots to produce an output. Figure 4.2 shows the slow processing design where the processing cycle equals one frame, i.e. 10 ms. In the slow processing case, we utilize only one verification circuit in parallel pipelined architecture where its cycle finishes before the third acquisition stage produces an output. In Figure 4.2, the employed verification circuit is ready and can be reassigned to verify the next results of the third stage at any time.

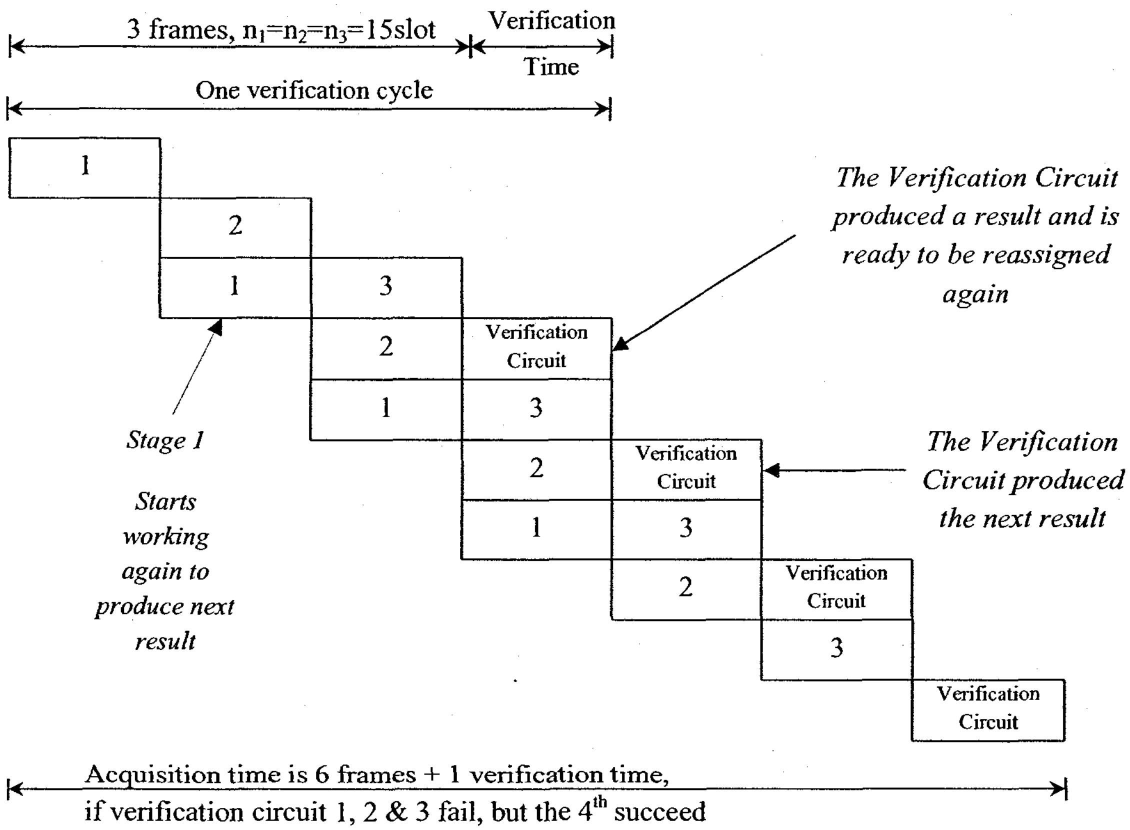
3 frames, n1=n2=n3=15slot
Verification Time
One verification cycle
1
2
1
3
2
1
Verification Circuit
3
2
1
Verification Circuit
3
2
Verification Circuit
3
Verification Circuit
Stage 1
Starts working again to produce next result
The Verification Circuit produced a result and is ready to be reassigned again
The Verification Circuit produced the next result
Acquisition time is 6 frames + 1 verification time, if verification circuit 1, 2 & 3 fail, but the 4th succeed

Figure 4.2: UTRA-FDD parallel pipelined operation with 1 verification circuits (slow processing case).

4.2 Signal and System Model

The first acquisition stage in UTRA-FDD systems is the most crucial stage since a large amount of uncertainty has to deal with. Finding the slot boundaries with a high degree of confidence is essential in order to reduce the total acquisition time in UTRA-FDD systems. In particular, we investigate many algorithms for combining the matched-filter results of each slot of the frame. Rayleigh fading, various amounts of frequency offsets and Additive White Gaussian Noise (AWGN) are assumed. The AWGN represents the data and control signals of all BSs and MSs except the PSC. Each one of the four BSs that are assumed to be in the vicinity of the applicable MS is simulated with four multipath signals, consequently producing 16 possible multi paths that can be detected.

At the BS transmitter QPSK modulation technique is used, so the data and control signals will be multiplied by the Quadrature carriers $cos(\omega_c t)$ and $sin(\omega_c t)$ in the Inphase and Quadrature channels respectively, where ω_c is the angular carrier frequency. The transmitted signal will be subjected to propagation delay, τ, and random phase error, ϕ, according to the ℓ^{th} taken path of a BS. The fading channel is represented by, g, the Rayleigh fading factor that depends on the ℓ^{th} taken path of a BS's multipath. All received samples of a given signal PSC, SSC, P-CPICH, ...etc of the same frame have the same random phase error, samples of 10 frames have the same Rayleigh fading envelope which corresponds to a Doppler shift of 10 Hz. The received signal delay with respect to local MS time does not change since the BS synchronization signals i.e. PSC, SSC and P-CPICH, are all transmitted synchronously with each other and in streaming mode.

The exact transmitted signals from all BSs and MSs are received at the intended MS. However, since we are working on the first acquisition stage of UTRA-FDD systems, we preserve the PSC signal and replace all interference signals by an equivalent AWGN. The AWGN, $n(t)$, represents the SSC, P-CPICH, other BSs signal, MPs, users' data ... etc whose variance depends on the load level. The received signal, $r(t)$, at the MS receiver representing the PSC signal plus the equivalent noise is written in (4.1). The double summations represent the 256 chips of the received PSC.

$$r(t)=\sum_{b=1}^{NBS}\sum_{\ell=1}^{NMP} g_{b\ell}c_p(t-\tau_{b\ell})[\cos(\omega_c(t-\tau_{b\ell})+\phi_{b\ell})+\sin((\omega_c(t-\tau_{b\ell})+\phi_{b\ell})]+n(t) \qquad (4.1)$$

NBS : The number of base stations.

NMP : The number of multi paths.

$g_{b\ell}$: The Rayleigh fading factor according to the ℓ^{th} taken path of a BS.

$c_p(t)$: The primary synchronization code PSC.

$\tau_{b\ell}$: The propagation delay according to the ℓ^{th} taken path of a BS.

ω_c : Carrier frequency.

$\phi_{b\ell}$: The random phase error according to the ℓ^{th} taken path of a BS which is uniformly distributed between $[0, 2\pi]$.

$n(t)$: The equivalent additive white Gaussian noise.

Figure 4.3 shows the MS's receiver which includes the carrier removal, the despreading stage using matched-filter and the peak analyzer for the first acquisition stage of the three stages cell search. The carrier recovery is applied to all signal cases, i.e. coherent and non-coherent signal cases. The receiver tries to acquire the slot timing by correlating the received signal with the P-SCH code using a matched-filter whose tab coefficient is matched to the PSC as in Figure 4.3. The received signal is correlated in the Inphase and Quadrature branches by the local MS carrier, then the signals are passed through low-pass filters to suppress the high frequency terms as in Figure 4.3. Two matched-filters, each of length 512 half-chips representing the over sampled PSC, operate on the low-pass filters' output signals resulting in x(t), and y(t) signals. Samples of the two matched-filters are combined coherently or non-coherently in different ways giving 5 cases; $x(t)$, $y(t)$, $x(t)+y(t)$, $x(t)-y(t)$ and $z(t)=\sqrt{x^2(t)+y^2(t)}$, which will be denoted by the values R = 1, 2, 3, 4 and 5 respectively. Since the Signal to Interference Ratio (SIR) is very low due to the load of the system, it is required to combine multiple correlation values from multiple slots to properly acquire the P-SCH code, consequently the slot timing. The obtained peaks at the output of the matched-filter indicate the existence and relative reception strength of the base stations in the vicinity of the user receiver. In addition, the peaks identify the slot boundaries for a specific base station and deliver estimations of the channel's amplitude and phase during a 256 chip block.

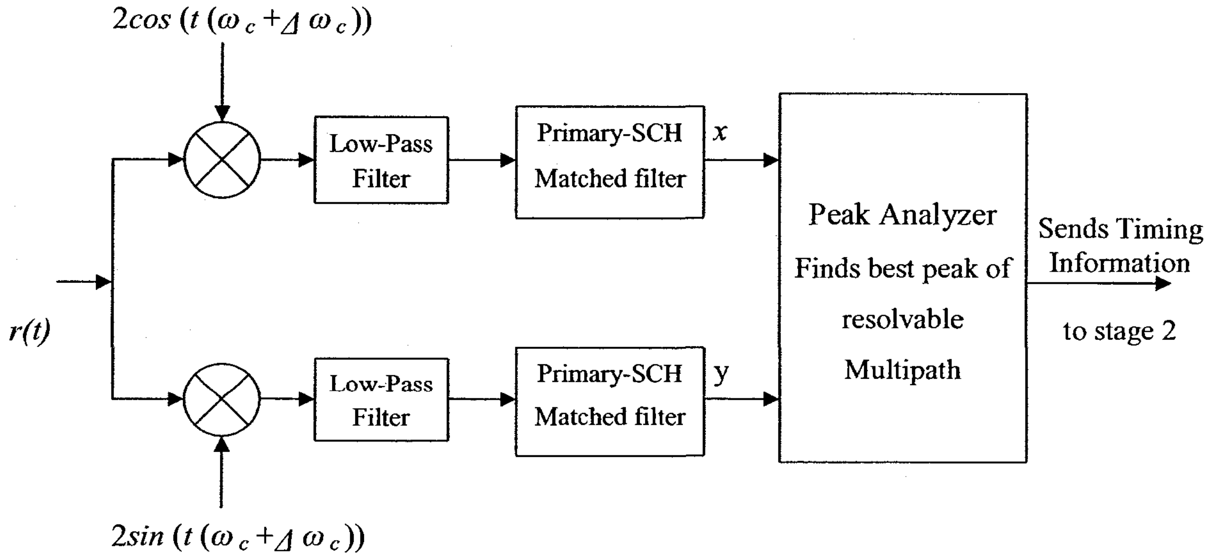
$2cos\,(t\,(\omega_c+\Delta\,\omega_c))$
$r(t)$
Low-Pass Filter
Low-Pass Filter
Primary-SCH Matched filter
Primary-SCH Matched filter
x
y
Peak Analyzer
Finds best peak of resolvable Multipath
Sends Timing Information to stage 2
$2sin\,(t\,(\omega_c+\Delta\,\omega_c))$

Figure 4.3: Carrier removal and dispreading prior to application of peak detection algorithm [33].

The outputs of the two matched-filters, in the Inphase and Quadrature channels, represent the received signal correlated with the PSC which are expressed in (4.2) and (4.3) as $x(t)$ and $y(t)$ respectively. The autocorrelation of the PSC is shown in Figure 4.4.

$$x(t)=\sum_{b=1}^{NBS}\sum_{\ell=1}^{MP} g_{b\ell}R_{PSC}(t-\tau_{b\ell})[\cos(\Delta\omega_c t+\omega_c\tau_{b\ell}-\phi_{b\ell})-\sin(\Delta\omega_c t+\omega_c\tau_{b\ell}-\phi_{b\ell})]+n(t) \tag{4.2}$$

$$y(t)=\sum_{b=1}^{NBS}\sum_{\ell=1}^{MP} g_{b\ell}R_{PSC}(t-\tau_{b\ell})[\cos(\Delta\omega_c t+\omega_c\tau_{b\ell}-\phi_{b\ell})+\sin(\Delta\omega_c t+\omega_c\tau_{b\ell}-\phi_{b\ell})]+n(t) \tag{4.3}$$

R_{PSC} : Autocorrelation function of PSC.

$\Delta\,\omega_c$: The angular frequency error (offset).

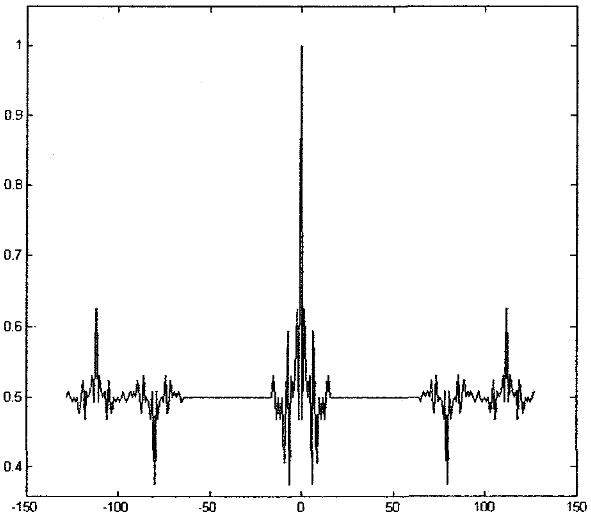

Figure 4.4: Primary Synchronization Code auto-correlation function.

4.3 Mean Acquisition Time Calculation

Now, at the MS peak analyzer and for all subsequent peak detection algorithms we have four coherent cases and one non-coherent case depending on whether the tested signal is $x(t)$, $y(t)$, $x(t)+y(t)$, $x(t)-y(t)$ or $z(t)$ respectively. Since the PSC is the same in all slots for all BSs, we introduce the sliding frame concept to obtain better averaging, i.e. each 15 consecutive slots are considered to be one frame and are processed according to the new peak detection algorithms. In addition, at the MS, a pipelined acquisition operation is assumed, i.e. the BS epoch (slot boundaries) of the previous frame is found at the end of each slot. For example, at the end of slot 16 (first slot of second frame) the applicable acquisition results for the previous first frame (slots 1-15) is found. At the end of slot 17 (second slot of second frame) the applicable acquisition results for slots 2-16 (15-slots frame window) is found and so on. The frames are sliding slot by slot as shown in Figure 4.5. Moreover, Figure 4.1 and Figure 4.2 illustrate the computation times or processing times of each stage and not the length of the actual received and processed signals. The TT number of true frames results in giving F number of sliding frames which is expressed in (4.4). Where each processed true frame will produce 15 results except the last frame will produce one.

$$F = ((TT - 1) \times 15) + 1 \qquad \text{Sliding Frames.} \qquad (4.4)$$

Each slot has 2560 chips as per the UTRA-FDD standard. We over sample the received signal by 2 in order to deal with the misalignment problem, i.e. a code phase error of ± ½ chip. Therefore, we obtain 5120 received signal samples in each slot. Multiplying this by 15 (total number of slots per frame) results in giving 76800 received signal samples per frame. A Large number of frames is used to simulate the signals. Then, we calculate the probability of slot boundaries detection and false alarm, P_{d1} and P_{fa1}, respectively based on the four peak detection algorithms that will explained be in the following sub-sections.

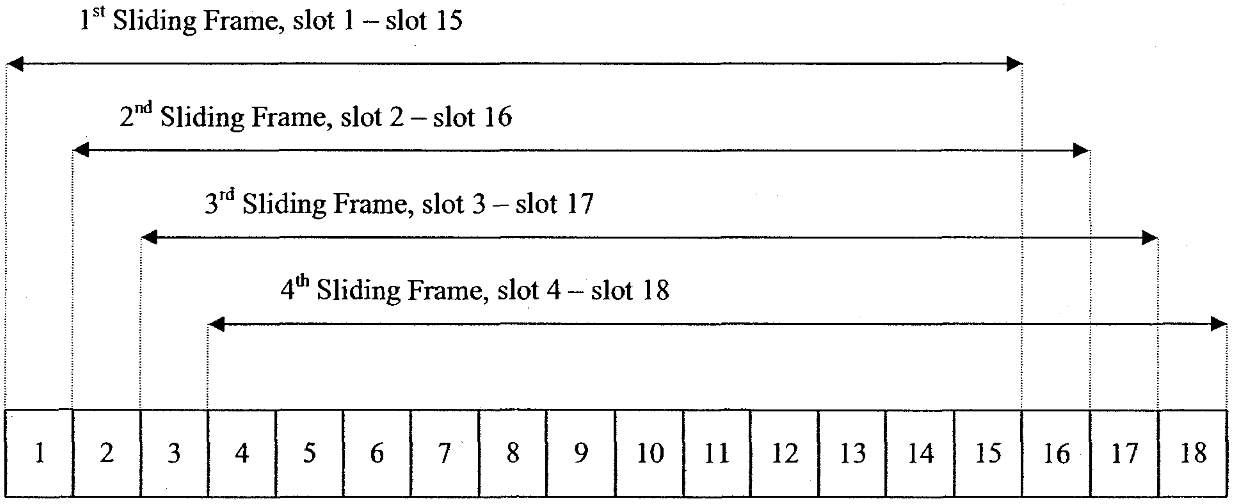
1st Sliding Frame, slot 1 – slot 15
2nd Sliding Frame, slot 2 – slot 16
3rd Sliding Frame, slot 3 – slot 17
4th Sliding Frame, slot 4 – slot 18
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

Figure 4.5: Running Sliding frames.

The main results in our simulation are only related to the first stage acquisition, i.e. slot boundaries detection. On the other hand, the detection and false alarm probabilities for the first stage are found and good bounds can be still computed for the three stages acquisition time. The following mean acquisition time computations apply to all of the four peak detection algorithms and all of the signal cases. The parallel pipelined acquisition systems are considered to be under steady state operation where all stages are engaged. Figures 4.1 and 4.2 illustrate the fast and slow processing cases respectively. The probabilities of success in the three acquisition stages are denoted as P_{d1}, P_{d2} and P_{d3}

respectively. The corresponding verification time is n_1n_4 slots, where n_1 is the processing cycle and n_4 is the number of verification circuit(s). The costly and fast processing case n_1=1 and n_4=4, and in the slower and less costly processing case n_1=15 and n_4=1 [33]. If the three acquisition stages are successful in their functions, where this event takes place with probability a = $P_{d1} \times P_{d2} \times P_{d3}$. Now, we assume that all of the three acquisition stages have the same processing time, i.e. n_1 slots, the mean acquisition time (T_{acq}) equals $3n_1$ slots to finish processing the signal in the three acquisition stages, each stage equals n_1, plus verification time equals n_1n_4 slots. Therefore, the total acquisition time for the success of all stages can be written as $n_1(3+n_4)$ slots in the fast and slow processing cases. If one or more of the first three acquisition stages were not successful, e.g. the first verification cycle is not successful but the second will be, this event take place with probability equal to $(1-a)a$ and T_{acq} is $n_1(3+ n_4) + n_1$ slots [33]. As we can see the cost is only one additional pipeline cycle of n_1 slots. If the first and second verification cycles failed but the third succeeded, another n_1 slots is added with probability equals to $(1-a)^2a$ and T_{acq} becomes $n_1(3+ n_4) + 2n_1$. This aspect is illustrated in both Figures 4.1 and 4.2. At any time, the acquisition system picks the result of the first successful verification circuit then declares acquisition and starts to listen to the pilot and broadcast channels of the BS. Averaging over all possibilities, the mean acquisition time can be computed as one verification cycle time, i.e. fill up time and final acquire time, added to the number of extra cycles, n_1 slots, in case of false alarm. This idea is expressed as follows [33]:

$$T_{acq} = n_1(n_4 + 3) + n_1 \sum_{i=0}^{\infty} ia(1-a)^i$$

$$= n_1(n_4+3) + \frac{n_1(1-a)}{a} \qquad \text{Slots–time} \qquad (4.5)$$

In this thesis, we will compute the results for n_1=1, i.e. pipeline cycle equals to one slot, or n=15, i.e. pipeline cycle equals to one 15-slot frame. The first value means the fast processing case with less acquisition time but is very demanding in terms of hardware and software. The second value means the slow processing case with more acquisition time but is not demanding in terms of hardware and software. In our simulation model, the fast processing case is possible for the first stage while using the sliding frame concept because the same PSC is transmitted in all slots over the entire cell. However, for the second and third stages of the UTRA-FDD cell search, more hardware would be required if the processing time is n_1=1 slot rather than the typical one frame processing time, i.e. n_1=15 slots (slower processing case). It is important to notice that Figures 4.1 and 4.2 show only processing times of the stages and not the length of the processed signals. For example, samples of a sliding frame may be processed within one slot time only or one frame time (15 slots).

We have assumed two scenarios in calculating the mean acquisition time where the fast and slow processing cases are applied in both scenarios. The first assumption is that the results of the three acquisition stages are independent, i.e. worst case assumption with $P_d = P_{d1} \times P_{d2} \times P_{d3}$. The obtained two mean acquisition times in the worst case are considered as an upper bound of the mean acquisition time. The second assumption is that the first stage performance is the limiting factor of the system, i.e. correlated best case assumption with $P_d = P_{d1} = P_{d2} = P_{d3}$. In the correlated situations, high interference values may lead to incorrect results from all of the three acquisition stages and more accurate results may lead to success of all three acquisition stages. In the correlated case, two more mean acquisition times are obtained from (4.5) by substituting $a=P_d$ and not $a=P_d^3$. These two additional results are considered as a lower bound of the mean acquisition time.

4.4 Peak Detection Techniques

The following peak detection algorithms are to be implemented in the peak analyzer block and applied to the output of matched-filters as shown in Figure 4.3. In order to proceed with the description of the peak detection algorithms, definition of some criteria should be introduced where they will be combined in various ways. Each received frame is processed according to Figure 4.6, where the corresponding k^{th} sample location in each

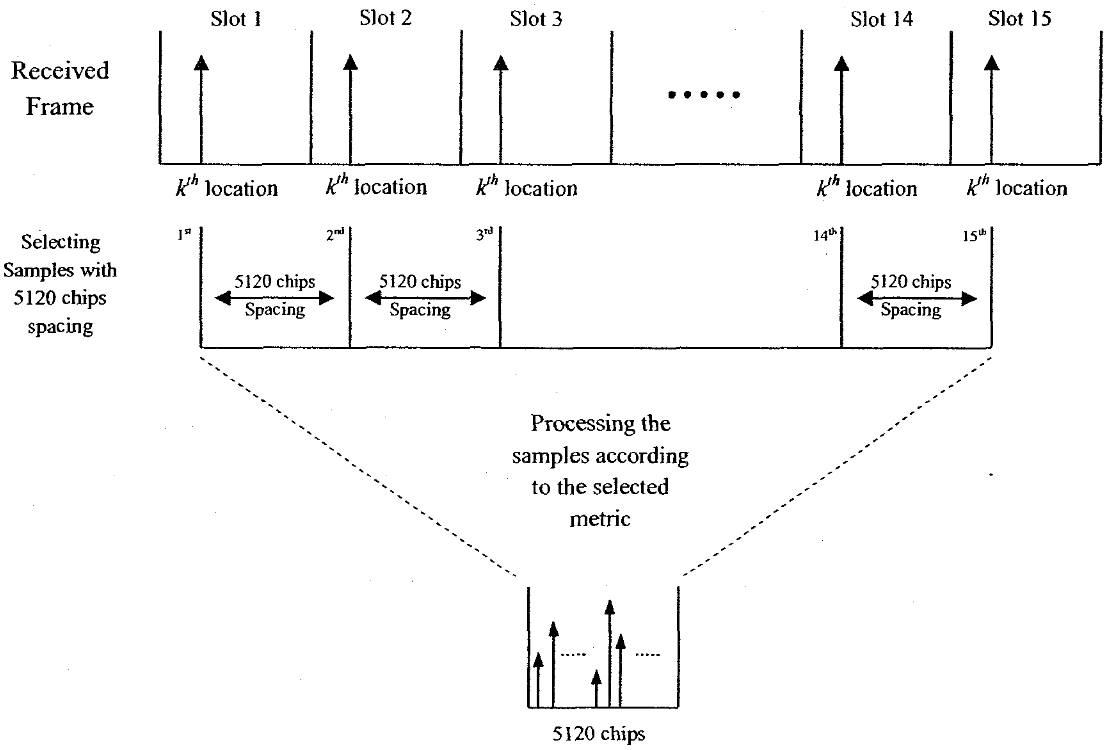

Figure 4.6: General criteria processing model

slot is selected and processed according to the selected criteria. Consequently, they will reflect the quality of the obtained samples. R on the right hand side denotes the investigated signal, and C on the left hand side denotes the signal number.

The first criteria is denoted as $CR1(k)$ where $CR1(k)$ reflects the relative strength of the received signal at the k^{th} candidate sample location i.e. $CR1(k)$ represents the number of high samples of different slots exceeding a threshold (TH) at a certain location k, where $k=1, 2,\ldots$ 5120 and j represents the slot number per frame.

$$CR1(k) = \sum_{j=1}^{j=14} MAX\left\{0, \text{sgn}(ABS(R(5120.j.Tc/2+(k-1).5120Tc/2)) - TH)\right\} \qquad (4.6)$$

For example for $R = 1$, if $x(t)$ samples at the 87th location inside the slot, i.e. $x(87)$, $x(87+5120)$, $x(87+2\times5120)$ ...etc all exceed the TH, then the peak analyzer considers sample 87 is a good candidate for slot start. If the samples vary from exceeding to not exceeding TH more frequently, then this sample location is not a good candidate for slot start. The TH is recalculated for each sliding frame, by averaging the sum of all samples of the absolute value of the investigated signal.

The second criteria is denoted as *CR2(k)* where $CR2(k)$ reflects the frequency of change of signs of consecutive samples from slot to slot, i.e. *CR2(k)* = 15− number of sign switching in successive R^{th} received signal samples of different slots corresponding to same k position of successive slots, $k=1, 2,\ldots$ 5120. For example for $R = 2$, if $y(t)$ samples at the 53rd location within the slot, i.e. $y(53)$, $y(53+5120)$, $y(53+2\times5120)$ ...etc all are positive or negative, then the peak analyzer considers sample location 53 is a good candidate for slot start. If the samples alternate from positive to negative more frequently, then this sample location is not a good candidate for slot start.

Now, combining $CR1(k)$ with $CR2(k)$ will give us a third criteria denoted as $CR3(k)$ where $CR3(k)$ reflects relative strength of the received signal at the k^{th} candidate sample location and the frequency of change of signs of consecutive samples from slot to slot, i.e. the most probable slot start is the candidate that exceed the TH and the sign is the same in all slots.

$$CR3(k) = CR1(k) + CR2(k) \tag{4.7}$$

4.4.1 Algorithm 1, Best of the Average of the Absolute

The absolute values of the samples at a certain location k are averaged over the 15 slots of each sliding frame thus obtaining only 5120 running average samples of the original 76800 samples of each sliding frame and denoted as $DR1(k)$. The peak analyzer subtracts 256 T_C from the sample location of the maximum of the 5120 samples to yield the most probable slot start. This timing information is then sent to the following SSC stage (2nd stage) of the overall acquisition system.

$$DR1(k) = \text{Average of the absolute of } R(k) \text{ over 15 slots} \tag{4.8}$$

The mean acquisition time bounds are calculated according to the algorithm shown in Figure 4.7. The selected sample location (best candidate for slot start) out of 5120 samples is compared against the known locations of the 2×16 *BS*s multi-paths, where the ×2 is attributed to over-sampling. If the location of the maximum value happened to coincide with one of the known BSs multi-path locations, the P_d counter is incremented by 1 for each of the sliding frames. P_d is finally obtained by dividing the P_d counter by the total number of the sliding frames.

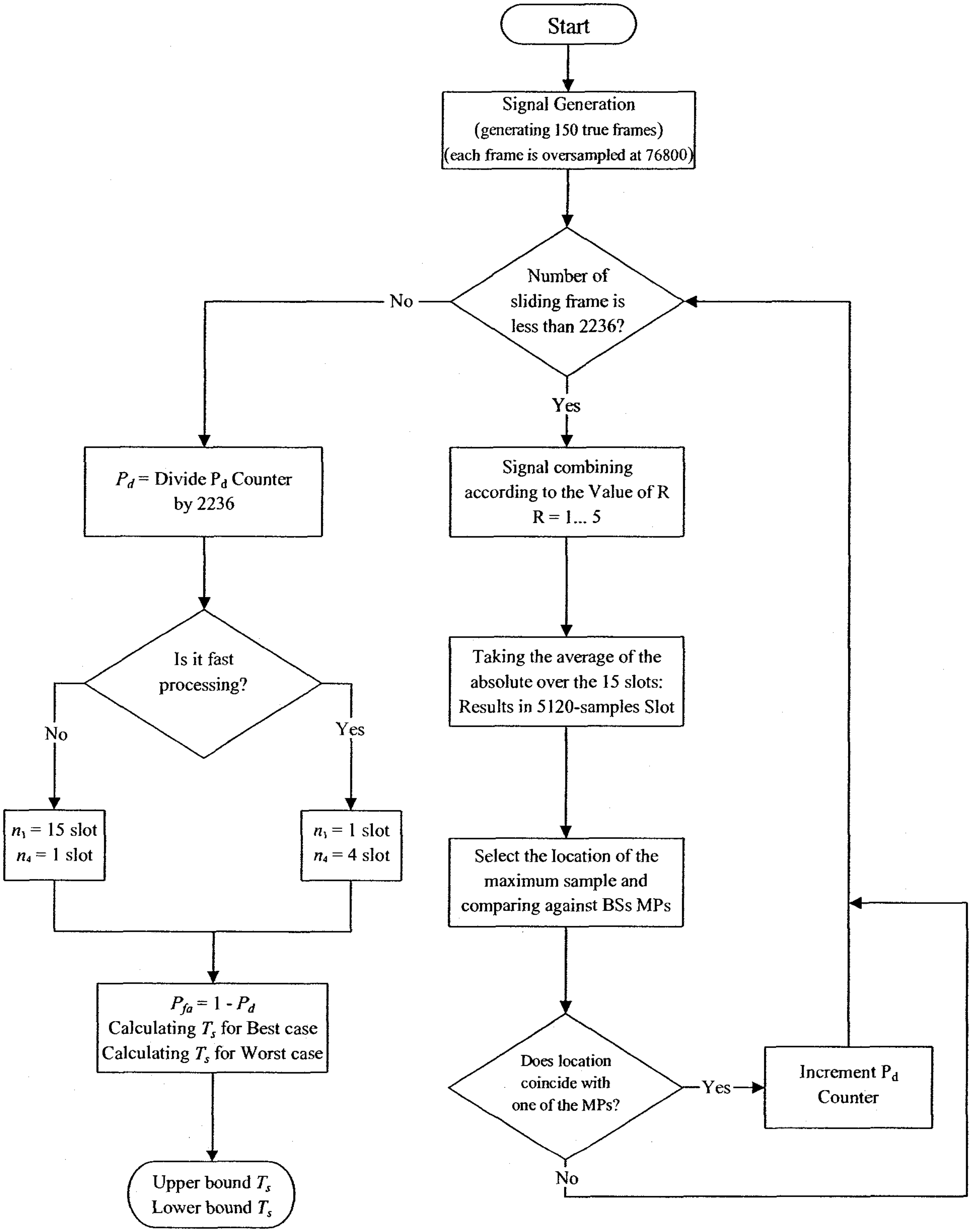
Start
Signal Generation
(generating 150 true frames)
(each frame is oversampled at 76800)
Number of sliding frame is less than 2236?
No
Yes
Signal combining according to the Value of R
R = 1... 5
Taking the average of the absolute over the 15 slots: Results in 5120-samples Slot
Select the location of the maximum sample and comparing against BSs MPs
Does location coincide with one of the MPs?
Yes
No
Increment P_d Counter
P_d = Divide P_d Counter by 2236
Is it fast processing?
No
Yes
n_1 = 15 slot
n_4 = 1 slot
n_1 = 1 slot
n_4 = 4 slot
P_{fa} = 1 - P_d
Calculating T_s for Best case
Calculating T_s for Worst case
Upper bound T_s
Lower bound T_s

Figure 4.7: A flowchart explaining Algorithm 1, Best of the Average of the Absolute.

The investigated signal varies depending on the case, i.e. it is $x(t)$, $y(t)$, $x(t) \pm y(t)$ or $z(t)$ corresponding to R = 1 ... 5. Once P_d and P_{fa} =1− P_d are found, the mean acquisition times are computed as in equation (4.5). We notice that case 5 of this algorithm, R = 5, corresponds to a modified version of the classic matched-filter detection, where the processed signal is $z(t) = \sqrt{x^2(t) + y^2(t)}$. However, this modified technique is TH independent. The adjustment of this TH, its variation and its optimal choices are delicate and complex issues that have been treated in [34].

4.4.2 Algorithm 2, Threshold Application and Amplitude Scaling

For all k, $CR1(k)$ is multiplied by the corresponding value of $DR1(k)$ for all k to yield:

$$CR4(k) = CR1(k) \times DR1(k) \qquad (4.9)$$

The peak analyzer subtracts 256 T_C from the sample location that yields maximum $CR4(k)$ over all k to yield the most probable slot start where the mean acquisition time bounds are calculated according to the algorithm shown in Figure 4.8. The selected sample location (best candidate for slot start) out of 5120 samples of $CR4(k)$ is compared against the known locations of the 2×16 BSs multi-paths, where the ×2 is attributed to over-sampling. If the location of the maximum value happened to coincide with one of the known BSs multi-path locations, the P_d counter is incremented by 1 for each of the sliding frames. P_d is finally obtained by dividing the P_d counter by the total number of the sliding frames. The investigated signal varies depending on the case, i.e. it is $x(t)$, $y(t)$, $x(t) \pm y(t)$ or $z(t)$ corresponding to R = 1 ... 5. The mean acquisition times are computed as per equation (4.5) after finding P_d and $P_{fa} = 1 - P_d$.

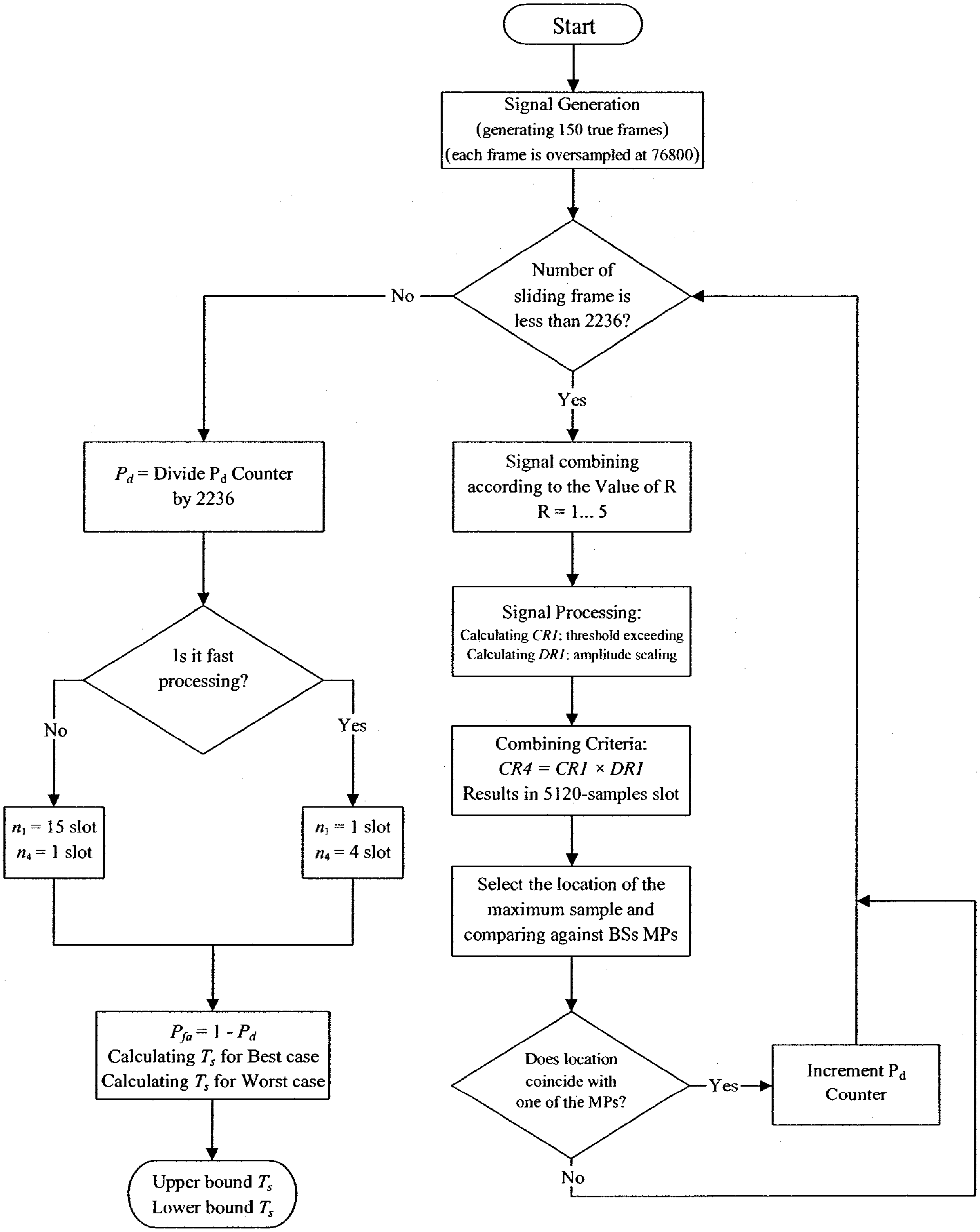
Start
Signal Generation
(generating 150 true frames)
(each frame is oversampled at 76800)
Number of sliding frame is less than 2236?
No
Yes
P_d = Divide P_d Counter by 2236
Signal combining according to the Value of R
R = 1... 5
Signal Processing:
Calculating *CR1*: threshold exceeding
Calculating *DR1*: amplitude scaling
Is it fast processing?
No
Yes
Combining Criteria:
CR4 = *CR1* × *DR1*
Results in 5120-samples slot
n_1 = 15 slot
n_4 = 1 slot
n_1 = 1 slot
n_4 = 4 slot
Select the location of the maximum sample and comparing against BSs MPs
P_{fa} = 1 - P_d
Calculating T_s for Best case
Calculating T_s for Worst case
Does location coincide with one of the MPs?
Yes
Increment P_d Counter
No
Upper bound T_s
Lower bound T_s

Figure 4.8: A flowchart explaining Algorithm 2, Threshold application and Amplitude Scaling.

4.4.3 Algorithm 3, Sign Switching, Threshold Application and amplitude scaling

$CR3(k)$ is defined as the addition of two criteria as in (4.7). If samples at the same location within each slot jump from positive to negative more frequently or samples have no relative strength as compared to TH, then this sample location is not a good candidate for slot start. Multiplying $CR3(k)$ by the corresponding value of $DR1(k)$ for all k yields:

$$CR5(k) = CR3(k) \times DR1(k) \qquad (4.10)$$

The peak analyzer subtracts 256 T_C from the sample location that yields maximum $CR5(k)$ over all k to yield the most probable slot start where the mean acquisition time bounds are calculated according to the algorithm shown in Figure 4.9. The sample location of the maximum (the detected slot start) of the 5120 samples of $CR5(k)$ is compared against the known locations of the 2×16 *BS*s multi-paths, where the use of ×2 is attributed to over-sampling. If the location of the maximum value coincides with one of the known *BS*s multi-path locations, the P_d counter is incremented by 1 for each of the sliding frames. Finally, P_d is computed by dividing the P_d counter by the total number of the sliding frames. The investigated signal varies depending on the case, i.e. it is $x(t)$, $y(t)$, $x(t) \pm y(t)$ or $z(t)$ corresponding to $R = 1 \ldots 5$. Once P_d and $P_{fa} = 1 - P_d$ are found, the mean acquisition times are computed as per equation (4.5).

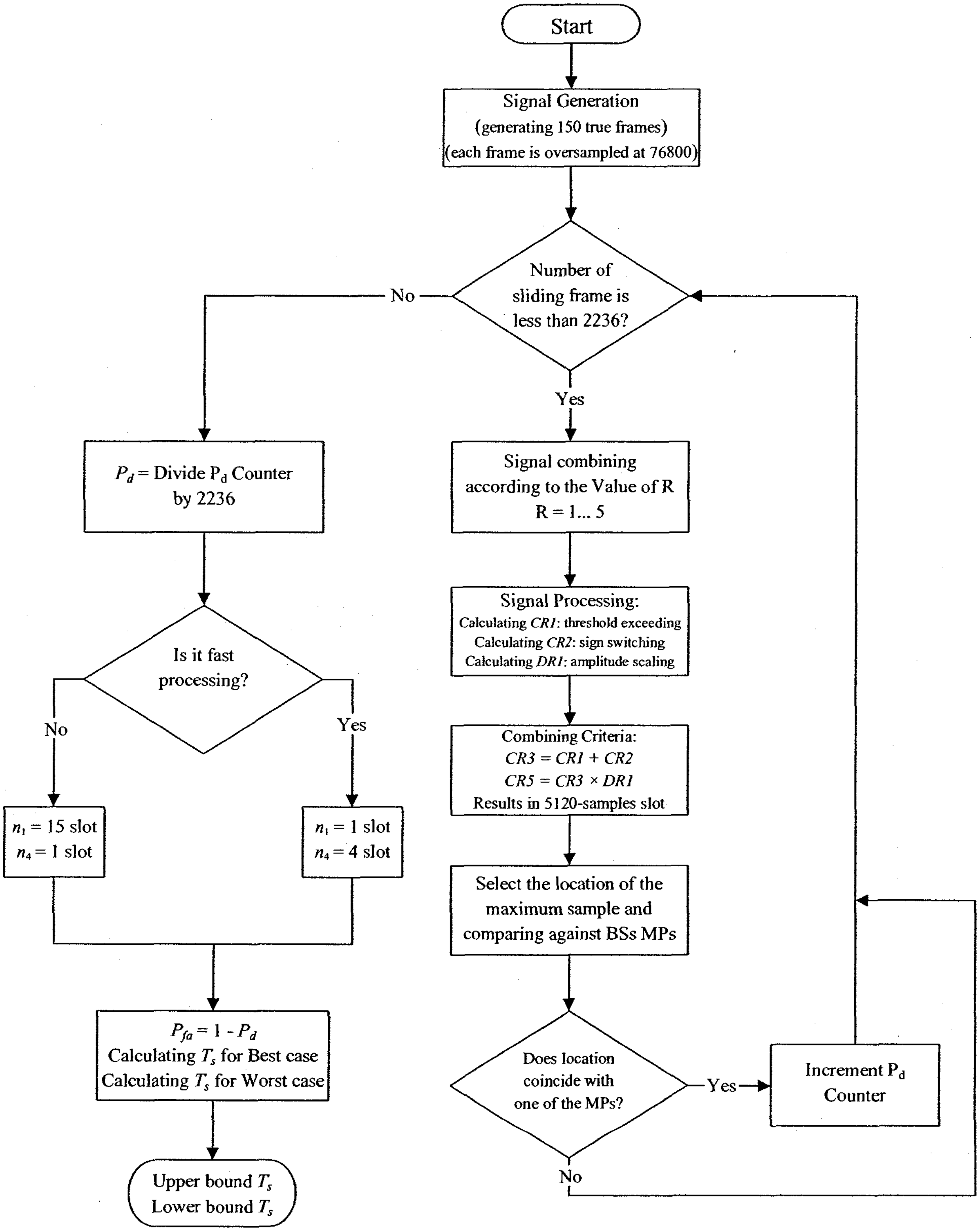
Start
Signal Generation
(generating 150 true frames)
(each frame is oversampled at 76800)
Number of sliding frame is less than 2236?
No
Yes
P_d = Divide P_d Counter by 2236
Signal combining according to the Value of R
R = 1... 5
Signal Processing:
Calculating $CR1$: threshold exceeding
Calculating $CR2$: sign switching
Calculating $DR1$: amplitude scaling
Is it fast processing?
No
Yes
Combining Criteria:
$CR3 = CR1 + CR2$
$CR5 = CR3 \times DR1$
Results in 5120-samples slot
n_1 = 15 slot
n_4 = 1 slot
n_1 = 1 slot
n_4 = 4 slot
Select the location of the maximum sample and comparing against BSs MPs
$P_{fa} = 1 - P_d$
Calculating T_s for Best case
Calculating T_s for Worst case
Does location coincide with one of the MPs?
Yes
Increment P_d Counter
No
Upper bound T_s
Lower bound T_s

Figure 4.9: A flowchart explaining Algorithm 3, Sign Switching, threshold application and amplitude scaling.

4.4.4 Algorithm 4, Best of the Positive Samples

The MS keeps positive valued samples of each sliding frame and replaces negative ones with zeros. Then, it averages samples at a certain location k over the 15 slots of each sliding frame. Thus, it obtains only 5120 running average samples (that call $CR6(k)$) of the original 76800 samples of each sliding frame. Where R refers again to the signal investigated, $R = 1 \ldots 4$. The peak analyzer subtracts 256 T_C from the sample location that yields the maximum $CR6(k)$ over all k to yield the most probable slot start where the mean acquisition time bounds are calculated according to the algorithm shown in Figure 4.10. As before, if the sample location of the maximum (the detected slot start) of the 5120 samples of $CR6(k)$ coincides with one of the known *BS*s multi-path locations, the P_d counter is incremented by 1 for each of the sliding frames. Finally, P_d is computed by dividing the P_d counter by the total number of the sliding frames. Dividing the P_d counter by the number of the sliding frames gives the P_d value. Finally, the mean acquisition times are computed as per equation (4.5).

Taking negative valued samples of each sliding frame instead of positive samples is the logical complement of this algorithm. The fourth algorithm applies only for the four coherent signal cases $x(t)$, $y(t)$, $x(t) \pm y(t)$, i.e. R= 1, 2, 3 and 4, since all samples of the non-coherent signal case $z(t)$, i.e. R =5, are positive.

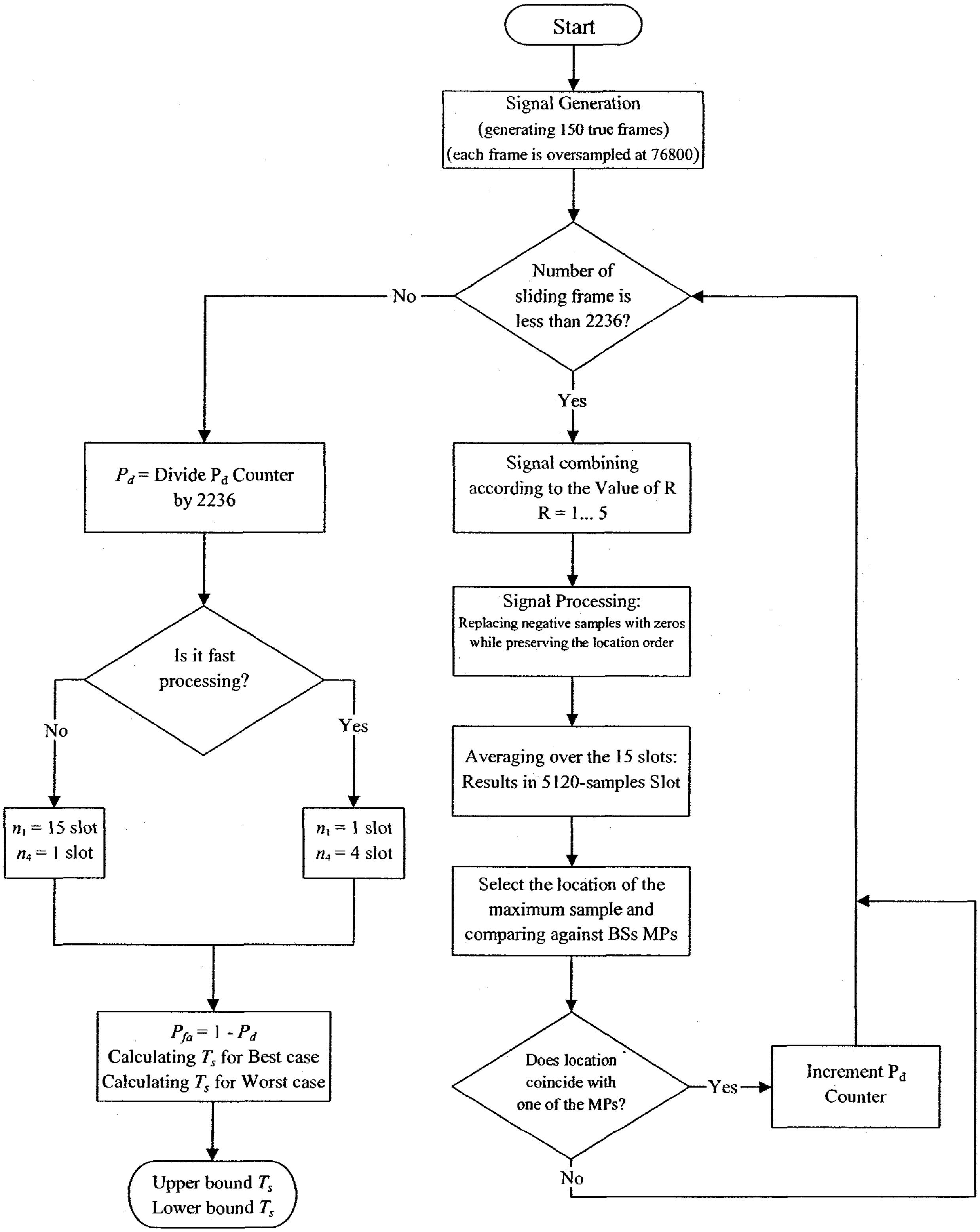
Start
Signal Generation
(generating 150 true frames)
(each frame is oversampled at 76800)
Number of sliding frame is less than 2236?
No
Yes
P_d = Divide P_d Counter by 2236
Signal combining according to the Value of R
R = 1... 5
Signal Processing:
Replacing negative samples with zeros while preserving the location order
Is it fast processing?
No
Yes
Averaging over the 15 slots: Results in 5120-samples Slot
n_1 = 15 slot
n_4 = 1 slot
n_1 = 1 slot
n_4 = 4 slot
Select the location of the maximum sample and comparing against BSs MPs
P_{fa} = 1 - P_d
Calculating T_s for Best case
Calculating T_s for Worst case
Does location coincide with one of the MPs?
Yes
Increment P_d Counter
No
Upper bound T_s
Lower bound T_s

Figure 4.10: A flowchart explaining Algorithm 4, Best of the Positive Samples.

4.4 Results and Comparisons

The simulation has been conducted with the purpose of testing the possibility of using a coherent detection scheme for the first acquisition stage, as well as the non-coherent case. Therefore, various signals were generated at the peak analyzer inputs as per section 4.2 at different SIR and fading level. Each signal case, i.e. $x(t)$, $y(t)$, $x(t) \pm y(t)$ or $z(t)$, was processed as per the specified four peak detection algorithms, where all of the criteria were computed for each signal sample. Subsequent application of the peak detection algorithm yielded the probability of correct slot detection P_d and the probability of false alarm P_{fa} as previously stated.

All signal cases were simulated at a Doppler bandwidth of 10Hz, fading signal envelope generated by standard normal distributions for fast processing, i.e. n_1=1, n_4=4 and slow processing, i.e. n_1=15, n_4=1, and for the worst case independent analysis ($a=P_d^3$) and the best case correlated analysis ($a=P_d$). All results were repeated at different carrier frequency errors Δf of 0 Hz, 100 Hz, 4 KHz and 20 KHz, and at each SIR value. Signals of TT = 150 true frames were simulated, which means F = 2236 sliding frames were averaged. The detection probabilities were obtained by simulation for the aforementioned peak detection algorithms while the mean acquisition times were computed from (4.5).

4.4.1 Probability of Detection Results

We have varied the carrier frequency errors to see the performance of the probability of detection for all of the peak detection algorithms. The minimum and maximum carrier errors are 0 Hz and 20 KHz respectively. The results obtained for 100 Hz and 4 KHz carrier errors came in between the limiting results of 0 Hz and 20 KHz. This verifies that many of the coherent techniques utilized in this thesis can work even at large carrier errors. The reason is the assumed presence of more than one BS and the utilization of the new peak detection algorithms which are resilient to the carrier errors.

Some of the obtained probability values are addressed in this sub section. Figures 4.11 through 4.13 show P_d values for the first peak detection algorithm at Δf = 0, 4 and 20 KHz respectively where first algorithm is based on the criteria $DR1(k)$ which is amplitude scaling. Figure 4.12 shows P_d values for all signal cases at Δf = 4 KHz where coherent case probabilities are slightly degraded than the non-coherent case. This difference increases with increasing Δf to 20 KHz as in Figure 4.13. Figure 4.13 shows a reasonable P_d values obtained for case 5 at Δf = 20 KHz (Matched-filter like), e.g. at SIR of -10 dB, P_d = 0.62, without even using the antenna diversity combining techniques.

Figures 4.14 through 4.16 show P_d values for the second peak detection algorithm at different Δf = 0, 4 and 20 KHz respectively where second algorithm is based on the criteria $CR4(k)$ which is threshold application and Amplitude scaling. The increase of the carrier frequency error from 0 Hz to 20 KHz results in decreasing P_d performance by 50% of their actual value for all signal cases. On the other hand, increasing the carrier error to 4 KHz leads to slight degradation in performance while the non-coherent case providing better values than the coherent case. Figure 4.15 shows this aspect.

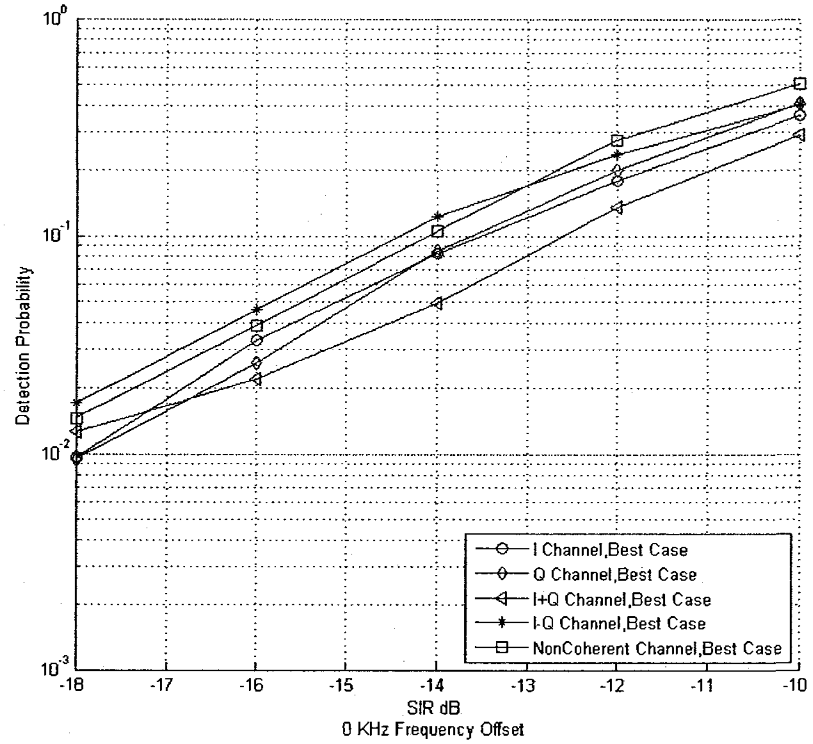

Figure 4.11: Algorithm 1, First stage detection probability @ Δ*f* = 0 Hz.

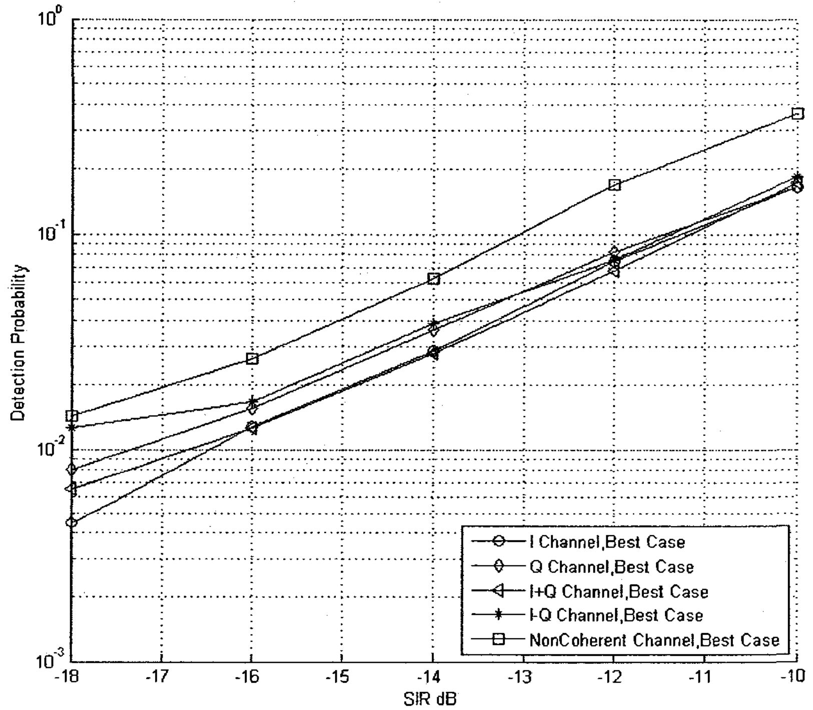

Figure 4. 12: Algorithm 1, First stage detection probability @ Δ*f* = 4 KHz.

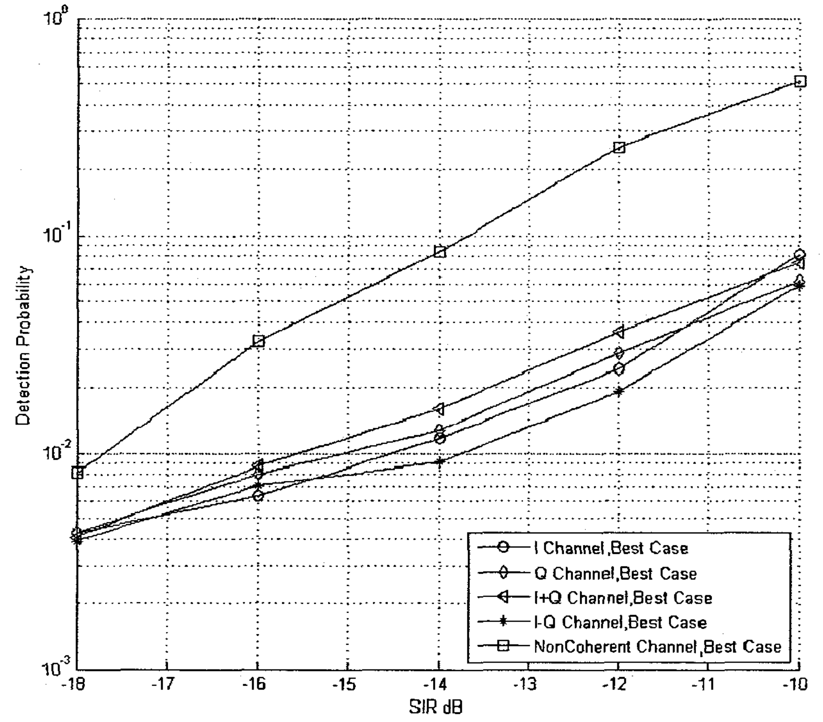

Figure 4.13: Algorithm 1, First stage detection probability @ $\Delta f = 20$ KHz.

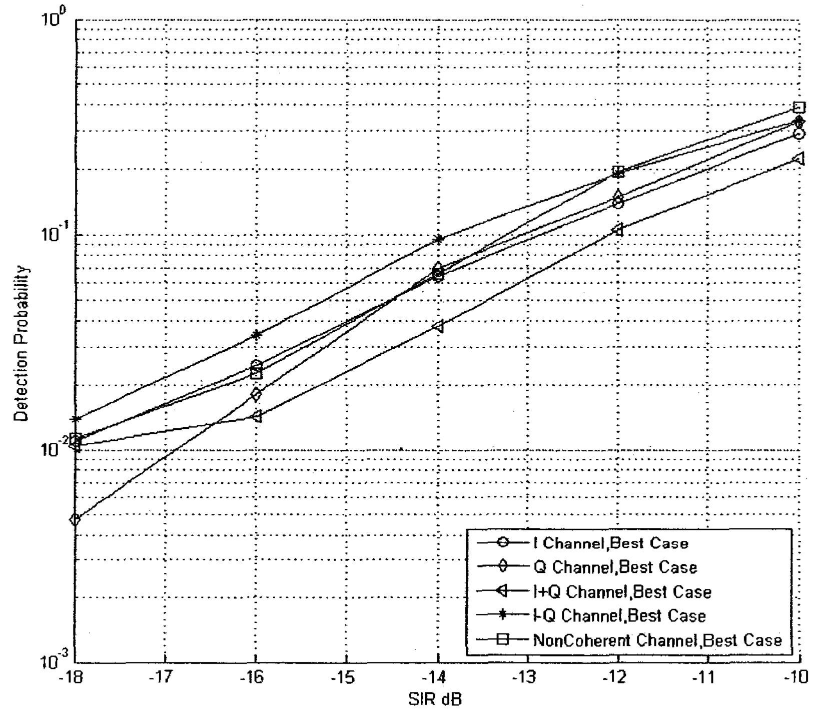

Figure 4.14: Algorithm 2, First stage detection probability @ $\Delta f = 0$ Hz.

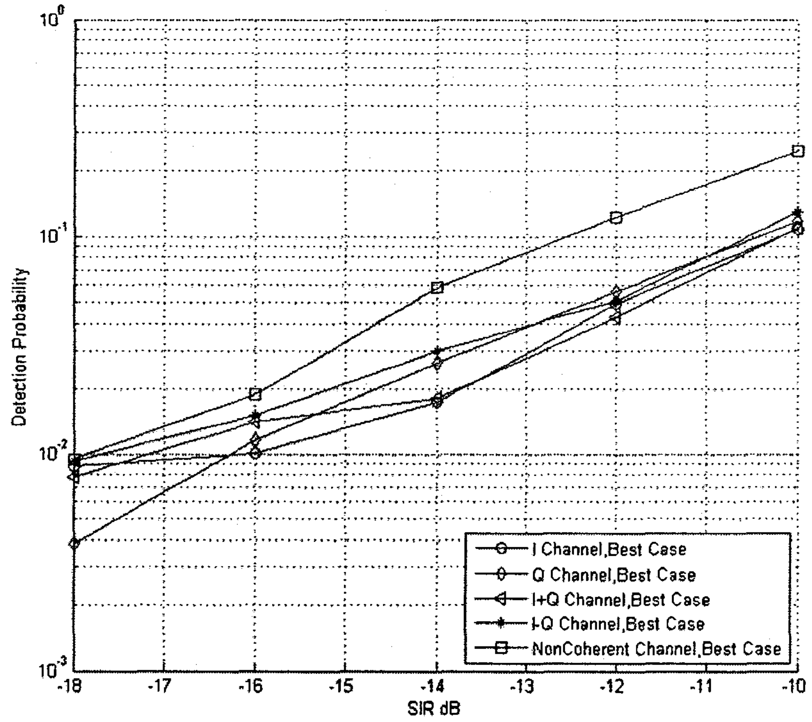

Figure 4.15: Algorithm 2, First stage detection probability @ Δf = 4 KHz.

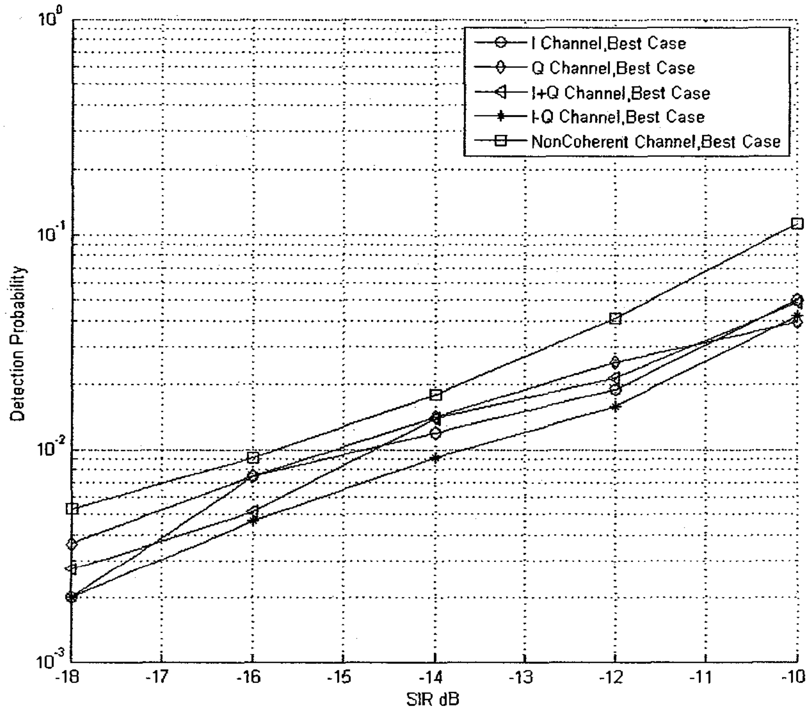

Figure 4.16: Algorithm 2, First stage detection probability @ Δf = 20 KHz.

The third peak detection algorithm is based on the criteria *CR*5(*k*) which is sign switching, threshold application and amplitude scaling. Figure 4.17 show P_d values for the third algorithm at carrier frequency error equals 20 KHz. The third algorithm P_d performance at carrier errors equal to 0 and 4 KHz is almost the same as the second algorithm at the same carrier errors with the second algorithm providing slightly better results. The second algorithm P_d values for all signal cases came in between the coherent case as lower bound and non-coherent case as upper bound of the third algorithm. A comparison of Figures 4.16 and 4.17 reveal this point.

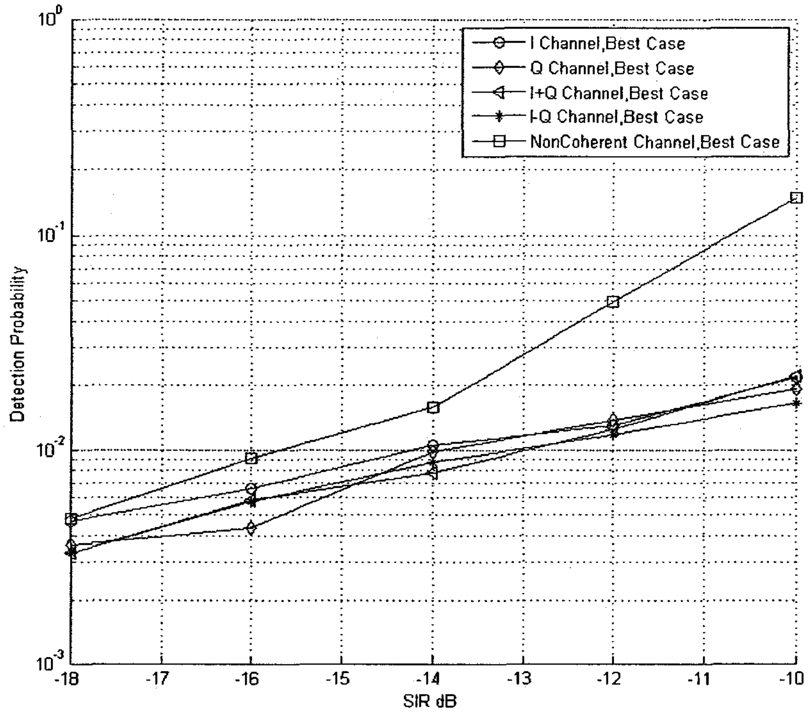

Figure 4.17: Algorithm 3, First stage detection probability @ Δf = 20 KHz.

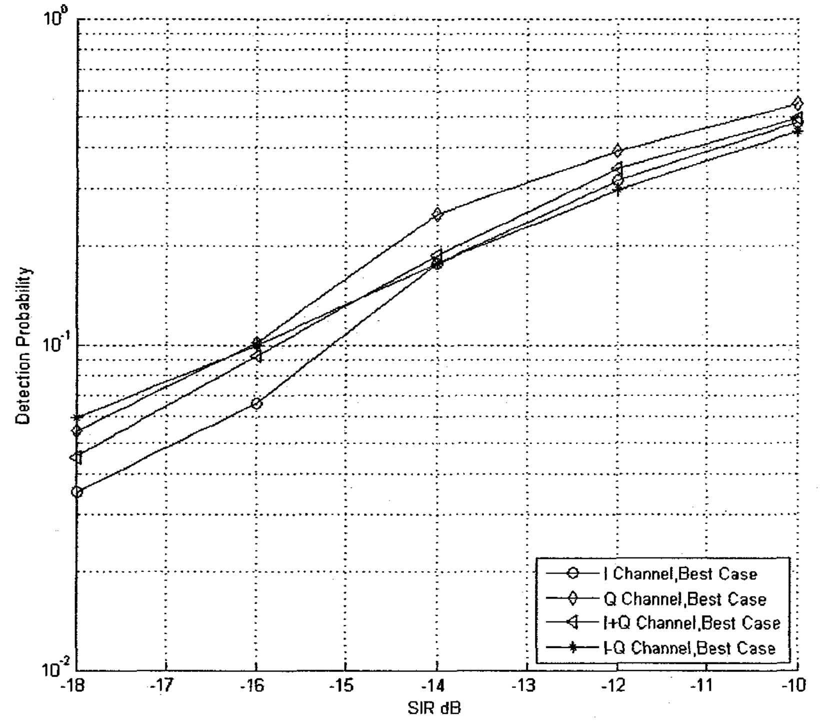

Figure 4.18: Algorithm 4, First stage detection probability @ $\Delta f = 0$ Hz.

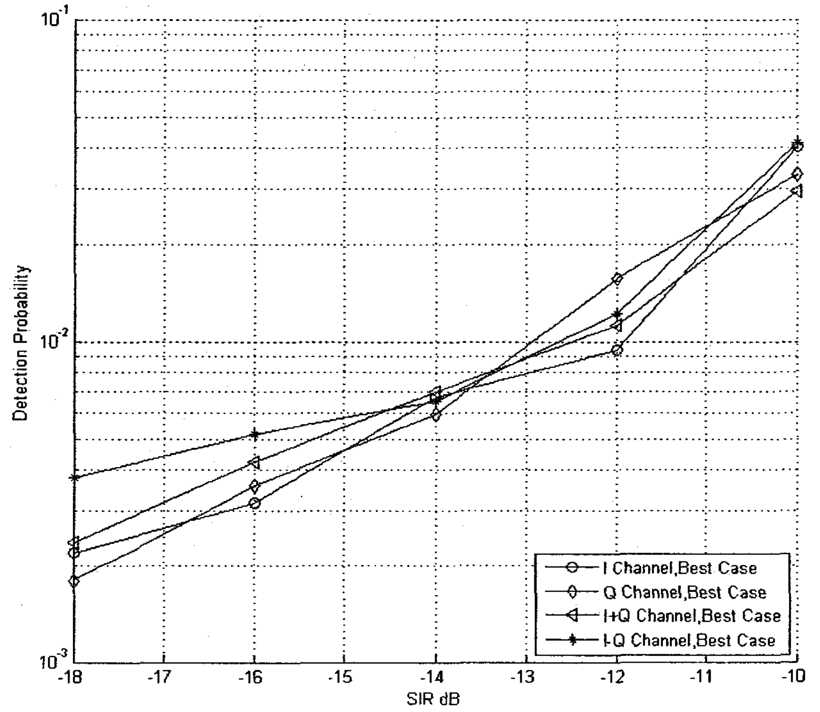

Figure 4.19: Algorithm 4, First stage detection probability @ $\Delta f = 4$ KHz.

Figures 4.18 and 4.19 show P_d values for the fourth peak detection algorithm at carrier frequency errors equal to 0 and 4 KHz respectively, where algorithm 4 is based on the criteria *CR*6(*k*), which is based on best of the positive samples. The increase of the carrier frequency error from 0 Hz to 4 KHz results in decreasing the P_d performance to 10% of their actual value for all signal cases.

4.4.2 Mean Acquisition Time Results

The following figures show the results obtained for different parameters, i.e. different peak detection algorithms, different carrier frequency errors and different pipeline architecture designs. The best correlated and worst independent acquisition times are shown in the following Figures for all four algorithms, where one notices the marked difference between the independent and dependent assumptions, i.e. upper and lower limits for mean acquisition time.

Figures 4.20 through 4.27 show the mean acquisition times for the first algorithm, i.e. best of the average of the absolute. Figures 4.20 and 4.24, slow and fast processing respectively, show that the first algorithm T_{acq} results are very close for all coherent and non-coherent signal cases at $\Delta f = 0$ (but with the coherent cases providing slightly better results). In Figure 4.20, we can see that independent signal cases provide acceptable mean acquisition time at SIR = −10 dB. However, all of the correlated signal cases at SIR = −18 dB came in between 1 sec and 600 msec, e.g. $x(t) - y(t)$ at 600 msec. Table 4.1 summarizes the correlated best case in Figures 4.20 and 4.24.

Table 4.1: First algorithm T_{acq} for the correlated best case in seconds at $\Delta f = 0$ Hz.

0 Hz Carrier error	Slow Processing			Fast Processing		
	−10 dB	−14 dB	−18 dB	−10 dB	−14 dB	−18 dB
$x(t)$	0.058	0.151	1.068	0.006	0.012	0.073
$y(t)$	0.055	0.148	1.090	0.005	0.012	0.075
$x(t)+y(t)$	0.064	0.233	0.826	0.006	0.018	0.057
$x(t)-y(t)$	0.054	0.111	0.616	0.005	0.009	0.043
$z(t)$	0.049	0.125	0.717	0.005	0.010	0.050

Increasing the carrier frequency error decreases T_{acq} for coherent cases compared to the non-coherent case for both independent and correlated assumptions. Figures 4.20 through 4.27 illustrate this point. In Figures 4.20 through 4.23, the mean acquisition times for the independent signal cases in the slow processing model become greater than 1 sec even at SIR $= -10$ dB when the carrier frequency error is greater than 0 Hz while the correlated signal cases gives acceptable results even at SIR $= -14$ dB and $\Delta f = 4$ KHz. On the other hand, in the fast processing model, the mean acquisition times for the independent signal cases is between 500 ms and 600 ms at SIR $= -11$ dB and $\Delta f = 4$ KHz. The mean acquisition times for all correlated signal cases in fast processing model come between 175 ms and 86 ms at SIR $= -18$ dB and $\Delta f = 20$ KHz. For $\Delta f = 20$ KHz, we can notice the difference between coherent and non-coherent cases where non-coherent case is approximately at 30% of coherent signal cases as shown in Table 4.2.

Table 4.2: First algorithm T_{acq} for the best case in sec at Δf = 20 KHz.

20 KHz Carrier error	Slow Processing			Fast Processing		
	−10 dB	−14 dB	−18 dB	−10 dB	−14 dB	−18 dB
$x(t)$	0.152	0.880	2.374	0.012	0.061	0.160
$y(t)$	0.193	0.817	2.374	0.015	0.057	0.160
$x(t)+y(t)$	0.163	0.656	2.489	0.013	0.046	0.168
$x(t)-y(t)$	0.200	1.133	2.594	0.015	0.078	0.175
$z(t)$	0.050	0.150	1.255	0.005	0.012	0.086

Figures 4.28 through 4.35 show the mean acquisition time for the second algorithm, i.e. threshold application and amplitude scaling. Comparison of Figures 4.28 through 4.35 reveal less mean acquisition time difference between coherent and non-coherent processed signals for different pipeline architectures and various carrier frequency errors. The non-coherent correlated case of the second algorithm provide the shortest acquisition time, T_{acq}= 56 ms for slow processing, at Δf = 0 Hz and SIR = −10 dB. However, in Figure 4.28, increasing the system load results in degradation of the non-coherent correlated case performance comparing the coherent correlated case where R = 4, i.e. $x(t)-y(t)$, gives T_{acq}= 300 ms at SIR = −16 dB and the non coherent case gives T_{acq}= 500 ms at same value of SIR. The independent signal cases' mean acquisition times for slow processing gives acceptable results at SIR = −10 dB and Δf = 0 Hz while the same performance level is achieved at SIR = −12 dB and Δf = 0 Hz in the fast processing model. Table 4.3 shows mean acquisition time values for slow and fast processing at Δf = 0 Hz.

Table 4.3: Second algorithm T_{acq} for the best case in sec at $\Delta f = 0$ Hz.

0 Hz Carrier error	Slow Processing			Fast Processing		
	−10 dB	−14 dB	−18 dB	−10 dB	−14 dB	−18 dB
$x(t)$	0.064	0.185	0.939	0.006	0.014	0.065
$y(t)$	0.060	0.173	2.158	0.006	0.014	0.146
$x(t) + y(t)$	0.075	0.295	1.001	0.007	0.023	0.069
$x(t) - y(t)$	0.059	0.136	0.749	0.006	0.011	0.052
$z(t)$	0.056	0.181	0.923	0.006	0.014	0.064

Increasing the carrier frequency error increases the difference between the coherent and the non-coherent cases, where non-coherent cases provides less mean acquisition time for independent and correlated assumptions. This result came into contrast to the case when $\Delta f = 0$ Hz. In Figures 4.29, 4.30 and 4.31, the mean acquisition time values of the independent signal cases belonging to the slow processing model are greater than 1 sec in the presence of carrier error even at SIR = −10 dB. While in Figure 4.34, the fast processing model the mean acquisition time values are between 200 ms and 500 ms for coherent cases, and 50 ms for non-coherent at SIR = −10 dB and $\Delta f = 4$ KHz. Figures 4.31 and 4.35 show slow and fast processing respectively for the second algorithm at $\Delta f = 20$ KHz and some of the mean acquisition time values are addressed in Table 4.4. In the presence of carrier error, we can notice that non-coherent correlated and independent signal cases are better than coherent signal cases for fast and slow processing models. Figures 4.29, 4.30, 4.31, 4.33, 4.34 and 4.35 illustrate this point.

Table 4.4: Second algorithm T_{acq} for the best case in sec at Δf = 20 KHz.

20 KHz Carrier error	Slow Processing			Fast Processing		
	−10 dB	−14 dB	−18 dB	−10 dB	−14 dB	−18 dB
$x(t)$	0.229	0.870	5.030	0.017	0.060	0.337
$y(t)$	0.283	0.736	2.782	0.021	0.051	0.188
$x(t) + y(t)$	0.236	0.746	3.689	0.018	0.052	0.248
$x(t) - y(t)$	0.266	1.133	5.030	0.020	0.078	0.337
$z(t)$	0.118	0.587	1.929	0.010	0.041	0.131

Figures 4.36 through 4.43 show the mean acquisition time for the third algorithm, i.e. sign switching and amplitude scaling. There is no difference between coherent and non-coherent signal cases of slow processing at Δf= 0 Hz and 100 Hz, as shown in Figures 4.36 and 4.37. The same observation can be concluded from Figures 4.40 and 4.41 for the fast processing. In Figure 4.34, independent coherent signals are between 280 ms and 390 ms while non-coherent signal is at 160 ms for carrier error equals 100 Hz, SIR = −10 dB and slow processing. The correlated signal cases in slow and fast processing models provide less mean acquisition time values for higher system loads and presence of carrier errors, e.g. mean acquisition time for coherent signal case R = 4 equals 300 ms at SIR = −15 dB, Δf = 100 Hz and slow processing as shown in Figure 4.37. However, increasing Δf to 4 KHz will result in increasing the difference between coherent and non-coherent signal cases. The non-coherent correlate and independent cases provide less T_{acq}, as shown in Figure 4.38 and 4.42 slow and fast processing models

respectively. In Figures 4.39 and 4.43 slow and fast processing respectively, $\Delta f = 20$ KHz results in less T_{acq} for non-coherent independent and correlated cases compared to coherent cases between SIR = −10 dB and −14 dB, but for SIR values greater than −14 dB the difference in T_{acq} is less between coherent and non-coherent signal cases.

Figures 4.44 through 4.51 show the mean acquisition time for the fourth algorithm, i.e. best of the positive samples. Algorithm 4 has four coherent cases where the non-coherent case, i.e. z(t), does not apply since all samples are positive. We can notice that the fourth algorithm in the slow processing case with the presence of carrier frequency errors results in relatively high mean acquisition time values for independent and correlated assumptions. Figures 4.45, 4.46 and 4.47 present this point. In the fast processing case, we obtain better results. The best correlated cases in Figures 4.48, 4.49, 4.50 and 4.51 can be summarized in Table 4.5 where the mean acquisition time values of all four coherent cases are included. Increasing the carrier error from 0 Hz to 100 Hz increases the mean acquisition time. However, increasing the carrier error to 4 KHz increases the mean acquisition time at SIR = −14 dB by 20 msec. Further increasing the carrier error to 20 KHz will increase the mean acquisition time as shown in Table 4.5.

Table 4.5: Fourth algorithm T_{acq} for the best case and fast processing in sec.

	−10 dB	−14 dB	−18dB
0 Hz	0.005	0.008	0.050 – 0.150
100 Hz	0.020	0.080	0.100 – 0.400
4 KHz	0.020	0.100	0.100 – 0.400
20 KHz	0.040	0.150	0.250 – 0.550

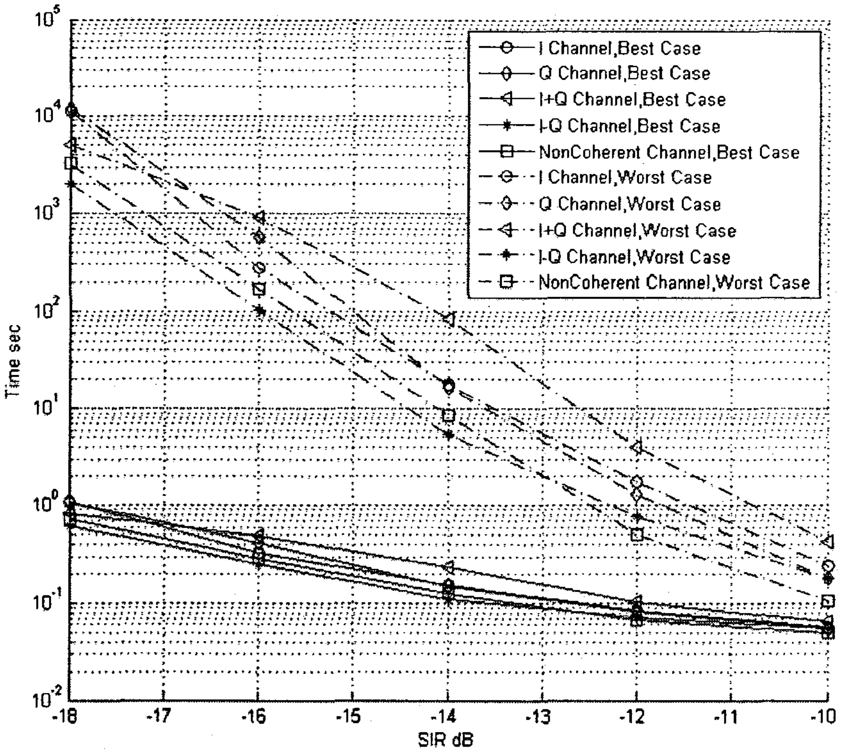

Figure 4.20: Algorithm 1, 0 KHz Frequency Offset, Slow Processing

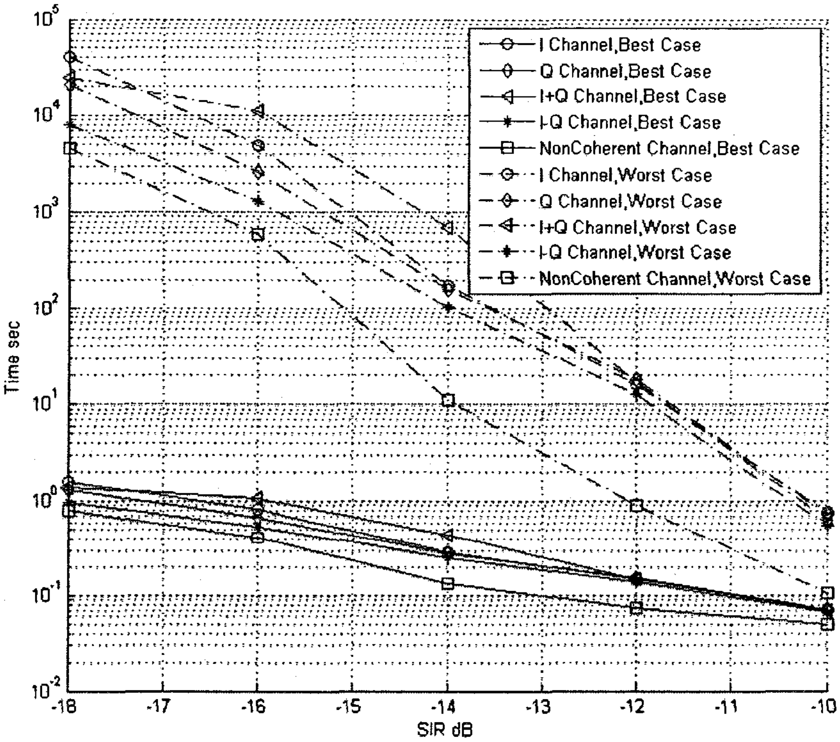

Figure 4.21: Algorithm 1, 100 Hz Frequency Offset, Slow Processing

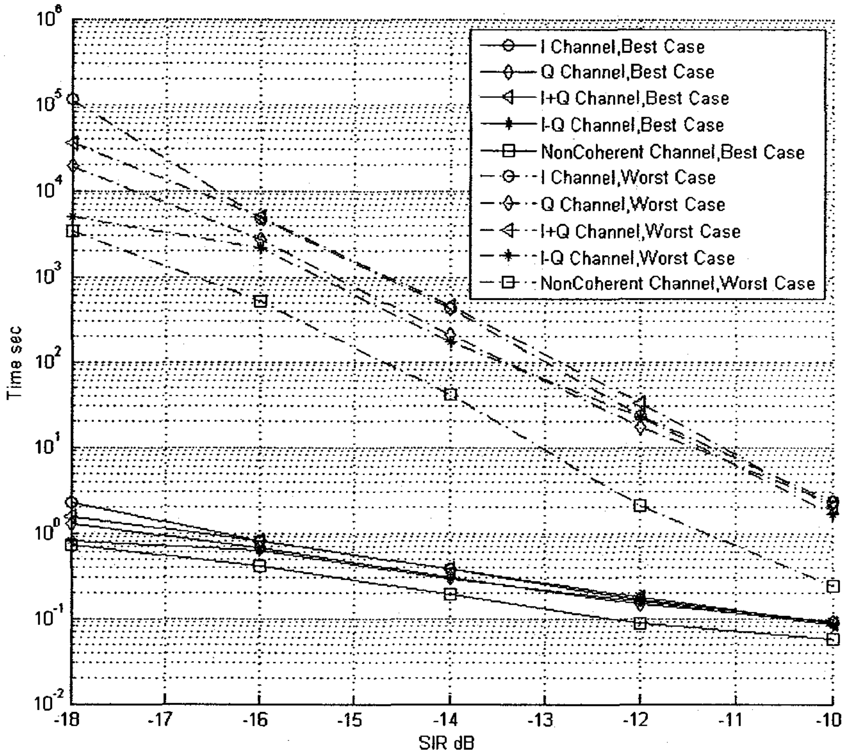

Figure 4.22: Algorithm 1, 4 KHz Frequency Offset, Slow Processing

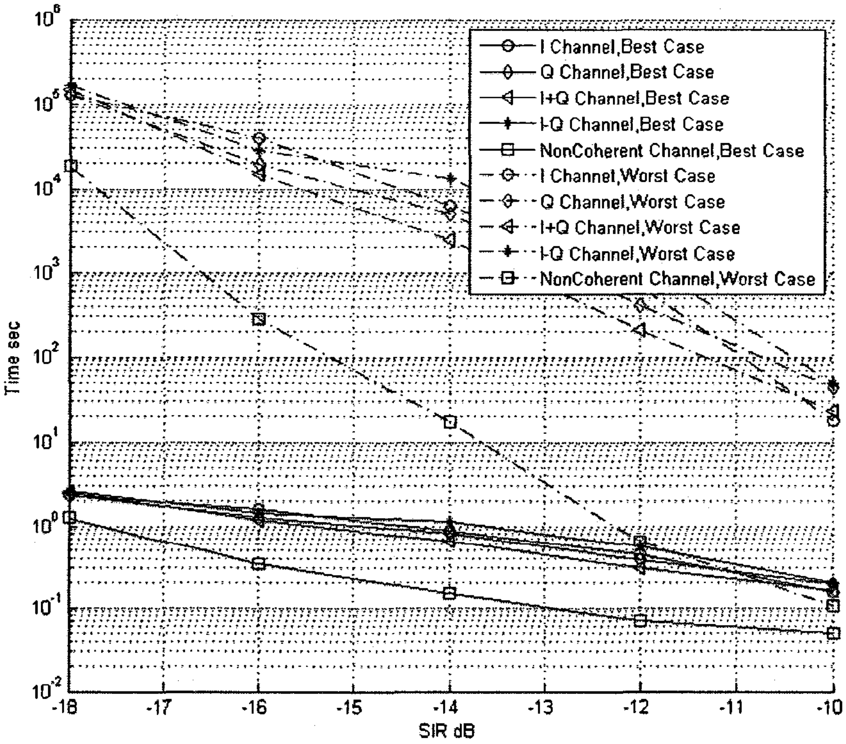

Figure 4.23: Algorithm 1, 20 KHz Frequency Offset, Slow Processing

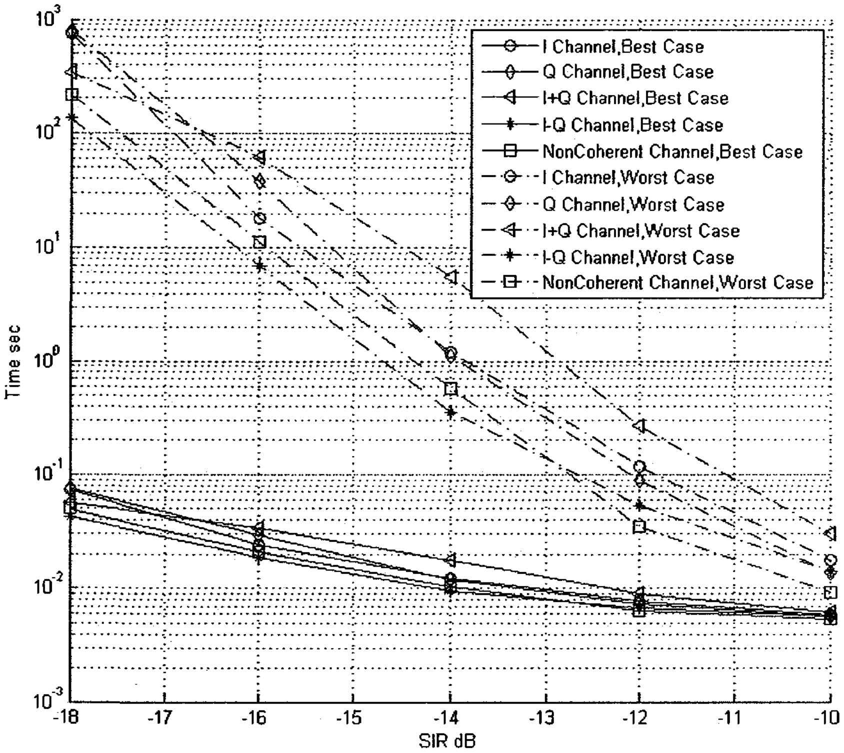

Figure 4.24: Algorithm 1, 0 KHz Frequency Offset, Fast Processing

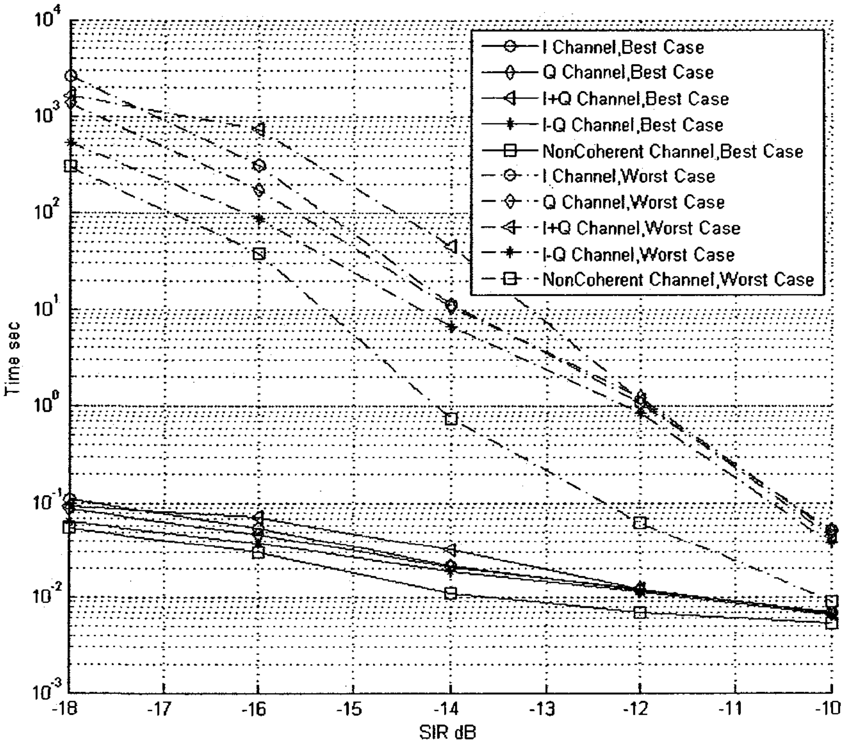

Figure 4.25: Algorithm 1, 100 Hz Frequency Offset, Fast Processing

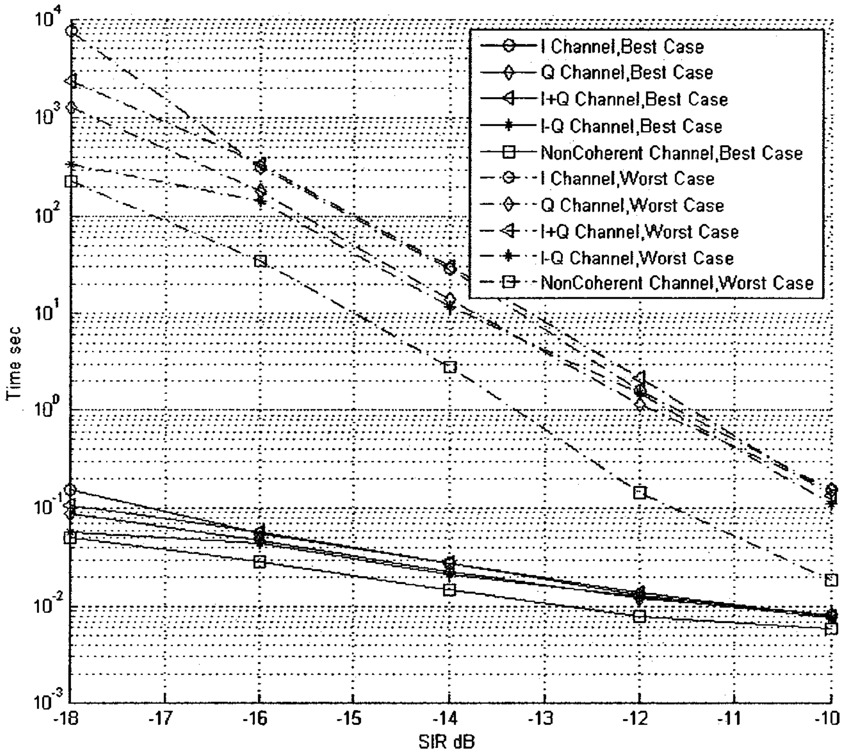

Figure 4.26: Algorithm 1, 4 KHz Frequency Offset, Fast Processing

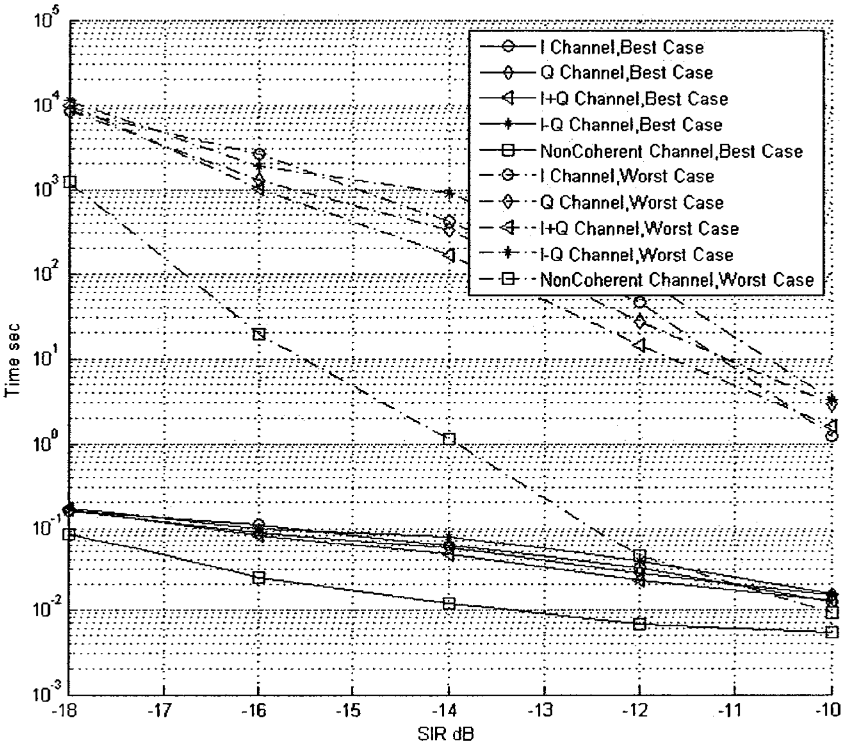

Figure 4.27: Algorithm 1, 20 KHz Frequency Offset, Fast Processing

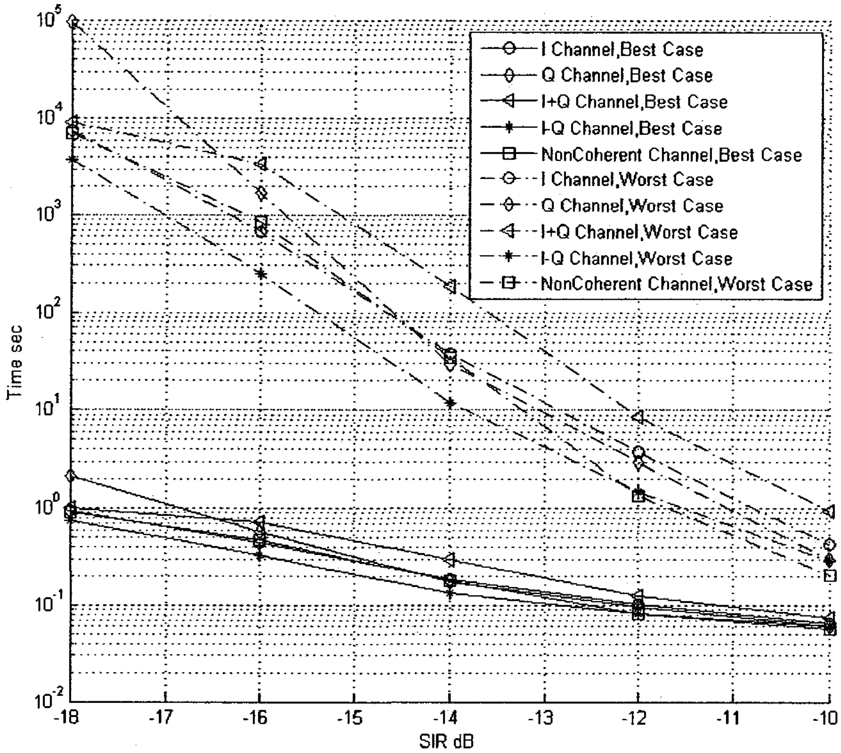

Figure 4.28: Algorithm 2, 0 KHz Frequency Offset, Slow Processing

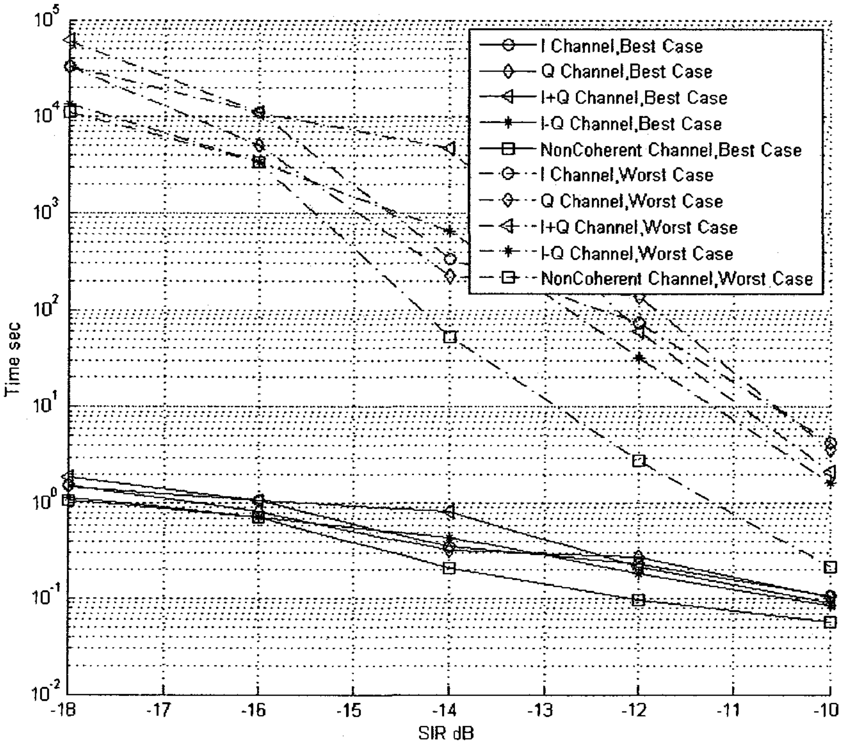

Figure 4.29: Algorithm 2, 100 Hz Frequency Offset, Slow Processing

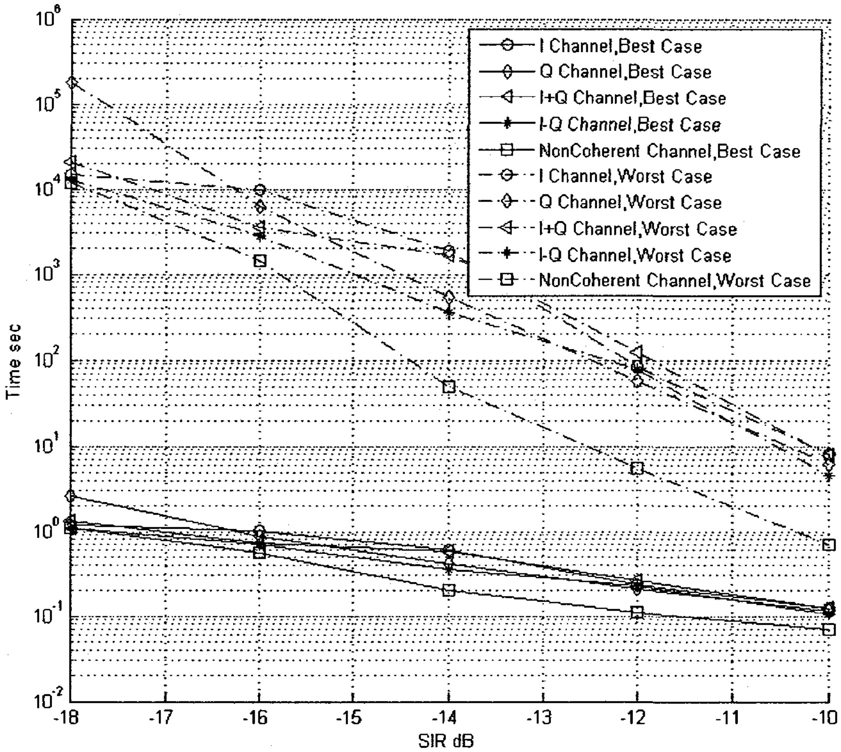

Figure 4.30: Algorithm 2, 4 KHz Frequency Offset, Slow Processing

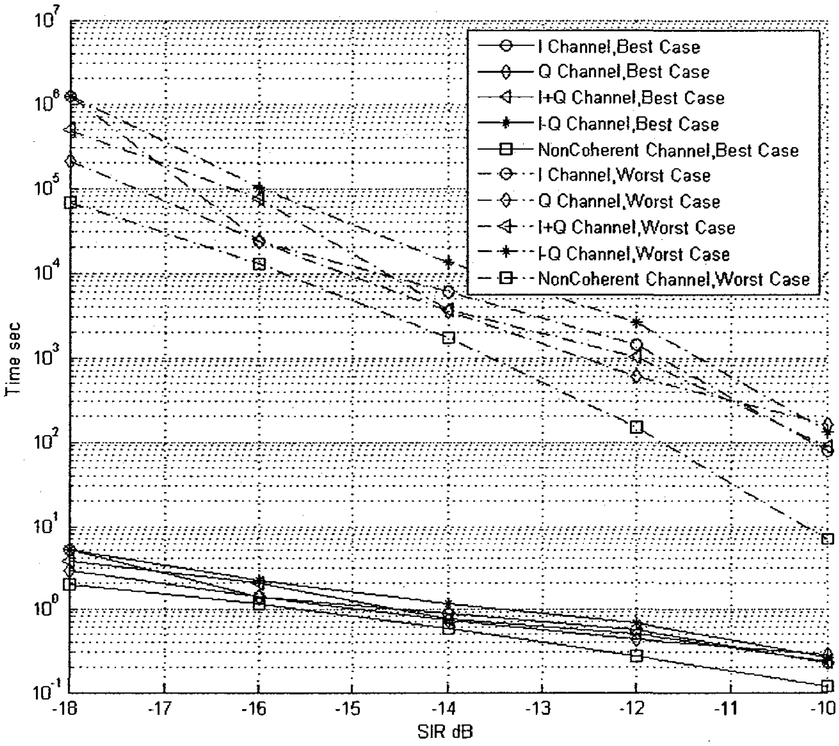

Figure 4.31: Algorithm 2, 20 KHz Frequency Offset, Slow Processing

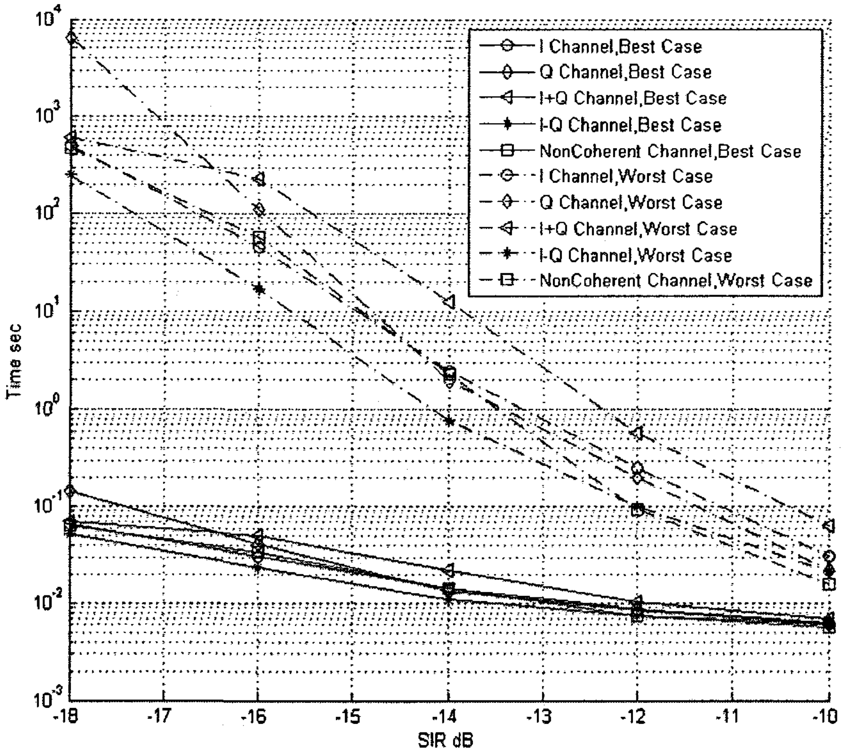

Figure 4.32: Algorithm 2, 0 KHz Frequency Offset, Fast Processing

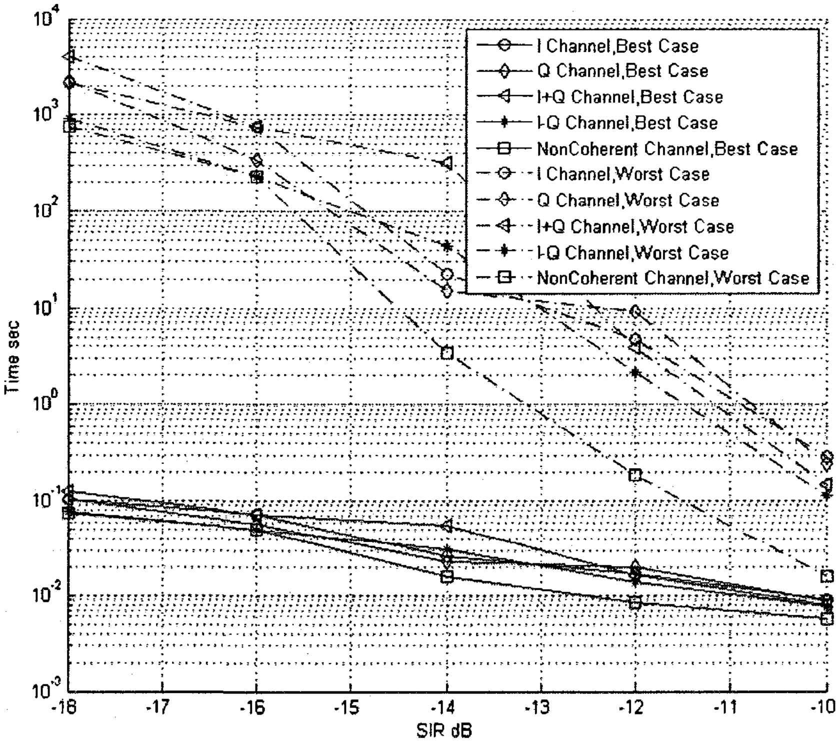

Figure 4.33: Algorithm 2, 100 Hz Frequency Offset, Fast Processing

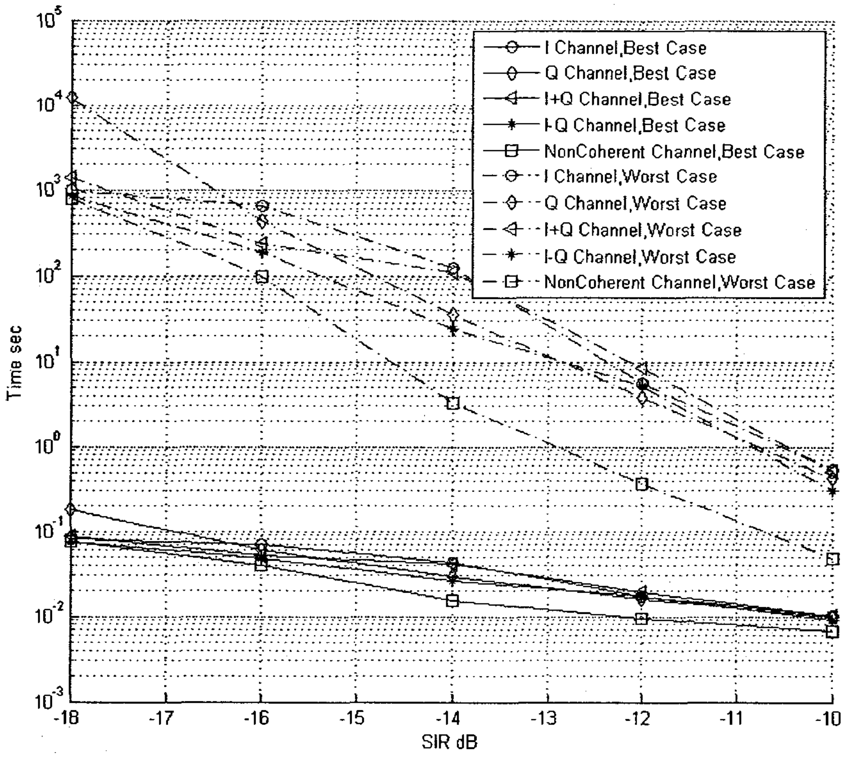

Figure 4.34: Algorithm 2, 4 KHz Frequency Offset, Fast Processing

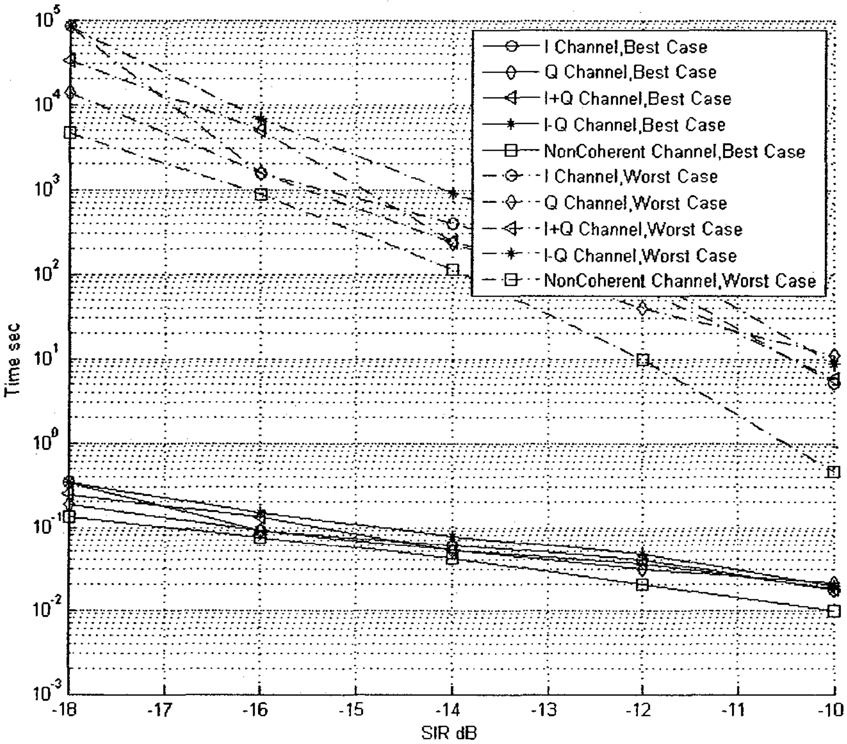

Figure 4.35: Algorithm 2, 20 KHz Frequency Offset, Fast Processing

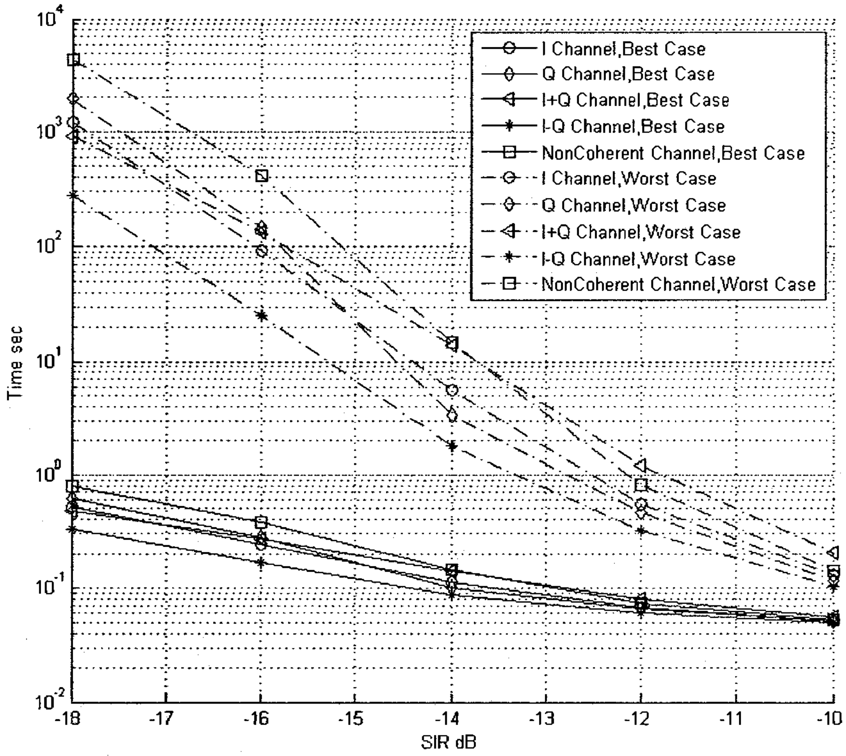

Figure 4.36: Algorithm 3, 0 KHz Frequency Offset, Slow Processing

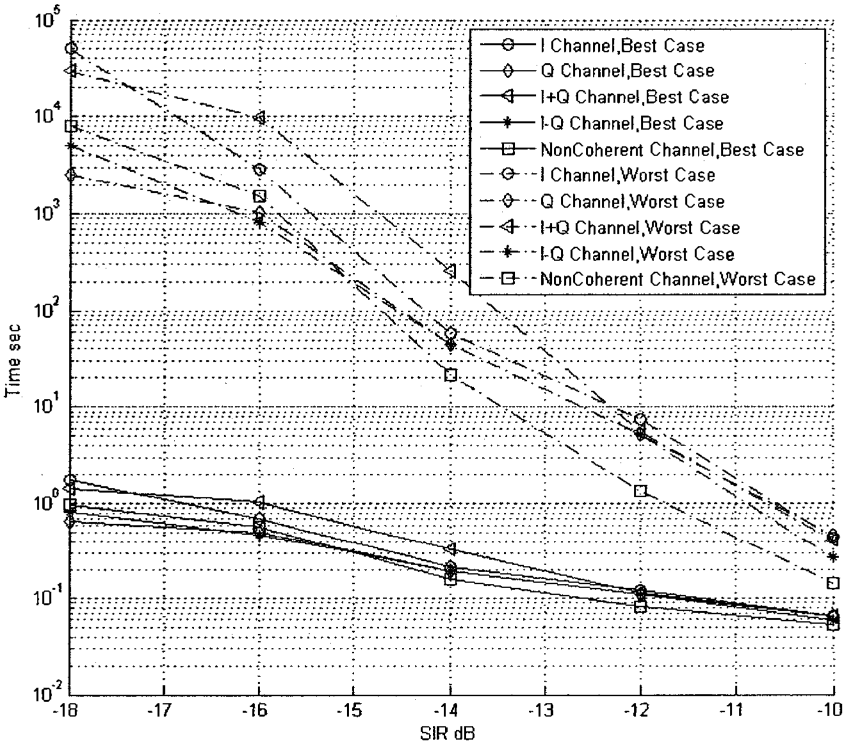

Figure 4.37: Algorithm 3, 100 Hz Frequency Offset, Slow Processing

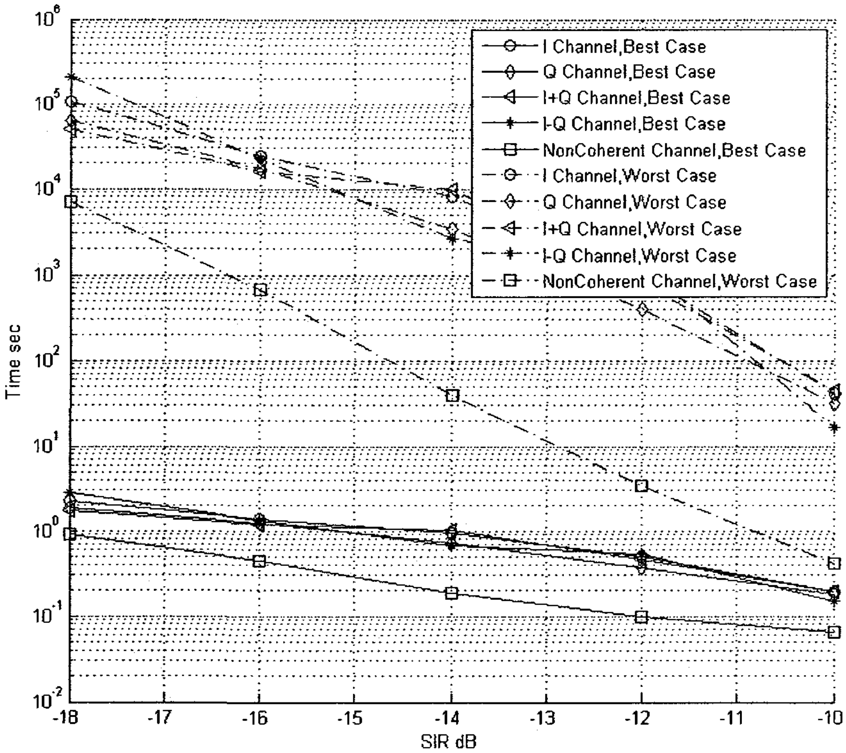

Figure 4.38: Algorithm 3, 4 KHz Frequency Offset, Slow Processing

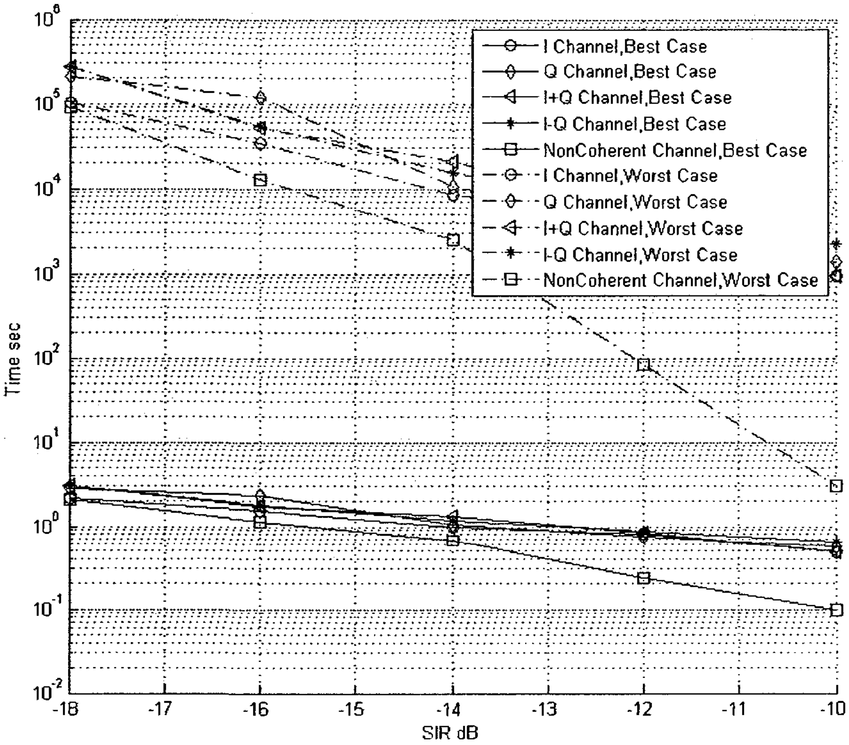

Figure 4.39: Algorithm 3, 20 KHz Frequency Offset, Slow Processing

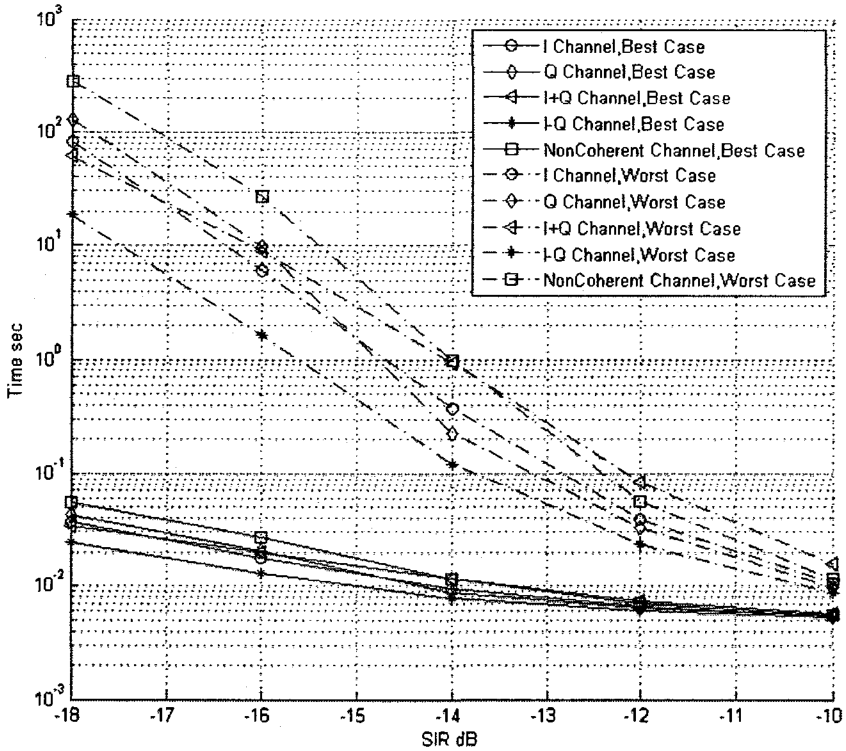

Figure 4.40: Algorithm 3, 0 KHz Frequency Offset, Fast Processing

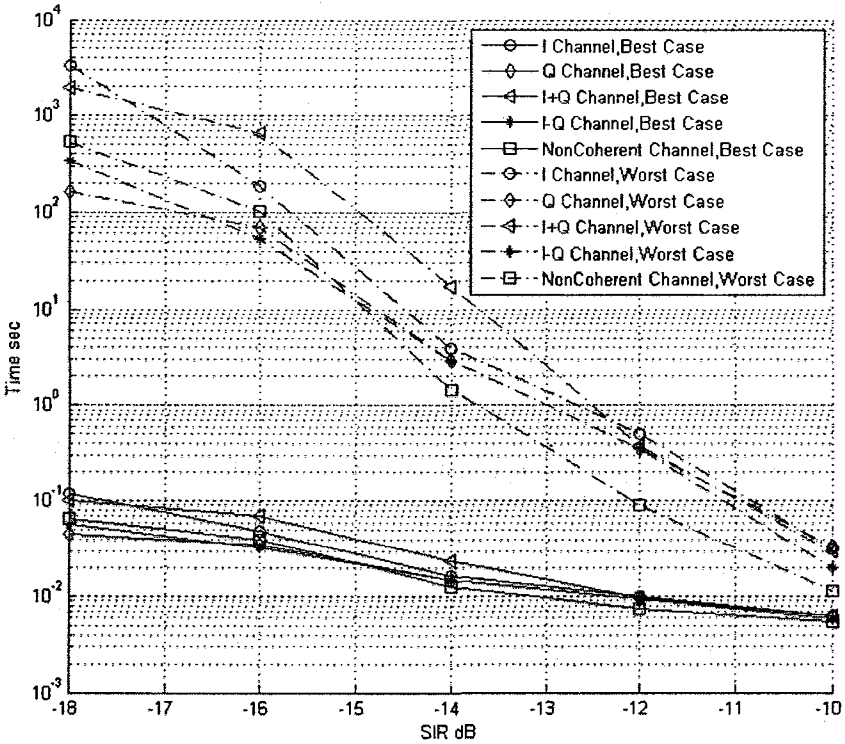

Figure 4.41: Algorithm 3, 100 Hz Frequency Offset, Fast Processing

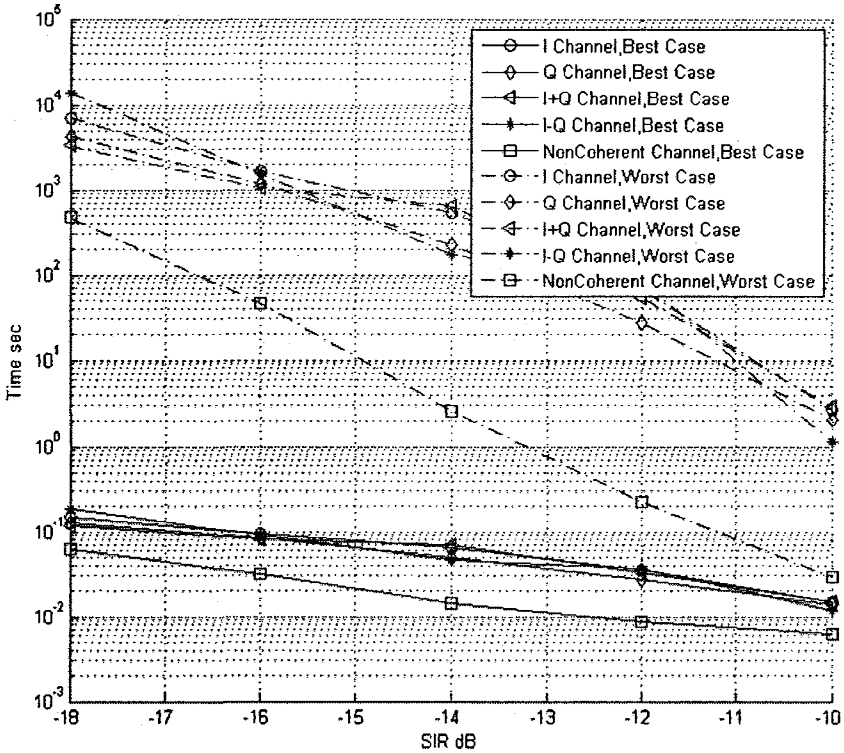

Figure 4.42: Algorithm 3, 4 KHz Frequency Offset, Fast Processing

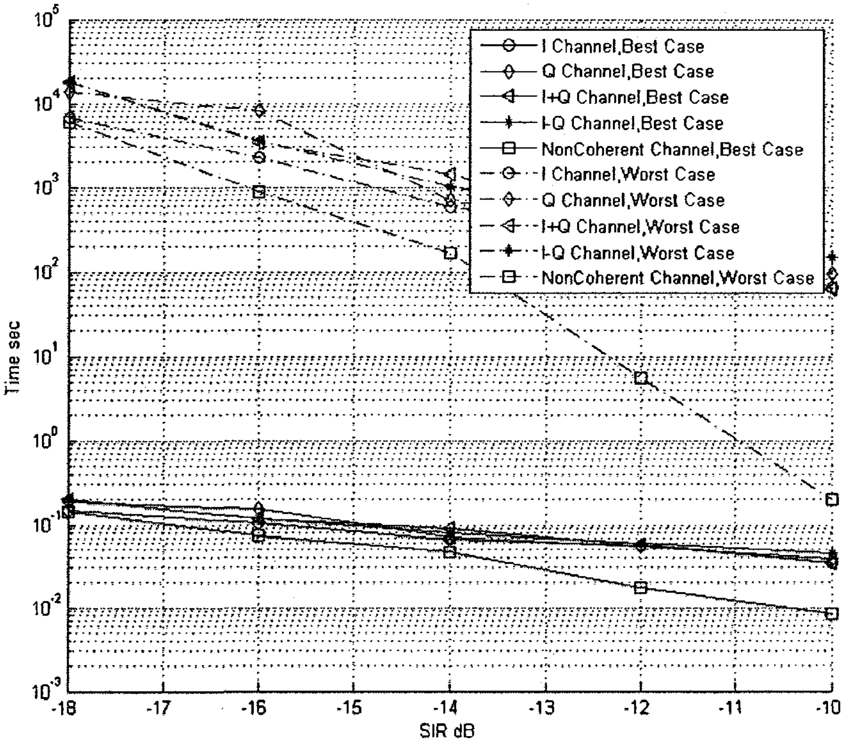

Figure 4.43: Algorithm 3, 20 KHz Frequency Offset, Fast Processing

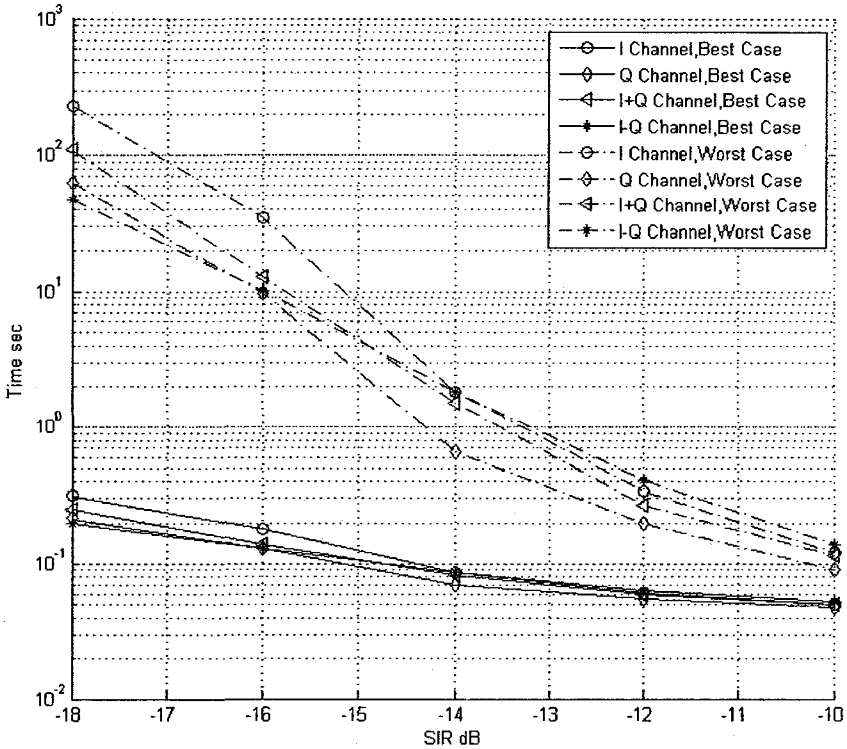

Figure 4.44: Algorithm 4, 0 KHz Frequency Offset, Slow Processing

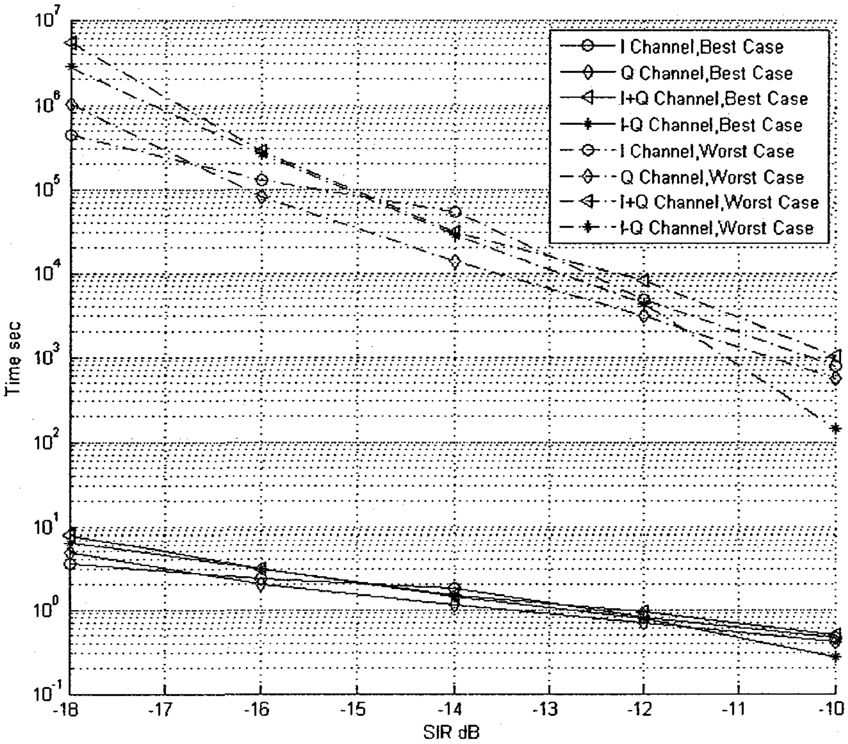

Figure 4.45: Algorithm 4, 100 Hz Frequency Offset, Slow Processing

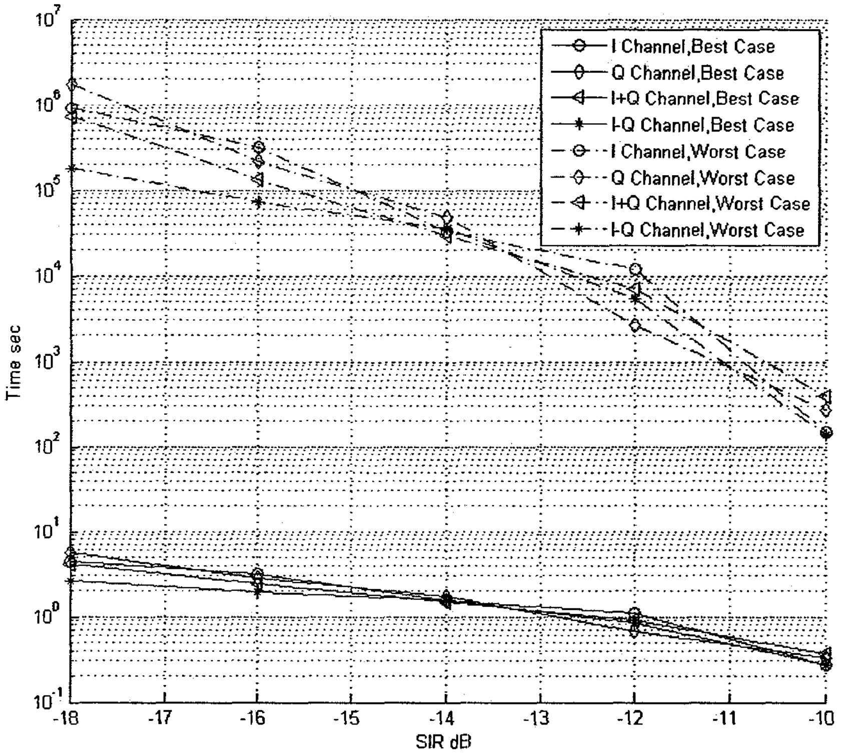

Figure 4.46: Algorithm 4, 4 KHz Frequency Offset, Slow Processing

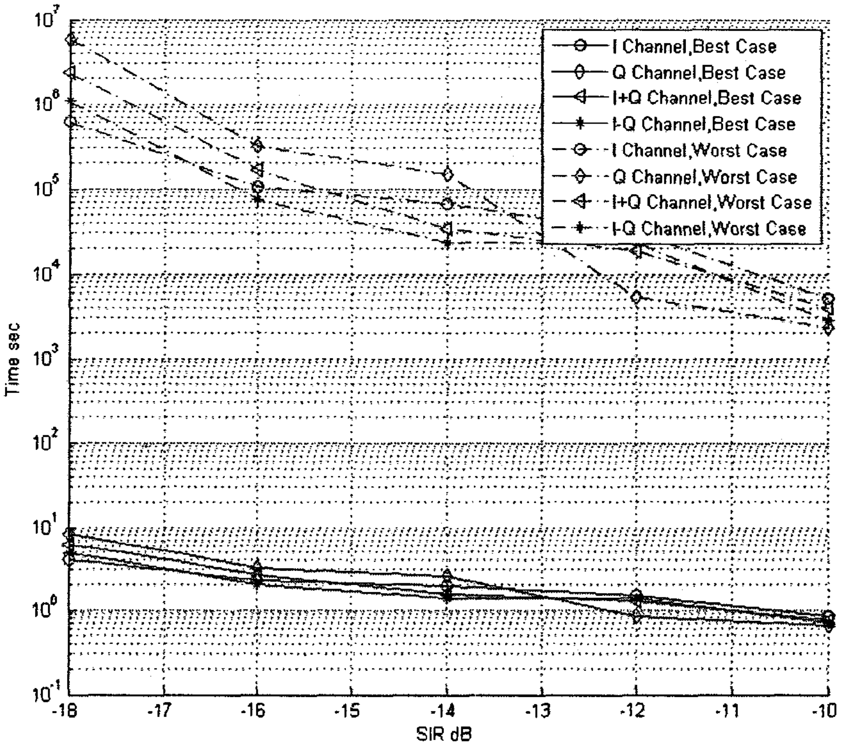

Figure 4.47: Algorithm 4, 20 KHz Frequency Offset, Slow Processing

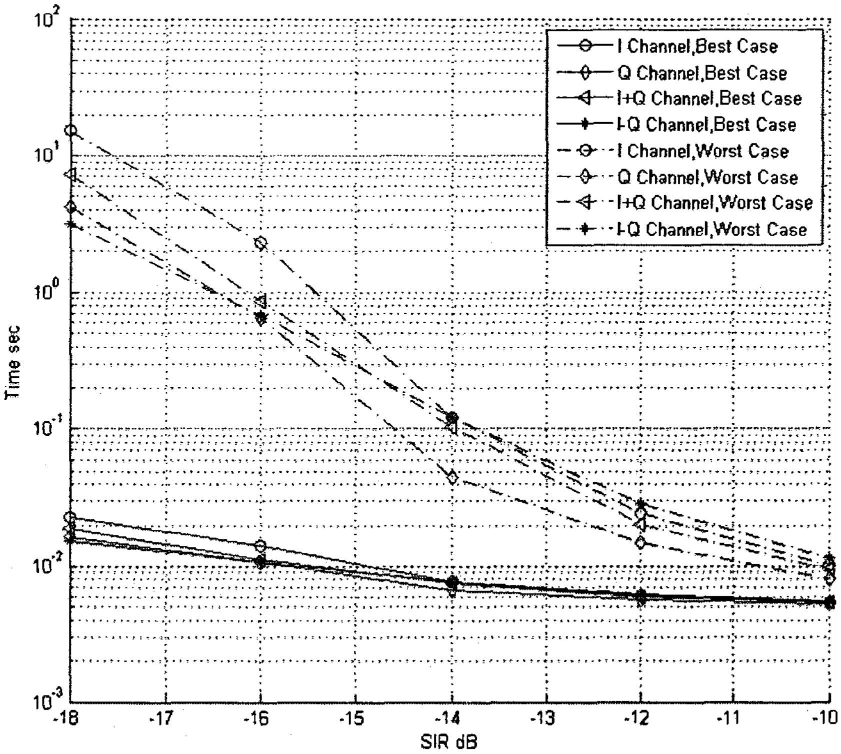

Figure 4.48: Algorithm 4, 0 KHz Frequency Offset, Fast Processing

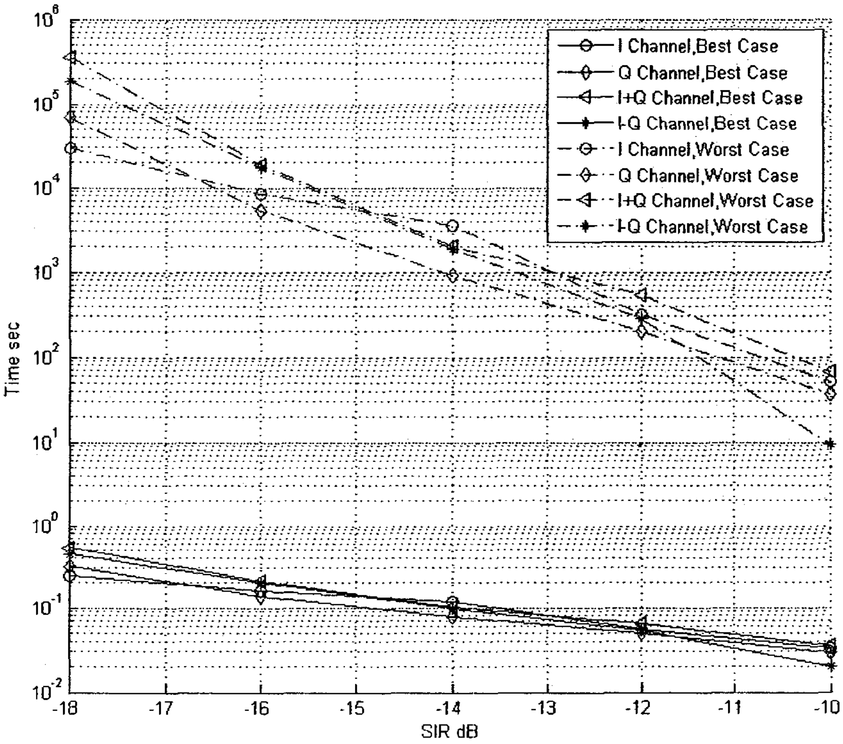

Figure 4.49: Algorithm 4, 100 Hz Frequency Offset, Fast Processing

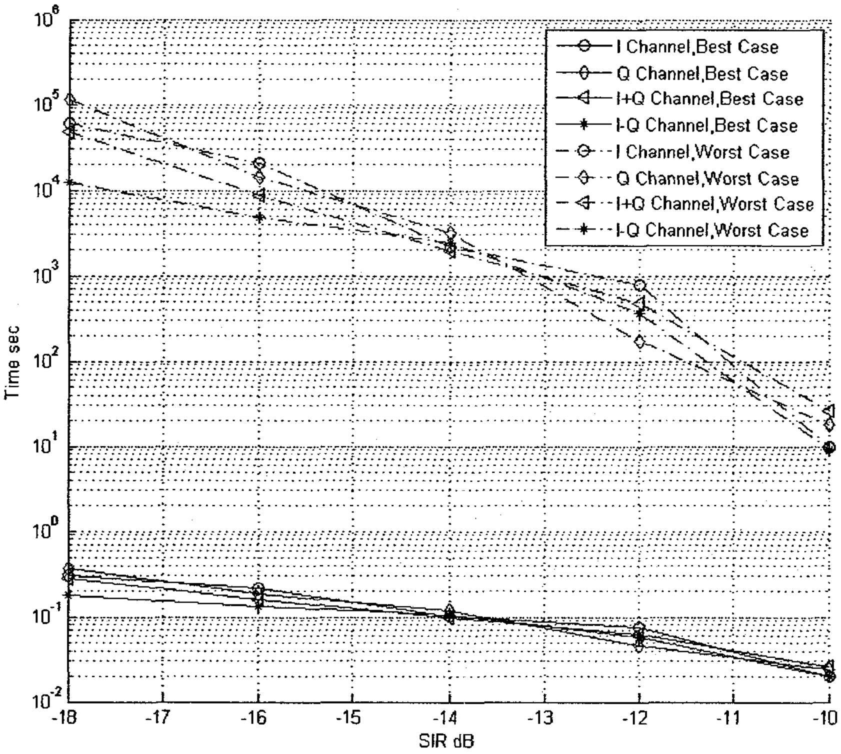

Figure 4.50: Algorithm 4, 4 KHz Frequency Offset, Fast Processing

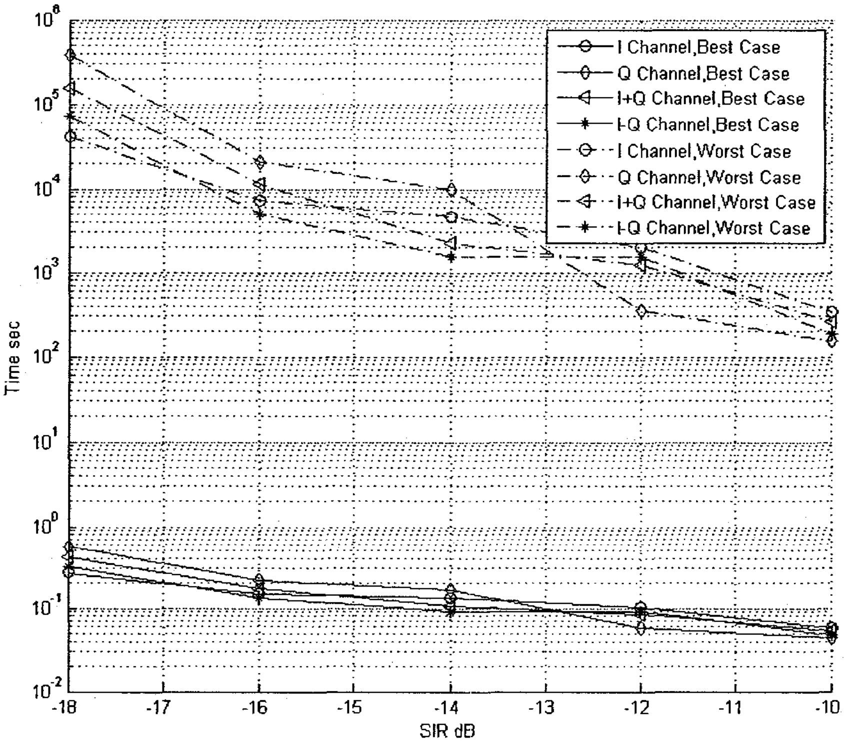

Figure 4.51: Algorithm 4, 20 KHz Frequency Offset, Fast Processing

Chapter 5

Conclusion and Future work

5.1 Conclusion

In this thesis, our intention was to focus on the first acquisition stage of the cell search of UTRA-FDD because it has to deal with the largest amount of uncertainty. The MS has no prior information of the BS(s) in its vicinity and we assume that the cell search starts when the mobile station is turned on. Moreover, the first acquisition stage is carried out at heavy channel loads due to other BSs' and users' signals. The MS uses the PSC to acquire slot boundaries in order to obtain timing information of the detected BS. Less mean acquisition time at very low SIR values implies that more number of users can be accommodated by the same number of BSs. Fortunately, the number of BSs that needs to be deployed will be less. Moreover, less mean acquisition time means that MS resolves timing and code uncertainty with less power consumption.

We have presented different coherent and non-coherent peak detection algorithms for the first stage of UTRA-FDD code acquisition where the currently used detection method is non-coherent. Along with the standard oversampling, we have assumed the presence of more than one base station with few multipath signals. Also, we have utilized more than one verification circuit in pipelining architecture. Both arrangements have helped to reduce the mean acquisition time.

The non-coherent signal case is processed non-linearly where squaring the X and Y signals in the I and Q branches respectively will produce extra undesired terms. However, the four coherent signal cases are linearly processed where we only use counters, which means it is more manageable especially at very low SIR values, i.e. more number of users.

We have used two processing models, i.e. slow processing and fast processing models. The fast processing model uses short processing cycles, i.e. one slot time. Therefore, we utilize four verification circuits because each verification circuit consume four processing cycles. On the other hand, the slow processing model uses longer processing cycles, i.e. one frame time. Therefore, we could use one verification circuit because the verification circuit produces an output before the processing cycle finishes. The fast processing model gave us very good mean acquisition results for the correlated acquisition stages, i.e. best case, even at low SIR values. However, the slow processing model gave us good mean acquisition results for the correlated best case at higher SIR values. Also, we introduced the concept of sliding frames to provide better average over the received samples.

Coherent combining results were similar to their non-coherent counterpart for low and moderate frequency offsets. The same was observed for very low SIR values. Since we have concentrated on the first acquisition stage of the UTRA-FDD; detection probabilities of all three stages were assumed to be the same, leading to a worst independent case assumption.

We have used MATLAB to run simulations in order to evaluate the performance of our four peak detection algorithms. We have varied several input parameters such as carrier frequency errors, processing model and signal to interference ratio (Channel load). We have observed the detection probability and mean acquisition time for all four peak detection algorithms.

The first three algorithms have coherent and non-coherent signal cases and the last algorithm has only coherent signal cases. For $\Delta f = 0$ and SIR $= -10$ dB in the first three algorithms, all four coherent signals' mean acquisition time are slightly better than their non-coherent counterpart. Moreover, at lower values of SIR coherent and non-coherent signal cases yield similar mean acquisition time results. However, for large carrier error values, non-coherent provides better results. Fast processing advantages are clear in all algorithms where all T_{acq} results are approximately 10% of their slow processing counter values. While the general trend of results yield better mean acquisition time values as we go from algorithm 4 to 3 to 2 to 1.

In Figure 4.27, the fast processing model for algorithm one at 20 KHz frequency offset reveals big difference between coherent signal cases and non-coherent one. However, in Figures 4.35 and 4.43 for the second and third algorithms respectively at the same frequency offset and fast processing model revile less difference. In the first algorithm, the detection is based on one criteria, i.e. $DR1(k)$ where the second algorithm is based on combining two criteria, i.e. $CR1(k)$ and $DR1(k)$, and the third one is based on combining three criteria, i.e. $DR1(k)$, $CR1(k)$ and $CR2(k)$. Therefore, probability of detection that is based on combining different criteria results in less difference for coherent and non-coherent signal cases.

We should avoid working with carrier errors higher than Δf = 20 KHz, since mean acquisition time can become very long at low SIR values, especially for a slow processing verification circuit.

5.2 Future Work

We advise the following points for future research work:

1. In our simulation model, we have concentrated on the first acquisition stage of the UTRA-FDD systems. We assumed that the performance of the three stages is equal. However, a detailed study of simulation and analysis for the second and third acquisition stages can be completed in future work.

2. In regard to the number of the verification circuit which lead to fast or slow processing model, it will be interesting to try different arrangements and study their effect on the mean acquisition time.

3. The three acquisition stages of the cell search are considered to have equal acquisition times. We recommend further research that studies the effect of unequal processing time for the three acquisition stages.

4. It is motivating to apply our peak detection algorithms along with antennas diversity because such technique will improve the quality and reliability of the channel. Therefore, an improvement in the mean acquisition time could be attained.

5. We have used Doppler shift of 100 Hz in our simulation model, which is equivalent to change of the channel every 100 msec. Testing different values of Doppler shift is essential in order to evaluate various environments.

References

[1] Hsiao-Hwa Chen, Mohsen Guizani, “Next generation wireless systems and networks”, John Wiley & Sons, 2006.

[2] Oliphant, M.W., “The mobile phone meets the Internet”, Spectrum, IEEE, Volume 36, Issue 8, August 1999 Page(s):20 – 28.

[3] Lawrence Harte, Morris Hoenig, Daniel McLaughlin and Roman Kikta, “CDMA IS-95 for Cellular and PCS: Technology, Applications, and Resource Guide”, McGraw-Hill Professional, 1999.

[4] Timo Halonen, Javier Romero and Juan Melero, “GSM, GPRS and EDGE performance: evolution towards 3G/UMTS”, Wiley, 2002.

[5] M. R. Karim and Mohsen Sarraf, “W-CDMA and Cdma2000 for 3G Mobile Networks”, McGraw-Hill Professional, 2002.

[6] John G. Proakis, “Digital Communications 4th edition”, McGraw-Hill, 2000.

[7] Hsiao-Hwa Chen, “The Next Generation CDMA Technologies”, John Wiley & Sons, 2007.

[8] Chren, W.A., Jr., “High-speed frequency synthesizer for spread spectrum communication systems ”,Tactical Communications Conference, 1994, Vol. 1. Digital Technology for the Tactical Communicator., Proceedings of the 1994.

[9] Haartsen, J.C., "Bluetooth ^{TM}: a new radio interface providing ubiquitous connectivity", Vehicular Technology Conference Proceedings, 2000 Tokyo. 2000 IEEE 51st, Volume 1, 15-18 May 2000 Page(s):107 – 111.

[10] Toskala, A.; Holma, H.; Muszynski, P., "ETSI WCDMA FOR UMTS", Spread Spectrum Techniques and Applications, 1998. Proceedings., 1998 IEEE 5th International Symposium on Volume 2, 2-4 Sept. 1998 Page(s):616 - 620 vol.2 Digital Object Identifier 10.1109/ISSSTA.1998.723859.

[11] Wheatley, C.; "CDMA's dependence on good timing and related issues", Frequency Control Symposium, 1996. 50th, Proceedings of the 1996 IEEE International. 5-7 June 1996 Page(s):1211 - 1218 Digital Object Identifier 10.1109/FREQ.1996.560315.

[12] "Physical layer - General description", 3GPP TS 25.201, V8.0.0 (2008-03).

[13] Harri Holma and Antti Toskala, "WCDMA for UMTS : radio access for third generation mobile communications", John Wiley & Sons, 2004.

[14] Zamzuri, H.B.; Ibrahim, M.B., "SIMULATION ON WCDMA FOR 3G MOBILE SYSTEMS", Telecommunication Technology, 2003. NCTT 2003 Proceedings. 4th National Conference on 14-15 Jan. 2003 Page(s):244 - 249

[15] "High Speed Downlink Packet Access", 3GPP TR 25.858, V5.0.0, (2002-03).

[16] "User Equipment (UE) radio transmission and reception (FDD)", 3GPP TS 25.101 V8.2.0 (2008-03).

[17] "Radio resource management strategies", 3GPP TR 25.922 V7.1.0 (2007-03).

[18] "Spreading and modulation (FDD)", 3GPP TS 25.213 V8.0.0 (2008-03).

[19] “Physical channels and mapping of transport channels onto physical channels (FDD)”, 3GPP TS 25.211 V8.0.0 (2008-03).

[20] Shin Shin-Lin Shieh; Shin-Yuan Wang; Wang, B.H., ”Soft-decision Reed-Solomon decoding strategies for code group identification and frame synchronization in UMTS WCDMA”, Vehicular Technology Conference, 2003. VTC 2003 - Fall. 2003 IEEE 58th Volume 4, 6-9 Oct. 2003 Page(s):2348 - 2351 Vol.4.

[21] Shu Lin, Daniel J. Costello Jr. “Error Control Coding” (2nd Edition), Prentice Hall, 2004.

[22] Raymond Steele, Chin-Chun Lee and Peter Gould, “GSM, cdmaOne, and 3G systems”, John Wiley, 2001.

[23] Rodger E. Ziemer, Roger L. Peterson, David E. Borth, “Introduction to Spread Spectrum Communications”, Prentice Hall Inc. March 1995.

[24] DiCarlo, D.; Weber, C., “Statistical Performance of Single Dwell Serial Synchronization Systems” IEEE Transactions Communications, Volume: 28, Issue: 8, Part 2 On page(s): 1382- 1388 - Aug 1980.

[25] Don Torrieri, “Principles of Spread-Spectrum Communication Systems”, Springer, November 2004.

[26] Ward, R., “Acquisition of Pseudonoise Signals by Sequential Estimation”, Communications, IEEE Transactions on [legacy, pre - 1988] Volume 13, Issue 4, Dec 1965 Page(s):475 – 483.

[27] Ward, R.; Kai Yiu;, "Acquisition of Pseudonoise Signals by Recursion-Aided Sequential Estimation", Communications, IEEE Transactions on [legacy, pre - 1988]Volume 25, Issue 8, Aug 1977 Page(s):784 – 794.

[28] Yamada, M.; Kamio, Y.; Wada, Y., "Acquisition of direct-sequence spread-spectrum signal with parallel matched filters", Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on Volume 2, 18-21 Sept. 2000 Page(s):1260 - 1264 vol.2.

[29] "Physical layer procedures (FDD)", 3GPP TS 25.214 V8.1.0 (2008-03).

[30] Wang, Y.-P.E.; Ottosson, T., "Cell search in W-CDMA", Selected Areas in Communications, IEEE Journal on Volume 18, Issue 8, Aug. 2000 Page(s):1470 – 1482.

[31] Moon, J.; Yong-Hwan Lee, "Rapid Slot Synchronization in the Presence of a Large Frequency Offset and Doppler Spread in WCDMA Systems", Wireless Communications, IEEE Transactions on Volume 4, Issue 4, July 2005 Page(s): 1325 – 1330.

[32] Iksoo Eo; Hoseok Lee; Kyungsu Kim, "High speed code acquisition for wideband CDMA system", VLSI and CAD, 1999. ICVC '99. 6th International Conference on, 26-27 Oct. 1999 Page(s):293 – 296.

[33] Hassan A., Chakfeh Y., and Elhakeem A. K., "Pipelined, Multi-target, Peaks Combining Techniques for WCDMA Signal Acquisition in Fading channels".

[34] Aissaoui, A.; Hammoudi, Z.; Farrouki, A.;, "Adaptive PN code acquisition using automatic censoring for DS-CDMA communication". IEEE Waveform Diversity and Design Conference 2007. International, June 2007 Page(s):169 – 173.