Acquisition and Tracking of Weak GPS Signals as Received by Cellular Telephones

A Thesis Submitted to the College of Graduate Studies and Research

in Partial Fulfilment of the Requirements

for the Degree of Master of Science

in the Department of Electrical and Computer Engineering

University of Saskatchewan

Saskatoon, Saskatchewan

by

Howard Grant

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Electrical and Computer Engineering
University of Saskatchewan
Saskatoon, Saskatchewan S7N5A9
Canada

OR

Dean
College of Graduate Studies and Research
University of Saskatchewan
107 Administration Place
Saskatoon, Saskatchewan S7N 5A2
Canada

ABSTRACT

This thesis investigates the suitability of global navigation satellite system (GNSS) signals for cellular phone location. The requirement is to determine and report the location of a phone during an emergency call.

The thesis analyzes acquisition and tracking techniques suitable for very weak signals as received by a cellular phone indoors. The L1 and L5 signals from GPS satellites and the L1 signal from Galileo satellites are considered. It is shown that long integration times and coherent integration are required for the weakest expected signals. Long coherent integration times require a precise knowledge of the Doppler shift due to the range rate of the satellite. The tolerance to Doppler shift can be increased by using FFTs in the analysis of the data. Non-coherent averaging techniques improve the Doppler tolerance but compared to coherent averaging, the loss of signal to noise ratio is too large for the weakest signals.

Coherent averaging of the GPS L1 signal requires data removal that can be accomplished with assistance from the cellular network. The GPS L5 and Galileo L1 signals include a data-less or pilot channel. The GPS L5 pilot channel includes a 20 bit Neuman Hoffman code with a bit period of 1 ms. This code has to be acquired or removed before coherent averaging. Similarly the Galileo pilot channel includes a 25 bit short code.

Once code acquisition has been accomplished, it is necessary to track the signals from at least four satellites for long enough to compute a position estimate. A discussion of tracking techniques is included to show the signal to noise ratio limitations for adequate tracking accuracy.

The results show that GNSS signals are suitable for cellular phone location in a large number of situations. Increased receiver sensitivity would permit location in additional situations. In rural situations GNSS may be the only available option.

AKNOWLEDGEMENTS

I wish to thank my wife for her patience and support during the years that I strugled with this work. She put up with the many hours that I spent on course work and on my thesis when there were many other things that needed to be done. When I was discouraged and ready to quit, she encouraged me to continue.

I wish to thank Professor David Dodds for his guidance and assistance in completing this document.

CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABREVIATIONS AND ACRONYMS

A/D	Analogue to Digital (converter)
AOA	Angle of Arrival
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BOC	Binary Offset Carrier
BOCc	BOC Cosine = sgn(cos(2*π*F_{BOC}))
BOCC	Combined BOC used in revised Galileo L1, see Appendix 1
BOCs	BOC Cosine = sgn(sin(2*π*F_{BOC}))
bps	bits per second
BPSK	Binary Phase Shift Keying
C/A	Coarse Acquisition
dB	decibel
dBi	decibels above isotropic
dBW	decibels above 1 Watt
DOP	Dilution of Position
EIRP	Effective Isotropic Radiated Power
$\mathcal{F}(\bullet)$	Fourier transform
FFT	Fast Fourier Transform
Galileo	European Satellite Navigation System
GLONASS	Russian Satellite Navigation System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HOW	Hand Over Word
Hz	Hertz (cycles per second)
ICD	Interface Control Document
IE	Information Exchange
IF	Intermediate Frequency
ITU	International Telecommunications Union
K	Kelvin (degrees Celsious above absolute zero)
k	Boltzman's constant
km	kilometer
LMU	Location Measurement Unit
LNA	Low Noise Amplifier
m	minutes or meters
m/s	meters per second
MA	Mean Anomoly
Mch	Mega chips
ms	millisecond
Msps	Mega samples per second
NH	Newman-Hoffman (code)
OCXO	Oven Controlled Crystal Oscillator
OTDOA	Observed Time Difference of Arrival
ppm	parts per million

$Q(x)$	Q function, cumulative normal distribution from x to infinity
$Q_M(\alpha,\beta)$	Marcum Q function, see Appendix 1
RA	Right Ascension Angle (Right Ascension of the Ascending Node)
RF	Radio Frequency
RMS	Root Mean Square
RTD	Real Time Difference
s	second
TCXO	Temperature Compensated Crystal Oscillator
UTDOA	Uplink Time Difference of Arival
Σ	summation

LIST of SYMBOLS

B, BW	Bandwidth
C/N	Carrier to Noise Ratio
C/No	Carrier to Noise Density Ratio
Eb/No	Energy per bit to Noise Power density ratio
G/T	Receiver Figure of merit, Ratio of antenna gain to system noise temperature
L_C	Code length (chips)
M	The number of correlations to be averaged
N	Noise Power
$N(\mu,\sigma)$	Normal distribution with mean = μ and standard deviation = σ
No	Noise power density
P_d	Probability of Detection
P_{fa}	Probability of False Alarm
S/N	Ratio of Signal power to Noise power in measurement bandwidth
S/No	Ratio of Signal power to Noise power density
Tc	Chip Time, duration of a code chip
T_L	duration of a code of L_C chips = $L_C Tc$
δ	symbol used to designate a small (incremental) value i.e. δF, $\delta\omega$
η_I, η_Q	In phase and quadrature values of noise voltage
λ	wavelength or non centrality factor of a non-central chi-squared distribution
μ	mean of a distribution
σ	standard deviation of a distribution

1.0 INTRODUCTION

1.1 STATEMENT of OBJECTIVES

The objective of this thesis is to show that global satellite navigation systems are suitable for locating cellular phones during an emergency call. For some time it has been a requirement in the USA to determine a cellular phone position during a 911 call. The requirement is detailed in E911. After several people died in Canada when their location could not be determined, a similar requirement has been placed on Canadian cellular phone operators. In environments where the portable phone can exchange signals with at least three towers, the location can be determined by cellular system transit time methods. However, in many parts of Canada coverage from only one or two towers is available. In these environments, systems like GPS are the only option for location.
There are two main steps to showing that GNSS systems are suitable for cellular phone location. The first step is to analyze the range of GNSS signal levels expected to be received by a cellular phone. The second step is to show that the weak signals that are expected can be acquired and tracked.

1.2 OUTLINE of THESIS

In this thesis GPS and Galileo systems are discussed. GPS is the well known system that was originally developed for the U.S. military and is commonly used for civil navigation. Galileo is the European system that is currently under development. Although not considered here, there is also the Russian system GLONASS. GLONASS is nearing full implementation and would have had a complete constellation of satellites were it not for the recent launch failure.

The characteristics of the GPS and Galileo satellite constellations are described in Sections 1.3.1 and 1.3.2. Sections 1.3.3 and 1.3.4 describe the signals transmitted by GPS and Galileo satellites. The satellite signals are modulated by spreading codes to enable precise transmission time measurements. Section 1.3.5 provides the characteristics of the spreading codes.

Section 1.4 is an important section. This section estimates the range of signal to noise ratios expected to be received by GPS receivers and cellular phones receiving GNSS signals. The first part of the estimation is to quantify the quality of the receivers as expressed as the figure of merit. This figure of merit is determined from the effective antenna gain and the system noise temperature. It is shown that the cellular phone antenna is not ideal for receiving GNSS signals. Using the estimated figures of merit and the signal levels at the surface of the earth the expected received signal to noise ratio for the GPS receivers and cellular phones are computed. Following this calculation, there is a discussion of attenuation of the signals expected in various environments. From all this data the expected range of received levels is computed and a minimum signal to noise ratio objective for cellular phones is chosen. I have not found any publication that defines the characteristics of cellular phones receiving GPS or other GNSS signals. This work is unique to this thesis.

Since time and frequency accuracy are important to successful operation of GNSS receivers, Section 1.5 describes the characteristics of reference oscillators used in these receivers.

Chapter 2 discusses various topics relating to acquisition of GNSS signals. Acquisition is primarily finding the correct alignment between the received code that is modulated on the signal and a locally generated code. The code offset is used to determine the precise transit time delay from the satellite to the receiver.

Section 2.1 discusses the required minimum sampling rates and the losses due to quantizing with a small number of bits. This is a review of published work although some analysis and simulation was performed to verify the results.

Section 2.2 provides the theory for correlation showing the signal to noise ratio improvement and the distribution of the non-aligned and aligned signal amplitudes. The

theory is widely published. Examples are presented to verify the theory. Correlation with GPS codes is simulated and the results are presented.

Section 2.3 summarizes the loss of correlation gain due to Doppler frequency offsets. The relationship between frequency offset and probability of detection is shown.

Section 2.4 describes the various correlation techniques. The relationship between correlation techniques and correlation time is discussed.

With the very weak signals that are encountered with cellular phones receiving GNSS signals, significant averaging of correlation results is required for successful acquisition. The principles of averaging have been extensively published. Section 2.5 of this thesis provides the theory and shows how these averaging techniques are used in acquisition of the GNSS signals. All the techniques were simulated and the results are shown. The importance of the distribution of the non-aligned and aligned distribution of signal amplitudes is shown. A method of comparing techniques is presented in section 2.5.5. This comparison appears to be novel and was first presented in our paper [20]. Section 2.5.6 shows how a similar comparison is applied to segmented correlation.

Section 2.6 describes the various strategies used to detect code alignment after correlation and averaging.

The time to acquire weak signals is important when reporting a position during an emergency call. Section 2.7 provides analysis of acquisition time. The theory and methodology for acquisition time when using parallel correlation is new to this thesis.

The results of the acquisition time analysis show that external assistance is needed when the signals are weak. Section 2.8 discusses assistance from the cellular base station and shows the improved acquisition times when assistance is available. The analysis of acquisition times for weak signals relates to the procedures detailed in Chapter 3.

Chapter 3 describes optimum procedures for acquiring weak signals from GPS and Galileo. The principles employed are described in Chapter 2 but the particular arrangement of these principles to obtain acceptable results is the result of this research.

Chapter 4 describes the techniques required to track the signals after initial correlation. The purpose of the chapter is to show that weak signals can be tracked for long enough to determine a position. Delay locked loops used to track code alignment are discussed and simulated. The results of the simulation with the three signals of interest are given. Since the frequency is constantly changing due to the motion of the satellite, frequency and phase tracking are required. Techniques for frequency/phase tracking are presented. Simulation results are given.

Finally, Chapter 5 summarizes the results of the thesis and provides the conclusions.

1.3 BACKGROUND

1.3.1 Orbits and Constellations

1.3.1.1 GPS Orbits and Constellation

The GPS satellite constellation [1] consists of a minimum of 24 operational satellites plus spares. At the present time there are 31 satellites in orbit. The minimum constellation has 24 satellites arranged in six planes with 4 satellites per plane. The orbits are nearly circular with a mean radius of 26560 km giving an orbit rate of 2 orbits per sidereal day. The eccentricity is specified to be less than .02. The inclination of the orbits is approximately 55 degrees. The orbit planes are arranged around the earth at approximately 60 degree intervals. In each plane there are two satellites about 30 degrees apart and the others are spaced between 92 and 130 degrees. For example, in the first orbit plane, the separations are 130, 98, 32 and 100 degrees for a total of 360 degrees.

With this constellation, a satellite that passes overhead may be visible for nearly seven hours. Other satellites will be seen at a lower elevation angle and will be visible for a

shorter time. At any one time there will be at least 8 satellites visible from any location on earth. Four satellites are required to accurately determine a location.

The orbits are defined in a coordinate system called the earth centered inertial coordinate system (ECI). The x axis of this system is in the equatorial plane and points from the center of the earth through the prime meridian at the time of the vernal equinox. This reference direction is fixed in space, not rotating with the earth. The z axis is the axis of rotation of the earth and points to true North. The y axis, also in the equatorial plane is orthogonal to the x and z axes to form a right handed system. Orbit planes are defined by the angle in the equatorial plane from the x axis to the point where the orbit intersects with the equatorial plane when the satellite is moving northward. This angle is called the right ascension angle (RA). The angle between the orbit plane and the equatorial plane is called the inclination. The position of a satellite in the orbit plane is determined from an angle known as the mean anomaly (MA). The mean anomaly is measured from the perigee (the closest point to the earth) to the position the satellite would be in were the orbit circular. It is called the mean anomaly because it is computed as the product of the mean motion (or average orbit rate) and the time from the perigee. The actual position of the satellite is computed from the mean motion and the eccentricity.

The constellation was modeled and the position of the satellites was computed as a function of time. Figure 1.3/1 shows the elevation angle of the 24 satellites as a function of time from a location near Saskatoon. In this figure the satellites are labeled by the right ascension angle (RA) and the mean anomaly (MA). The time axis is the time in minutes from the epoch. The epoch is the time the orbits were initially defined. The time reference is only given to show the relative position of the 24 satellites. Twelve hours or 720 minutes are shown because the orbits repeat approximately twice a day.

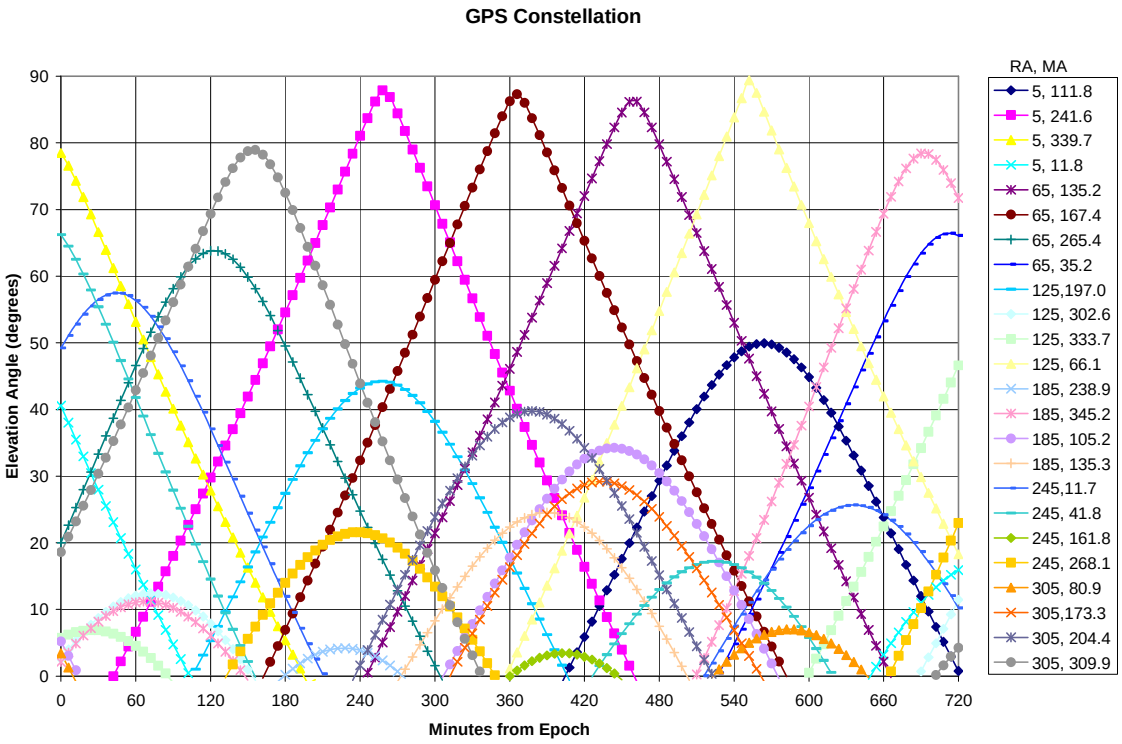

Figure 1.3/1: Elevation Angle vs. Time for GPS Satellites

1.3.1.2 Galileo Orbits and Constellation

The Galileo constellation [8] consists of 27 operational satellites plus 3 spares. The satellites will be in 3 planes spaced evenly around the earth with 9 satellites per plane. The semi major axis of the orbit is approximately 29600 km and the orbits are nearly circular. The orbit inclination is 56 degrees.

A satellite that passes overhead will be visible for almost nine hours. Figure 1.3/2 shows the elevation vs. time in minutes for a typical satellite.

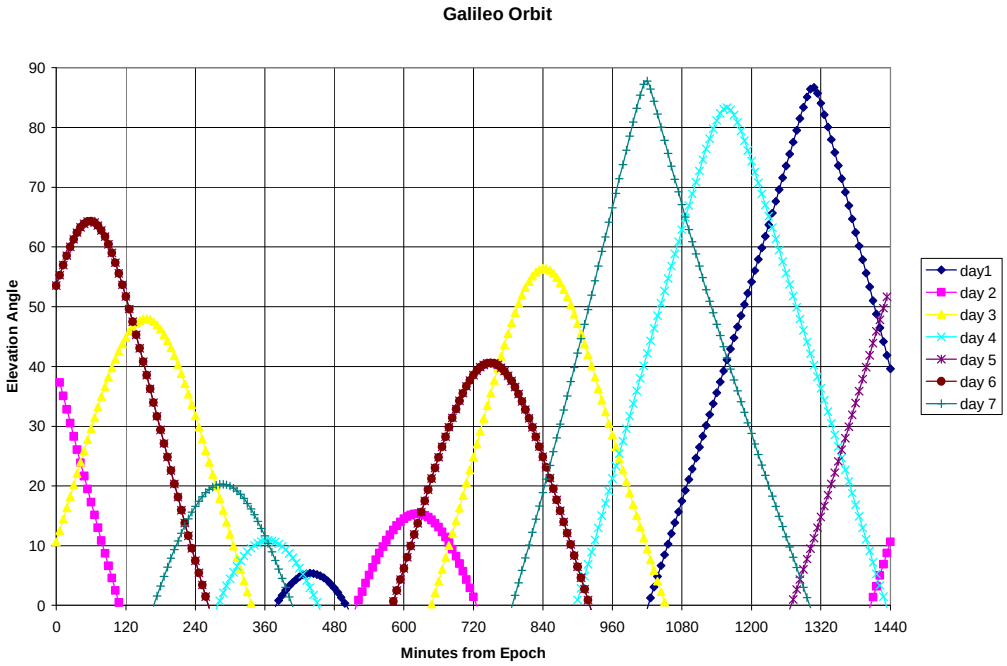

Figure 1.3/2: Elevation Angle vs. Time for a Galileo Satellite

1.3.2 Range and Range Rate

The closest approach of a GPS satellite to a location on the earth surface is 20182 km and occurs when a satellite is directly overhead. The maximum distance occurs when a satellite is on the horizon and is approximately 25911 km. These distances result in a transmission time range of 67.3 ms to 86.4 ms giving a difference of 19.1 ms.

For a Galileo satellite the closest approach, approximately overhead, is about 23230 km and the longest distance is about 28900 km resulting in a range of transit time of about 19 ms.

Figure 1.3/3 shows the range rate vs. time for the same constellation of satellites as shown in Figure 1.3/2. Range rate (*RR*) is the rate that the distance from the satellite to the receiver (*R*) changes as a function of time: $RR = dR/dt$. The range rate is not worst case. The worst case is for a satellite with the maximum eccentricity and when the

receiver is in the plane of the orbit. The maximum range rate occurs when a satellite is near the horizon. For GPS the maximum is about 850 m/s. Doppler shift is computed as:

$$D_f = \frac{1}{\lambda}\frac{dR}{dt} = \frac{f_c}{c}\frac{dR}{dt} \text{ or } D_f = RR\frac{f_c}{c}$$

For L1 the Doppler shift is -5.25 times the range rate giving a maximum Doppler close to 4500 Hz. With the higher orbit, the maximum range rate for Galileo is about 722 m/s giving a maximum Doppler shift of about 3800 Hz.

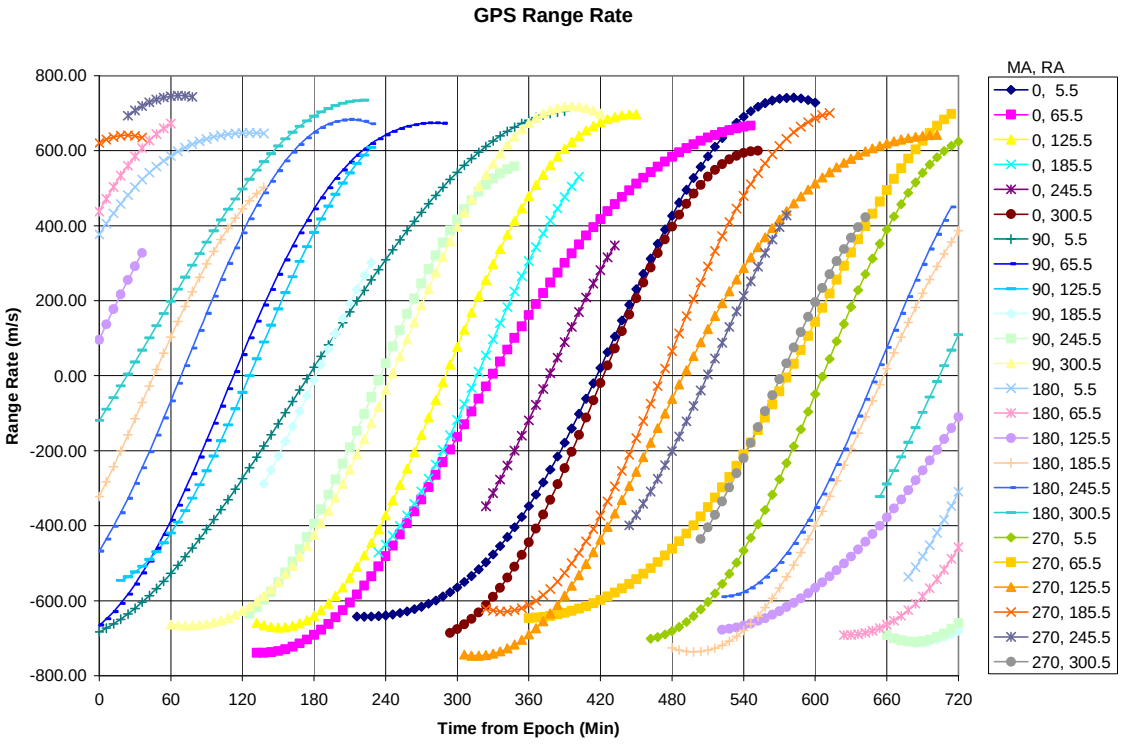

Figure 1.3/3: Range Rate vs. Time for GPS Satellites

1.3.3 GPS L1 and L5 Signal Structure

Each of the GPS II and IIR [6] satellites transmit signals on two frequencies. These are L1 at 1575.42 MHz and L2 at 1227.6 MHz. The L2 signal is not considered in this work. The GPS IIF satellites [7] will also transmit the L5 signal at 1176.45 MHz. The first GPS IIF satellite has been launched and is undergoing tests.

The signals include a time reference that is the time the signal originated from the satellite and information defining the position of the satellite. The GPS L1 information

rate is 50 bits per second. The L1 signal is spread at a rate of 1.023 mega chips per second (Mch/s) with a Gold code that is unique to the satellite. This spreading code is called the C/A code or coarse acquisition code. The C/A code is BPSK modulated onto the carrier after modulation by the data. There is a second spreading code called the P code with a chip rate of 10.23 Mch/s. The two codes are modulated in quadrature. The P code is a long pseudo-random sequence. The P code is the exclusive-or of two sequences called X1 and X2. The X1 code has a period of 1.5 seconds and is used as the timing reference for the data. Each satellite has unique spreading codes. The codes are defined in the interface control document (ICD) [6].

Figure 1.2/4 shows the frame structure. Each frame consists of 5 sub-frames of 300 bits with duration of 6 s. The start of the frame coincides with the X1 code epoch. With the X1 code epoch occurring every 1.5 s there are 4 epochs per sub-frame.

Sub-frames 4 and 5 are sub-commutated so that 25 frames are required to obtain all the information. However, sub-frames 1, 2 and 3 include all the information required to calculate a position locus that includes the satellite ephemeris. Sub-frames 4 and 5 include almanac information for all the satellites including ephemeris data, with reduced accuracy, and satellite health.

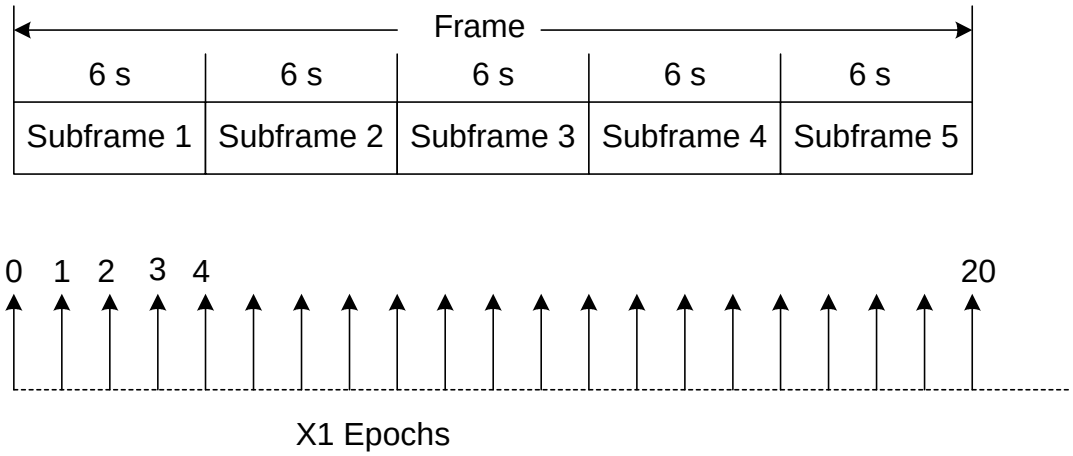

Figure 1.3/4: GPS Data Structure

To get sufficient time information to compute a position locus, code, bit and frame synchronization must be obtained. Code synchronization gives the time to a fraction of a chip period but within a 1 ms code cycle. Bit synchronization defies the transit time to within a 20 ms bit period. To finally resolve the time ambiguity sub-frame synchronization is required. Word 2 in all sub-frames is the handover word (HOW). The HOW word starts with the 17 most significant bits of the 19 bits defining the number of X1 epochs since the beginning of the week. This count, known as the truncated time of the week (TOW) count defines the exact time of the start of the transmission of the next sub-frame.

If the satellite ephemeris has recently been stored, it is possible to obtain a position locus without frame synchronization. As stated in section 1.3.2, the range of travel times from the satellite to the earth is 19.1 ms (67.3 ms to 86.4 ms). Since the range is less than the bit period, the travel time can be established from the code and bit timing. Given a reasonably accurate clock in the receiver, the time of transmission can be determined and the satellite position computed.

The GPS L5 signal has code rates of 10.23 Mch/s and the codes are 10230 chips long (1ms). The in-phase component of the signal is modulated by the spreading code followed by a 10 bit Neumann-Hoffman (NH) code and data at 50 bps that is ½ rate FEC coded (100 symbols per second). The quadrature component is modulated with a spreading code and a 20 bit NH code but no data. The codes are generated as the moulo-2 sum of two independent sequences in a similar manner to the generation of Gold codes. The data structure is similar to L1.

1.3.4 Galileo L1 Signal Structure

The Galileo satellites [8] are planned to transmit three signals labeled L1, E5 and E6. Only L1 is considered in this work. The carrier frequency (f_C) for L1 is 1575.42 MHz, the same as GPS L1. To decrease the interference with GPS, the L1 signal is modulated by binary offset carriers (BOC) to create pairs of spectral lobes above and below the carrier frequency.

There are three components to the Galileo L1 signal, designated L1a, L1b and L1c. L1a is modulated in phase and L1b and c are modulated in quadrature. The Galileo L1b and c codes have a rate of 1.023 Mch/s and are 4092 chips long (4 ms). The L1a code has a rate of 2.5575 Mch/s (2.5x1.023). The spreading sequence is the modulo-2 sum of the spreading code and a binary offset carrier (BOC). The BOC is a square wave with amplitudes ±1 and frequency equal to the BOC rate. The BOC rate *(F_{BOCa})* for L1a is 15 x 1.023 MHz. The BOC for L1a is described as:

$$BOC_{\cos} = \text{sgn}(\cos(2\pi F_{BOCa} t)) \quad (1.1)$$

The BOC rate for L1b and c is 1.023 MHz, designated F_{BOCb}. The BOC for L1b and c is:

$$BOC_{\sin} = \text{sgn}(\sin(2\pi F_{BOCb} t)) \quad (1.2)$$

The L1a signal is for government use only and is not of interest to civilians.

The spreading codes are described as memory codes. These codes cannot be reproduced in the way that Gold codes are but are simply stored as a set of binary sequences unique to each satellite. The L1b and L1c codes repeat synchronously every 4 ms. The L1a code is also synchronous with the L1b and L1c codes but the length is classified. It is specified that there are an integer number of L1a codes for each L1b and L1c code. The L1b signal is modulated by the modulo-2 addition of the spreading code and the BOC and data at 120 bps rate ½ coded. There are 10 synchronization symbols resulting in a symbol rate of 250 symbols per second. The L1c signal is modulated by the modulo-2 addition of the spreading code and the BOC and a 25 bit short code. The symbol rate for the short code is 250 symbols per second.

There are three components to the signal that is used to modulate the two phases of the carrier:

$$\begin{aligned} E_{L1a} &= C_{L1a} D_{L1a} BOC_{\cos}(15) \\ E_{L1b} &= C_{L1b} D_{L1b} BOC_{\sin}(1) \\ E_{L1c} &= C_{L1c} BOC_{\sin}(1) \end{aligned} \quad (1.3)$$

The numbers in the brackets denote the BOC frequency in multiples of 1.023 MHz. The symbols C_{L1a} and C_{L1b} are the spreading codes, C_{L1c} is the product of the spreading code and the short code. D_{L1a} and D_{L1b} represent the data bits.

The L1 signal modulation is then:

$$S_{L1} = \frac{1}{3}\left\{\left[\sqrt{2}E_{L1b} - \sqrt{2}E_{L1c}\right] + j\left[2E_{L1a} + E_{L1a}E_{L1b}E_{L1c}\right]\right\} \quad (1.4)$$

The in-phase modulation is the difference of the L1b and L1c signals. The quadrature modulation is the L1a signal plus the product of the three signals. The product term is added to place the symbols on the unit circle. The signal constellation has 6 states as shown by Figure 1.3/5.

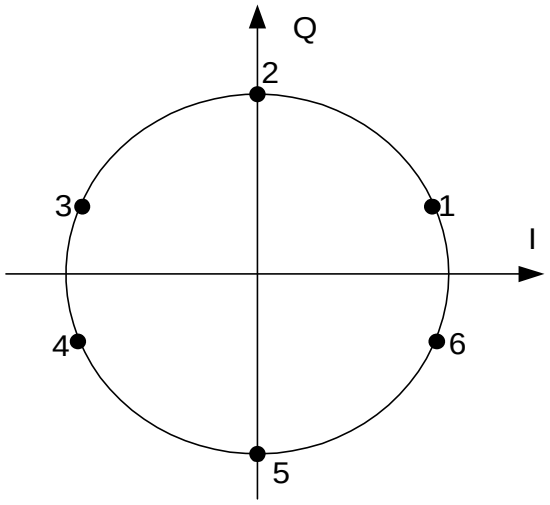

Figure 1.3/5: Galileo L1 Signal Constellation

The power sharing on the carrier is given in table 1.3/1.

Table 1.3/1: Signal Sharing for Galileo L1

Signal	Power Share
L1a	4/9
L1b	2/9
L1c	2/9
Product	1/9

The values in the table represent the unfiltered sharing. After filtering, the power in L1a is reduced as it has higher frequency components than L1b or L1c.

The spectrum resulting from the modulation has main lobes on both sides of the carrier at ±15.345 MHz due to the L1a BOC and at ±1.023 MHz due to the L1b and L1c BOC. The modulation creates a null at the carrier frequency.

The Galileo L1b data is transmitted in pages. Each data page has 120 data bits that are rate ½ coded then 10 synchronization bits are added. The resulting 250 bits are transmitted every second. The data are organized in sub-frames of 30 pages. The navigation data for the satellite transmitting the data is contained within each sub-frame. To obtain all the almanac data 24 sub-frames are needed (120 x 30 x 24 = 86400 bits).

1.3.5 Code Characteristics

GPS uses Gold codes for the coarse acquisition (C/A) spreading code of length 1023. Each satellite has a unique code. Gold codes are constructed by taking the exclusive-or of the outputs of two m-sequences. The GPS code generator is shown by Figure 1.3/6.

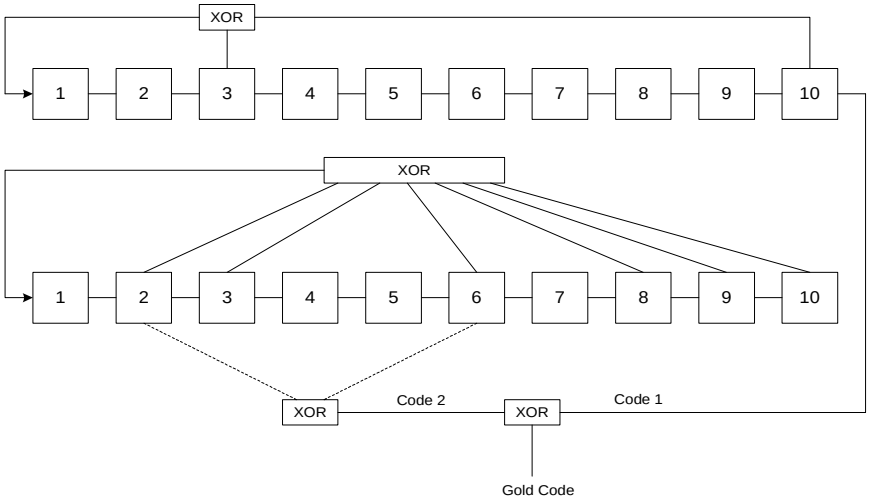

Figure 1.3/6: Gold Code Generator for GPS Code 1

The polynomials for the two m-sequences are $X^{10} + X^3 + 1$ and $X^{10} + X^9 + X^8 + X^6 + X^3 + X^2 + 1$. The different codes for the satellites are generated by taking the exclusive or of code 1 with different phases of code 2. The figure shows the connections for GPS code 1 where the phase is determined by the XOR of registers 2 and 6 of the code 2 generator. GPS code 31 uses the exclusive or of registers 3 and 8.

The cross correlation of any 2 GPS L1 C/A codes takes on only three values. These values are -1, -t and $t-2$. For codes of length 1023, $t = 2^{(l+2)/2}+1 = 65$ where $l = 10$. This gives false peaks of 63 and -65. The auto-correlation of a Gold code of length 1023 has the value of 1023 when aligned and otherwise has the values -1, 63 or -65 as shown by Figure 1.3/7.

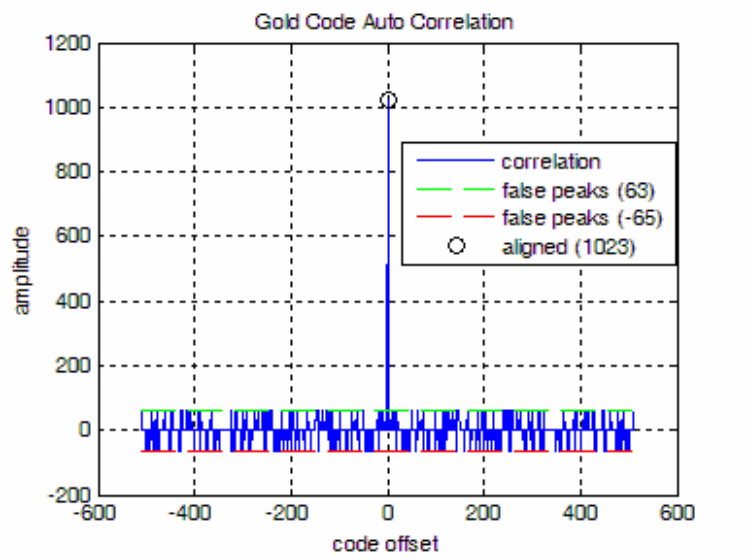

Figure 1.3/7: Gold Code Auto-Correlation

Figure 1.3/8 shows the auto-correlation of a GPS L5 code. The GPS L5 code, with length 10230, has relatively smaller auto correlation for non aligned phases as shown by the figure.

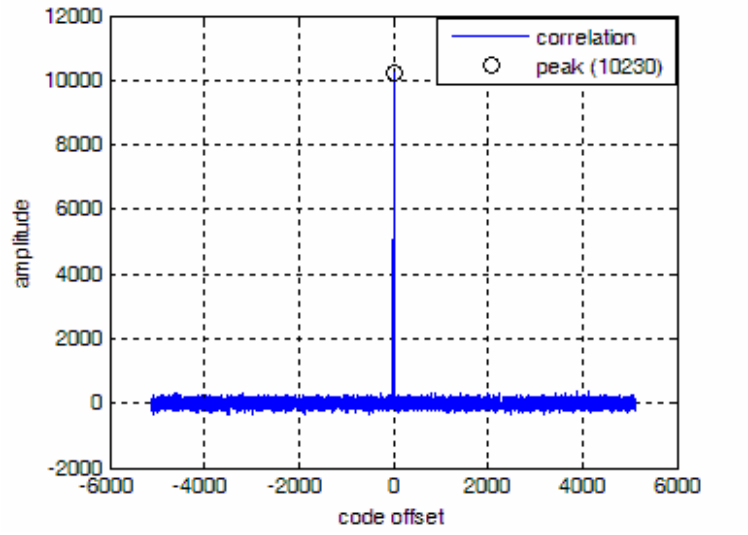

Figure 1.3/8: GPS L5 Code Auto-Correlation

The Galileo L1 codes are described as memory codes. These codes cannot be reproduced in the way that Gold codes are and are simply stored as a set of binary sequences unique to each satellite. The length of the Galileo L1b and L1c codes is 4092 chips (4 ms at a rate of 1023 Mchps). Figure 1.3/9 shows the autocorrelation of a Galileo L1 code. This is the auto-correlation of the code itself and does not show the effect of the BOC signal. The effect of the BOC is considered in Chapter 4.

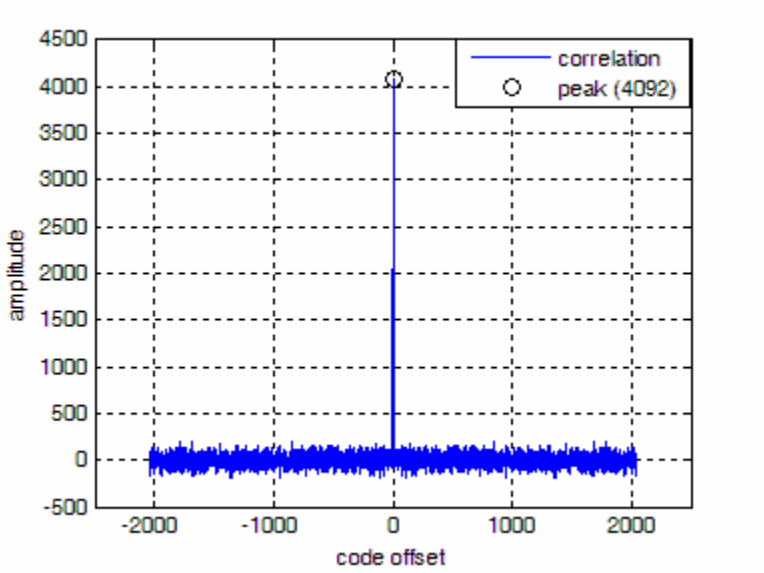

Figure 1.3/9: Galileo L1 Code Auto-Correlation

1.3.6 Dilution of Position (DOP)

The position of a receiver on the surface of the earth computed from each satellite is determined by an arc with a radius equal to the distance from the satellite. There is a region of uncertainty on each side of the arc due to the inaccuracy. If the time reference was accurate, the horizontal position could be determined, within the region of uncertainty, from a second satellite at right angles to the first. Since the receiver is unlikely to be on the surface of a sphere with the mean earth radius, a satellite at a high elevation is desired to obtain elevation information. Finally, to resolve the absolute time error, a fourth satellite is required.

The horizontal position error is degraded by the secant of the elevation angle. Furthermore, if the satellites are not well spaced in longitude, the position locus will be elongated.

The best position accuracy is obtained with three satellites at low elevation angles spaced well in azimuth and a fourth satellite at a high elevation angle.

Dilution of precision is covered in some detail in Chapter 5 of [1].

1.4 SIGNAL LEVELS

Signal levels from the satellites are given as the signal strength at the surface of the earth. All the satellite signals are right hand circularly polarized (RHCP). This signal strength is equivalent to the signal received by an RHCP antenna with unity (0 dBi) gain. When received by a linearly polarized antenna, there is a 3 dB loss. The GPS ICDs, [6] and [7], quote the signal strength as that received by a linearly polarized antenna with 3 dBi of gain, that is the same as received by a circularly polarized antenna with 0 dBi gain.

The GPS signal strengths are quoted for each signal component. The Galileo signal strengths are quoted as the total signal for each carrier. These levels are given in table 1.4/1.

Table 1.4/1: Signal Levels at the Surface of the Earth

Satellite	Signal	Carrier Power (dBW)	Correlation Loss (dB)	Signal Power (dBW)
GPS	L1 C/A			-158.5
GPS	L5I			-157.9
GPS	L5Q			-157.9
Galileo	L1b	-152	6.2	-158.2
Galileo	L1c	-152	6.2	-158.2

These levels are the guaranteed minimum and actual values will be somewhat higher. Fisher [19] shows the measured values of the L1 C/A code signal from block II/IIA satellites to be -156.7 dBW. This value will be used in the analysis. For the other

signals, the specified levels [7], [8] shown in Table 1.4/1 will be used until measured values are available.

At this time a note regarding signal to noise ratio measurements and designations is in order. The simplest concept is carrier to noise ratio, *C/N*. *C/N* represents the ratio of total power of all signals modulated on a carrier to the total noise in the measurement bandwidth. This bandwidth is defined by a filter. The filter is normally the input filter of the receiver. Another frequently used term is the carrier power to noise power density ratio, *C/No*. *C/No* is the ratio of the carrier power to the noise in a 1 Hz bandwidth. The relationship between *C/N* and *C/No* is as follows: $C/No = C/N - 10\log_{10}(BW)$ (1.5) where *BW* is the bandwidth of the filter in Hz. If the filter is perfect, with unity gain over the pass-band and zero gain elsewhere, the bandwidth is the width of the pass-band. However a perfect filter is unrealizable. A real filter has variable gain across the pass-band. The bandwidth used to calculate *C/No* is a parameter known as noise bandwidth (B_N). Noise bandwidth is the bandwidth of a perfect filter that would allow the same noise to pass through as our filter:

$$B_N = \frac{1}{Ho}\int_{f_0-F}^{f_0+F} H(f)df \qquad (1.6)$$

where *Ho* is the power response at the center of the filter (f_0), *H(f)* is the response at frequency *f* and ±*F* represents the range of frequencies for which the filter has a measurable response. Theoretically, ±*F* should be ±∞. The GNSS carriers discussed here are each modulated with more than one signal. To be clear the power in the relevant signal is referred to as *S* and the signal to noise ratio as *S/N*. The signal to noise power density ratio is *S/No*.

The signal to noise ratio in a system changes from the input to the output. In this system, the signal to noise ratio is improved by filtering, correlation and by averaging. The signal to noise ratio at the reference point will be described in the relevant sections and the improvement will be quantified. Given a signal level (*S*) such as defined in Table 1.4/1, the received signal to noise power density ratio is determined as:

$$S/No = S + G_A - 10\log_{10}(kT_S) \tag{1.7}$$

and the signal to noise ratio is

$$S/N = S + G_A - 10\log_{10}(kT_S B_N) \tag{1.8}$$

where S is the signal level at the input to the antenna, G_A is the net gain of the antenna, k is Boltzmann's constant, T_S is the system noise temperature and B_N is the noise bandwidth of the filter.

The signal to noise ratio in the receiver depends on the antenna gain and polarization, the receiver noise figure and the antenna noise temperature. The equation for *S/No* can be rewritten as

$$S/No = S + 10\log_{10}\left(\frac{G_{AR}}{T_S}\right) - 10\log_{10}(k) \tag{1.9}$$

where G_{AR} is the antenna gain as a number (not in dB). Clearly,

$$S/N = S/No - 10\log_{10}(B_N)\,. \tag{1.10}$$

A parameter, known as the figure of merit, is often used to express the net effect of antenna gain and the system noise figure. The figure of merit, *G/T*, is the ratio of the net antenna gain divided by the system noise temperature. This parameter is usually expressed in dB and properly has the unit dB K. Given the signal level and the *G/T*, the *S/No* is determined as follows:

$$S/No = S + G/T + 228.6 \tag{1.11}$$

The term S is the signal strength at the input to the antenna as shown in Table 1.4/1. The number 228.6 is the negative of Boltzmann's constant expressed in dB. In this work the term *S/No* is used to define the power in the wanted signal to the noise power density. The term *C/No* refers to the total power of all signals modulated on an RF carrier.

Table 1.4/2 provides an estimate of the *G/T* for a professional GPS receiver, a hand held GPS receiver and a cellular phone receiving GPS. For a *G/T* analysis, a reference point for the antenna gain and system noise temperature is required. In this analysis, the reference point is the input to the LNA.

Table 1.4/2: *G/T* Analysis

Parameter	GPS Receiver Professional	GPS Receiver Hand Held	Mobile Phone	units
Antenna Gain at 20 deg. Elev.	2	1	-2.5	dBi
Polarization Loss	1	1	3	dB
Net Antenna Gain	1	0	-5.5	dBi
Antenna Noise Temperature	130	150	290	K
Loss to LNA	1	1	1	dB
Noise Temp. from Loss	59.64	59.64	59.64	K
Antenna Noise Temp at LNA	103.26	119.15	230.36	K
LNA Noise Temperature	110	129	129	K
System Noise Temperature	272.91	307.79	419.00	K
System *G/T*	-24.36	-25.88	-32.72	dB K

The values in table 1.4/2 are typical and actual performance will vary from receiver to receiver. The antenna gain for the phone is estimated as the mean over the upper hemisphere. At some angles the gain may be positive. However, at other angles the gain will be lower than the -2.5dB given in the table.

There are two significant factors resulting in the performance of the phone being worse than the GPS receivers. The first is the polarization. In North America, cellular signals are vertically (linearly) polarized. This results in a 3 dB polarization loss. The second factor is the antenna pattern. A GPS antenna is designed to receive signals from overhead. A phone antenna is designed to receive signals from a horizontal direction. Often a phone antenna receives signals from all directions. This characteristic reduces the gain in the direction of a satellite and results in a worse antenna noise temperature.

A low cost GPS antenna may have a linearly polarized antenna but the more sensitive units have circularly polarized antennas. There are basically two types of circularly polarized antennas used in hand held GPS receivers, patch antennas [48] and quadrifilar helix antennas [47]. The patch antenna requires a horizontal ground plane approximately 70 mm square. The quadrifilar helix antenna is about 14 mm diameter. Neither would fit in a modern cellular phone.

Phone antennas are either monopole or patch antennas. Rowley and Waterhouse [32] show typical performance for antennas designed for the 1800 MHz band. When the

antenna is in free space, the gain in the horizontal direction varies between 0 and -5 dBi. When placed near the human head the pattern is distorted and the gain towards the head is lower. The elevation patterns are generally symmetrical above and below horizontal. The antenna then sees the hot earth resulting in a much higher antenna noise temperature. The higher noise temperature is a red herring when comparing the performance of a phone and a GPS receiver under faded conditions as the "sky temperature" increases as the loss increases.

$$T_{Sky} = \frac{T_{Clear}}{L} + \frac{(L-1)}{L} 290 \qquad (1.12)$$

We see that as the fading loss (L) gets large, the sky temperature approaches 290. Thus in a severely attenuated environment all antennas will have a temperature approaching 290 K.

I have not found any reference to *G/T* of GPS receivers in the literature but, from reported signal levels and received *S/No*, the values for the GPS receivers are accurate to within 1 dB. Furthermore, I have not found any reference to actual *S/No* or *G/T* values for cellular phones receiving GPS signals.

Using the received signal levels and the *G/T* values, Table 1.4/3 provides expected *S/No* values for the three signals. The receivers are 1) a professional GPS receiver, 2) a hand held GPS receiver and 3) a cellular telephone equipped to receive GPS signals. These values are valid at a low elevation angle (20 deg) and with no obstruction.

Table 1.4/3: S/No Values for Receivers

S/No For Receivers	Professional	Hand Held	Phone	units
GPS L1 C/A	47.54	46.02	39.18	dB Hz
GPS L5	46.34	44.82	37.98	dB Hz
Galileo L1b/c (L1-6.2 dB)	46.04	44.52	37.68	dB Hz

If there is any obstruction in the path, due to foliage or buildings, the signal will be lower. The paper by Bryant [14] gives a rule of thumb for parking garages. He reports approximately 10 dB attenuation for each floor below the surface. He also reports signal levels of -180 to -185 dBW four floors down. Van Diggelen [16], [17] also reports signal levels down to -185 dBW in parking garages. The paper by NavSync [46] reports signal

levels under heavy foliage of -157 to -182 dBW, in a car trunk of -171 to -182 dBW and in a windowless office of -160 to -187 dBW. The results from these three papers show attenuations of up to 30 dB. The signals were received and measured with a hand held GPS receiver. With these receivers, a signal level of -188 dBW is close to the practical limit for acquisition and tracking. A cellular phone receiving GPS has a *G/T* that is about 6 dB worse. Considering this degradation, an objective of receiving GPS with a signal level of -182 dBW is practical. A signal level of -182 dBW corresponds to an attenuation of 25 dB. It can be seen that with 25 dB of attenuation, the *S/No* for the three signals as received by a cellular phone will be 13 to 14 dB Hz. Although this is not the worst signal that is likely to be encountered, this will be shown to be close to the practical limit for acquisition and tracking.

1.5 REFERENCE OSCILLATOR

Receivers employ a 10 MHz or 10.23 MHz oscillator as the reference for all frequencies including the carrier frequency and code rate. High quality GPS receivers have oven controlled crystal oscillators (OCXO) to provide a stable and accurate reference. A hand held unit generally has a temperature compensated crystal oscillator (TCXO). A phone may have a simple clock oscillator that is uncompensated for temperature. A TCXO will have frequency stability in the order of 1 parts per million (ppm). The stability of a clock oscillator may be 20 to 50 parts per million. Another factor is the short term stability. The stability of a TCXO or clock oscillator is in the order of 10^{-9} in one second. With a carrier frequency of 1.5 GHz the short term variation is about 1.5 Hz. This short term variation limits the integration time for acquisition as we will see in Section 2. The warm up time is also a concern. All oscillators require a warm up time before the accuracy is met. During the warm up time there are significant changes in frequency even for a high quality oscillator. The warm up time for a TCXO or clock oscillator could be several minutes.

2.0 SIGNAL ACQUISITION

2.1 SAMPLING and QUANTIZATION

All modern receivers use digital processing. Generally, the front end includes an analogue filter and a low noise amplifier (LNA). After the LNA, the signal may be down converted to an intermediate frequency or directly sampled at the RF frequency [10]. The exact configuration does not materially affect the result. The filter must be wide enough to pass most of the power in the signal but not so wide as to pass excessive noise.

The sampling rate for the digitization must satisfy the Nyquist criterion. The sampling could be real only or real plus imaginary (I and Q). For real only sampling the sample rate must be at least twice the bandwidth to satisfy the Nyquist criterion. Complex sampling can be considered as two samples and a sampling rate equal to the bandwidth is acceptable.
In the balance of this work, complex sampling is assumed. However real sampling is equivalent to complex sampling as long as the rate criteria are satisfied.

Generally the GNSS signal is digitized with a small number of bits, one to eight. There is a loss in signal to noise ratio associated with a small number of bits. Van Dierendonck [1] shows the loss due to 1 to 5 bits with various conditions. The loss depends on the filter bandwidth and the ratio of RMS noise to the maximum digitization threshold. The theoretical minimum loss for one bit quantization is 1.96 dB but can be as high as 3.5 dB in a narrow bandwidth. Two bit quantization under the best conditions could have a loss as low as 1 dB but could be as bad as 3.5 dB, the one bit loss. The loss for three bit quantization, with optimal loading of the quantizer, can be as low as 0.3 dB. With 8 bits the loss is negligible if the levels are set appropriately. Three bits are generally considered adequate.

The bandwidth of the main lobe of the GPS L1 C/A signal is twice the chip rate or 2.046 MHz. The bandwidth of the GPS L5 signal is 10 times that rate or 20.460 MHz. In this work the GPS L1 signal is digitized at 8.184 Msps, digitally filtered to a bandwidth of

2.046 MHz and down sampled to 2 complex samples per chip (2.046 Msps). The GPS L5 signal is directly sampled with 2 complex samples per chip (20.460 Msps).

For Galileo, the BOC signals have the effect of creating lobes, one on each side of the carrier. The L1a BOC creates lobes 15.345 MHz above and below the carrier frequency. The L1b and L1c BOC create lobes 1.023 MHz from the carrier frequency. The resulting band width of the L1b and L1c signals is at least 4 times the chip rate. Since we are not interested in the L1a signal, we can filter out the lobes centered at ±15.345 MHz around the carrier.

Figure 2.1/1shows the filtered and unfiltered Galileo L1 spectrum. The sampling rate for this plot was 30.69 MHz. This would be too low if the L1a signal was of interest as only half of the main lobe is captured at plus and minus the BOC rate. An odd symmetric FIR filter with 17 taps and a cut off frequency of .575 radians per sample was used, giving a two sided bandwidth of 5.5 MHz. For correlation, the filtered digitized spectrum was down-sampled to 5 complex samples per chip (5.115 Msps).

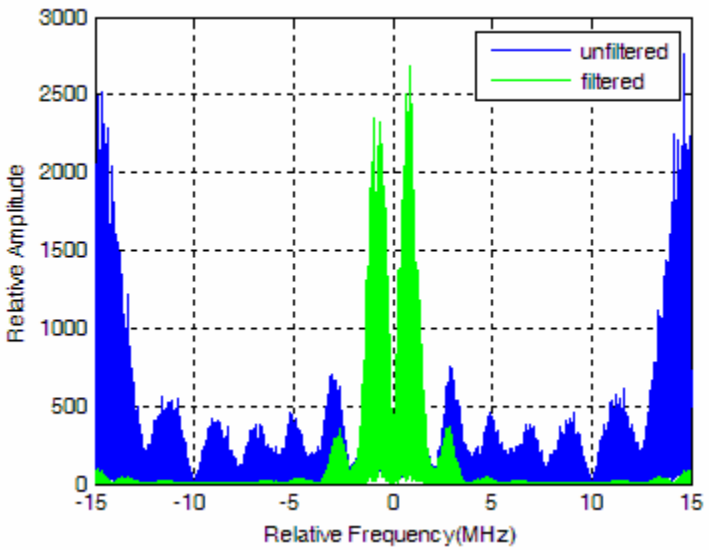

Figure 2.1/1: Galileo L1 Spectrum

2.2 BASIC CORELATION

The process of correlation is to align the coded received signal with a locally generated code. The process is to multiply the received signal by the locally generated code, element by element, and sum the result. The offset (delay) is changed and the multiply and sum process is repeated. After all possible delays have been tested the best alignment is determined as the offset that gives the largest output.

The following discussion is for a single signal component. Although all carriers are modulated by more than one signal component, the components are separated by their cross polarization properties and by filtering. The received input signal is represented as:

$$v(t) = AC(t+T)d(t)(\cos(\omega t+\theta)+ j\sin(\omega t+\theta))+\eta_I(t)+ j\eta_Q(t) \qquad (2.1)$$

Where *A* is the amplitude, *C(t + T)* is the delayed spreading code and *d(t)* is the data. *C(t + T)* and *d(t)* are binary signals with amplitudes ±1. Since we do not know the phase of the received signal, in-phase and quadrature components are required.

The terms η_I and η_Q are the in phase and quadrature components of the noise and both have the distribution $\mathrm{N}\left(0,\frac{\sigma_N}{\sqrt{2}}\right)$ where $\sigma_N{}^2$ is the total noise power at the input to the correlation.

Before correlation, the signal is mixed down as close as possible to zero frequency, leaving a residual frequency offset denoted $\delta\omega$. The resulting signal is:

$$v_2(t) = AC(t+T)d(t)(\cos(\delta\omega t+\theta)+ j\sin(\delta\omega t+\theta))+\eta_I + j\eta_Q \qquad (2.2)$$

At this time we will assume that the data modulation (*d(t)*) is constant for the period of the correlation. This is a good assumption as data changes occur synchronously at the start of every 20th code.

This signal is multiplied by the locally generated code, $C(t+\tau)$ to perform the correlation sum:

$$R(\tau) = \frac{1}{T_C}\int_{to}^{to+L_C T_C} (C(t+\tau)AC(t+T)d(t)(\cos(\delta\omega t+\theta)+ j\sin(\delta\omega t+\theta))+\eta_I + j\eta_Q)dt \tag{2.3}$$

L_C is the length of the code in chips and T_C is the chip period. When $T = \tau$ the codes are aligned and the output is:

$$R(T) = L_C Ad\int_{to}^{to+L_C T_C} (\cos(\delta\omega t+\theta)+ j\sin(\delta\omega t+\theta))dt + \eta_{LI} + j\eta_{LQ} \tag{2.4}$$

The term *d* is the data value that is not a function of time for the period of the correlation and has a value of ±1.

The expectation of *R(0)* is:

$$E(v(0)) = L_C A\frac{\sin(\pi\delta FLTc)}{\pi\delta FLT_C}\exp(j\theta) \text{ where } \delta F = 2\pi\delta\omega \tag{2.5}$$

When the codes are not aligned the integral of $C(t+\tau)C(t+T)$ is a small value, depending on the cross correlation properties of the code (see section 1.2.5). In this case the result is primarily complex noise.

Whether aligned or not the noise remains complex Gaussian.

In modern systems, the signal processing is digital. The received signal is band limited, sampled and digitized. In some cases further band limiting using digital filtering is employed. The main lobe of the GPS C/A code has bandwidth twice the chip rate or 2.046 MHz. If we sample at twice the chip rate, taking complex samples, the noise samples are independent and uncorrelated. The input signal is then the sampled version of *v(t):*

$$v(k) = AC\left(\frac{k}{2}+\frac{T}{T_C}\right)d(t)\left(\cos\left(\delta\omega\frac{kT_C}{2}+\theta\right)+ j\sin\left(\delta\omega\frac{kT_C}{2}+\theta\right)\right)+\eta_I + j\eta_Q \tag{2.6}$$

With two samples per chip, $t = \frac{kT_C}{2}$.

After correlation with the locally generated code $C(\frac{k}{2}+\frac{\tau_k}{2})$ where τ_k is an integer with values between *0* and $2L_C$*-1*, the result is:

$$R(\tau_k) = \sum_{k=1}^{2L_C} C(\frac{k}{2} + \frac{\tau_k}{2})\left[AC\left(\frac{k}{2} + \frac{T}{Tc}\right)d(t)\left(\cos\left(\delta\omega\frac{kTc}{2} + \theta\right) + j\sin\left(\delta\omega\frac{kTc}{2} + \theta\right)\right) + \eta_I(k) + j\eta_Q(k)\right]$$

(2.7)

$$R(\tau_k) = AK(\tau_k)d\frac{\sin(\pi\delta FL_C T_C)}{L_C\sin(\pi\delta FT_C)}\exp(j\theta) + \sum_{1}^{2L}\left(\eta_I(k) + j\eta_Q(k)\right) \quad (2.8)$$

The factor $K(\tau_k)$ depends on the filtering of the received signal and the alignment. The maximum value of $K(\tau_k) = 2L_C$ when the codes are aligned and the minimum is zero. When the codes are not aligned the product of the codes alternates between positive and negative values. The sum is then a small number. When the codes are perfectly aligned and unfiltered, the sum is $2L_C$. Since there are only two samples per chip, the best alignment could be off by one-quarter of a chip. A number of simulations were run with different filtering and alignment. The output with the worst loss was found to be 0.85 of the maximum output amplitude or -1.5 dB. The loss is a function of the filter characteristics and the number of samples per chip. Tsui [4] suggests that the number of samples per chip should not be an integer so that the alignment is constantly changing.

Multiplying the noise by the code does not change the statistics of the noise. The separate sum of the in phase or quadrature noise values are each Gaussian with the distribution: $\mathrm{N}(0,\sigma_L) = \mathrm{N}\left(0, \sqrt{2L_C}\frac{\sigma_N}{2}\right)$. (2.9)

The total noise power is then $N_L = 2L_C\sigma_N{}^2 = 2L_C N$ (2.10)

The signal power when aligned is ideally $S_L = A_L{}^2 = (2L_C A)^2$ (2.11)

In this case, correlation results in a signal to noise improvement of $2L_C$. This is the expected result.

The question now is what to use as a decision metric [12] in order to decide the best alignment. It is difficult to compare complex numbers especially when we do not know

the angle (θ). There are two possibilities, the magnitudes can be compared or the square of the magnitudes can be compared. Computing the magnitude requires the addition of a square root function. Computationally it is more efficient to compute the square.

Given the result is a complex number of the form $a + jb$, the square of the magnitude is $a^2 + b^2$. The aligned and non aligned cases have to be considered separately. It is assumed that the auto-correlation for the non-aligned case is negligible when compared to the received noise. When not aligned the sum is primarily noise that is Gaussian with zero mean. The sum of the squares of the real and imaginary parts has a chi-squared distribution with 2 degrees of freedom. The chi squared distribution with 2 degrees of freedom is negative exponential.

$$f(x) = \frac{1}{2\sigma^2}\exp\left(\frac{-x}{2\sigma^2}\right) \tag{2.12}$$

The variable x is the square of the magnitude or the power. The standard deviation σ in this equation is the standard deviation of the in-phase or quadrature noise terms at the output of the correlator (σ_L): $\sigma_L = \sqrt{2L_C}\,\frac{\sigma_N}{\sqrt{2}} = \sqrt{L_C}\,\sigma_N$. (2.13)

When the codes are aligned the distribution is non central chi-squared with the non centrality factor λ and 2 degrees of freedom (k).

$$\lambda = \sum_1^2 \frac{S_i}{\sigma_i^2} \text{ where} \tag{2.14}$$

S_i is the square of the signal amplitude of the in-phase or quadrature term at the output of the correlator and σ_i is the standard deviation of the in-phase or quadrature noise term.

$$\lambda = \sum_1^2 \frac{S_i}{\sigma_i^2} = \frac{(2L_C A)^2\left(\cos^2(\theta) + \sin^2(\theta)\right)}{2L_C\,\sigma_N^2/2} = \frac{(2L_C A)^2}{L_C\sigma_N^2} = \frac{4L_C A^2}{\sigma_N^2} \tag{2.15}$$

The mean of the distribution of the aligned signal amplitudes is:

$$\mu = k + \lambda \tag{2.16}$$

The variance is:

$$V = 2k + 4\lambda \tag{2.17}$$

To determine the correct alignment, a threshold is chosen above the noise power that has a low probability of being exceeded by the noise. This probability is called the probability of false alarm (P_{fa}).

$$P_{fa} = \int_{Th}^{\infty} \frac{1}{2\sigma^2} \exp(-\frac{x}{2\sigma^2}) dx = \exp(-\frac{Th}{2\sigma^2}) \tag{2.18}$$

Finally, substituting $\sigma_L = \sqrt{L_C}\sigma_N$, Equation (2.13), the threshold can be computed as a function of the probability of false alarm.

$$Th = -2L_C\sigma_N{}^2 \ln(P_{fa}) \tag{2.19}$$

The correct alignment is then determined by the delay that results in an output that exceeds the threshold. The probability of exceeding the threshold is known as the probability of detection *(P_d)*.

The probability of detection is computed as:

$$P_d = \int_{Th}^{\infty} f(v^2) d(v^2) \tag{2.20}$$

where v^2 is the sum of the squares of in-phase and quadrature voltage.

This probability can be computed using the Marcum Q function (Appendix A1).

$$P_d = Q_M(\alpha, \beta) = Q_M\left(\sqrt{\lambda}, \sqrt{\frac{Th}{\sigma_L{}^2}}\right) \tag{2.21}$$

Consider an example. In this example the unfiltered signal is sampled at 8.184 Msps then digitally filtered to a bandwidth of 2.046 MHz. This filtering removes unwanted signals and leaves only the main lobe of the GPS C/A signal. After filtering, the sample rate is down-sampled to 2 complex samples per chip. The input noise power (N_R) is set to 1 and the signal amplitude is set to the square root of the signal to noise ratio. The filtering reduces the noise power by a factor of four because the noise bandwith (equation 1.6) has been reduced to ¼ of the input bandwidth. At the output of the correlator, the standard deviation of the noise is computed using Equation (2.13) and allowing for the filter:

$$\sigma_L = \sqrt{L_C}\sigma_N = \sqrt{1023}\sqrt{\frac{1}{4}} = \frac{\sqrt{1023}}{2}$$

The signal amplitude at the output of the correlator is determined as the square root of the signal power as determined by Equation (2.11):

$$V_S = 2L_C A = 2046\sqrt{S/N}$$

For a given probability of false alarm the threshold can be calculated using Equation (2.19). For this example the probability of false alarm is set to .001.

$$Th = -2L\sigma_N{}^2 \ln(Pfa) = 511.5\ln(.001) = 3533$$

Given the threshold the probability of detection can be computed using the Marcum Q function. First we have to calculate the arguments of the Q function. The second argument (β) in Equation (2.21) is as follows:

$$\beta = \sqrt{\frac{Th}{\sigma^2}} = \sqrt{\frac{Th}{L}}\frac{1}{\sigma_N} = 2\sqrt{\frac{3533}{1023}} = 3.7 \qquad (2.22)$$

The probability of detection depends on the signal to noise ratio. The first argument of the Q function depends on the signal to noise ratio. In this example the amplitude of the signal (A) is the square root of the signal to noise ratio. The argument (α) is as follows:

$$\alpha = \sqrt{\lambda} = \frac{2A\sqrt{L}_C}{\sigma_N} \qquad (2.23)$$

Values of λ and P_d were computed for signal to noise ratios from -26 to -31 dB. Table 2.2/1 shows these values.

Table 2.2/1: Probability of Detection vs. *S/N*

S/N (dB)	*S/No* (dB Hz)	*A*	$\sqrt{\lambda}$	P_{fa}	P_d calculated	P_d simulated
-26	43.1	.05	6.4	.001	.997	1
-27	42.1	.045	5.76	.001	.983	.96
-28	41.1	.04	5.12	.001	.93	.90
-29	40.1	.035	4.48	.001	0.8	.75
-30	39.1	.0316	4.04	.001	.66	.6
-31	38.1	.028	3.6	.001	.495	.43

This correlation was simulated and the probability of detection was obtained after multiple tries. Figure 2.2/1 shows the computed and simulated probability of detection vs. signal to noise ratio. The signal to noise ratio was determined in the input bandwidth of 8.184 MHz.

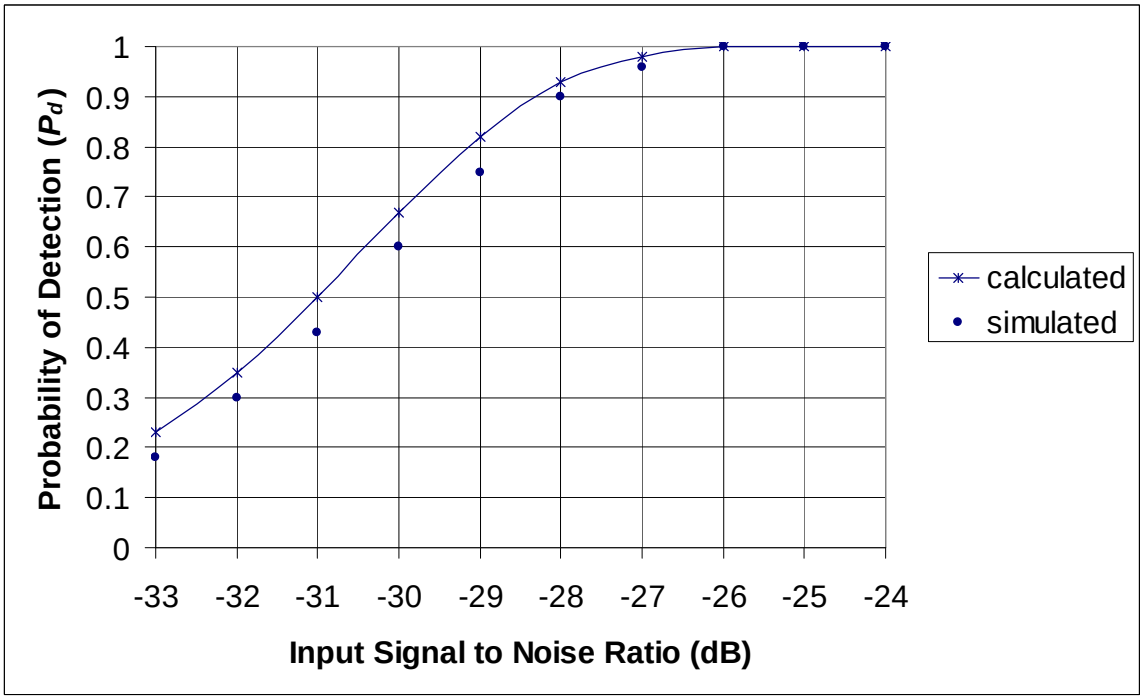

Figure 2.2/1: Computed and Simulated Probability of Detection.

It can be seen that the simulated performance is 0.5 dB worse than calculated. This is due to the filtering loss that was not included in the calculation.

Figure 2.2/2 is the output of the correlation. As can be seen, two noise values exceed the threshold. This is not unexpected as the mean number exceeding is:

$$n_P = 2L_C P_{fa} = 2.046. \tag{2.24}$$

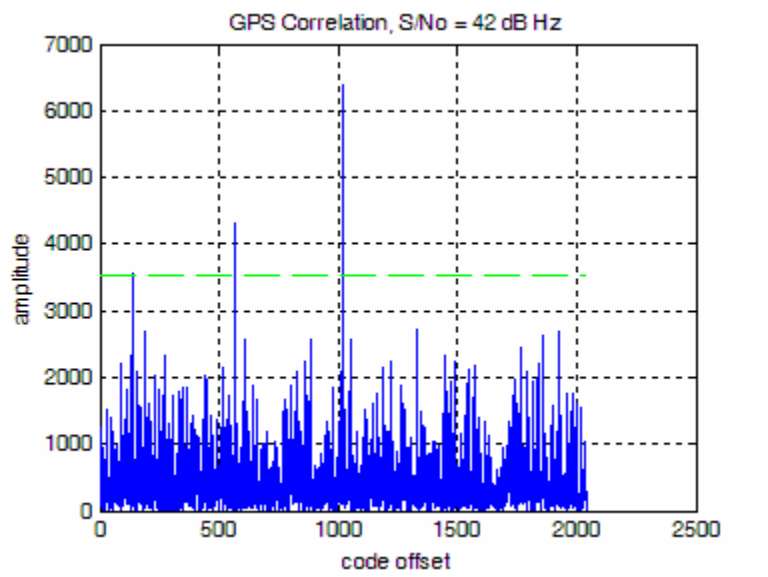

Figure 2.2/2: GPS L1 Correlation

Figure 2.2/3 shows the probability of values exceeding the threshold for various values of P_{fa}.

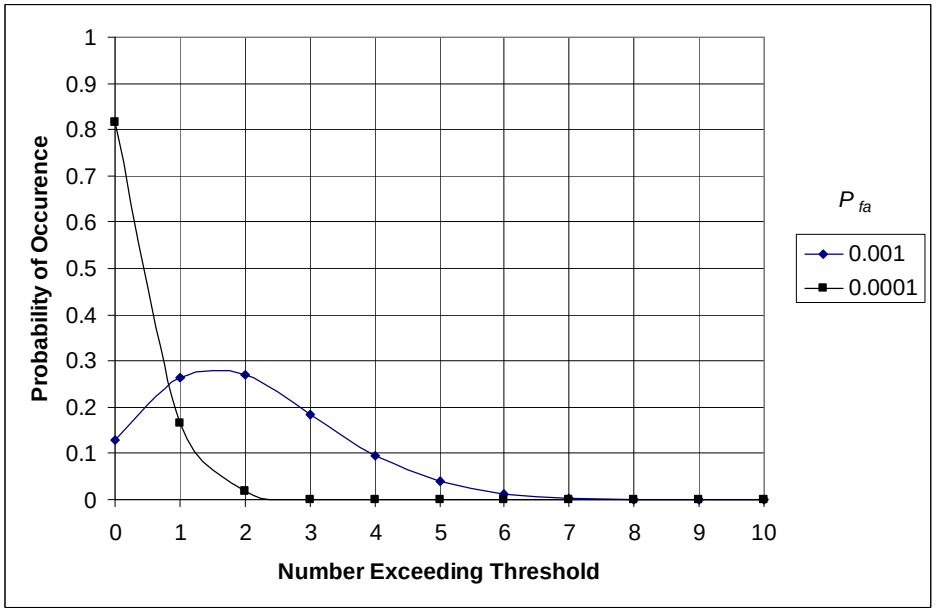

Figure 2.2/3: Distribution of Values Exceeding the Threshold vs. P_{fa}

2.2.1 Relationship between *C/N, BW, C/No and* L_C

The input signal to noise ratio is:

$$\left(\frac{S}{N}\right)_{IN} = \frac{A^2}{\sigma_N{}^2} = \frac{S}{NoBW} \tag{2.24}$$

The output signal to noise ratio, ignoring filtering losses, is:

$$\left(\frac{S}{N}\right)_{OUT} = \frac{n_S LA^2}{\sigma_N{}^2} = \frac{n_S LS}{NoBW} \tag{2.25}$$

n_S is the number of samples per chip, $n_S = 2$ for this work.

The bandwidth is a factor (F) times the chip frequency or:

$$BW = F\left(\frac{1}{T_C}\right) \tag{2.26}$$

Given that one complex sample is equivalent to two real-only samples, the Nyquist sampling rate is equal to the sampling rate and we conclude that $F=n_S$ in most cases. Following this argument:

$$\left(\frac{S}{N}\right)_{OUT} = \frac{S}{No}\frac{n_S L_C}{BW} = \frac{S}{No}\frac{n_S L_C}{n_S \diagup Tc} = \frac{S}{No} L_C T_C \tag{2.27}$$

The factor $L_C T_C$ is the code period and is 1/code rate.

Finally we conclude that the output signal to noise ratio is the *S/N* in a bandwidth equal to the code rate. Expressed in decibels:

$$\left(\frac{S}{N}\right)_{OUT} = \frac{S}{No} - 10\log_{10}\left(\frac{1}{L_C Tc}\right) = \frac{S}{No} + 10\log_{10}(L_C Tc) \tag{2.28}$$

2.3 EFFECTS OF CARRIER AND CODE DOPPLER

Referring to Section 2.2, there is a loss if the frequency of the carrier is not exactly known. We often refer to the frequency error (also known as frequency offset) as the Doppler error but it is partly due to the inaccuracy of the reference oscillator. The relative loss (i.e. reduced correlation gain) due to a frequency error (δF) is:

$$L_D = \frac{1}{L_C T_C}\int_0^{L_C T_C} \left(\cos(2\pi\delta F t) + j\sin(2\pi\delta F t)\right) dt \tag{2.29}$$

$$L_D = \frac{1}{L_C T_C}\int_0^{L_C T_C} \exp(j2\pi\delta F t)dt = \frac{1}{L_C T_C}\left[\frac{1}{2\pi\delta F}\exp(j2\pi\delta F t)\right]_0^{L_C T_C} \quad (2.30)$$

$$L_D = \frac{\sin(\pi\delta F L_C T_C)}{\pi\delta F L_C T_C}\exp(j\pi\delta F L_C T_C) \quad (2.31)$$

The length of the code is L_C chips and the chip time is T_C giving a code length of $L_C T_C$.

The loss then is determined by the well known $\sin(x)/x$ function. There is a phase shift but this does not affect the magnitude of the loss.

When the processing is digital and the integral is replaced by a summation. The exact expression of the loss is:

$$L_D = \frac{\sin(\pi\delta F L_C T_C)}{L_C \sin(\pi\delta F T_C)} \quad (2.32)$$

When L_C is large and $\delta F T_C$ is small the difference in the results is negligible and the linear function is used.

Figure 2.3/1 shows the loss in dB as a function of the frequency error expressed as a fraction of the code frequency. For example, if the code length is 1023 chips and the chip rate of 1.023 Mch/s then the code frequency is 1 kHz. A frequency error of 500 Hz results in a loss of 3.92 dB.

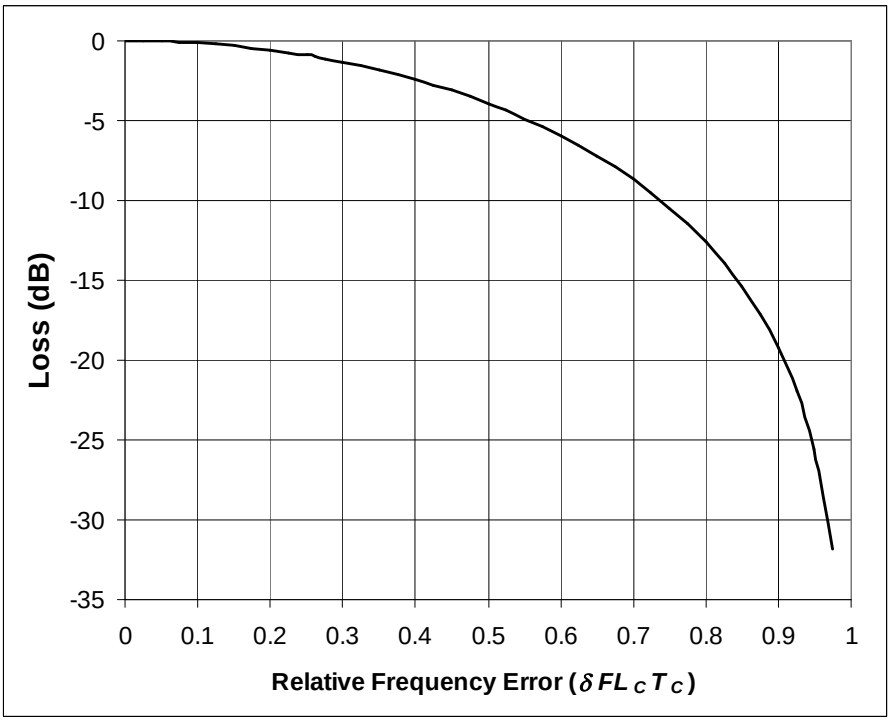

Figure 2.3/1: Loss due to Frequency Error

Figure 2.3/1 shows the loss as a function of the frequency offset assuming the signal frequency is exactly at that offset. However, when we search for the signal we can only bound the range of offsets. It is fair to assume that the signal frequency is equally likely to be anywhere in the range of possible offsets. Furthermore it is instructive to examine the affect of frequency offset on the probability of detection. Figure 2.3/2 shows the probability of detection at the edge and the mean as a function of the relative frequency offset. As can be seen, if the probability of detection with no offset is 0.95 then at the edge of a ±0.3 bin it is 0.65 but the mean over the range is 0.87. A relative frequency offset of 0.3 when correlating with a code length of 1 ms is an offset of 300 Hz. If the probability of detection with no offset is raised to 0.99, the mean over the 0.3 offset is 0.95. In the balance of this work, 0.3 will be used for the limit of relative frequency offset.

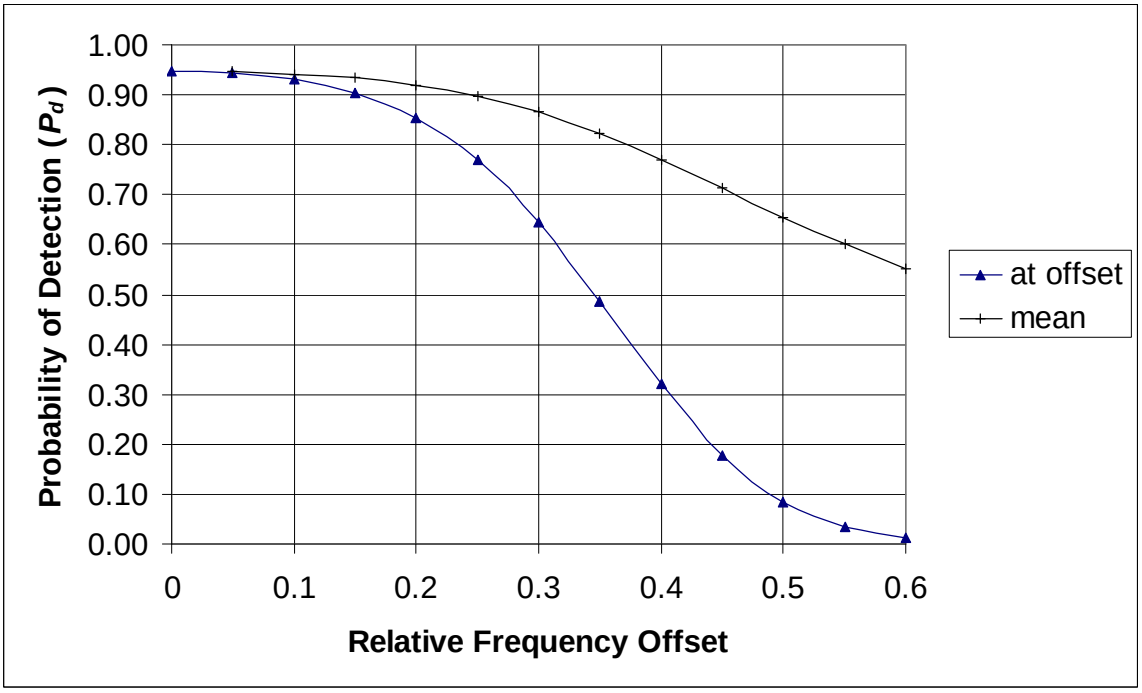

Figure 2.3/2: Probability of Detection vs. Relative Frequency Offset

Code Doppler is an error in the chip rate of the code. The relationship between the carrier frequency and the chip rate is exact as transmitted by the satellite. For GPS L1 C/A code and Galileo L1 the carrier frequency is 1540 times the chip rate. For GPS L5 the carrier frequency is 115 times the chip rate. The product of the code Doppler rate and the correlation time gives the error in chips. When the error is a substantial part of a chip, the correlation peak will be spread over more than one time offset. When correlating over a single code period, the code Doppler is negligible. However, we will see later, that when averaging over long periods, the code Doppler can be significant.

Figure 2.3/3 shows the frequency error tolerance as a function of integration time for 0.5 chip spread.

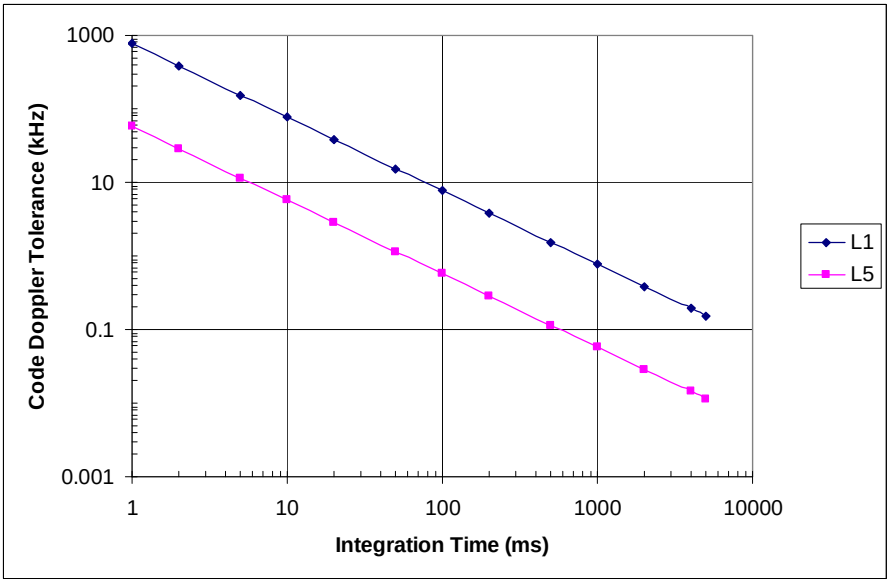

Figure 2.3/3: Code Doppler Tolerance vs. Integration Time

2.4 CORRELATION TECHNIQUES

2.4.1 Serial Acquisition

Although serial acquisition is too slow to meet our requirements, it is discussed here because the fundamental concepts are common to other techniques. Figure 2.4/1 is a typical correlator configuration.

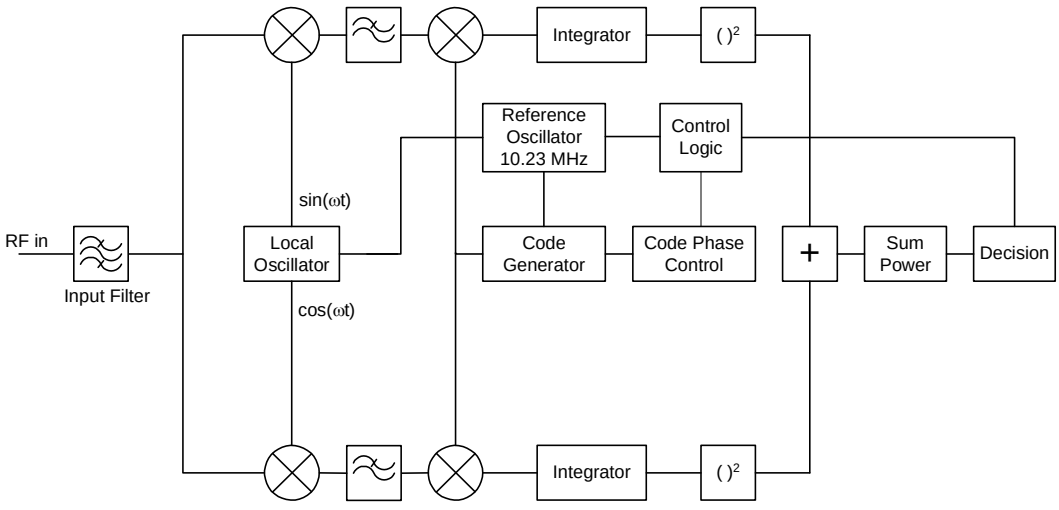

Figure 2.4/1: Basic Coherent Demodulator and Correlator

The diagram shows an analogue process. All modern systems will digitize the signal after the input filter but the process is the same. In the analysis of section 2.2, we assumed that the carrier phase and chip timing were known. Since the carrier phase is not known, both in phase and quadrature channels are processed in parallel. After integration over one code period the outputs are squared and added before further integration. Because the chip timing is not known the code phase is advanced in fractions of a chip (1/2 or 1/4).

Serial acquisition uses a single correlator and multiplies one sample at a time. Assuming the code phase is advanced by ½ chip at a time, this technique would require 2 x 1023 code periods or 2.046 seconds to test all 2046 code offsets. When further averaging is required, the time is multiplied by the number of averages. In the presence of low signal to noise ratios the time is significantly increased.

2.4.2 Parallel Acquisition

Parallel acquisition [36] uses more than one correlator in parallel to shorten the acquisition time accordingly. There are many ways parallel acquisition could be arranged. We could use one correlator for each satellite, or one for each Doppler offset. A common technique would be to have one correlator for each code chip so that one code phase can be tested in one chip time or in M times the chip time where M is the number averaged.

Any or all of the above can be combined.

2.4.3 Acquisition using a Matched Filter

A matched filter is a way of implementing parallel acquisition. A matched filter has a transfer function that is the complex conjugate of the signal to which it is matched. In this context the filter is matched to the spreading code. Figure 2.4/2 shows one implementation of the matched filter.

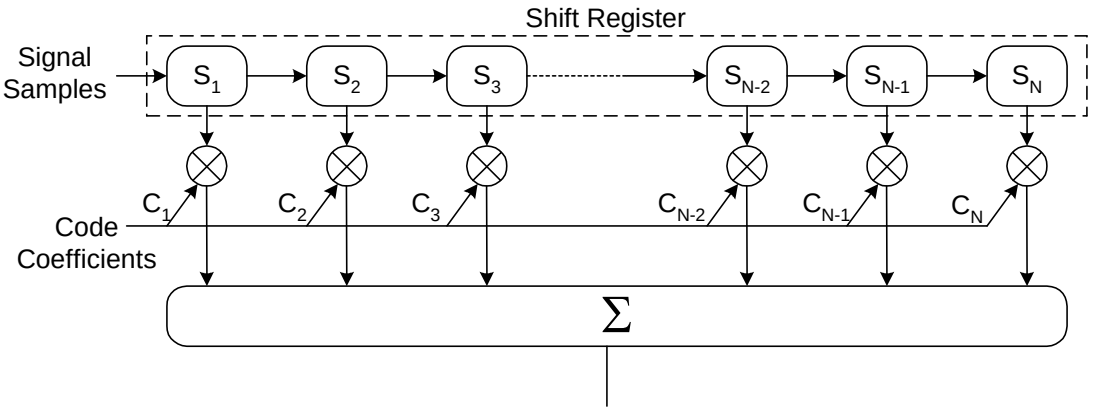

Figure 2.4/2: Matched Filter

If the shift register is as long as the code, correlation for one code phase can be performed in a single chip time. Compared to serial search, the correlation time is reduced by the code length (1023). If further averaging is required, the output is squared and added for M chip times. Figure 2.4/3 shows an I-Q matched filter with power averaging.

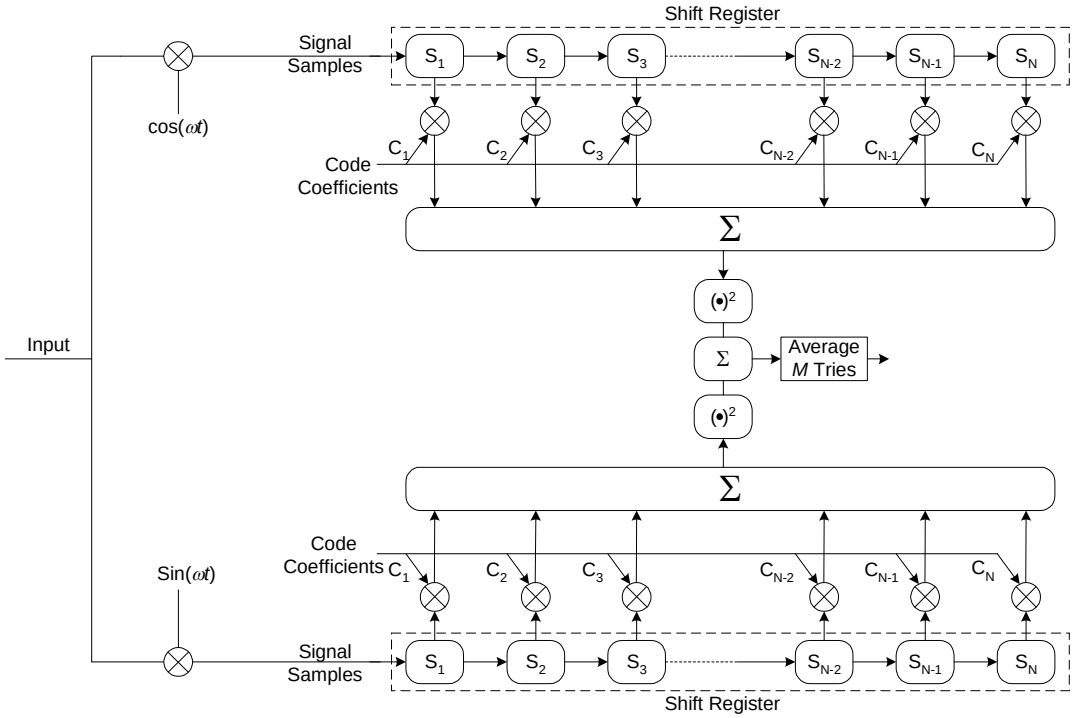

Figure 2.4/3: IQ Matched Filter with Power Averaging

The Doppler frequency offset immunity can be improved by segmenting the matched filter. In a segmented matched filter the products of the code and signal are summed

coherently for each segment. The power of each segment is summed. The probabilities of detection and false alarm for segmented correlation are discussed in section 2.5.6. The segmented matched filter is presented in [31].

The mean acquisition time with a matched filter of length 1023 is approximately 1/1023 of the mean acquisition time for serial acquisition. Using this technique, even with a signal to noise ratio as low as -40 dB, the acquisition time will be about 1.2 seconds. The exact result depends on the search and verification strategy.

The matched filter can be used with differential detection instead of power averaging. Correlating with a matched filter is computationally efficient but requires specialized hardware.

2.4.4 Correlation Using Fourier Transforms

If we have two sequences ***x*** and ***y*** of the same length, then the correlation theorem says:

$$\Im(x \otimes y) = \Im(x)\Im(y) \tag{2.33}$$

Using this theorem we can take the Fourier transform of the sequence of received samples X_k and the code samples C_k, multiply the transforms (element by element) then take the inverse transform to obtain the cross correlation. This process is computationally efficient compared to serial correlation. Another advantage of this process is special hardware is not required.

2.5 AVERAGING TECHNIQUES

Four techniques are discussed.

- Coherent averaging
- Power averaging
- Differential Averaging
- Averaging using the FFT

Following the discussion of the various techniques, a comparison of the techniques is presented.

2.5.1 Coherent Averaging

The output of the correlator or matched filter may be averaged to give better performance for weak signals.

The input to the averaging process is a complex combination of signal and noise:

$$v_i = A_L\left(\cos(\theta) + j\sin(\theta)\right) + \eta_I + j\eta_Q \tag{2.34}$$

The terms η_I and η_Q are noise voltages with distribution $\mathrm{N}(0,\sigma_L) = \mathrm{N}\left(0,\frac{\sigma_{NL}}{\sqrt{2}}\right)$ where σ_{NL} is the standard deviation of the total (in-phase plus quadrature) noise at the correlator output: $\sigma_{NL} = \sqrt{2}\sigma_L$. (2.35)

The signal to noise ratio at the input to the averaging process is $\frac{S}{N} = \frac{A_L{}^2}{\sigma_{NL}{}^2}$ (2.36)

where A_L refers to the signal amplitude at the output of the basic correlator.

Two states are considered, H_0 when the codes are not aligned and H_1 when the codes are aligned.

In state H_0 the value of A is small and the noise is dominant.

In coherent averaging we sum the voltages

$$v_M = \sum_{i=1}^{M} v_i = MA_L\left(\cos(\theta) + j\sin(\theta)\right) + \sum_{i=1}^{M}\left(\eta_{Ii} + j\eta_{Qi}\right) \tag{2.37}$$

assuming A_L is constant for the duration of the averaging.

The sum of each of the noise terms is Gaussian with zero mean and distribution

$$\mathrm{N}(0,\sigma_M) = \mathrm{N}\left(0,\sqrt{M}\sigma_L\right) \tag{2.38}$$

In this analysis we set $\sigma_{NL} = 1$ then $\sigma_M = \sqrt{\frac{M}{2}}$ (2.39)

To evaluate the probabilities we take the square of the magnitude.

In state H_0, the sum of the squares of the real and imaginary noise voltages has a chi-squared distribution with 2 degrees of freedom that is negative exponential.

$$f(x) = \frac{1}{2\sigma_M{}^2}\exp\left(\frac{-x}{2\sigma_M{}^2}\right) = \frac{1}{2M\sigma_L{}^2}\exp\left(-\frac{x}{2M\sigma_L{}^2}\right) \tag{2.40}$$

The probability of false alarm is determined as $P_{fa} = \exp\left(-\frac{Th}{2M\sigma_L{}^2}\right)$ (2.41)

Solving for the threshold we obtain

$$Th = -\ln(P_{fa})2M\sigma_L{}^2 \quad (2.42)$$

In state H_1 the distribution of the square of the magnitude is non-central chi squared with 2 degrees of freedom and non centrality factor:

$$\lambda = M\frac{A_L{}^2}{\sigma_L{}^2} = 2M\frac{S}{N} \quad (2.43)$$

The probability of detection can then be computed using the Marcum Q function

$$P_d = Q_M(\alpha, \beta) \quad (2.44)$$

where $\alpha = \sqrt{\lambda}$ and $\beta = \sqrt{\frac{Th}{M\sigma_L{}^2}}$ (2.45)

In state H_1 the normalized mean of the output is $\mu_{CA} = 2 + 2M\frac{S}{N}$ (2.46)

and the variance is $V_{CA} = 4 + 8M\frac{S}{N}$. (2.47)

In state Ho the normalized mean of the output is $\mu_o = \frac{2M\sigma_L{}^2}{M\sigma_L{}^2} = 2.$ (2.48)

The normalized variance is 4.

We now consider the case where the input is the output of the basic correlator.

Note that the output of the basic correlator, Equation (2.8), is:

$$R(\tau_k) = AK(\tau_k)d\frac{\sin(\pi\delta F L_C T_C)}{L_C\sin(\pi\delta F)}\exp(j\theta) + \sum_1^{2L}\left(\eta_I(k) + j\eta_Q(k)\right)$$

For coherent averaging the output of the correlator is summed for *M* tries and the result is determined from the sum. After summing *M* tries the result is:

$$R_M(\tau_k) = MAK(\tau_k)\frac{\sin(\pi\delta F L_C M T_C)}{L_C M\sin(\pi\delta F)}\exp(j\theta)\sum_{m=1}^{M} d(m) + \sum_{m=1}^{M}(\eta_{\mathrm{Im}} + j\eta_{Qm}) \quad (2.49)$$

The distribution of each of the sums of noise values is

$$\mathrm{N}(0, \sqrt{M}\sigma_L) = \mathrm{N}(0, \sqrt{L_C M}\sigma_N). \quad (2.50)$$

Consider the case where δF is small and the data value (d) is constant for the integration period. When the codes are not aligned, $K(\tau_k)$ is small and the noise is dominant. The noise power is

$$N_M = MN_L = M(2L_C\sigma_N{}^2) = 2ML_CN\,. \tag{2.51}$$

When the codes are aligned, the signal at the output of the correlator is $S_L = (2L_CA)^2$ In this case the signal power after averaging is $S = (2ML_CA)^2$. From this result and equation 2.51 we see that the signal power has increased by M^2 and the noise power has increased by M for a signal to noise ratio improvement equal to M.

There are two other factors to consider, first, since the correlation is now over a period of ML_CT_C instead of L_CT_C the effect of frequency offset is M times worse. The result is a significantly reduced tolerance to frequency offset.

The second factor to consider is the data value $d(m)$. For coherent integration to be successful, measures must be taken to ensure that the data value is constant for the length of the integration [28].

After averaging and squaring, the distributions are the same except for the factor M. The distribution of noise amplitudes is:

$$f(x) = \frac{1}{2\sigma^2}\exp\left(-\frac{x}{2\sigma^2}\right) = \frac{1}{2M\sigma_L{}^2}\exp(-\frac{x}{2M\sigma_L{}^2}) \tag{2.52}$$

The distribution of aligned signal amplitudes is non-central chi-squared with 2 degrees of freedom and the non centrality factor:

$$\lambda = \frac{4ML_CA^2}{\sigma_N{}^2} = 4ML_C\left(\frac{S}{N}\right) \tag{2.53}$$

where $\frac{S}{N}$ is the signal to noise ratio at the input to the correlator.

Figure 2.5/1 shows the non-aligned and aligned distributions for M = 100 and signal to noise ratio at the input to the averaging process (i.e. signal to noise ratio at the output of

the correlator) equal to -7 dB. With an improvement of 20 dB ($10\log_{10}(M)$) due to averaging, the resultant signal to noise ratio is 13 dB.

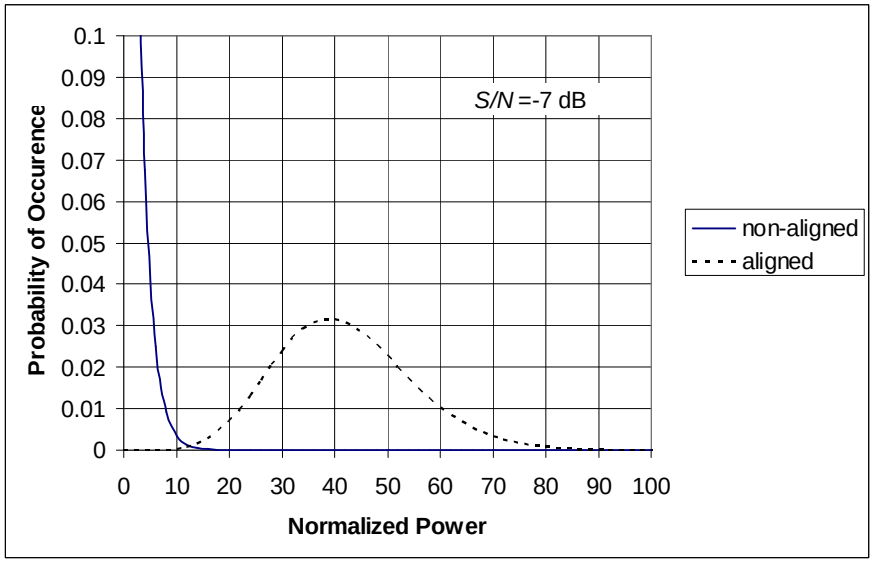

Figure 2.5/1: Non-aligned and Aligned Distributions for Coherent Averaging

The probability of detection is then determined as:

$$P_d = Q_M(\alpha, \beta) \tag{2.54}$$

The term $\alpha = \sqrt{\lambda}$ and $\beta = \sqrt{\dfrac{Th}{2LM\sigma_L{}^2}} = \sqrt{\dfrac{Th}{LM}}\dfrac{1}{\sigma_N}$. (2.55)

Using the same example as for basic correlation except for the addition of coherent averaging over 100 tries, the sensitivity is improved by 20 dB. Figure 2.5/2 shows the correlation with an *S/N* at the input to the receiver equal to -47 dB. The bandwidth is 8.184 MHz giving an $S/No = S/N + 10\log_{10}\left(8.184\times10^6\right) = -47 + 69 = 22$ dB Hz.

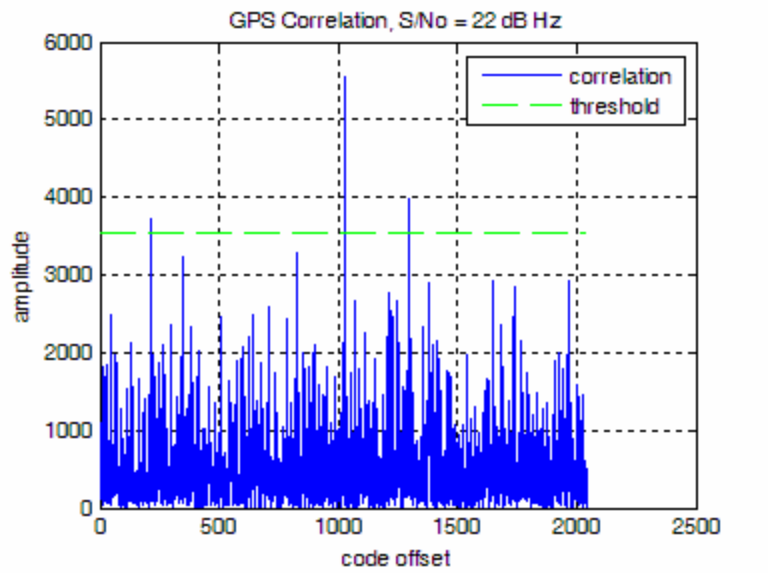

Figure 2.5/2: Correlation with averaging of 100

Figure 2.5/3 shows the computed and simulated probabilities of detection as a function of signal to noise ratio.

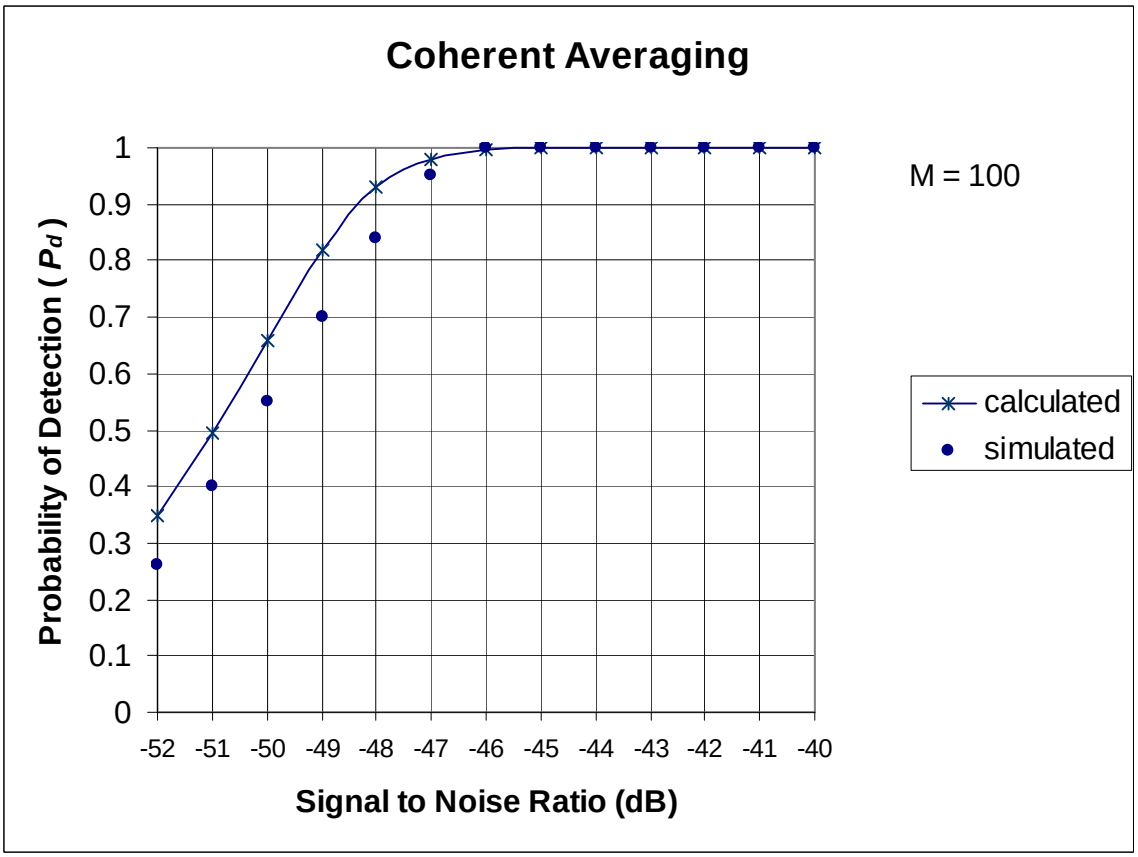

Figure 2.5/3: Probability of Detection vs. Signal to Noise Ratio for Coherent Averaging

2.5.2 Power Averaging

Power averaging is achieved by squaring (multiplying by the complex conjugate) before summing.

The input to the averaging process is a complex combination of signal and noise, Equation (2.34):

$$v_i = A_{Li}\left(\cos(\theta_i) + j\sin(\theta_i)\right) + \eta_{Ii} + j\eta_{Qi} \tag{2.56}$$

The terms η_I and η_Q are noise voltages with distribution $\mathrm{N}(0,\sigma_L) = \mathrm{N}\left(0,\frac{\sigma_{NL}}{\sqrt{2}}\right)$ (2.57)

The signal to noise ratio, Equation (2.36), is $\frac{S}{N} = \frac{{A_L}^2}{{\sigma_{NL}}^2}$ (2.58)

After squaring, the output of the k^{th} offset is of the form:

$$p_k = {A_L}^2\left(\cos^2(\theta) + \sin^2(\theta)\right) + 2A_L\cos(\theta)\eta_{IL} + 2A_L\sin(\theta)\eta_{QL} + {\eta_{IL}}^2 + {\eta_{QL}}^2 \tag{2.59}$$

When the codes are not aligned the value of A is small and the sum of M values has a chi-squared distribution with $2M$ degrees of freedom. The normalized mean is $\mu_{PN} = 2M$ and the variance is $V_{PN} = 4M$.

The normalized mean is obtained by dividing the mean by σ^2 and the normalized variance by dividing by σ^4.

When the codes are aligned, the sum of M values has a non-central chi-squared distribution with $2M$ degrees of freedom and $\lambda = \frac{M{A_L}^2}{{\sigma_L}^2} = 2M\frac{S}{N}$. (2.60)

The normalized mean is $\mu_{PA} = 2M + \lambda = 2M + 2M\frac{S}{N}$ (2.61)

and the normalized variance is $V_{PA} = 4M + 8M\frac{S}{N}$. (2.62)

Figure 2.5/4 shows these distributions for the same conditions as Figure 2.5/1. In this Figure, both distributions are shifted to put the mean of the non-aligned distribution at zero. Note the large overlap between the aligned and non-aligned distributions.

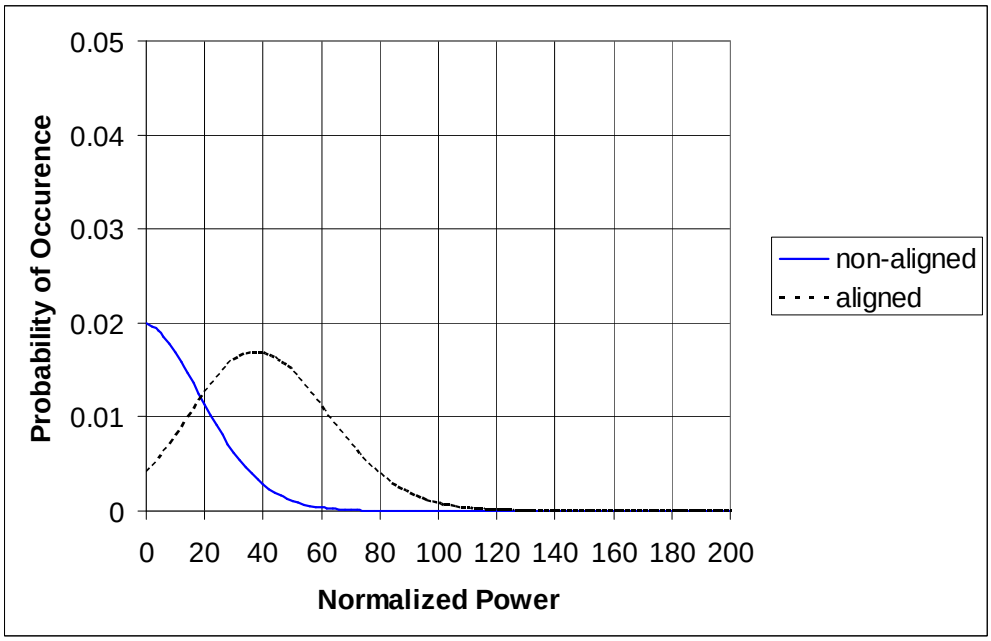

Figure 2.5/4: Non-aligned and Aligned Distributions for Power Averaging with *S/N* = -7 dB

To obtain the same probability of detection the signal to noise ratio has to be raised by 5 dB. Figure 2.5/5 shows the distributions with an input *S/N*=-2 dB.

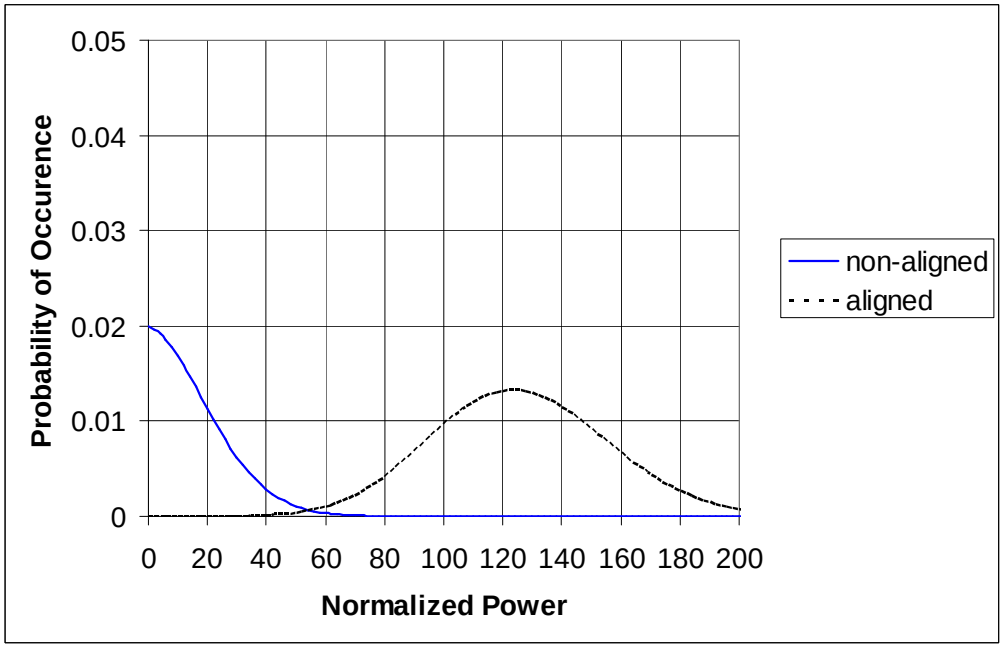

Figure 2.5/5: Non-aligned and Aligned Distributions for Power Averaging with *S/N*=-2 dB

Consider the same correlator as in the example above followed by power averaging for 100 times. In this example M = 100 and the number of degrees of freedom k = 200. When not aligned, the output has a chi-squared distribution with 200 degrees of freedom. The normalized threshold is computed as the value for which the integral of the chi-squared distribution from that value to infinity is equal to the desired probability. In this work I used the inverse cumulative distribution in EXCEL (CHIINV(P_{fa},$2M$)).
For P_{fa} = .001 is and M = 100, Th_N = 267.45. The actual threshold is then

$$Th = Th_N \sigma^2 = Th_N L_C \sigma_{NL}{}^2 = 68400$$

When aligned, the distribution of the output is non central chi-squared with non centrality factor $\lambda = 4ML_C \frac{A^2}{\sigma_{NL}{}^2}$. The Marcum Q function, $Q_M(\alpha, \beta)$ was used to evaluate the probability of detection for various signal to noise ratios. First, $\beta = \sqrt{Th_N} = 16.35$.
Secondly, $\alpha = \sqrt{\lambda}$ was calculated for various signal to noise ratios. Note that the input signal to noise ratio is determined in the input bandwidth that is 8.184 MHz (the sampling rate in the simulation). The $S/No = S/N + 69$ dB. This situation was also simulated. The results with M=100 are shown by Figure 2.5/6. It can be seen that the simulated results are about 0.5 dB worse than the calculated results. This as before is due to the filtering and sampling loss.

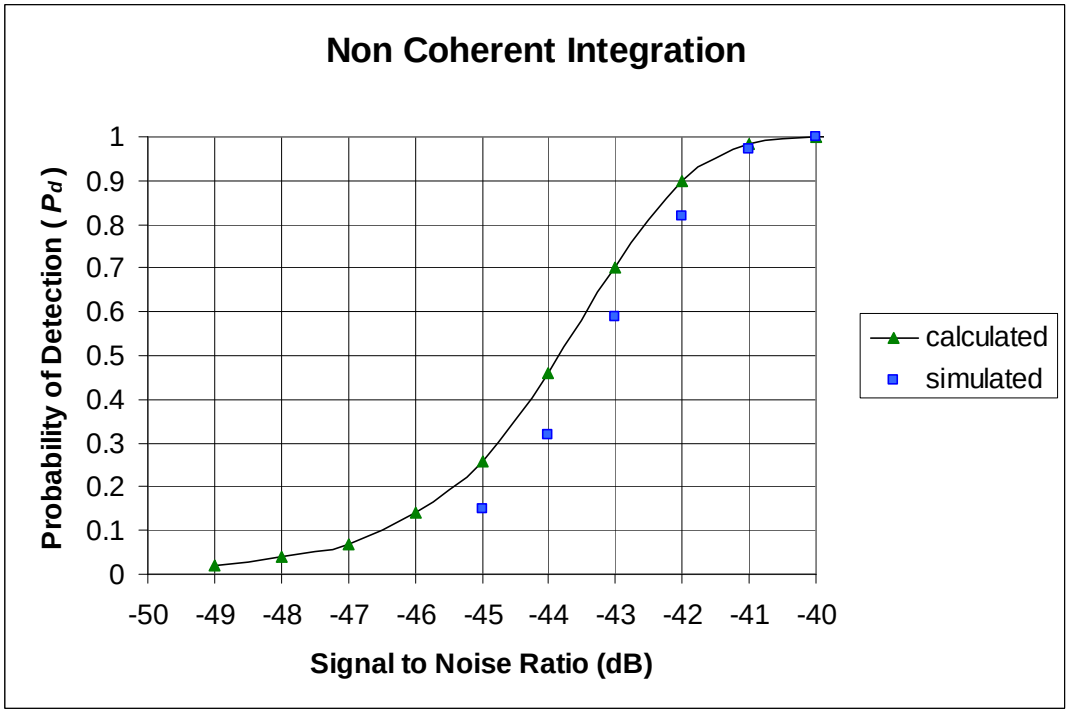

Figure 2.5/6 Probability of Detection vs. Signal to Noise Ratio for Power Averaging

Comparing to Figure 2.5/3 we see a loss for power averaging of about 5 dB.

2.5.3 Differential Detection

The value of the kth offset of the ith correlation at the input to the averaging can be represented as:

$$v_k(i) = A_{Lk}\{\cos(\theta + i\delta) + j\sin(\theta + i\delta)\} + \eta_I(i) + j\eta_Q(i) \qquad (2.63)$$

In this equation, A_L is the amplitude of the signal component, θ is the phase offset and δ is the phase change from one correlation to the next. It is assumed that the amplitude (A_L) is constant from one correlation to the next. The noise terms η_I and η_Q are Gaussian random variables with distribution $\mathrm{N}\left(0, \frac{\sigma_{NL}}{\sqrt{2}}\right)$ and σ_{NL} is the square root of the noise power at the output of the correlator (N_L).

$$\delta = 2\pi\delta F L_C T_C \qquad (2.64)$$

Where δF is the frequency offset and $L_C T_C$ is the code period.

In differential averaging [33] the result is the sum of the value of the k^{th} offset of the i^{th} correlation multiplied by the conjugate of the kth offset of the $(i\text{-}1)^{th}$ correlation.

$$P_k = \sum_{i=1}^{M} v_k(i) v_k(i-1)^* \tag{2.65}$$

Re-arranging and dropping the subscript k we obtain:

$$v(i) = A_L \cos(\theta + i\delta) + \eta_I(i) + j\{A_L \sin(\theta + i\delta) + \eta_Q(i)\} \text{ and} \tag{2.66}$$

$$v(i-1)^* = A_L \cos(\theta + (i-1)\delta) + \eta_I(i-1) - j\{A_L \sin(\theta + (i-1)\delta) + \eta_Q(i-1)\} \tag{2.67}$$

For the values of *k* where the codes are not aligned, *A* is small compared to the noise values and is assumed to be zero. In this case the sum of *M* products is:

$$P = \sum_{i=1}^{M} \{\eta_I(i-1)\eta_I(i) + \eta_Q(i)\eta_Q(i-1) + j\eta_I(i-1)\eta_Q(i) - j\eta_I(i)\eta_Q(i-1)\} \tag{2.68}$$

The minus sign does not change the statistics of the noise products simply rotating the result by pi radians.

With $M = 1$, the sum of each real or imaginary pairs of products has the distribution

$$p(x) = \frac{1}{2\sigma_L^2} \exp\left(-abs\left(\frac{x}{\sigma_L^2}\right)\right) \tag{2.69}$$

The sum of *M* terms approaches a Gaussian distribution as *M* gets large. Considering *p(x)* is symmetrical *M* does not need to be very large. Figure 2.5/7 shows the result for $M = 10$. The distribution of the real and imaginary sums approaches

$$\mathrm{N}\left(0, \sqrt{M}\,\frac{\sigma_{NL}^2}{\sqrt{2}}\right). \tag{2.70}$$

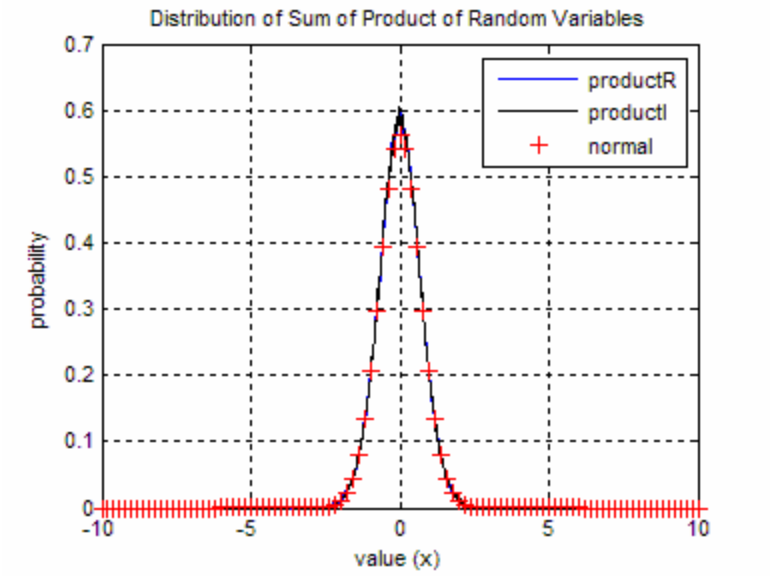

Figure 2.5/7: Distribution of Sum of Product of Random Variables

The result of the sum is circular complex Gaussian noise.

When the codes are aligned the value of A_L becomes significant. In this case the trigonometry becomes complex. To get an understanding of the process we take a simpler case. If we set θ and $\delta = 0$ the equations become:

$$v(i) = A_L + \eta_I(i) + j\eta_Q(i) \text{ and} \tag{2.71}$$

$$v(i-1)^* = A_L + \eta_I(i-1) - j\eta_Q(i-1)\,. \tag{2.72}$$

The product gives:

$$\begin{aligned} v(i)v(i-1)^* = A_L{}^2 + A_L\{\eta_I(i) + \eta_I(i-1) + j(\eta_Q(i) - \eta_Q(i-1))\} + \\ \eta_I(i-1)\eta_I(i) + \eta_Q(i)\eta_Q(i-1) + j\eta_I(i-1)\eta_Q(i) - j\eta_I(i)\eta_Q(i-1) \end{aligned} \tag{2.73}$$

The sum of the terms in A_L gives the following result:

$$A_L\begin{Bmatrix} \eta_I(1) + \eta_I(0) + j(\eta_Q(1) - \eta_Q(0)) + \\ \eta_I(2) + \eta_I(1) + j(\eta_Q(2) - \eta_Q(1)) + \\ \eta_I(3) + \eta_I(2) + j(\eta_Q(3) - \eta_Q(2)) + \\ \bullet\bullet\bullet \\ \eta_I(M-1) + \eta_I(M-2) + j(\eta_Q(M-1) - \eta_Q(M-2)) + \\ \eta_I(M) + \eta_I(M-1) + j(\eta_Q(M) - \eta_Q(M-1)) \end{Bmatrix} \tag{2.74}$$

After summing like terms we get:

$$A_L\left\{\eta_I(0)+\eta_I(M)+\sum_{i=1}^{M-1}2\eta_I(i)+j\left(\eta_Q(0)+\eta_Q(M)\right)\right\} \tag{2.75}$$

The variance (V) of the real component is:

$$V_I = 2A_L{}^2 * \left(\frac{\sigma_N}{\sqrt{2}}\right)^2 + A_L{}^2(M-1)\left(\frac{2\sigma_N}{\sqrt{2}}\right)^2 = A_L{}^2(2M-1)\sigma_{NL}{}^2 \tag{2.76}$$

The variance of the imaginary component is

$$V_Q = A_L{}^2\sigma_{NL}{}^2 \tag{2.77}$$

The variance of the noise product terms is the same as for the non-aligned case.

The total variance of the real component is then:

$$V_R = \sigma_x{}^2 = (2M-1)A_L{}^2\sigma_{NL}{}^2 + M\sigma_{NL}{}^4 \tag{2.78}$$

The variance of the imaginary component is:

$$V_I = \sigma_y{}^2 = A_L{}^2\sigma_{NL}{}^2 + M\sigma_{NL}{}^4 \tag{2.79}$$

We see that when not aligned, the distribution of output values is complex circular Gaussian and when aligned is complex Gaussian but not circular.

Finally the mean value when aligned is $A_L{}^2$.

The above analysis was for the simple case where θ and δ are zero. It can be shown that the value of θ does not change the result as we are comparing the result to the previous result. However, the change in angle, δ, caused by a frequency offset has a twofold effect. Since δ is due to a frequency error, δF, it has the sin(x)/x effect on the amplitude. Of more interest here, the rate of change of phase, δ, rotates the output in the complex plane.

Figure 2.5/8 is a scatter plot showing the locus of output values in the complex plain. In this plot, the value of δ is 0. The circular scatter, cetered on zero, represents the non aligned case, while the oval scatter represents the aligned case.

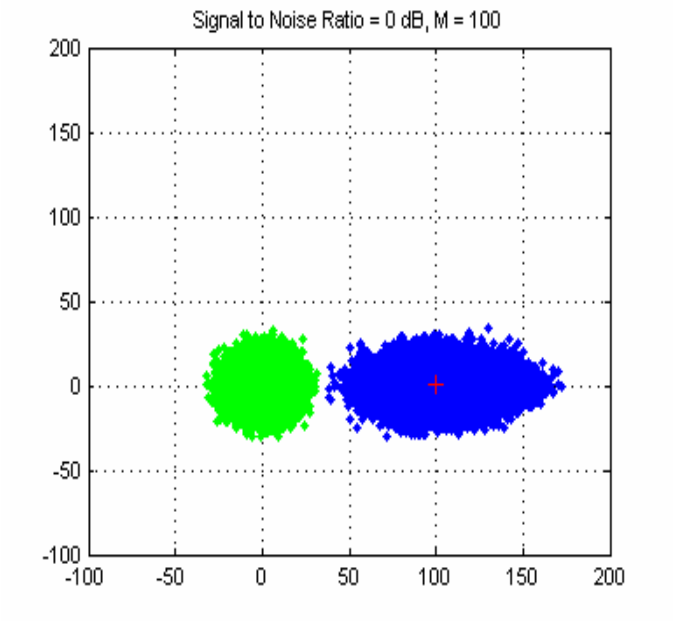

Figure 2.5/8; Scatter Plot with $\delta = 0$

The next plot, Figure 2.5/9, shows the same conditions except $\delta = \pi/4$. Note that the plot is the same except it has been rotated. The real axis from the previous plot is now at 45 degrees.

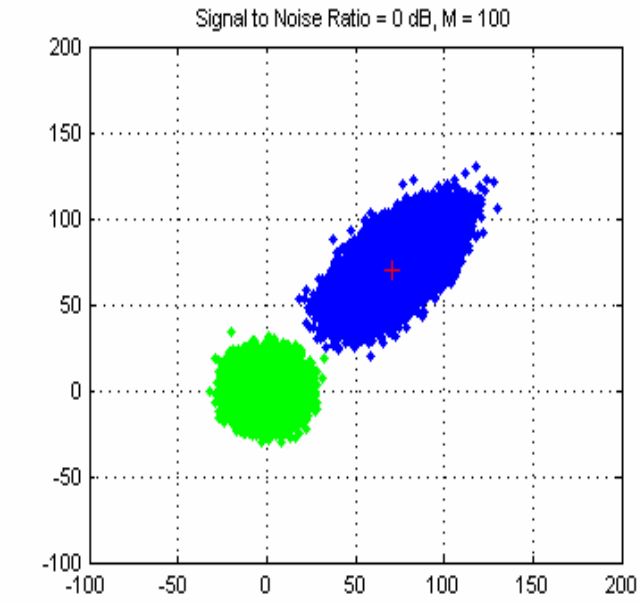

Figure 2.5/9; Scatter Plot with $\delta = \pi/4$

As with basic correlation a decision variable has to be chosen. We could use the square of the magnitude but the dimensions would be power squared whereas the decision variable for the other techniques is power. For this reason I have chosen to use the absolute value or the magnitude.

The magnitude of the complex Gaussian distribution from the non-aligned case has a Rayleigh distribution. The probability of false alarm (P_{fa}) is determined from the tail of the Rayleigh distribution.

$$P_{fa} = \exp\left(-\frac{Th^2}{2\sigma^2}\right) \tag{2.80}$$

If the threshold is set to 3.7σ, the probability of false alarm is .001. The Rayleigh assumption is valid for large M. However for small values it is not a good fit. Figure 2.5/10 shows the distribution with M=2.

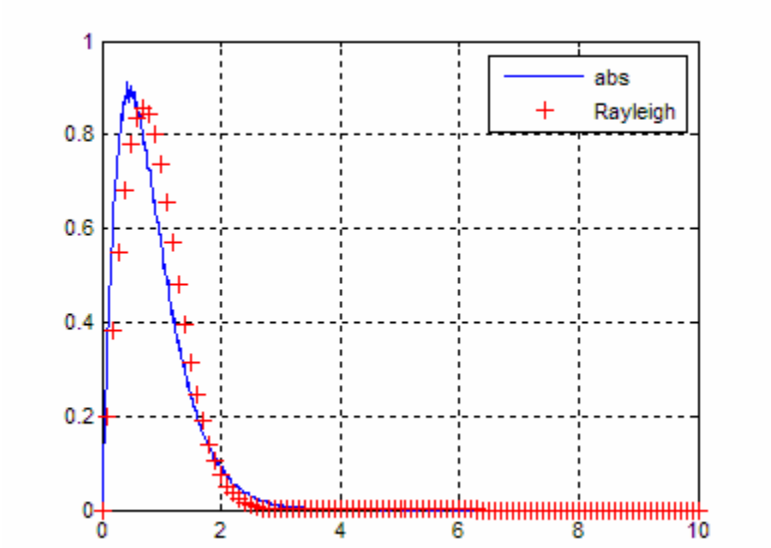

Figure 2.5/10: Distribution of Magnitude of Noise with M=2

When aligned the distribution of the magnitude is complex Normal with mean $\mu = MA^2$ as follows:

$$p(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y}\exp\left(-\frac{1}{2}\left(\frac{(x-\mu_X)^2}{\sigma_X{}^2} + \frac{y^2}{\sigma_Y{}^2}\right)\right) \tag{2.81}$$

In this equation x represents the radial direction and y the tangential direction.

The absolute value is the distance from the origin, $d = \sqrt{x^2 + y^2}$. (2.82)

The mean distance is slightly longer than the mean value of x. The distribution of the distance can be approximated by a normal distribution with $\sigma = \sigma_X$ and

$\mu = \sqrt{\left(MA^2\right)^2 + {\sigma_X}^2}$. (2.83)

This is a reasonable fit for cases where the probability of detection is high. For lower values, the distribution becomes skewed and diverges from normal.

Figures 2.5/11 and 2.5/12 show 2 examples of this fit. These graphs also show the Rayleigh distribution for the non aligned case. These examples were chosen as they represent cases where the aligned and non aligned distributions have very little overlap giving a low value of P_{fa}.

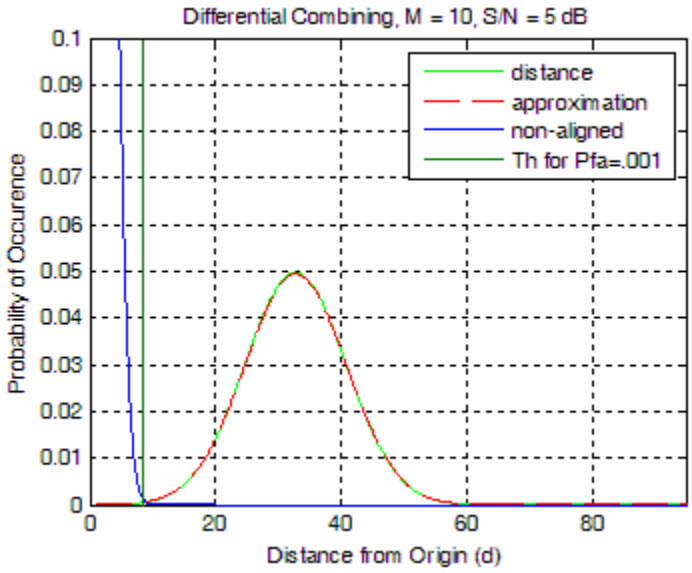

Figure 2.5/11: Distribution with *M*=10

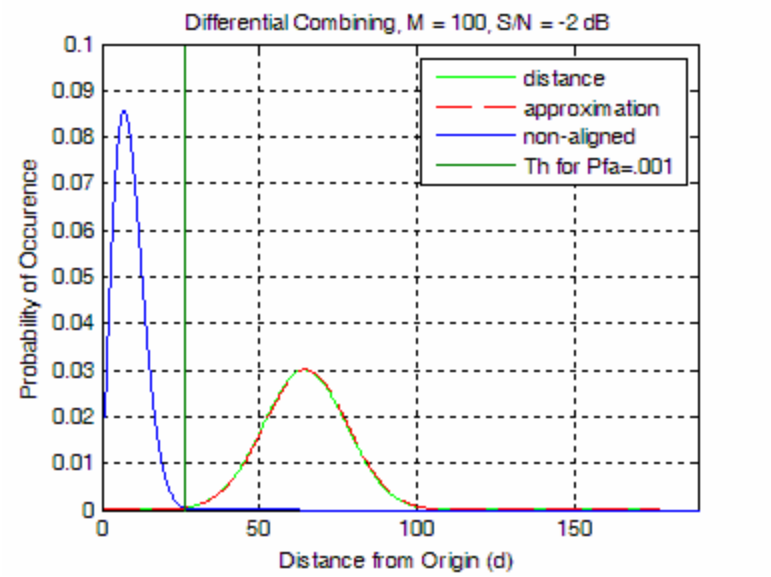

Figure 2.5/12: Distribution with *M*=100

Consider the same example as used for coherent and power averaging.

Note that the output of the basic correlator, Equation (2.8), is:

$$R(\tau_k) = AK(\tau_k)d\,\frac{\sin(\pi\delta F L_C T_C)}{L_C \sin(\pi\delta F T_C)}\exp(j\theta) + \sum_{1}^{2L_C}\left(\eta_I(k) + j\eta_Q(k)\right)$$

The distribution of each of the sums of noise values is $\mathrm{N}\left(0, \sqrt{L_C}\sigma_N\right)$ and the variance of the noise is ¼ that is $\sigma_N = \frac{1}{2}$ after filtering. As with the other examples, we assume the data value (d) is constant and $K(\tau_k) = 2L_C$ that is the maximum value. We also assume δF is small. In this example M = 100.

First we consider the non-aligned case. The total noise power is $N_L = 2L_C{\sigma_N}^2$ and

$$\sigma_{NL} = \sqrt{2L_C}\sigma_N = \sqrt{2046}\,\frac{1}{2} = 22.6$$

After summing over M values the standard deviation of the in-phase and quadrature values is $\sigma_d = \frac{\sqrt{M}{\sigma_{NL}}^2}{\sqrt{2}} = 3617$

The probability of false alarm for a Rayleigh distribution is

$$P_{fa} = \exp(\frac{-x^2}{2\sigma^2}) \tag{2.84}$$

Solving for x with P_{fa} = .001 we get Th = 13444.

At the input to the averaging process, the amplitude of the aligned signal is

$$A_L = \sqrt{S/N}\, 2L_C \tag{2.85}$$

Using these values and assuming the normal distribution described above, the probability of detection was computed as $P_d = Q\left(\frac{(\mu - Th)}{\sigma_X}\right)$ (2.86)

where μ, σ_x and σ_y are as defined by Equations (2.78), 2.79) and (2.83) .

$$\mu = \sqrt{\left(MA_L^{\,2}\right)^2 + \sigma_X^{\,2}}$$

$$\sigma_X = \sqrt{(2M-1)A_L^{\,2}\sigma_{NL}^{\,2} + M\sigma_{NL}^{\,4}}$$

$$\sigma_Y = \sqrt{A_L^{\,2}\sigma_{NL}^{\,2} + M\sigma_{NL}^{\,4}}$$

The probability of detection was calculated for signal to noise ratios of -40 to -49 dB. The results are shown by Figure 2.5/13 along with the results of the simulation. The simulated results are about 0.5 dB worse due to the filtering and sampling loss.

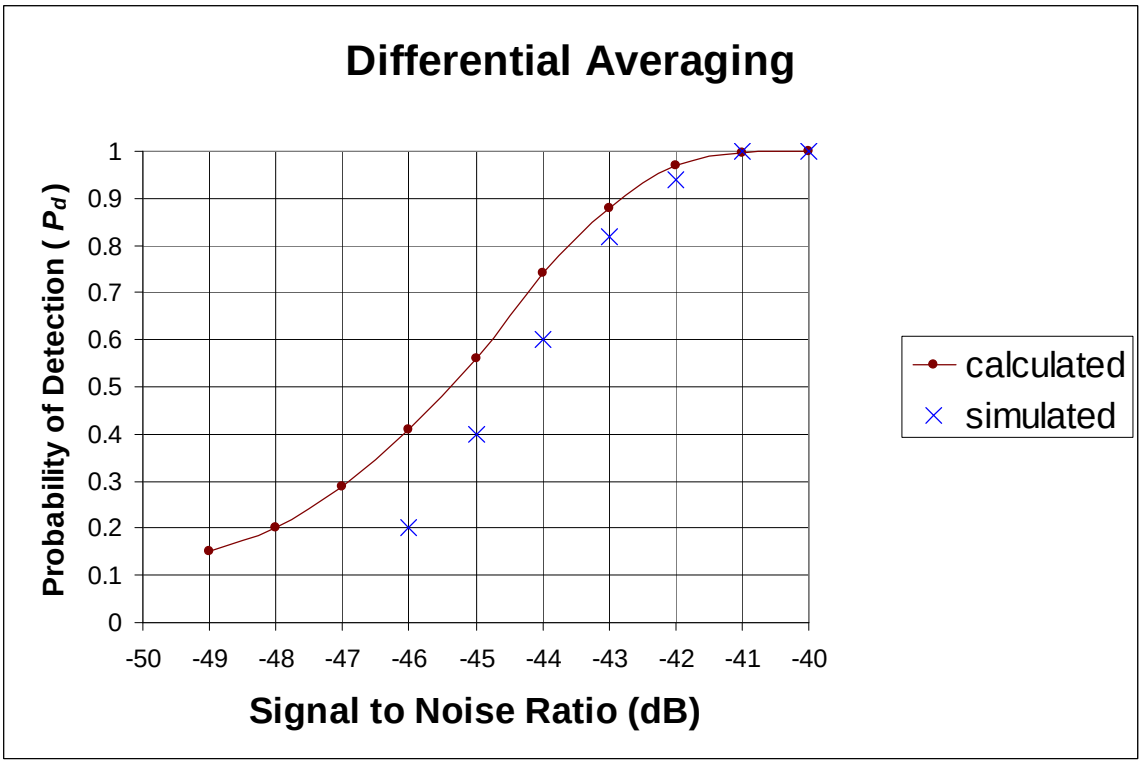

Figure 2.5/13: Probability of Detection vs. Signal to Noise Ratio for Differential Averaging

2.5.4 Averaging with the aid of Fourier Transforms

The procedure described here [30], [34] is a variation of coherent averaging that has a larger tolerance to Doppler shift.

After long coherent averaging the tolerance to frequency offset is small. However, multiple frequency bins can be created spanning a much larger frequency range.

Consider a code of length L_C. The output of the correlation is a vector of L_C code offsets. Instead of summing the vectors, a matrix of M vectors can be constructed. This matrix has L_C code offsets by M correlations. If the FFT of each column is computed, a new matrix is generated with L_C code offsets by M Doppler frequencies as shown by Figure 2.5/14.

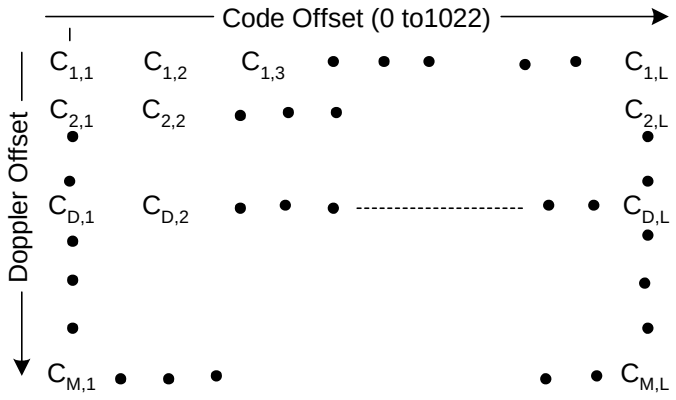

Figure 2.5/14: Matrix of Doppler Offset vs. Code Offset

Consider the correlation output to be a sinusoidal signal with AWGN in the aligned bin and noise only in the other bins. The frequency response of a FFT bin is

$$H(\omega) = \exp(j\frac{\omega}{2}(M-1))\frac{\sin(\omega M/2)}{\sin(\omega/2)} \tag{2.87}$$

The frequency ω is the frequency offset from the center of the bin measured in radians per sample.

$$\omega = 2\pi(f - f_B)Tc \tag{2.88}$$

$$f_B = \frac{m-1}{MT_C} \tag{2.89}$$

The bin frequency (f_B) is the center frequency of the m^{th} bin. The relative response of the first and second bins is shown by Figure 2.5/15.

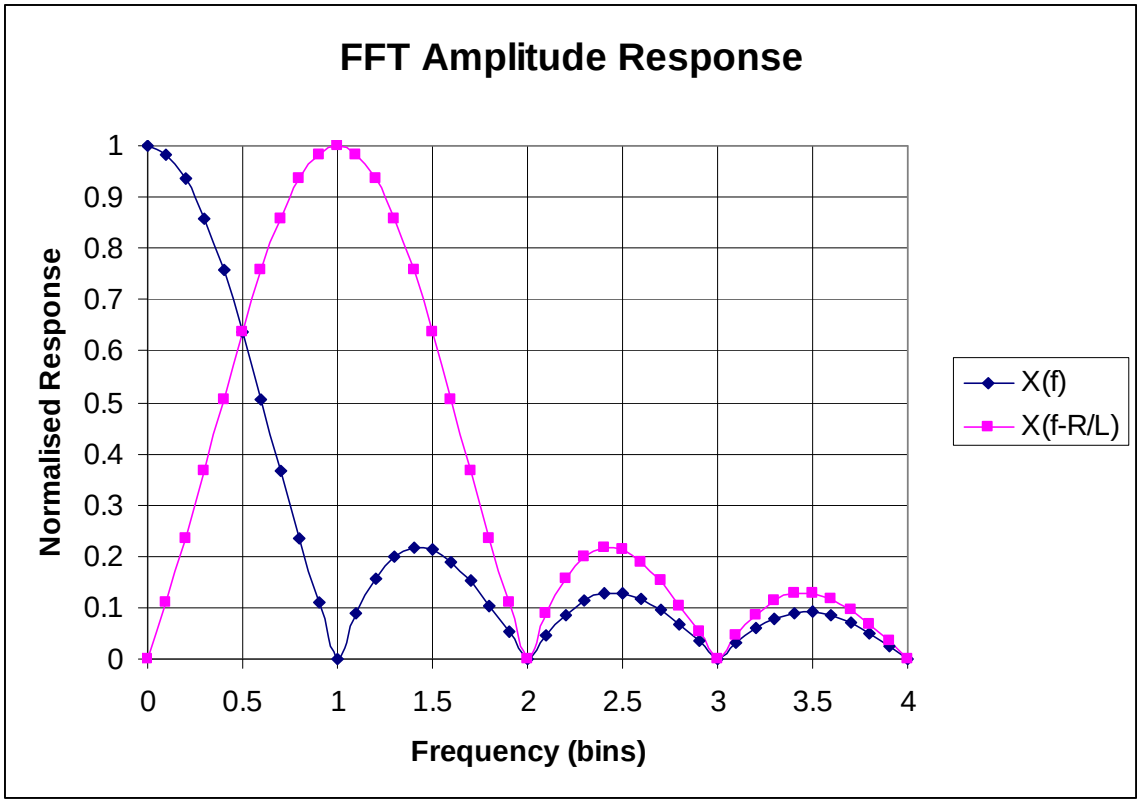

Figure 2.5/15: FFT Amplitude Response

The Amplitude at the bin center is *MA* where *A* is the amplitude of the signal. The response at the intersection of two bins is reduced to 0.63 of the bin center or 3.9 dB down.

The noise amplitude in a bin is:

$$\eta_n = \sum_{k=1}^{M} x(k) \exp\left(- j2\pi k\, n/M\right) \tag{2.90}$$

The noise samples are complex Gaussian and statistically independent. The rotation does not change the distribution. Thus the noise power in a bin is the sum of the power of noise samples. The variance (*V*) of the noise in an FFT bin is *M* times the variance of the noise after correlation.

$$V_M = \sigma_M{}^2 = M\sigma_L{}^2 = ML_C\sigma_N{}^2 \tag{2.91}$$

At the center of the bin, the improvement in signal to noise ratio is the same as achieved with coherent summing. The loss of 3.9 dB at the intersection of two bins is significant. This loss can be reduced by windowing. However, windowing widens the main lobe of the frequency response. Consider a Hanning window:

$$W(n) = 0.5 - 0.5\cos(2\pi n / M) \qquad (2.92)$$

Figure 2.5/16 shows the response of the center bin with and without windowing for a 64 point FFT.

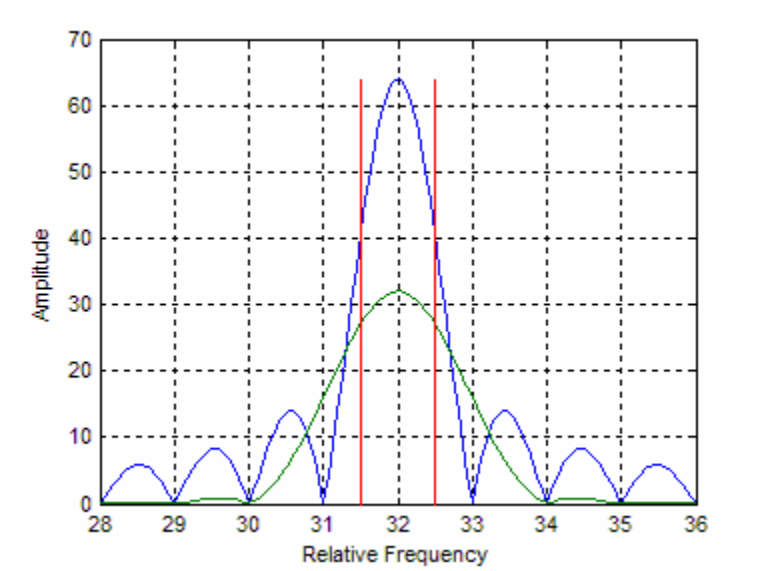

Figure 2.5/16: FFT Response With and Without Windowing

The response with windowing is down 1.42 dB at the bin edge. The noise bandwidth is increased by a factor of 1.5 or 1.76 dB. Windowing gives a net improvement at the bin edge of 3.92 - 1.42 - 1.76 = 0.74 dB.

2.5.5 Comparison of Averaging Techniques

The analysis of probability of detection and false alarm as a function of signal to noise ratio and averaging is covered in sections 2.1 to 2.5. The object of this section is to compare the various techniques. Figure 2.5/17 shows the Probability of detection vs. input signal to noise ratio for M=100. The input signal to noise ratio is the signal to noise ratio at the output of the correlator and the input to the averaging process. The simulated results are also shown.

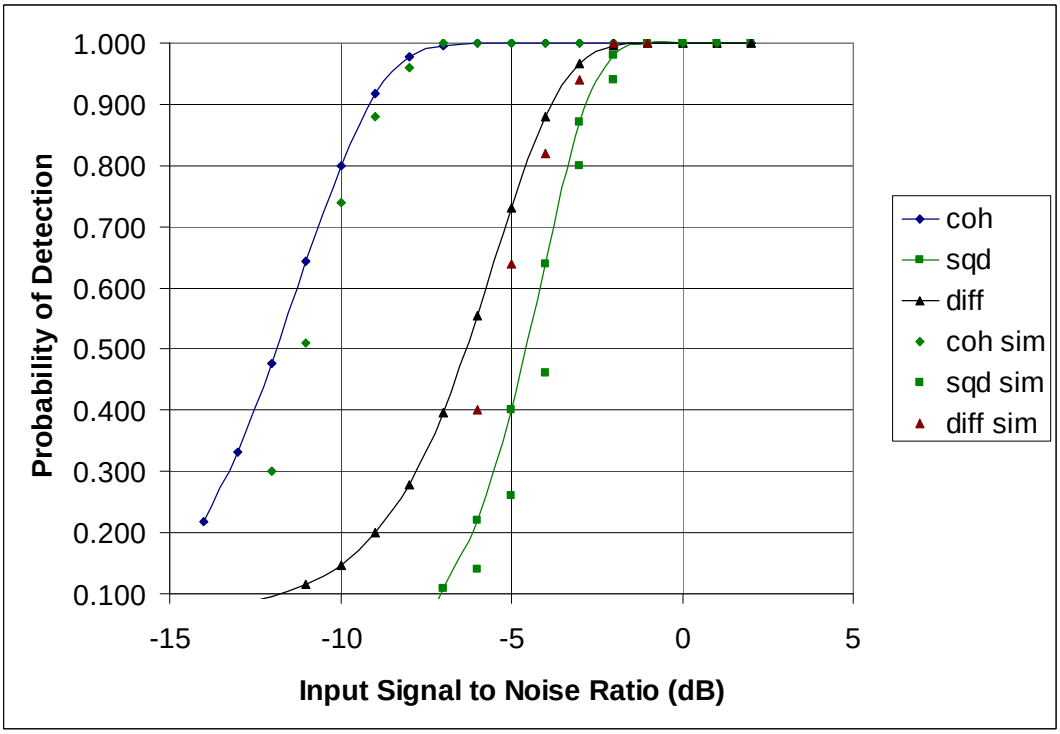

Figure 2.5/17: Probability of Detection vs. Input *S/N* for Averaging Techniques

This is a useful graph but only applies to averaging with M=100. A comparison for all practical values of M is desired.

Others have compared averaging techniques by computing the input and output signal to noise ratios [12]. This is not a good comparison as the distributions are quite different. In this work we compare techniques on the basis of probability of detection and false alarm. The question is, given a set of probabilities, what is the tradeoff between input signal to noise ratio and required number to average.

A useful way of comparing techniques is to determine the input signal to noise ratio required for successful detection given the averaging (type and value of *M*). The result is dependant on values of P_d and P_{fa}. The value of P_{fa} is influenced by the number of samples used in the correlation. The objective is to keep the number of false alarms small. Comparing averaging techniques for GPS L1 C/A code correlation, P_{fa}=.0001 is chosen to get an average of 0.2 false alarms in a code length of 2046 samples. We can now determine the input *S/N* as a function of *M* to obtain the desired probability of detection. In this example P_d = .95 was chosen. Figure 2.5/18 shows the result. Note that the coherent result also applies when Fourier transforms are used in the averaging process, as described in Section 2.5.4

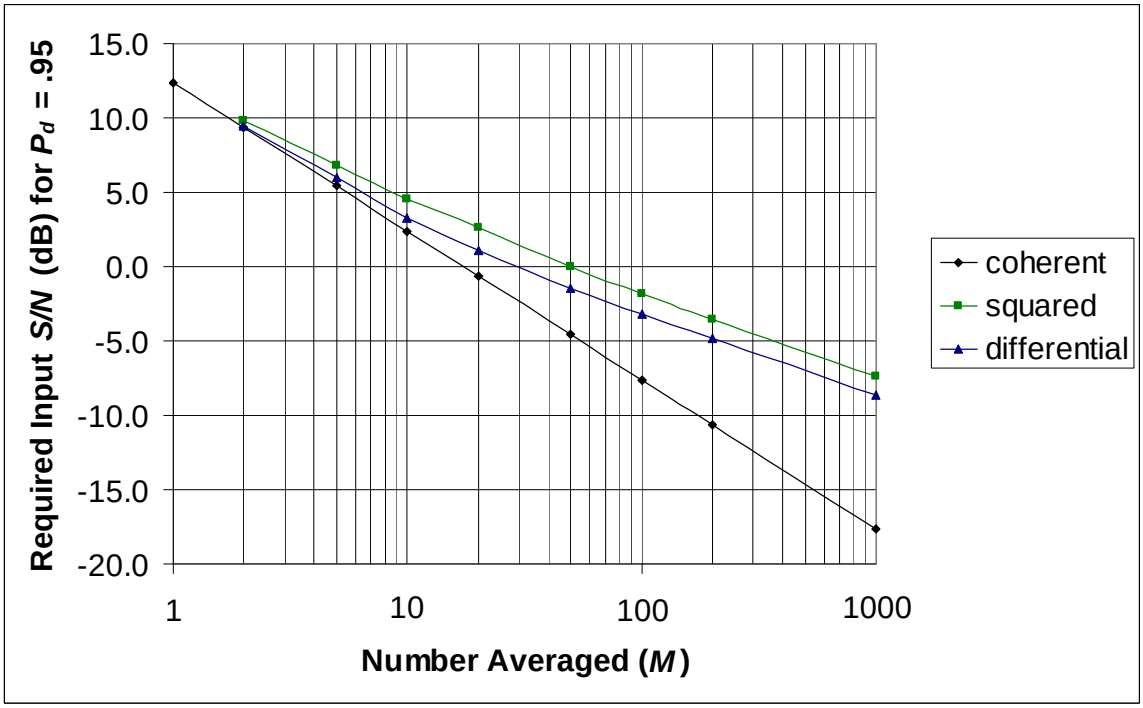

Figure 2.5/18: Comparison of Averaging Techniques for GPS L1 C/A Code

For GPS L5 and Galileo L1 the number of samples needed for correlation is much longer. For GPS L5 2x10230=20460 samples are required. For the Galileo L1b and c signals, 5 samples per chip for 4092 chips or 20460 samples are also used. To keep the number of false alarms per correlation less than one, P_{fa}=.00001 was chosen. Figure 2.5/19 shows the comparison.

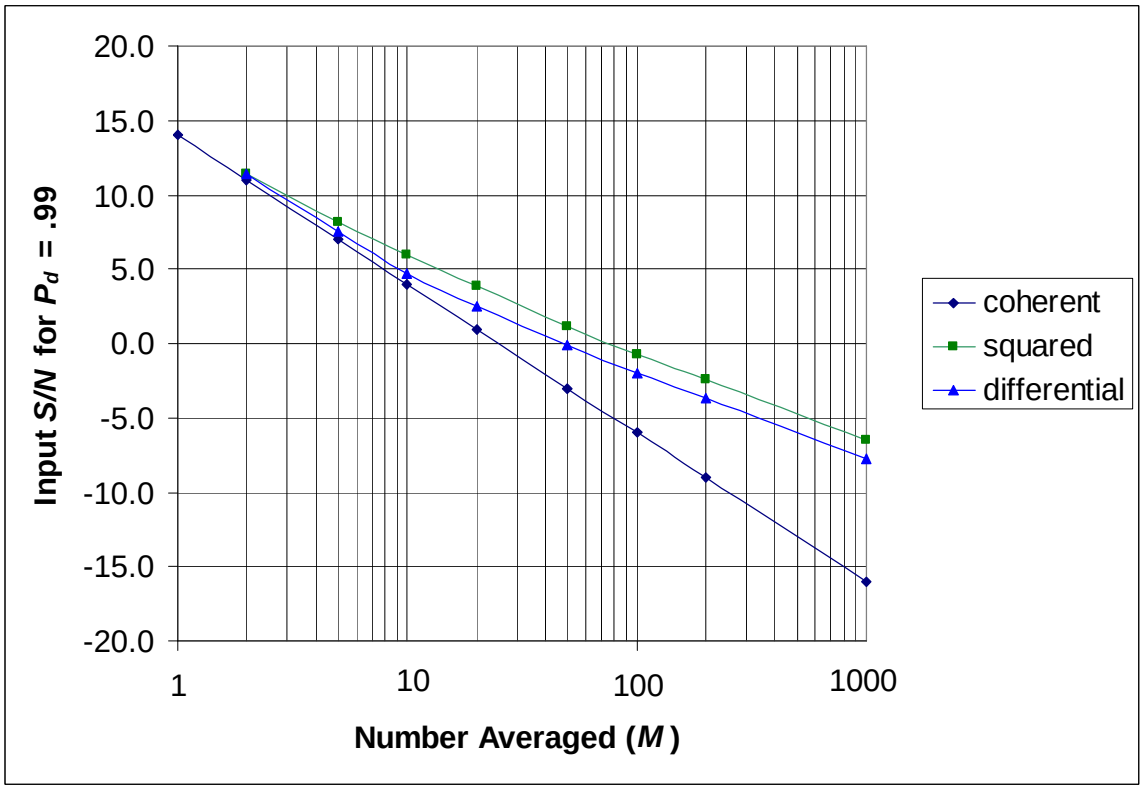

Figure 2.5/19: Comparison of Averaging Techniques for GPS L5 or Galileo L1b or c

As can be seen, the lower value of P_{fa} results in a loss of sensitivity of about 1 dB. The shape of the curves remains constant.

The result is interesting and is different from the result obtained by simply applying a squaring loss [37]. As expected, the required input signal to noise ratio for coherent averaging is inversely proportional to *M*. For large values of *M*, the improvement in sensitivity for power averaging becomes asymptotic to the square root of *M*. However at low values, the curves converge. This result is reasonable if we understand the distributions. Consider the non-aligned case. The distribution of the noise in both cases is chi-squared. The difference is in the number of degrees of freedom (*k*). For coherent averaging *k=2* and for power averaging *k=2M*. Figure 2.5/20 shows the chi-squared distribution for various values of *M*. As can be seen, as *M* decreases, the shape of the distribution approaches that for *k=2* (negative exponential).

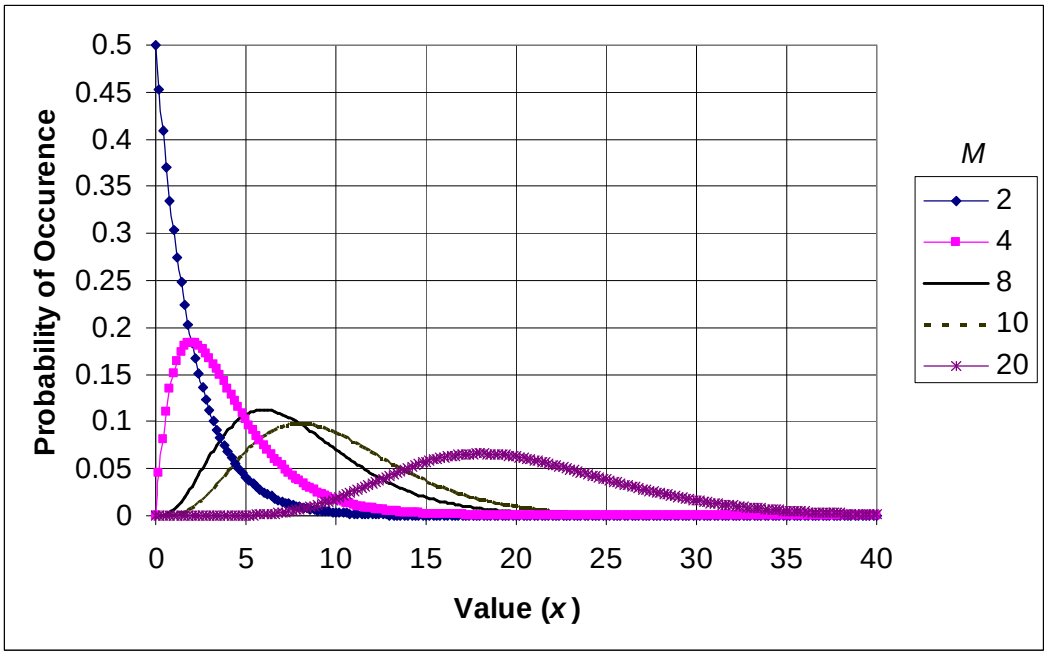

Figure 2.5/20: Chi-squared Distribution

The performance of differential averaging is about 1.3 dB better than power averaging except for small values of *M* where it falls between coherent and power averaging. Obviously these curves can also be used to determine the required averaging for a given input *S/N*.

There are other factors to consider in the comparison. First there is the data value. Coherent integration depends on the data value being constant for the duration. Non-coherent or squared averaging eliminates the data value by multiplying the correlation output by its conjugate. Differential averaging can survive some data changes provided the probability of reversals is small. This is the case with GPS where the data rate is 50 bps and the code length is 1 ms.

Another important parameter is the tolerance to frequency offset, due to unknown Doppler shift. Coherent integration is sensitive to the phase change due to frequency offset in the integration period. Non-coherent and differential integration are sensitive only to the phase change in the period of the input. In the simplest case this is the period of the basic correlation. If coherent integration precedes the non-coherent integration, it is the period of the coherent integration.

For coherent integration the code Doppler is negligible compared to the carrier Doppler. However, for long non-coherent integration, the code Doppler can become significant.

2.5.6 Averaging After Segmented Correlation

A segmented correlator [31] performs coherent correlation for fractions of the code then incoherently adds the outputs. The signal to noise ratio at the output of each segment of the correlator is reduced by the number of segments. Then the result after summing the power for the complete code is improved by the same factor as power averaging over the number of segments.

Consider a correlator with *K* segments. The signal to noise ratio at the output of each segment is reduced by *1/K* compared to an un-segmented correlator. If further averaging is performed for *M* complete codes the output will have a non-central chi-squared distribution with *2KM* degrees of freedom. We see then that the segment output is reduced by *1/K* but the averaging is increased by *K*. When *M* is large the improvement for power averaging is as the square root of *M*. From this we can conclude that the net loss in performance for segmentation is as the square root of the number of segments. However, when *M* is small, the loss for power averaging decreases. Figure 2.5/21 shows the required input *S/No* for various numbers of segments as a function of the number of complete codes that are averaged. The probability of false alarm for this comparison is .001 and the probability of detection is 0.95.

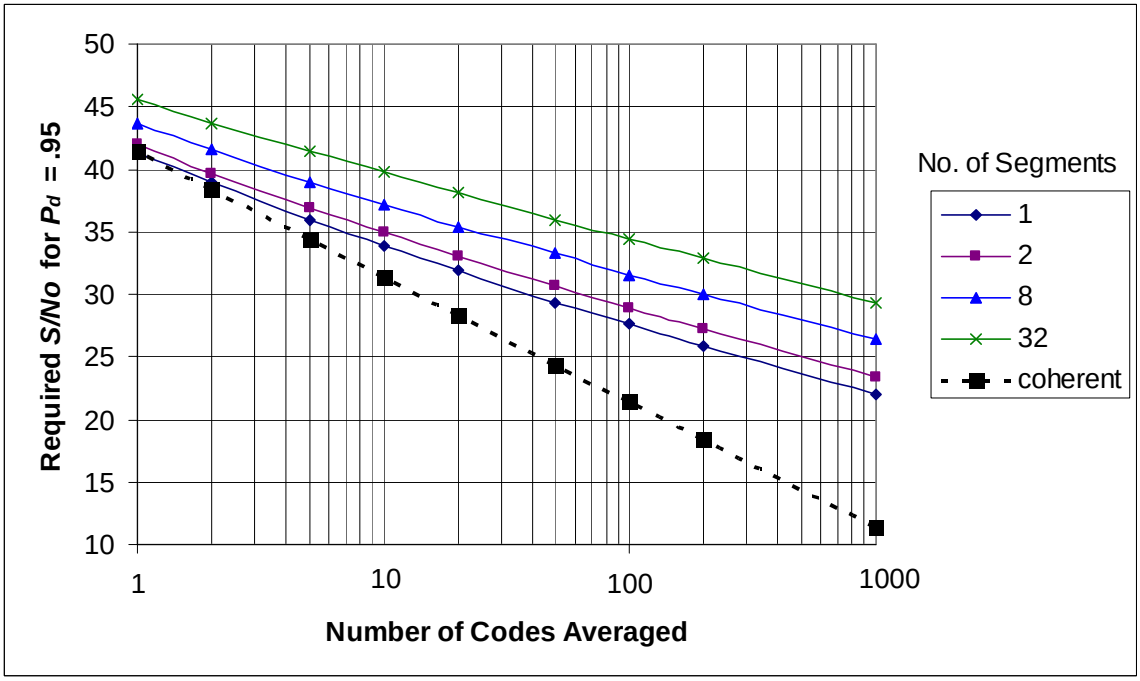

Figure 2.5/21: Averaging After Segmented Correlation

For large values of *M*, the increase in required *S/No* approaches the square root of the number of segments.

2.6 SEARCHING STRATEGIES

2.6.1 Correlation/Averaging Techniques and Searching Strategy

The optimum searching strategy depends on the correlation and averaging technique that is selected [11]. Three strategies are discussed here.

1. Serial Search
2. Hybrid Search
3. Maximum Search

2.6.2 Serial Search

Serial search is applicable when the search space has only one dimension that is delay offset. This strategy was the basis for the discussion of probability of false alarm (P_{fa}) and probability of detection (P_d) in the preceding sections.

In this strategy, a threshold is chosen that has a small probability of being exceeded by the noise. Then each cell is tested to see if the power exceeds the threshold. If the threshold is exceeded, a further test known as verification is performed to determine if the selected cell corresponds to the correct alignment. If true the process is complete and the alignment is found. If false the next cell is tested. The process continues until the correct alignment is found or all cells have been tested.

The probabilities depend on the aligned and non aligned distributions.

There are two variations to this procedure depending on the correlation method used as described in section 2.4. Using serial correlation, one alignment is tested at a time. After the correlation is completed for that alignment, the result is tested against the threshold. If the result is below the threshold or the verification fails, the next delay is tested.

If the correlation tests all delays at the same time as when FFTs are used, each delay is checked against the threshold but a new correlation is not necessary when the result is below the threshold.

The acquisition time when using these two techniques is analyzed in section 2.7.

2.6.3 Hybrid Search

A hybrid search method is applicable when the search space spans two dimensions. An example is Doppler offset and delay as described in section 2.5.4. In this example rows give the output as a function of delay for a given Doppler offset. Columns provide the output as a function of Doppler offset for a given delay. One method to search this space is to first find the maximum in each column. After this operation we have a one dimensional search space for the correct delay. The correct delay is then the offset with the maximum output. Once the delay is found, the corresponding column is searched for the correct Doppler offset defined by the maximum output.

2.6.4 Maximum Search

Maximum search simply finds the global maximum of all cells. The order of the search is not significant. The search can be row by row or column by column. The cell with the maximum output is assumed to correspond to the correct delay and Doppler shift.

There is a problem with this simple procedure. If the search space does not include a valid signal, the highest noise peak will be chosen. To solve this problem, a threshold is set in the same manner as for serial search. Only signals exceeding the threshold are considered as valid candidates for the maximum. In the following analysis, the initial threshold is labeled *Tho*.

For this strategy, the probability of detection is the probability that the maximum output corresponds to the correct alignment and Doppler shift.

The probability that the aligned value is the maximum value is the probability that all non aligned values are below the threshold and the aligned value is above the threshold for all thresholds. The probability that all non aligned values are below the threshold is:

$$P_0(Th) = (1 - P_{fa}(Th))^n \tag{2.93}$$

The number *n* is the number of independent samples per correlation. The probability of detection is then:

$$P_{\text{det}} = \int_{Th=Tho}^{\infty} P_0(Th)\, f_{\text{det}}(Th)\, dTh \tag{2.94}$$

The term $f_{det}(Th)$ is the probability density function for the aligned cell. After correlation or coherent averaging, the distribution of the aligned cell is non-central chi-squared with 2 degrees of freedom ($k=2$) and non centrality parameter λ. The probability distribution function is:

$$f_{\text{det}}(\lambda, Th) = \frac{1}{2}\exp\left(-\frac{(Th+\lambda)}{2}\right) I_0\left(\sqrt{\lambda Th}\right) \tag{2.95}$$

From the section on averaging techniques we know $\lambda = 2M\frac{S}{N}$. The constant *M* is the number of outputs coherently summed. The signal to noise ratio (*S/N*) is the signal to noise ratio after the correlation. In this section as in the section on averaging, *Th* and λ are the normalized values:

$$Th_{Normalized} = \frac{Th}{\sigma^2} \text{ and } \lambda = \sum_{i=1}^{k} \frac{A_i^2}{\sigma^2} \tag{2.96}$$

Furthermore, the non aligned distribution is chi-squared with two degrees of freedom or negative exponential.

$P_{fa} = \exp\left(-\frac{Th}{2\sigma^2}\right)$ and using the normalized threshold $P_{fa} = \exp\left(-\frac{Th}{2}\right)$. (2.97)

Using numerical integration the value of the probability of detection was computed as a function of λ. The required λ for P_{det} = 0.99 is 41. The require signal to noise ratio with no averaging is:

$\frac{S}{N} = 10\log_{10}\frac{\lambda}{2} = 13.1\,dB$.

With coherent averaging the required signal to noise ratio reduces as *1/M*.

$\frac{S}{N} = 10\log_{10}\frac{\lambda}{2M}$

Figure 2.6/1 shows the probability of detection vs. signal to noise ratio when 2046 cells are searched and when 128 x 2046 cells are searched. Note that a higher signal to noise ratio is required as the number of cells increases as there are more ways that a false peak could exceed the true peak.

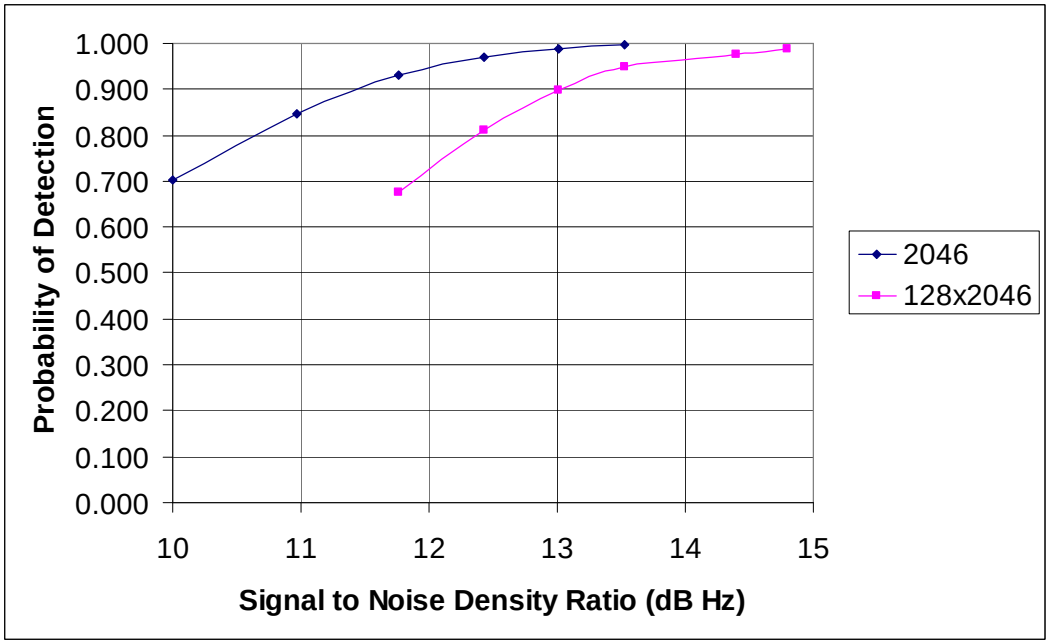

Figure 2.6/1: Probability of Detection for Maximum Search

If non coherent or power averaging is used the distribution of non-aligned values is chi-squared with 2*M* degrees of freedom.

$$P_{fa} = \int_{x=Th}^{\infty} \frac{x^{\left(\frac{k}{2}-1\right)} \exp\left(-\frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right) 2^{\left(\frac{k}{2}\right)}} dx \qquad (2.98)$$

In this work I used the chi-squared distribution function in EXCEL.

Pfa=CHIDIST(Th,k) = CHIDIST(Th,2M)

The aligned probability density function is non-central chi-squared with $k = 2M$ degrees of freedom and non centrality parameter λ. As with coherent integration $\lambda = 2M \frac{S}{N}$.

$$f_{\det}(Th) = \frac{1}{2} \exp\left(-\frac{(Th+\lambda)}{2}\right)\left(\frac{Th}{\lambda}\right)^{\left(\frac{k}{4}-\frac{1}{2}\right)} I_{\left(\frac{k}{2}-1\right)}\left(\sqrt{\lambda Th}\right) \qquad (2.99)$$

Again numerical integration was used to obtain the required value of λ to obtain a 99% probability of detection. The required *S/N* is plotted as a function of *M* in Figure 2.6/2. For this plot the number of cells searched is 20460. Note that the required signal to noise ratio is close to the value shown by Figure 2.5/19.

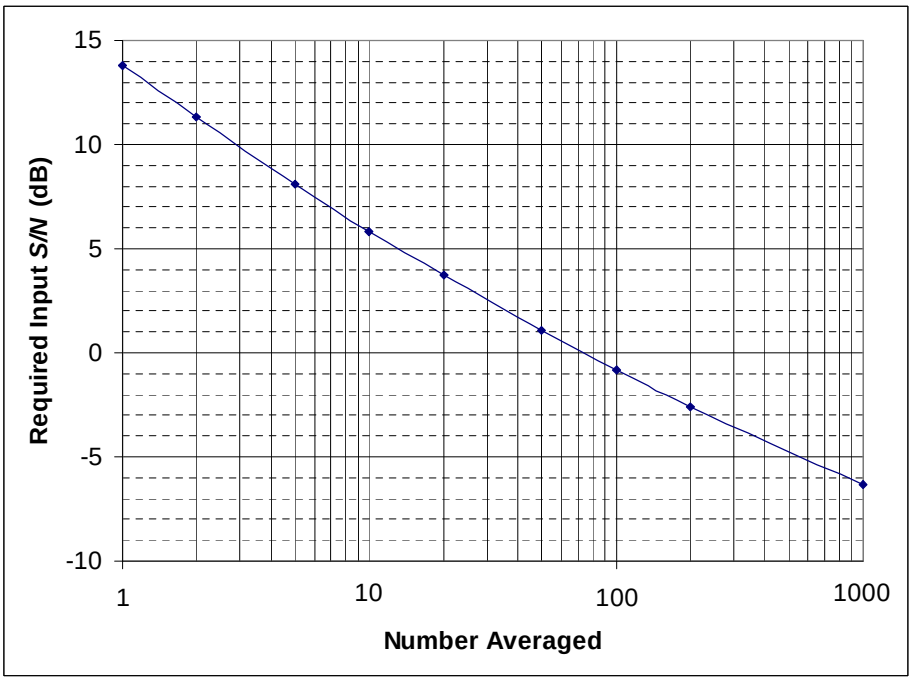

Figure 2.6/2: Required *S/N* vs. *M* for Power Averaging and Maximum Search

2.7 ACQUISITION TIME

The time to acquire code alignment depends on the basic correlation time, the required averaging and the number of frequency offsets (Doppler bins) that have to be searched. The time also depends on the correlation technique, the penalty for a false alarm, the probability of false alarm and the probability of detection.

Simon et al [5] give the mean acquisition time for serial correlation as:

$$Tacq = \left[\frac{2 + (2 - P_d)(L-1)(1 + KP_{fa})}{2P_d} \right] \tau_D . \qquad (2.100)$$

The value of *L* is the number of code phases tested and is equal to the code length divided by the step size. The factor τ_D is the dwell time at each step and is equal to the code

period times the number averaged. The factor *K* is the penalty for a false alarm and is the number of dwell times it takes for the verification process to determine that a false peak has been detected.
This is the acquisition time for a single Doppler bin assuming the signal is found within that bin.

The acquisition time for parallel acquisition is presented in the following paragraphs.
In this work it is assumed that correlation is performed using Fourier transforms. Using this technique all delays are tested at the same time. The processing time is fast compared to the time to collect sufficient samples for the analysis. Once sufficient samples are collected, correlation and averaging are performed. A threshold is set and each delay is tested for exceeding the threshold. The testing time is only the computation time that is assumed to be much smaller than the correlation time. If a sample exceeds the threshold, a verification procedure is invoked. If the sample is verified to be true it is assumed that the alignment is found and the process is complete. If the verification is false, the search continues.

The verification process takes longer than the correlation and averaging time. Thus the verification time and the probability of false alarm have a significant effect on the acquisition time.

Figure 2.7/1 is a Markov model of this process.

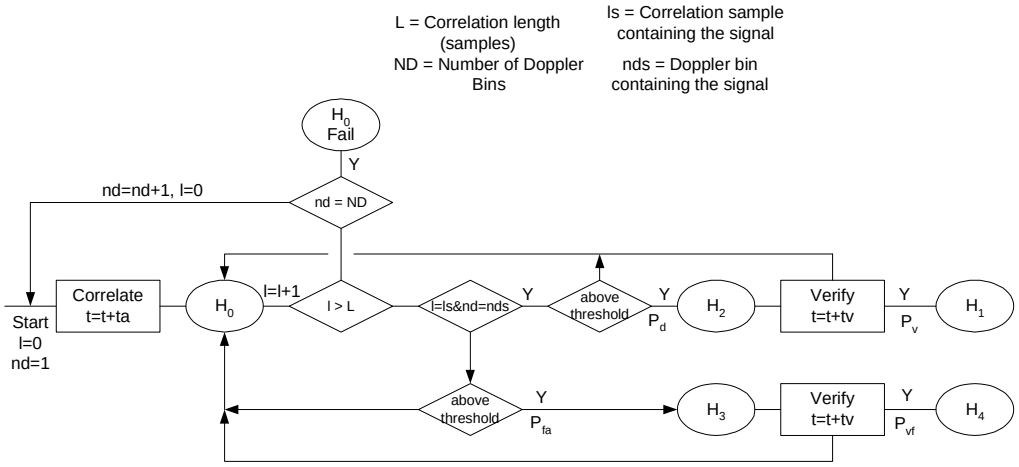

Figure 2.7/1: Markov Model of Acquisition Process

The non-aligned state is labeled H_0. The aligned state is H_1. There are two intermediate states, H_2 and H_3. These states correspond to a sample exceeding threshold for the correct delay and incorrect delay respectively. There is also a terminal state, H_4, for a false verification. The process is terminated when the state is H_1 or H_4 or when all delays and Doppler bins have been tested and alignment is not found (the state remains as H_0).

The probability of the sample corresponding to the true alignment exceeding the threshold is the probability of detection (P_d). The probability of a non aligned sample exceeding the threshold is the probability of false alarm (P_{fa}). If the true alignment is successfully found, it will, on average be found after half the Doppler bins (ND) have been tested. In this case the search time is:

$$T_S = \frac{ND}{2}Ta + \left(1 + P_{fa}L\frac{ND}{2}\right)Tv \tag{2.101}$$

The times Ta and Tv are the averaging time and verification time respectively and L is the correlation length in samples.

If the signal is not detected, all bins are tested and the time taken is:

$$T_S = \left(NDTa + NDP_{fa}LTv\right) \tag{2.102}$$

The total search time is then:

$$T_S = P_d\left(\frac{ND}{2}Ta + \left(1 + P_{fa}L\frac{ND}{2}\right)Tv\right) + \left(1 - P_d\right)\left(NDTa + NDP_{fa}LTv\right) \tag{2.103}$$

Unfortunately there is a probability (1-P_d) that the true alignment will not be found after a single search. If the process is continued until the signal is detected, the average acquisition time will be:

$$T_Q = \frac{T_S}{P_d} \tag{2.104}$$

Finally, collecting terms and substituting $Pv = KTa$ the acquisition time is:

$$\frac{\left(1 + LP_{fa}K\right)\left(2P_dK + ND\left(2 - P_d\right)\right)Ta}{2P_d} \tag{2.105}$$

The Markov chain was modeled in MATLAB™ and the results agreed well with those calculated with the equation.

Table 2.7/1 shows results for the following case:

L=2046

ND=200

Ta=50 ms

Tv=100 ms (*K*=2)

Table 2.7/1: Acquisition Time as a Function of P_d and P_{fa}

P_d	P_{fa}	*Ts*	*Tf*	Total	simulated	% error
0.95	0.001	24282	2546	26828	27141	-1.15324
0.95	0.0001	6788.7	704.6	7493	7396	1.315576
0.99	0.0001	7074.54	140.92	7215	7171	0.619997
0.99	0.001	25304.4	509.2	25814	25899	-0.32974
0.99	0.00001	5251.554	104.092	5356	5358	-0.04393

In this table, the times for successful acquisition and failed acquisition are shown separately. It is assumed that the process terminates after a single attempt.
It can be seen that the probability of false alarm has a large affect on the total time. The probability of detection has less effect. Although not shown by this example, the penalty for false alarm, *K*, also has a large affect. Raising the threshold to decrease the probability of false alarm decreases the sensitivity. The penalty for raising the threshold to decrease the probability of false alarm from .001 to .0001 is about 0.7 dB for the same probability of detection.

Acquisition times were analyzed for four conditions of GPS L1 signals. These conditions are:

1. The signal is strong enough to acquire without averaging, *S/No* = 44 dB Hz.
2. The minimum signal level for reliable data demodulation, *S/No* = 27 dB Hz.
3. A weak signal, *S/No* = 20 dB Hz.
4. A very weak signal close to the minimum detectable, *S/No* =14 dB Hz.

For the first case, the correlation time is 1 ms and there is no averaging. The penalty for false alarm is chosen to a generous value of 20. With 1 ms correlation time the search bandwidth is about 600 Hz.

For the second case there are two possibilities. The first is to correlate in 1 ms and power average 200 correlations. In this case the search bandwidth is 600 Hz. The second is to use coherent averaging for 10 ms followed by power averaging of 5 times. This results in a shorter time but the search bandwidth is only 60 Hz.

The third case requires coherent averaging for 10 ms followed by power averaging for 50 times. The search bandwidth is 60 Hz.

The final case requires coherent averaging for 10 ms followed by power averaging for 600 times. The search bandwidth is also 60 Hz.

The following parameters were used for all cases:

L = 2046

Maximum Doppler Offset = 4500 Hz

Oscillator Error (1 ppm) = 1500 Hz

P_{fa} = .0001

P_d = .95

Table 2.7/2 shows the mean acquisition time as a function of *S/No* for serial correlation. The times are analyzed for the four cases described above.

Table 2.7/2: Mean Acquisition Time for Serial Correlation

	coherent		Dwell		Mean Acquisition Time	
S/No	time (ms)	Averaging	time (ms)	*K*	seconds	minutes
44	1	1	1	20	1.13	0.02
27	10	5	50	4	56.58	0.94
20	10	50	500	2	565.71	9.43
14	10	600	6000	2	6788.46	113.14

The times shown in the table are for a single acquisition. If the Doppler shift is not known, this process has to be repeated many times for each satellite. Even if the Doppler

shift is known acquisition of at least 4 satellites is required. As can be seen, except for the strongest signal, the times are too long.

The time to acquire code timing and Doppler offset for a single satellite using parallel correlation is shown by table 2.7/3:

Table 2.7/3: Time to Acquire Code Timing and Doppler Offset

S/No	coherent time ms	averaging	verification time ms	acquisition time ms	search bandwidth	searches required	search time s
44	1	1	20	23	600	20	0.076
27	10	5	200	262	60	200	10.249
27	1	200	400	624	600	20	3.515
20	10	50	1000	1562	60	200	78.876
14	10	600	12000	18690	60	200	946.522

In the absence of assistance, on average, half the satellites will have to be searched for resulting in the times shown by table 2.7/4.

Table 2.7/4: Total Search Time

S/No (dB Hz)	Search Time per Satellite (s)	Number of Satellites	Total Time (s)
44	.076	15	1.14
27	10.249	15	153.74
27	3.515	15	52.73
20	78.876	15	1183.15
14	946.522	15	14197.83

As can be seen, the search time is very long for the last two cases. These times are only the code acquisition times. After code timing is acquired, tracking must be initiated and acquired. Navigation data has to be obtained for each satellite.

The analysis was also performed for GPS L5 and Galileo L1. With GPS L5, the data is rate ½ coded and requires an *Eb/No* = 5 dB for successful demodulation. For this reason the analysis was done with a *S/No* of 22 dB Hz instead of the 27 used for the C/A code. Table 2.7.5 shows the total acquisition time.

Table 2.7/5: Total Acquisition Time for GPS L5

S/No (dB Hz)	Search Time per Satellite (s)	Number of Satellites	Total Time (s)
44	0.065	15	0.98
22	71.39	15	1070.85
20	142.78	15	2141.70
14	1427.8	15	21417

With Galileo, the data is also rate ½ coded but the data rate is higher than GPS. The data rate is 120 bps and an Eb/No of 5 dB is required. The required *S/No* is then 26 dB Hz. Table 2.7/6 shows the acquisition times for Galileo.

Table 2.7/6: Total Acquisition Time for Galileo L1

S/No (dB Hz)	Search Time per Satellite (s)	Number of Satellites	Total Time (s)
44	0.518	14	7.25
26	4.521	14	63.29
20	91.668	14	1283.35
14	2756.168	14	38586.35

2.8 ASSISTED ACQUISITION

From the acquisition times presented in table 2.6/4 it is apparent that some effort is required to reduce acquisition times. Even with an *S/No* of 27 dB Hz, the best acquisition time is close to one minute. If we add to this the time to acquire tracking and gather navigation data the total can exceed 2 minutes. Two minutes is long compared to the average time of an emergency call. Furthermore, with signals weaker than 27 dB Hz, it is not possible to successfully demodulate the navigation data. Similar conclusions can be drawn for GPS L5 and Galileo L1.

The first problem to consider is the large number of frequency bins to be searched. There are two contributions to the range to be searched, the Doppler range and the reference oscillator error.

In a cellular system, the frequency reference of a mobile phone can be locked to the base station reference giving a frequency accuracy of .05 ppm.

If the range rate of the satellite is approximately known the frequency range can be further reduced. The Doppler offset within a cell will not vary significantly from the offset at the center of the cell.

Another contributor to the long acquisition time is the number of satellites to be searched for. If the position and identity of the visible satellites is known, as few as 4 satellites could be searched for and acquired.

The availability of navigation data is another problem. The navigation data can be collected at the base station and communicated to the mobiles. There is a significant amount of data and it has to be updated regularly.

Another alternative is for the mobile to acquire the signal and communicate the measured transit times (known as pseudo ranges) to a central system where the position is calculated. Table 2.8/1 is a comparison of the two types of assistance. The data in this table is reproduced from the paper by Zhao [40]. In this table UE refers to the user equipment or phone, IE is the information exchange and LMU is the location measurement unit. The LMU is the software that computes the location from the measured pseudo ranges.

Table 2.8/1: Comparison of UE Based and UE Assisted GPS

Pros/Cons	UE-Based GPS	UE-Assisted GPS
Advantage	Relatively short uplink IE	Relatively short downlink assistance IE if code phase and Doppler are used
	Assistance IE valid 2-4 hours or up to 12 hours at the UE if ephemeris life extension feature is used (less signaling)	Network in control of position determination
	Good for tracking/navigation applications	Need less computing power and memory at the UE
	Can be used as a standalone GPS receiver	
	Do not need LMU	
Disadvantage	Relatively long downlink assistance IE	Relatively long uplink IE
	Need more computing power and memory at the IE	Assistance IE valid for a few minutes at the UE if code phase and Doppler are used (more signaling for tracking/navigation applications)
		Need accurate timestamps at the UE
		Certain event trigger mechanisms will not work
		Need LMU for certain assistance data

Acquisition times were analyzed given a reference oscillator accuracy of .05ppm and knowledge of the Doppler frequency within 1 %. For GPS the resulting frequency

uncertainty is then plus or minus 45 + 79 = 124 Hz. It is also assumed that as many as 5 satellites may be searched for. The five satellites allow for one to be shadowed or not to be in a favorable position for accurate position determination.

Table 2.8/2 shows the acquisition time for GPS L1 with assistance.

Table 2.8/2: Total Search Time for GPS L1 with Assistance

S/No (dB Hz)	Search Time per Satellite (s)	Number of Satellites	Total Time (s)
44	0.023	5	0.11
27	0.556	5	2.78
20	2.947	5	14.73
14	35.363	5	176.82

Table 2.8/3 shows the acquisition time for GPS L5 with assistance.

Table 2.8/3: Total Search Time for GPS L5 with Assistance

S/No (dB Hz)	Search Time per Satellite (s)	Number of Satellites	Total Time (s)
44	0.023	5	0.11
22	2.469	5	12.35
20	4.938	5	24.69
14	49.38	5	246.9

Table 2.8/4 shows the acquisition time for Galileo L1 with assistance.

Table 2.8/4: Total Search Time for Galileo L1 with Assistance

S/No (dB Hz)	Search Time per Satellite (s)	Number of Satellites	Total Time (s)
44	0.053	5	0.37
27	0.361	5	2.52
20	5.692	5	39.84
14	64.743	5	453.21

From these numbers, it can be seen that acquisition times with assistance for signals with *S/No* greater than 20 dB Hz are acceptable. However at 14 dB Hz the acquisition times are getting too long.

There are ways of reducing the time somewhat. Further refining the accuracy of the Doppler estimate would help but the reference oscillator error is already dominant. Improving the reference frequency beyond .05 ppm is unlikely as that is the specification for the base station reference. More intelligent search procedures can shorten the search time. The paper by Pan [29] et al describes a method called offset-Z search. This

method assumes that the most likely result is near the centre of the search range. Although the paper was written for serial search for the correct delay, the principles can be extended to the search for the Doppler offset. If the most likely offset is near the centre of the search range, the average number of bins searched can be reduced.

For example, the GPS L1 acquisition requires 5 Doppler bins to be searched and the mean search time is 35.4 seconds. If we assume a truncated normal distribution with the peak frequency offset equal to twice the standard deviation (2σ) the mean acquisition time for the optimum search technique is reduced to 32.2 seconds. If the peak deviation is 3σ the optimum search technique results in an average search time of 28.2 seconds. Even with the best result, the total search time for 5 satellites is more than 2 minutes.

The times quoted assume a single acquisition process resulting in the total time being multiplied by the number of satellites. It is possible to have more than one acquisition process but this is a burden for a cellular phone.

Differential averaging instead of power averaging, after the 10 ms coherent averaging, reduces the number of correlations that need to be averaged. Differential averaging does not help for GPS L1 because the data can change for every data bit.

The use of FFT's in the averaging process can result in all required frequency bins being evaluated in the time taken for a single averaging process. This procedure is shown to be applicable to GPS and Galileo L1.

Chapter 3 discusses acquisition procedures applicable to the three signals.

3.0 PROPOSED ACQUISITION PROCEDURES FOR EACH SIGNAL

In this section we consider acquisition of weak signals. Two benchmark signal strengths are considered. These are *S/No*=20 dB Hz and *S/No*=14 dB Hz.

3.1 ACQUISTION PROCEDURES FOR WEAK GPS SIGNALS

The first objective of this procedure is to acquire a signal with an *S/No* of 20 dB Hz. The GPS C/A signal has a chip rate of 1.023 Mch/s and a length of 1023 chips or 1 ms. We could consider power averaging after basic correlation. The signal to noise ratio after basic correlation is 20 - 30 =-10 dB. The graph (Figure 2.5/18) does not show the required power averaging for an *S/N* of -10 dB. However, with *M* = 1000, the required input *S/N* is -7.3 dB. If we use the rule that the improvement in sensitivity for large values of *M* is proportional to the square root of *M*, we determine that power averaging for 3500 correlations or 3.5 s is required for detection. This is a possible solution but more coherent averaging would decrease the averaging time. The signal includes data at 50 bps. We see that there is a period of 20 ms between data bits. If we knew the bit timing we could integrate coherently for 20 ms. It is generally considered that without bit timing, the limit for coherent averaging is 10 ms or 10 code periods [42].

With an input *S/No* of 20 dB Hz the *S/N* after coherent integration for 10 ms is 0 dB. Referring to Figure 2.5/18 we see that with an input signal to noise ratio of 0 dB, power averaging for 50 correlations is required for detection. The resulting averaging time is 0.5 s and that is acceptable.

The second objective of this procedure is to acquire a signal with an *S/No* of 14 dB Hz or 6 dB lower. Using the rule that the improvement in sensitivity for large values of *M* is proportional to the square root of *M*, we determine that power averaging for 4^2 x 3500 = 56000 correlations or 56 s is required for detection. There are several problems with this long correlation time. The first problem is the code Doppler tolerance. The Doppler error for 0.5 chips is 15 Hz. This small tolerance would require up to seventeen searches

to cover the frequency range, even with assistance. The number of searches together with the long averaging time would result in an overall acquisition time of nearly 90 minutes. Another problem is the stability of the reference oscillator. The rate of change of Doppler can also be a problem as it can be as high as .5 Hz/s. Differential averaging can be used because the data can only change every 20th code period. However, averaging for 30000 code periods is still needed. As with a *S/No* of 20 dB Hz, more coherent averaging is required.

With an input *S/No* of 14 dB Hz the *S/N* after coherent integration for 10 ms is -6 dB. Referring to Figure 2.5/18 we see that with an input signal to noise ratio of -6 dB, power averaging for 600 correlations is required for detection. Figure 3.1/1 shows the result of the simulation.

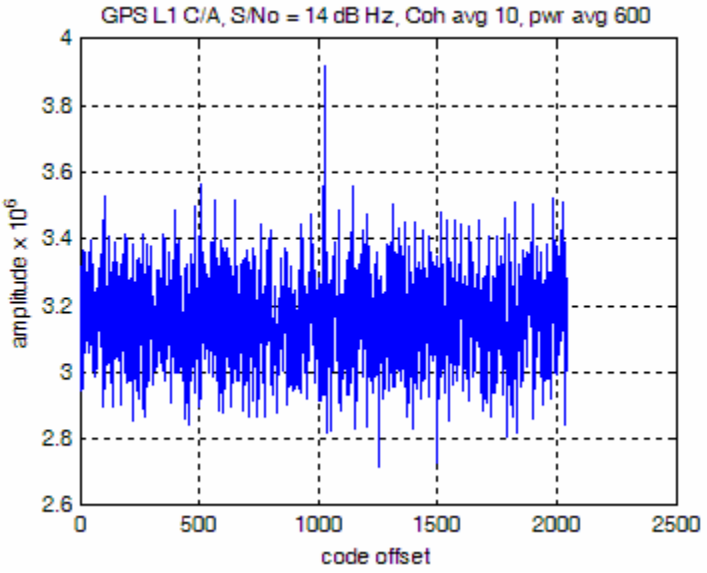

Figure 3.1/1: Correlation with Power Averaging, *M*=600

The Doppler tolerance for 1 dB degradation after coherent integration for 10 ms is 30 Hz.

Differential averaging after correlation for 10 ms does not give any improvement because the data could change for every sample.

Averaging for 6 seconds appears to be the only solution but the overall acquisition time is then over 4 minutes. The time could be improved if data aiding was available from the base station. If the bit timing was known coherent integration for 20 ms could be performed. The resulting *S/N* after coherent integration would be -3 dB. Further power averaging of 200 would be needed resulting in an averaging time of 4 s.

If the data as well as the timing is available coherent integration could be extended. The Doppler tolerance is very small after the required integration time. A potential solution is to use the Fourier transform method described in Section 2.5.4. Instead of summing the results of the 20 ms averages, 20 ms averages by M_F rows are saved to form a matrix. The minimum value of M_F required is 40 but a power of 2 simplifies the analysis. Assume 64 rows are saved for a total time of 1.28 s. After the FFT is performed on the columns, we have 64 frequency bins .78 Hz apart. The loss at the boundary between adjacent bins is almost 4 dB.

The disadvantage of this procedure is the extremely narrow frequency bins. The short term frequency stability alone will spread the signal over at least two bins. If there is any motion of the receiver the additional Doppler will spread the signal even more.

Another way of using the FFT procedure is to save a number of 1 ms correlations [30]. If 128 correlations are saved and the FFT performed, the resulting frequency resolution is 7.8 Hz that is within the capture range of the tracking loop described in section 3.3. After coherent correlation for 128 ms, the signal to noise ratio is improved by 21 dB. The resulting signal to noise ratio is 5 dB. Additional power averaging of at least 10 times is required for detection. Averaging with $M = 10$ is acceptable if the frequency is centered in the Doppler bin but a loss of up to 3.9 dB can occur. After averaging with $M = 30$, the probability of detection is acceptable even with the worst frequency offset. Figure 3.1/2 shows the result of the correlation and averaging. The *x* axis is the code delay and the *y* axis is the maximum in each column. Figure 3.1/3 shows the amplitude vs. frequency bin for the column corresponding to the correct delay.

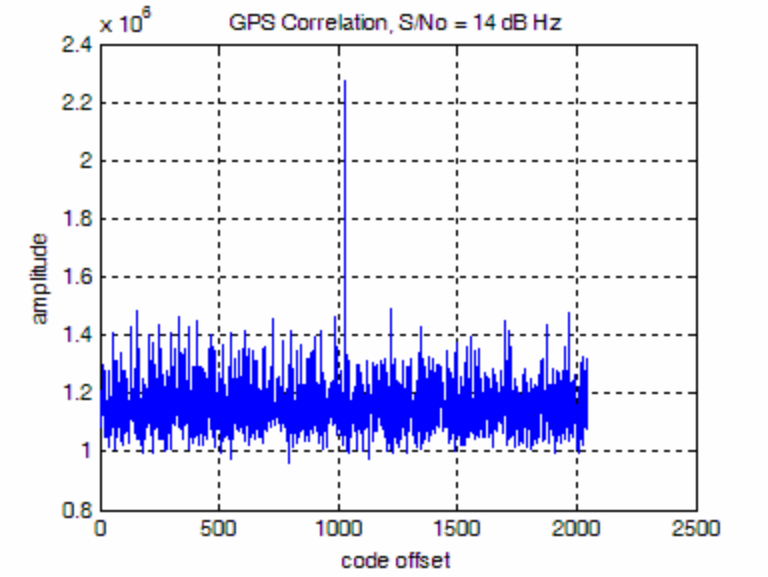

Figure 3.1/2: Amplitude vs. Delay for 128 pt FFT & *M*=30

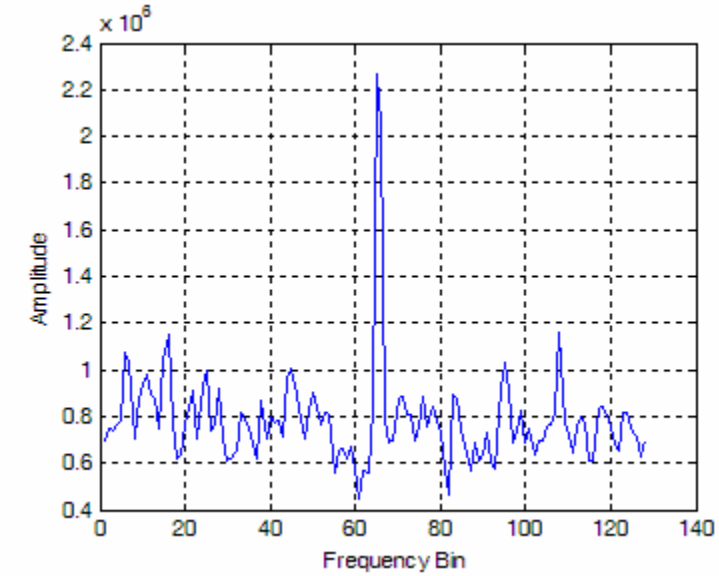

Figure 3.1/3: Amplitude vs. Frequency Bin for 128 pt FFT & *M*=30

The resulting correlation time is 3.8 seconds. This does not seem to be a big improvement over the 6 seconds of the first procedure. However, considering the 128 frequency bins, the overall frequency tolerance is ±500 Hz. With the frequency range described in section 2.7, only a single correlation is required instead of the 5 required by the earlier procedure.

The only disadvantage of this procedure is the need for data assistance from the base station in near real time.

It is fair to ask if the sensitivity of 14 dB Hz is the lower limit. The answer is no, if the data is known, it is possible to extend the power averaging to achieve greater sensitivity at the expense of acquisition time. If the averaging is increased from 30 to 60 times the sensitivity is improved by 1.5 dB. Also if the frequency offset is near the center of the bin, the sensitivity will be 2 to 3 dB better.

3.2 ACQUISITION PROCEDURES FOR WEAK GALILEO L1 SIGNAL

The Galileo L1c code is 4092 chips long at a code rate of 1.023 Mch/s. Considering the bandwidth, the simulation uses 5 samples per chip. From Figure 2.5/19 we see that with no averaging an *S/N* of 14 dB in the correlation bandwidth (250 Hz) is required for a .99 probability of detection. Using these values we see that the required *S/No* at the input to the correlator is:

$$\left.\frac{S}{No}\right)_{REQ} = 14 + 10\log_{10}(250) = 38\,dB\ Hz \tag{3.1}$$

As with GPS L1, signal acquisition requires significant averaging.

The first objective is to acquire a signal with an *S/No* of 20 dB Hz. The signal to noise ratio after the 4 ms correlation is 20 – 24 = -4 dB. Referring to Figure 2.5/19, we see that power averaging with *M*=400 could solve the problem. The total averaging time of 1.6 s is marginally acceptable.

Considering the objective of acquiring a signal with a *S/No* as low as 14 dB Hz, more averaging is required. Coherent averaging for one second (250 code lengths) would give an improvement of 24 dB. That would be close but the frequency tolerance would only be ±0.3 Hz.

To perform coherent averaging for longer than one code length, we must acquire the timing of the 25 bit short code. Remember, we know the code but do not know when it starts. There are basically two ways this can be accomplished without assistance. We could correlate with a long code consisting of the 4092 chip spreading code and the 25 bit short code. A second method is to perform the correlation 25 times using successive 4 ms samples, save the results then find the short code alignment by testing start points until a maximum is reached. There is a complication for the second technique. The short code bit could change in the middle of the 4 ms samples that we take from the received signal. To overcome this problem we take 25 8 ms samples, advancing the start by 4 ms for each sample. The correlation is performed with zero padded sequences consisting of one complete spreading code followed by 4 ms worth of zeros [41].

Both systems have been simulated and give similar results. If the correlation is performed using FFT's, we can compare the number of operations required for each technique. It is known that the number of operations for a n point FFT varies as $n\log_2(n)$. Using this knowledge the comparison is shown by table 3.2/1.

Table 3.2/1: FFT Time Comparison

	n	$\log_2(n)$	FFT time	No. of times	Total time
Long code	511500	18.96437	9700277.7	1	9700278
Spreading Code	20460	14.32052	292997.81	25	7324945
2 Code Lengths	40920	15.32052	626915.62	25	15672890

In this table, time is measured in operations. The time depends on the speed of the computation. We see that one long correlation is faster than 25 shorter correlations of 2 code lengths (8 ms). This is an approximation as the $n\log(n)$ rule is only accurate if n is a power of two.

After coherent correlation for the length of the short code, the bandwidth is 10 Hz. The required *S/No* for detection is 14 + 10 =24 dB Hz. We could continue coherent correlation but the Doppler tolerance after 1 second of coherent correlation is .3 Hz for 1

dB of loss. Considering that the rate of change of Doppler offset can exceed 0.5 Hz/s and the short term frequency tolerance exceeds 1 Hz, this is far too tight.

Power averaging for 20 times would achieve the same result as coherent averaging for 10. Another option is differential averaging. We know that differential averaging performs a little more than 1 dB better than power averaging. Differential averaging for 15 times would then achieve the same result. Either form of averaging leaves the Doppler tolerance in the order of ±3 Hz. Obviously a prior knowledge of the range rate is required from external assistance or through the use of the almanac.

Figure 3.2/1 shows the result of the simulation.

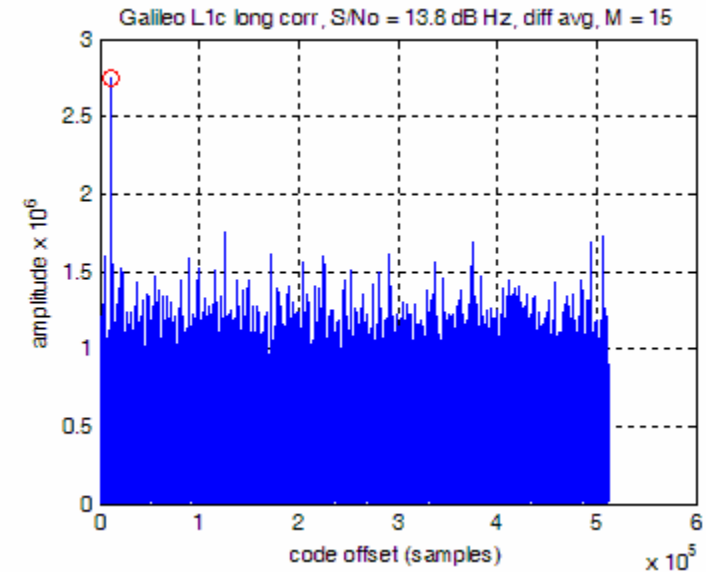

Figure 3.2/1: Differential Averaging for Galileo L1

The achieved sensitivity is close to the objective. Figure 2.5/19 was created for a probability of detection of 0.99. With a frequency offset of 3 Hz the loss in signal is 1.3 dB. With this loss, P_d at the bin edge is 0.84 dB and the average over the bin is .96. No further allowance for Doppler loss is required.

The disadvantage of this procedure is the small frequency tolerance and the resulting large number of bins to be searched. With assistance the frequency unknown is ±117 Hz

(see section 2.8). The number of bins to be searched is 39 resulting in a long acquisition time.

A potential solution is to use the FFT procedure described in section 2.5.4. If we save the result of 32 4 ms correlations we have a 20460 x 32 matrix. After performing the FFT on the columns, the frequency bin spacing is 7.8 Hz. However, considering the 32 bins, the overall frequency tolerance is ±125 Hz. With assistance, the frequency uncertainty is 116 Hz and can be covered in a single correlation.

To meet the requirement of acquisition with an *S/No* = 14 dB Hz, additional power averaging of about 30 times is needed resulting in a correlation time of 3.8 seconds. Considering that a single correlation covers the frequency range, there is a considerable improvement in acquisition time.

The only disadvantage of this procedure is the need to know the timing of the 25 bit short code.

3.3 ACQUISITION PROCEDURES FOR WEAK GPS L5b SIGNALS

The GPS L5 signal has two components. L5a is the data channel and is modulated in-phase. L5b is the pilot channel and is modulated in quadrature.
The GPS L5a and b codes are 10230 chips long and the chip rate is 10.23 Mch/s giving a code length of 1 ms. The L5b code is followed by a 20 chip Neuman Hoffman (NH) code. The L5a code is followed by a 10 bit NH code then data at 50 bps that is rate ½ coded.

As with the other signals, our first objective is to acquire a signal with *S/No*=20 dB Hz. After coherent integration for 1 ms the *S/No* is -10 dB. Power averaging with *M*=5000 would give the desired probability of detection but 5 s is getting long. With a 5 s averaging time, the code Doppler frequency tolerance for 0.5 chips is 11.5 Hz.

Coherent averaging for the length of the NH code results in a signal to noise ratio of 3 dB with an input *S/No* of 20 dB Hz. Further power averaging for 30 times or differential averaging for 20 times is required. The Doppler tolerance for 20 ms coherent averaging is ± 15 Hz.

Knowledge of the NH code alignment is required in order to coherently average for 20 ms. Without prior knowledge, there are basically two ways alignment can be achieved. Conceptually, the simplest is to correlate with one long code including the spreading code and the NH code (20 x 10230 = 204600 chips). With 2 samples per chip this requires a very long matched filter or 409200 point Fourier transforms. The FFT is not computationally difficult but a lot of memory is required. The alternative is to save the results of 20 correlations then search for the NH code alignment that gives the highest output. When the coherent averaging is followed by power or differential averaging it is necessary to save the sums of the 20 trial alignments. The result of the averaging is a matrix of 20 offsets by 20460 delays. With a low signal to noise ratio, it will not be possible to detect the correct alignment until the averaging is completed. There is a further complication with the second technique. The NH code bit will change somewhere in an arbitrary 1 ms sample of the received code. To overcome this problem, we correlate a 2 ms sample with a locally generated code (20460 samples) padded with 20460 zeros.

As with GPS L1 and Galileo L1 the faded signal could have as weak an *S/No* as 14 dB Hz. After correlation of one code length (1 ms) the *S/N* is -16 dB. Since an *S/N* after correlation of 14 dB is required for reliable detection, significant averaging is required. Referring to Figure 2.5/19 we see that coherent averaging for 1 second or 1000 codes would achieve the desired result. This would result in an impossibly tight Doppler tolerance.

Power averaging would overcome the Doppler problem and it would not be necessary to acquire the NH code alignment. However, we see that after averaging for 1000 codes we are still short 10 dB of the objective. The power averaging gain becomes asymptotic to the square root of the number averaged. To gain another 10 dB we need 100 times the

number or 100,000 codes. The code Doppler tolerance for 100 second averaging is 0.57 Hz. Differential averaging requires a somewhat shorter time but is still excessive. Clearly, longer coherent averaging is required.

Coherent averaging for the length of the NH code results in a signal to noise ratio of -3 dB with an input *S/No* of 14 dB Hz. Further power averaging for 300 times or differential averaging for 200 times is required. The Doppler tolerance for 20 ms coherent averaging is ± 15 Hz. After differential averaging for 4 seconds the code Doppler tolerance is ±14 Hz. This appears to be a reasonable compromise.

3.4 ACQUISITION PROCEDURES FOR COMBINED GPS L5a and L5b

Since half the power in the L5 signal is in-phase and half in quadrature, it would seem possible to obtain a 3 dB improvement by correlating with the combined codes.

$v_C = C_I + jC_Q$

If the signal to noise ratio is high enough that no additional averaging is required, the gain will be 3 dB.

Consider the case where correlation for the length of the quadrature NH code (20 ms) is to be performed. In this 20 ms interval, one quadrature NH code will be included and two in-phase codes. The in-phase codes have phase reversals due to the data. The code relationship is shown by Figure 3.4/1:

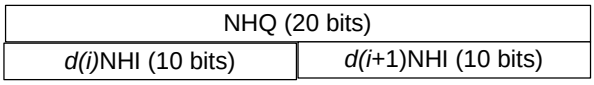

Figure 3.4/1: NH Code Relationship

As can be seen, in the duration of the quadrature code, the in-phase code has four states, plus-plus, plus-minus, minus-plus and minus-minus. If the data is known, a gain of 3 dB can be achieved.

If the data bits are not known, the correlation is performed for all four data bit combinations and the results are saved. The results are then searched for the data bit pair that gives the highest maximum output. The result is a 3 dB improvement if the correct data is found. This technique can be extended to averaging up to four quadrature NH codes. With $M = 2$, there are 16 bit combinations. An exhaustive search yields close to a 3 dB improvement. As M increases, the number of bit combinations increases rapidly and exhaustive searches become less practical. A modified hill climbing search technique was simulated. Since there are data errors resulting from the imperfect search technique, the gain is less than 3 dB. Simulation has shown an improvement of about 1.5 dB with $M = 4$. Beyond 4 there are too many possible data combinations and the improvement is small if any. With $M = 4$ it is possible to detect a signal with a combined *C/No* of 25 dB Hz or an *S/No* for L1b of 22 dB Hz.

One problem with this technique when averaging more than one NH code is it extends the coherent period and results in a tighter frequency offset tolerance. It is worth considering combining the results of in-phase and quadrature power outputs. Consider coherent correlation for the length of the quadrature NH code then power averaging for M codes. The correlation with the spreading code plus quadrature NH code yields a single maximum at the code delay. In the length of the quadrature NH code, the in-phase NH code is repeated twice. If the correlation is performed with repeated NH codes, there are two possible outcomes. If the data does not change, there will be two peaks. If the data changes, the two peaks will cancel and the output is only noise. To overcome this problem, the correlation is performed with one code length padded with zeros. The resulting correlation has two peaks with ½ the amplitude or 1/4 of the power. In state H_0 the noise will have half the power of the noise in the quadrature correlation. To get the benefit of the two peaks, the two halves are added. As the correct alignment could be in the second half, the doubled output is repeated. This process is illustrated by Figure 3.4/2. The resulting distribution is non central chi-squared with twice the degrees of freedom and non centrality factor (λ) twice the value at the correlator output. The probability of detection of the in-phase code is increased by about 1 dB.

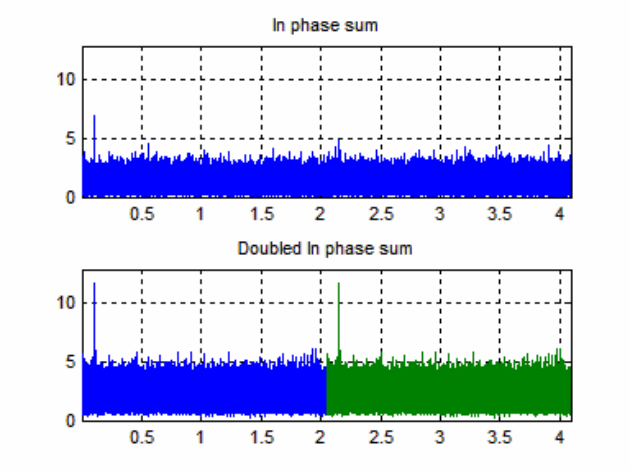

Figure 3.4/2: In-Phase Correlation

The doubled in-phase correlation is added to the quadrature correlation before detection. The result is illustrated by Figure 3.4/3. As can be seen the peak of the quadrature correlation is barely above the noise but when the in-phase combination is added, the peak is easily detected. The improvement gained by this process is not as large as hoped. The improvement over correlating with the quadrature code only is about 1 dB.

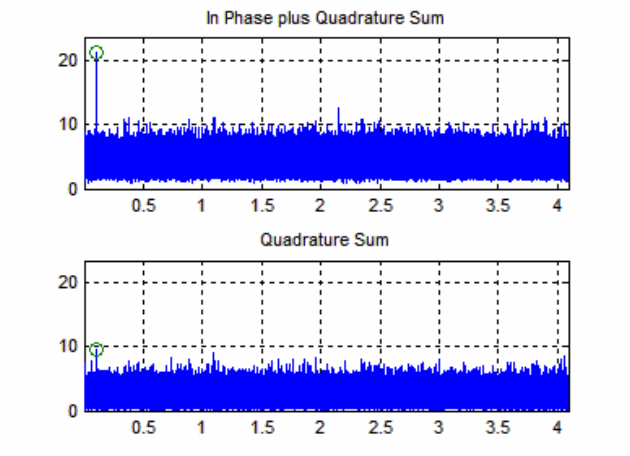

Figure 3.4/3: Quadrature and Combined Correlations

There is a further increase in sensitivity if differential averaging is used for the quadrature code. Using this technique simulation shows that a signal with an *S/No* = 20 dB Hz can be detected with averaging 10 NH codes. A signal with *S/No* = 14 dB Hz can be detected after averaging 100 NH codes. The result is acquisition in half the time that would be required when correlating and power averaging with the quadrature signal only.

4.0 SIGNAL TRACKING

4.1 SIGNAL TRACKING

There are two parts to tracking, delay tracking and phase or frequency tracking. Both of these components have two phases, an acquisition phase and a tracking phase. The acquisition process previously discussed in Chapter 2 finds the initial code alignment to within half of a chip and finds the frequency within bounds depending on the correlation process. The delay tracking loop refines the delay estimate then adjusts the delay as the distance from the satellite varies. Similarly the frequency or phase locked loop refines the received frequency and follows the frequency changes due to Doppler. If the signal to noise ratio is adequate, the phase is accurately tracked and facilitates demodulation of the navigation data.

4.2 DELAY LOCKED LOOPS

The purpose of the delay locked loop is to track the received code to estimate the delay from the satellite to the receiver. Figure 4.2/1 shows the principle of the delay locked loop. After RF and IF processing the received signal, Equation (2.1), is represented as:

$$v(t) = C(t)A(\cos(\delta Ft + \theta) + j\sin(\delta Ft + \theta)) + \eta_I(t) + j\eta_Q(t)$$

The received signal is multiplied by early (E), prompt (P) and late (L) versions of the code. The value of ΔT should be ½ chip or less. The three product signals are accumulated and averaged as required

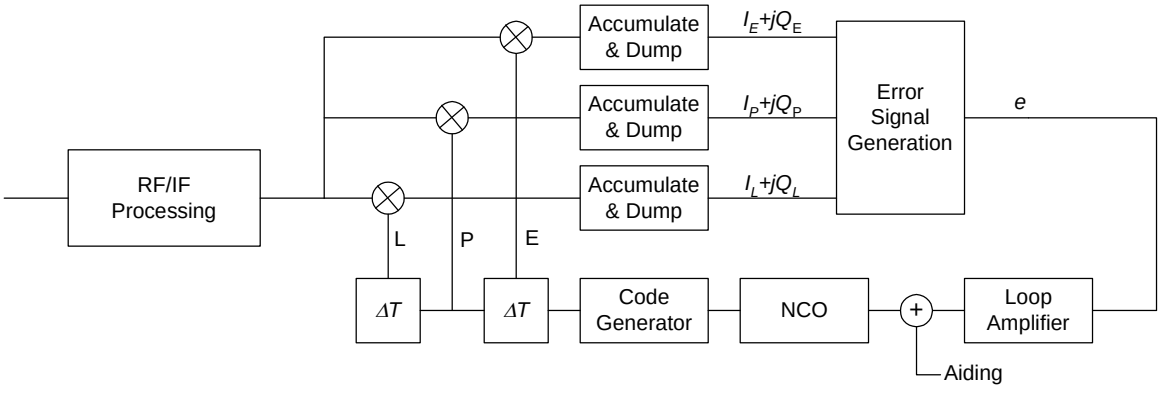

Figure 4.2/1: Delay Locked Loop

The error signal is generated by processing the three averaged outputs. The process used in this work is known as early-late power and is as follows [13]:

$$e = \left(I_L{}^2 + Q_L{}^2\right) - \left(I_E{}^2 + Q_E{}^2\right) \tag{4.1}$$

The error signal has to be scaled to be proportional to the delay and not to the signal strength. In this work the error signal is scaled to the average prompt power.

The loop amplifier filters the signal and multiplies it by a constant to obtain the required loop dynamics. After processing by the loop amplifier, the error signal is used to vary the frequency of a numerically controlled oscillator (NCO). The nominal frequency of the NCO is the code rate multiplied by the number of samples per chip. The NCO then controls the code generation rate. The output of the code generator is twice delayed by ΔT to obtain three versions of the code separated by a fraction of a chip.

4.2.1 Delay Locked Loop for GPS L1 C/A Code

The following error curves were generated with 8 samples per chip. Figure 4.2/2 shows the error curve, often called the S-curve, with early-late spacing of one chip or ΔT = ½ chip (±4 samples). If the signal was unfiltered (with rectangular chips) the curve would be a pair of triangles with maximum ±1 with delay offset of ±4 samples (1/2 chip).

Figure 4.2/3 shows the error curve with early-late spacing of ½ chip. As can be seen the slope is steeper but the peak to peak amplitude is lower. With a poor signal to noise ratio, the 1 chip spacing performs better.

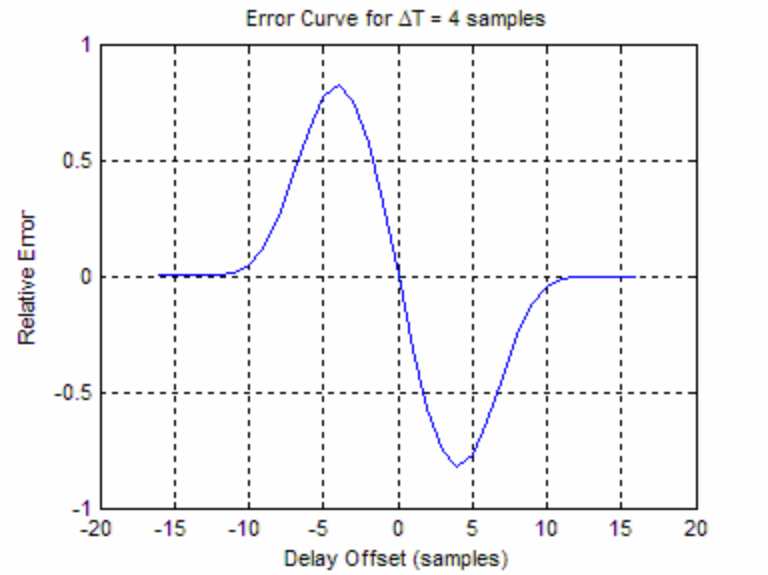

Figure 4.2/2: Error Curve with 1 Chip Spacing

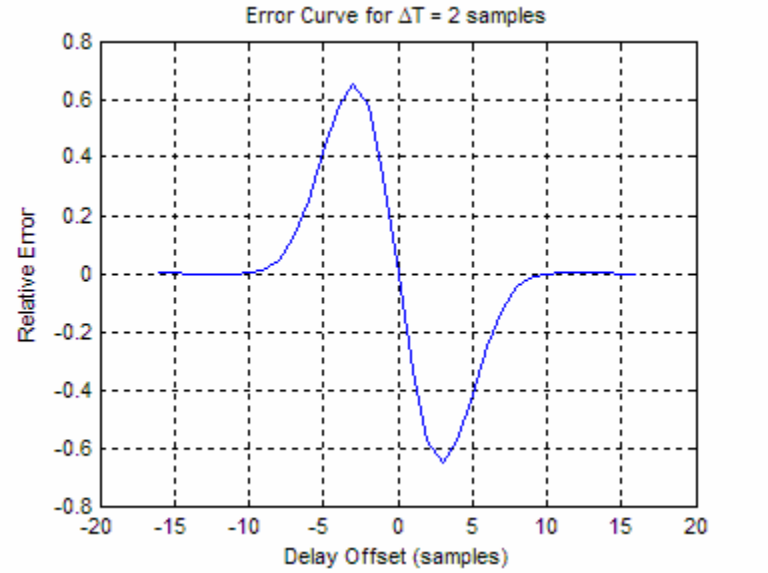

Figure 4.2/3: Error Curve with ½ Chip Spacing

The above curves show the response without noise. In the presence of noise, the curve gets distorted. Figure 4.2/4 shows the result with a good signal to noise ratio (*S/No*=44 dB Hz) and with one chip early-late spacing. With a poor signal to noise ratio, significant averaging is required before the error curve can be used.

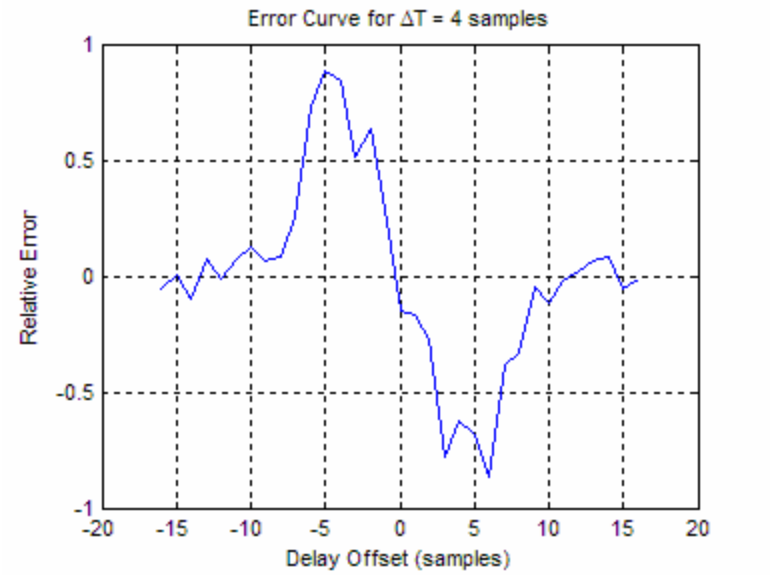

Figure 4.2/4: Error Curve with Good Signal to Noise Ratio

When the signal is weak the tracking loop must be preceded by coherent averaging. A common procedure for GPS L1 is to average for 10 ms. With a *S/No* of 20 dB Hz the *S/N* at the input to the tracking loop is 0 dB. The distribution of noise as a result of early-late power process has the form. $p(x) = \frac{1}{2\sigma^2}\exp\left(\frac{-abs(x)}{\sigma^2}\right)$ (4.2)

The parameter σ is the standard deviation of the noise and is:

$$\sigma = \frac{1}{\sqrt{S/N}} \tag{4.3}$$

This distribution was determined by experiment as shown by Figure 4.2/5 that shows the distribution with a signal to noise ratio of 0 dB.

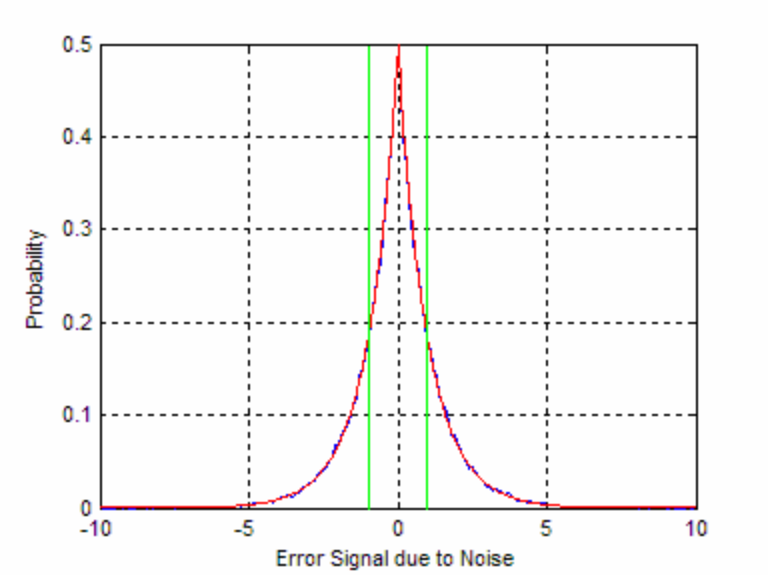

Figure 4.2/5: Distribution of Error due to Noise

The green lines show the maximum theoretical signal due to a delay offset (±1). When the noise exceeds this value it swamps the error signal. Table 4.2/1 shows the fraction of the time that the magnitude of the error exceeds 1 vs. the signal to noise ratio at the input of the tracking loop.

Table 4.2/1: Probability of Error Greater than 1.0

S/N dB	*S/N*	Probability of Error>1
10	10.00	0.00
6	3.98	0.02
3	2.00	0.14
0	1.00	0.37
-3	0.50	0.61
-6	0.25	0.78
-10	0.10	0.90

When the *S/No* is as low as 14 dB Hz longer coherent averaging is required. If the bit timing is known, integration for 20 ms can be performed. It is not necessary to know the data values, only the timing. With an *S/No* =14 dB Hz and coherently integrating for 20 ms the *S/N* at the input to the tracking loop is -3 dB.

The loop amplifier can have a number of different forms. It could simply multiply the error signal by a gain value, it could include filtering and it could include an integrator. The NCO acts as an integrator. Including a second integrator could cause stability problems and compensation would be required. A second integrator would allow the loop to follow a ramping delay. It is conventional in GPS receivers to use aiding as shown by Figure 4.2/1 obviating the need for a second integrator. The aiding can be derived from knowledge of the range rate or can be derived from the carrier frequency and the ratio of carrier frequency to chip rate.

The tracking loop was simulated and measurements were made with various signal to noise ratios. In this simulation the delay resolution is 1/8 of a chip. The trial begins with an offset of 4 samples or ½ chip. Figure 4.2/6 shows the result with *S/No*=20 dB Hz and 20 ms averaging.

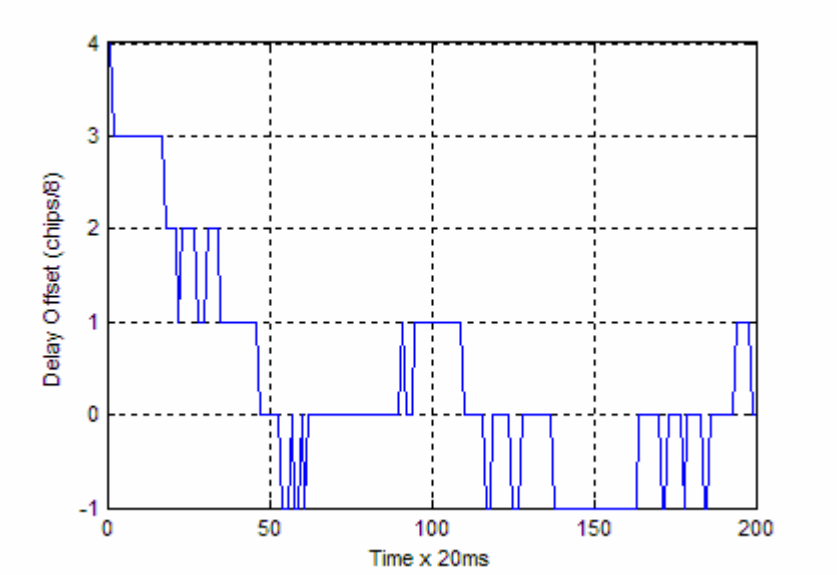

Figure 4.2/6: Delay Locked Loop Simulation with *S/No*=20 dB Hz

At this signal to noise ratio the tracking error is less than ±1 sample or 1/8 chip. Figure 4.2/7 shows the result of the same simulation with *S/No*=14 dB Hz and 20 ms averaging. The tracking error is occasionally as high as 5 samples or 5/8 of a chip. The mean for the example shown is in error by .023 chips which corresponds to an error of 6.75 m. When the errors significantly exceed 0.5 chips the loop can lose lock.

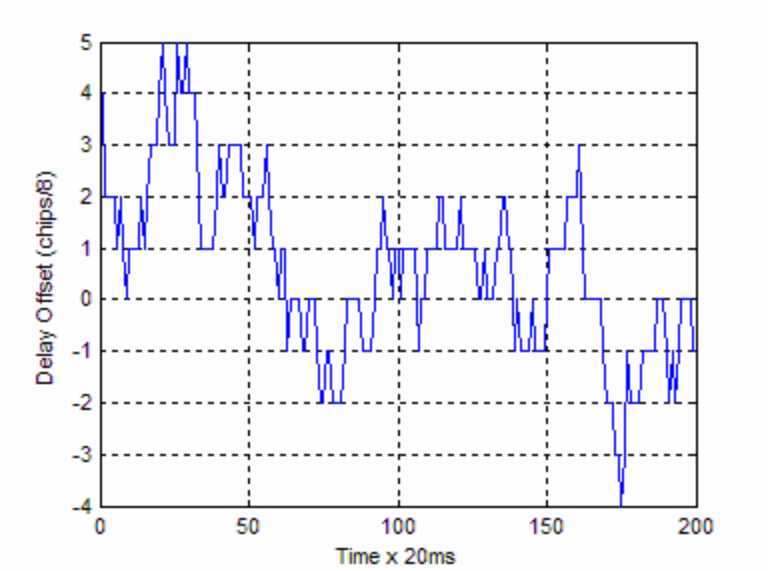

Figure 4.2/7: Delay Locked Loop Simulation with *S/No*=14 dB Hz

When the tracking loop is closed, the error signal is filtered by the loop bandwidth. The closed loop tracking error depends on the input *S/N*, the loop band width and the early-late spacing.

A. J. Dierendonck shows the variance of the tracking error to be:

$$\sigma^2 = \frac{B_L d}{2\,S/N}\left(1+\frac{2}{(2-d)\,S/No\,T_I}\right) \tag{4.4}$$

B_L is the loop bandwidth, d is the early-late spacing in chips and T_I is the integration time preceding the tracking loop.

Table 4.2/2 shows a small sample of calculated and simulated tracking variances. The integration time is 20 ms, the early-late spacing is 1 chip and the loop bandwidth is 0.5 Hz.

Table 4.2/2: Delay Locked Loop Tracking Variance

S/No dB	variance	simulated
29	0.000354	0.0016
24	0.00139	0.00225
20	0.005	0.0036
14	0.0495	0.059

With fairly short tests, 4 s, the results vary significantly from test to test. However the simulations show fairly good agreement with the theory.

4.2.2 Delay Locked Loop for GPS L5

The delay locked loop for GPS L5 is similar to that for the GPS L1 C/A code. The major difference is the chip rate is 10 times faster and the code is 10230 chips long instead of 1023 chips. The quadrature component is modulated by a 20 bit NH code but no data. If coherent averaging is used for the length of the NH code, the integration time at the input to the tracking loop is 20 ms as it is for L1.

4.2.3 Delay Locked Loop for Galileo

Galileo L1 has three components. L1a is for government use only, L1b is the data signal and L1c is the pilot channel. Initial acquisition and tracking is performed using L1c as there is no data modulation. The integration time when correlating for the length of the short code is 100ms.

The effect of the BOC on L1c is to produce a correlation amplitude response as shown by Figure 4.2/8. This Figure shows the ideal response with unfiltered signals and no noise. The correlation amplitude has a relative value of -.5 with ½ chip offset then returns to zero with one chip offset.

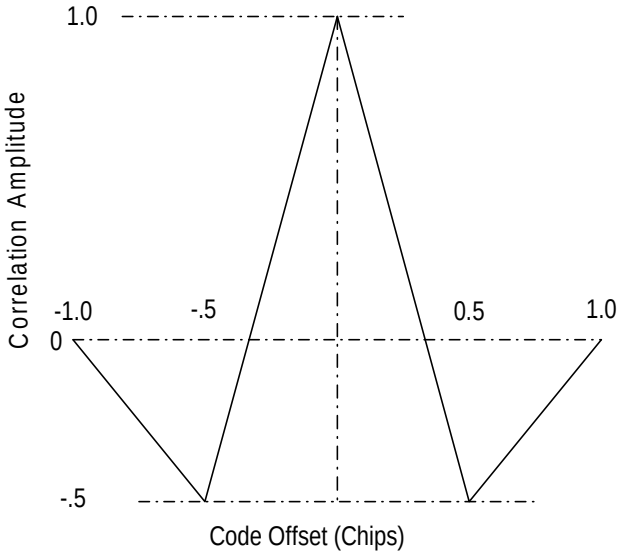

Figure 4.2/8: Correlation for Galileo L1c

When the signal is filtered the power response is as shown by Figure 4.2/9.

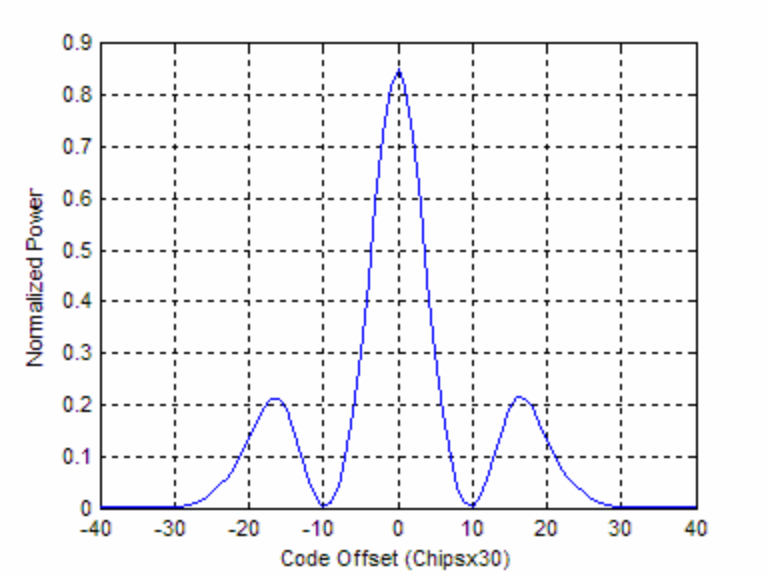

Figure 4.2/9: Correlation Power Response

With the early late spacing equal to ½ chip the S-curve has the shape shown by Figure 4.2/10. If the early-late spacing is greater than ½ chip, the S-curve has a kink in the middle as shown by Figure 4.2/11. This, of course is undesirable limiting the early-late spacing to ½ chip.

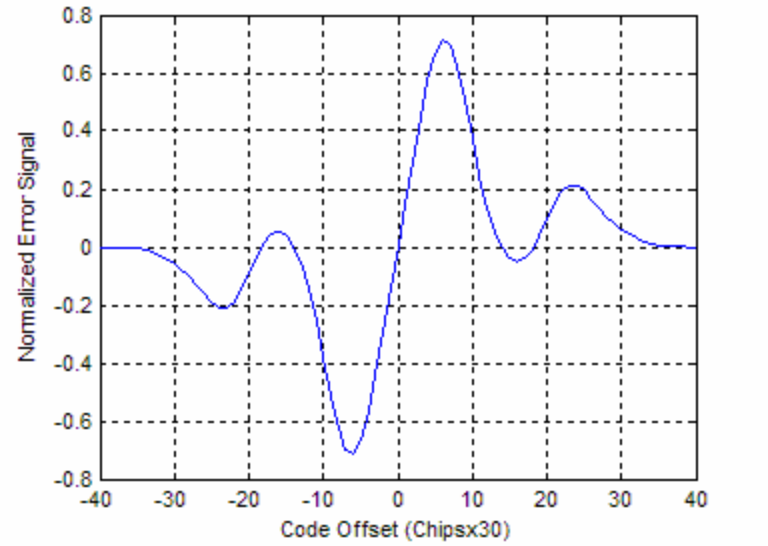

Figure 4.2/10: S-curve with Early-Late Spacing 1/2 Chip

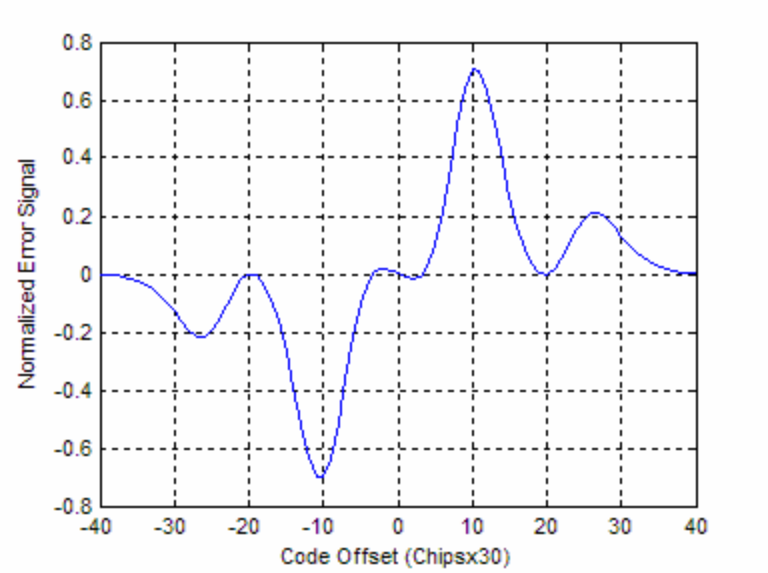

Figure 4.2/11: S-curve with Early-Late Spacing 2/3 Chip

Figure 4.2/12 shows the tracking error as a function of time. In this simulation there are 30 samples per chip. A peak error of 4 samples is .133 chips or about 40 m. The average error is about 1 m.

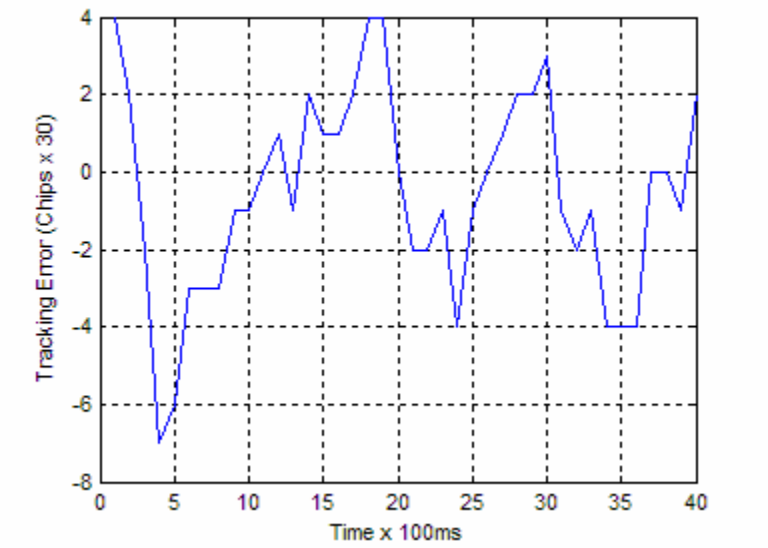

Figure 4.2/12: Delay Error as Function of Time with *S/No* =14 dB Hz

4.3 PHASE AND FREQUENCY TRACKING

The discussion and results for delay tracking assumed there was no frequency error. Since the carrier frequency is varying with time, frequency tracking is required. Considering that it is desirable to demodulate the data, phase tracking is often employed.

Frequency tracking is used to reduce the frequency error to allow the phase locked loop to acquire. Frequency tracking is more robust than phase tracking with low signal to noise ratios.

4.3.1 Phase and Frequency Tracking for GPS L1

The first requirement for frequency tracking is a frequency offset detector. In a digital system, frequency is detected by computing the rate of change of phase or the phase change between samples. This is implemented by comparing the phase of a sample to the phase of the previous sample. The most robust implementation is as follows:

$$ferr = RI_P - IR_P \tag{4.5}$$

R and I are the real and imaginary components of the current sample, R_P and I_P are the real and imaginary components of the previous sample. With GPS L1, if we know the data bit timing we can sample at 50 bps. The frequency error curve is as shown by Figure 4.3/1.

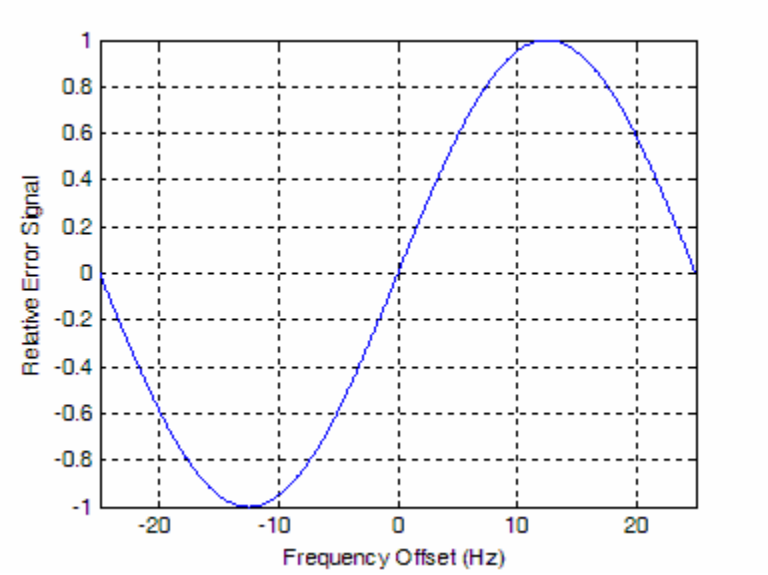

Figure 4.3/1: Frequency Error Curve for 50 samples/s

The frequency offset range is ±sample rate divided by two or ±25 Hz. The curve peaks at ±12.5 Hz. The problem with this implementation is that data changes between samples reversing the sense of the error signal. The data problem can be overcome by modifying the process as shown by the following equation:

$$ferr = \left(RI_P - IR_P\right)\mathrm{sgn}(II_P + RR_P) \tag{4.6}$$

The addition of the sgn term solves the data problem but reverses the sense of the error signal when the frequency offset exceeds the sampling rate divided by four. This effect is shown by Figure 4.3/2.

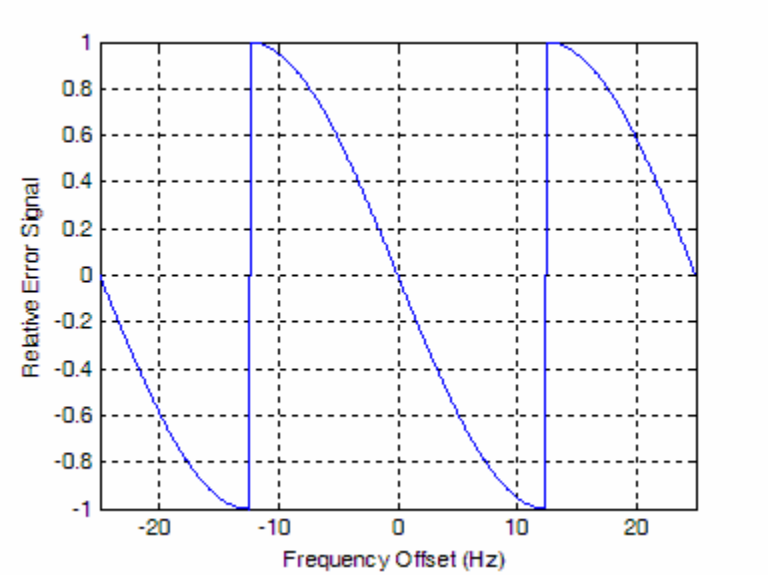

Figure 4.3/2: Frequency Detector with Sign Term

The frequency locked loop was simulated with a single integrator. The rate of change of Doppler frequency is low enough that a tracking loop with a single integrator is adequate. Figure 4.3/3 shows the time response of the loop with an initial frequency offset of 10 Hz.

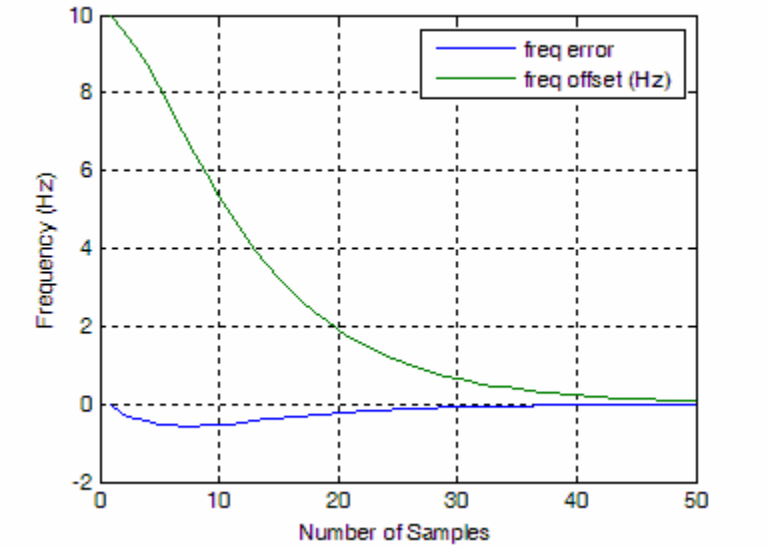

Figure 4.3/3: Time Response of Frequency Tracking Loop

The frequency offset reduces smoothly to a low value. The error signal begins with a negative value then reduces to zero when thefrequency error nears zero.

Figure 4.3/4 shows the result of the same loop without the sgn term in the frequency detector. The data modulation reverses the sense of the error signal alternately driving the frequency offset higher and reducing it.

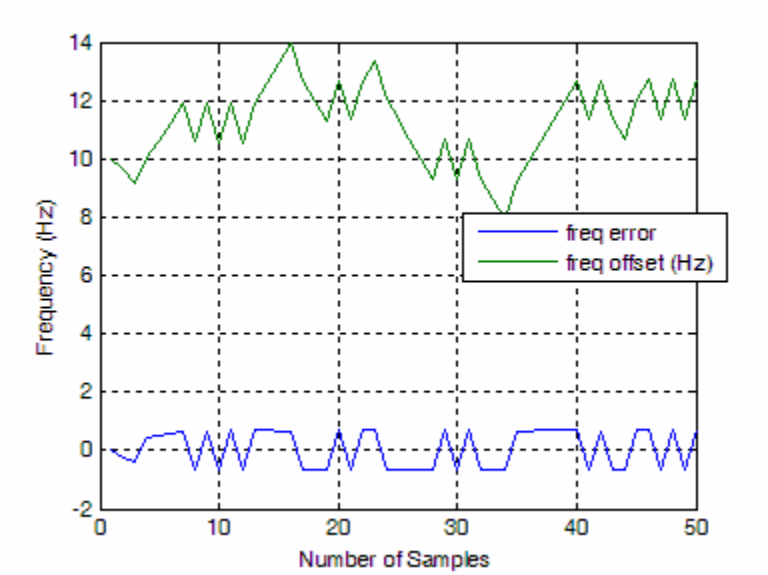

Figure 4.3/4: Time Response of Frequency Tracking Loop without Data Compensation

Figures 4.3/3 and 4.3/4 showed the response with a high signal to noise ratio. Figure 4.3/5 shows the time response with *S/No*=20 dB Hz. The noise adds errors to the frequency error signal. Sometimes the errors are large enough to prevent the loop from locking.

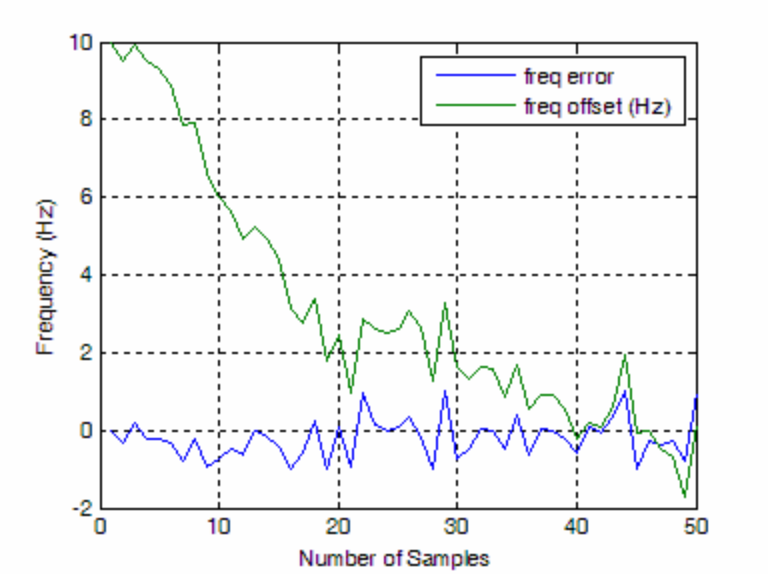

Figure 4.3/5: Time Response of Frequency Tracking Loop with *S/No*=20 dB Hz

When the signal is strong enough it is desirable to add a phase locked loop. Phase lock permits data demodulation and facilitates more accurate frequency control. There are two derivations of phase error signal that can be used. The first is simply the real (or imaginary) component of the signal normalized by dividing by the amplitude. The second is the product of real and imaginary components of the signal divided by the square of the amplitude. The real-only error signal is sensitive to data modulation as the slope of the error curve is reversed for a phase change of π radians. The product signal (RxI) is insensitive to a change of π radians as the slope remains the same. Figure 4.3/6 shows the response of the two error signals as a function of phase error.

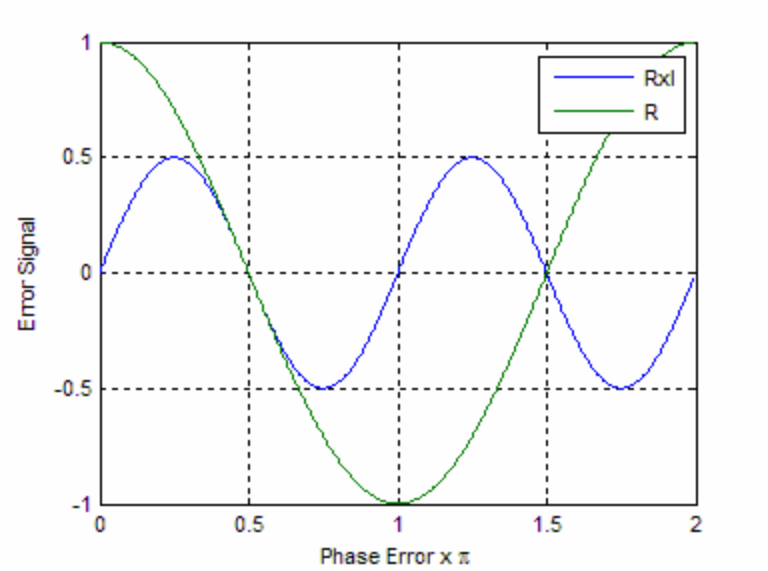

Figure 4.3/6: Phase Error vs. Phase Offset

Figure 4.3/7 shows one implementation of the phase locked loop.

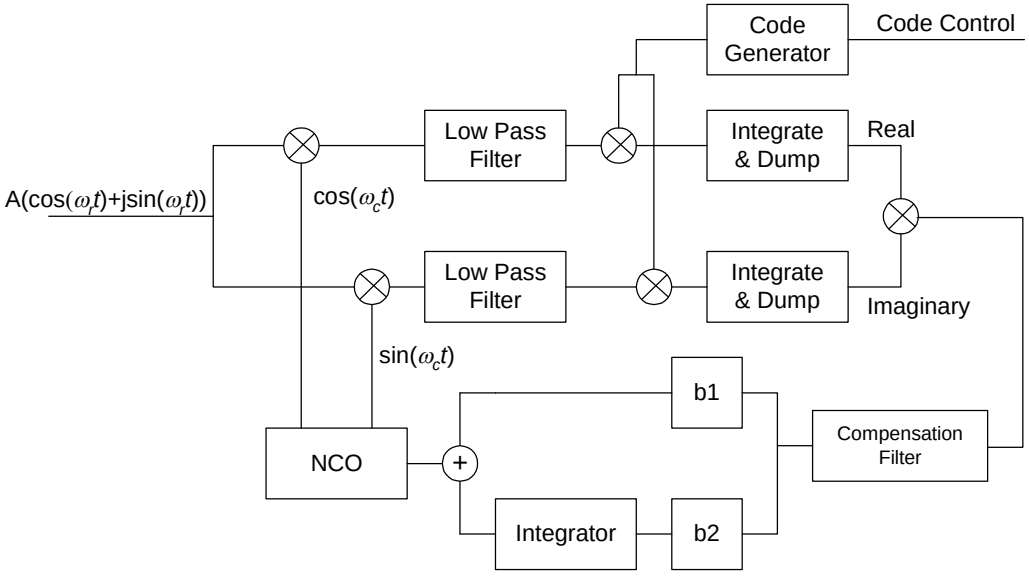

Figure 4.3/7: Phase Locked Loop

The received signal is split and one branch is multiplied by the cosine of the local oscillator signal and the other branch is multiplied by the sine. The signals are filtered then multiplied by the locally generated code. The product signals are coherently integrated for the averaging period. The code timing is controlled by the delay locked loop described in section 4.2.

The phase error signal is generated by multiplying the real or in-phase signal by the imaginary or quadrature signal. The error signal is processed then used to control the frequency of the NCO. The NCO behaves as a phase integrator. A second integrator with gain b_2 is required to track frequency offsets. With the two integrators, compensation is needed to prevent oscillation. The compensation used is a simple network with zeros at $a(\cos(\theta)+ j\sin(\theta))$ and $a(\cos(\theta)- j\sin(\theta))$. In the simulation a = 0.2 and θ = π/12. This simple network adds a phase lead compensating for part of the 2π phase lag from the integrators.

For the GPS L1 signal, the correlator outputs are coherently integrated for 20 ms before computing the phase error signal. Figure 4.3/8 shows the time response with an *S/No* of 29 dB Hz. The frequency offset starts at 10 Hz and rapidly reduces to a small value. There are some excursions of ±4 Hz when the phase tracking begins. The blue dots show the exclusive-or of the demodulated data and the transmitted data. As expected, there are no bit errors.

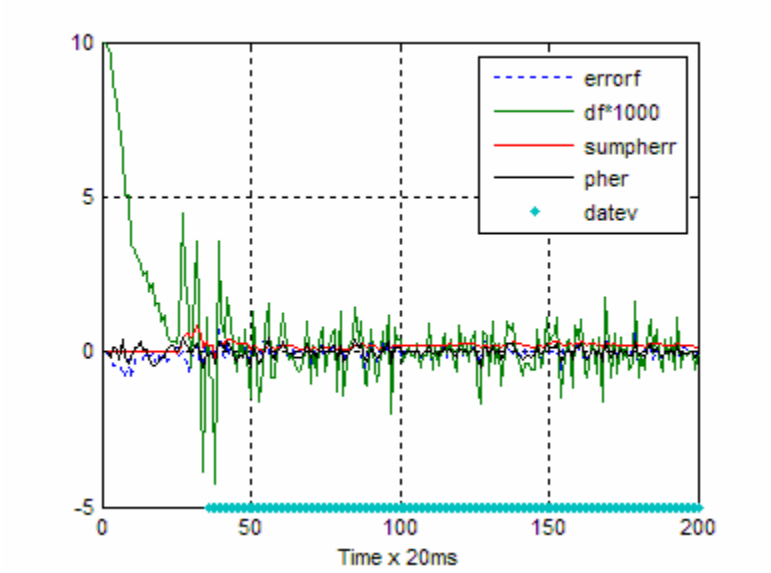

Figure 4.3/8: Time Response of Phase Locked Loop with *S/No*=29 dB Hz

Figure 4.3/9 is a scatter plot showing the locus of the samples in the complex plane. The few samples that appear in the real quadrants occur during the acquisition period.

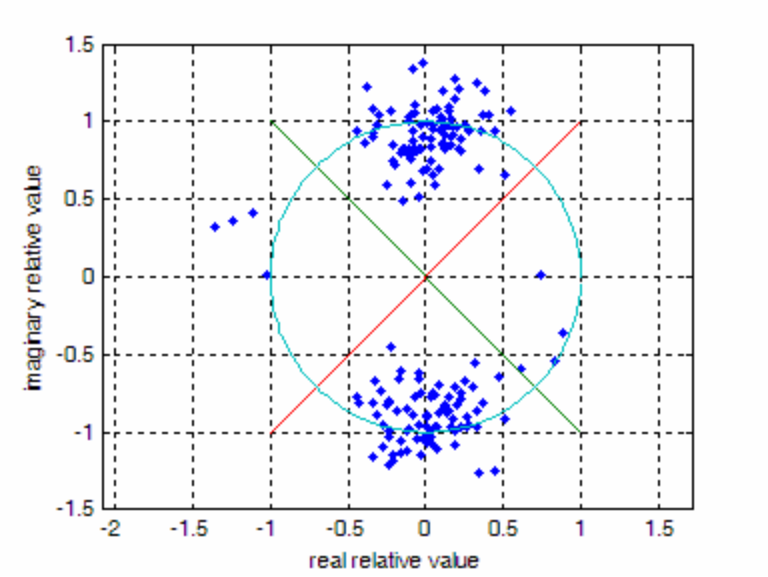

Figure 4.3/9: Scatter Plot with *S/No*=29 dB Hz

Figure 4.3/10 shows the time response with *S/No*=20 dB Hz. The frequency offset has larger excursions and we see there are significant bit errors. The Doppler loss for a 4 Hz frequency offset with integration time of 20 ms is negligible.

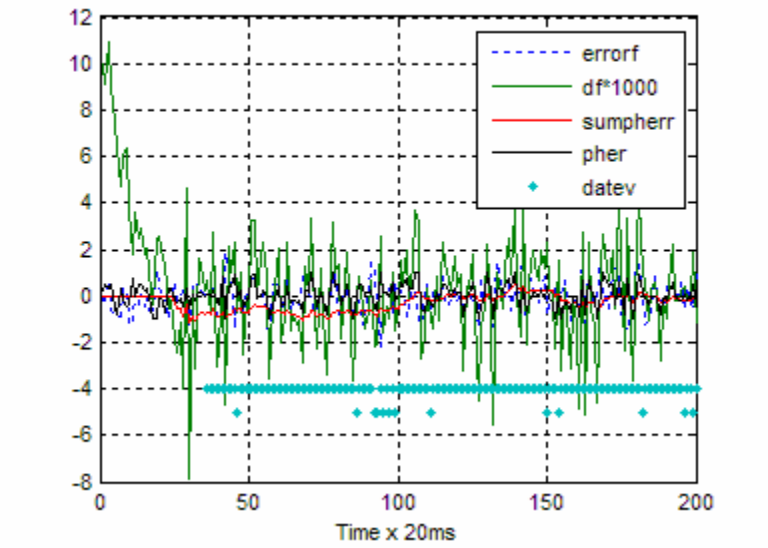

Figure 4.3/10: Time Response of Phase Locked Loop with *S/No*=20 dB Hz

With S/No as low as 14 dB Hz the probability of the loop locking to the signal is about 50%. However if the loop is locked at a higher *S/No* it will continue to track the signal as the *S/No* reduces to 14 dB Hz. Figure 4.3/11 shows the time response as the *S/No* reduces from 29 to 14 dB Hz in the period from sample 50 to 200.

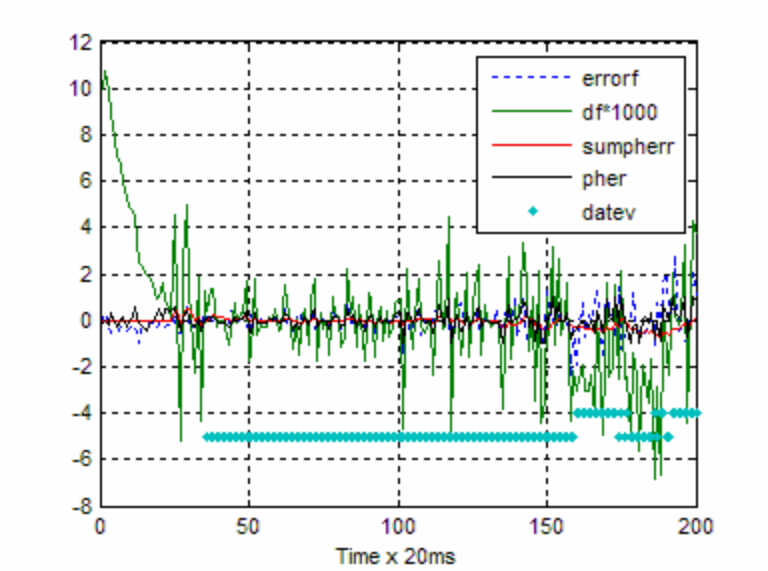

Figure 4.3/11: Time Response with *S/No* Reducing from 29 to 14 dB Hz

Table 4.3/1 summarizes the results of the combined tracking loops. Two entries are included for *S/No*=14. The first try the loop locked to the signal. In the second try the loop obtained a false frequency lock. Recall that no usefull output occurs when BER = 0.5

Table 4.3/1: Tracking Loop Performance

S/No (dB Hz)	Mean Delay Offset (chips)	Variance of Delay Offset (chips^2)	RMS Frequency Offset (Hz)	Bit Error Ratio
44	-.011	.00125	.146	0
29	-.015	.0017	.895	0
20	0.0125	.0068	1.98	.079
14	-.023	.066	4.15	.48
14	-.1	.066	29.5	.47

4.3.2 Phase and Frequency Tracking for GPS L5

The frequency and phase of the quadrature signal can be tracked after averaging for the length of the NH code. This results in a sampling rate for the tracking loop of 20 ms. The frequency error can be detected without the sign term as there is no data modulation. The error curve is as shown by Figure 4.3/1. The loop will acquire the signal with a frequency offset of ± 15 Hz. Figure 4.3/12 shows the time response of the loop with *S/No*=25 dB Hz. Figure 4.3/13 shows the response of the same loop with *S/No*=20 dB Hz. The loop will acquire and remain locked with *S/No* as low as 14 dB Hz as shown by Figure 4.3/14.

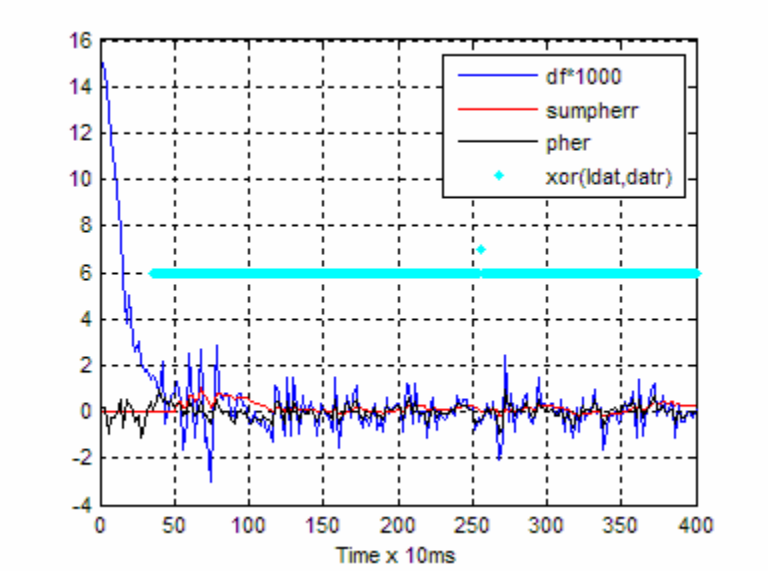

Figure 4.3/12: Time Response of GPS L5 Tracking Loop with *S/No*=25 dB Hz

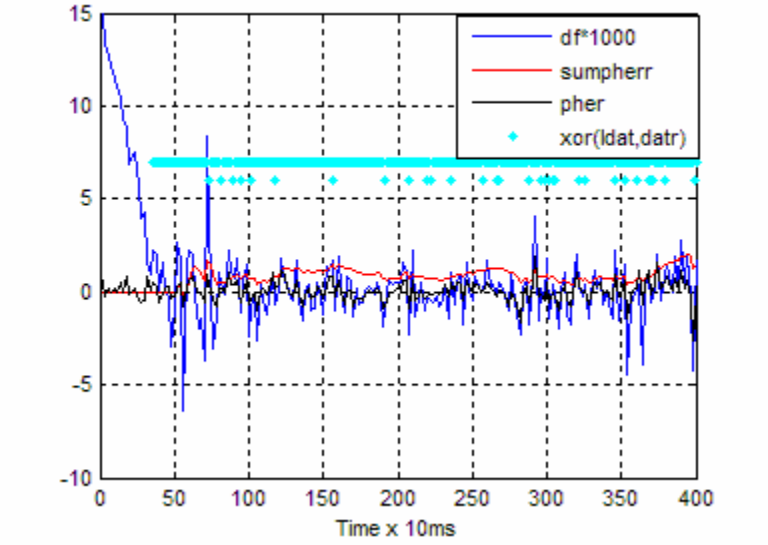

Figure 4.3/13: Time Response of GPS L5 Tracking Loop with *S/No*=20 dB Hz

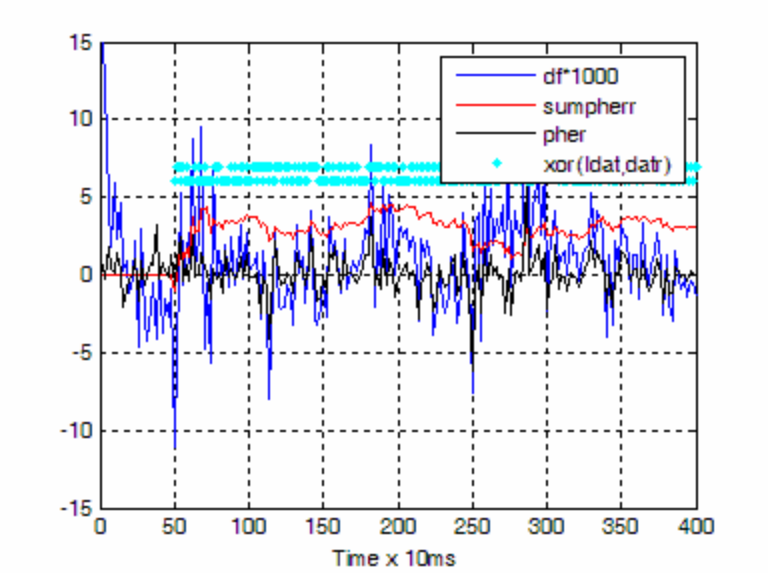

Figure 4.3/14: Time Response of GPS L5 Tracking Loop with *S/No*=14 dB Hz

Once the tracking loop has acquired phase lock on the quadrature channel, the data on the in-phase channel can be detected. Figure 4.3/15 is a scatter plot of the samples shown on the complex plane. In this example the blue dots represent the samples of the quadrature signal after the phase has been acquired. The red dots are taken during the acquisition period. The blue dots are clustered around +1 on the real axis. The cyan dots are samples of the in-phase signal taken every 10 ms. The dots are clustered around ± 0.5 as the averaging time is half of the quadrature channel. The data is detected as the sign of the imaginary part of the in-phase samples.

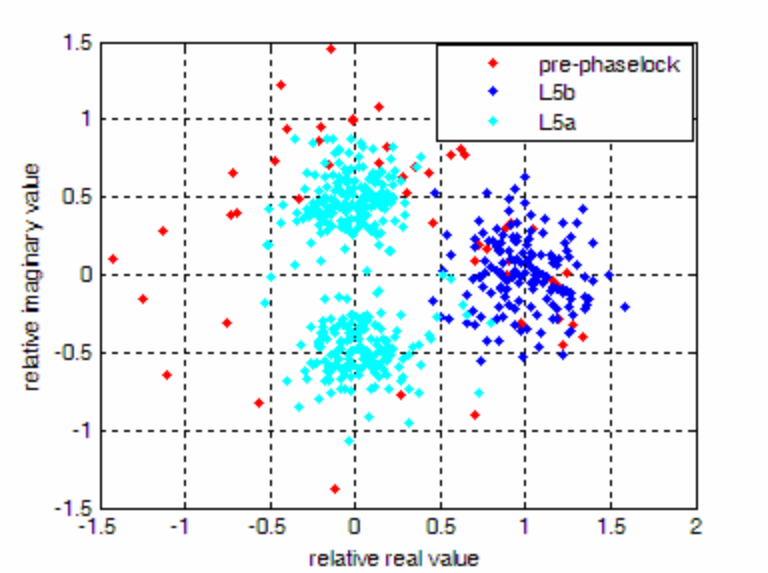

Figure 4.3/15: Scatter Plot of GPS L5 Samples

4.3.3 Phase and Frequency Tracking for Galileo L1

If the Galileo L1c correlator outputs are coherently integrated for the length of the short code the integration time is 100 ms and the sample rate is 10 sps. Figure 4.3/16 shows the frequency error curve for this sample rate. The sign correction term is not required because there is no data modulation on L1c. The ±3 Hz frequency offset used for acquisition is just beyond the peak of the error curve. The frequency locked loop will acquire with this offset.

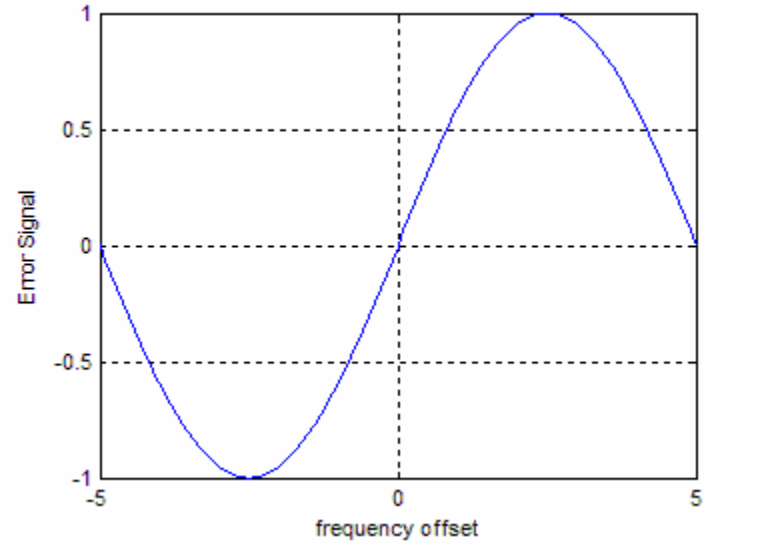

Figure 4.3/16: Frequency Error Curve with Sampling Rate 10 sps

Figure 4.3/17shows the time response of the frequency locked loop with *S/No*=29 dB Hz. The loop will acquire and track a signal with *S/No* as low as 14 dB Hz. Since the L1b signal is modulated on the same carrier and the code timing is the same, we can track the phase of L1c and decode the data on L1b.

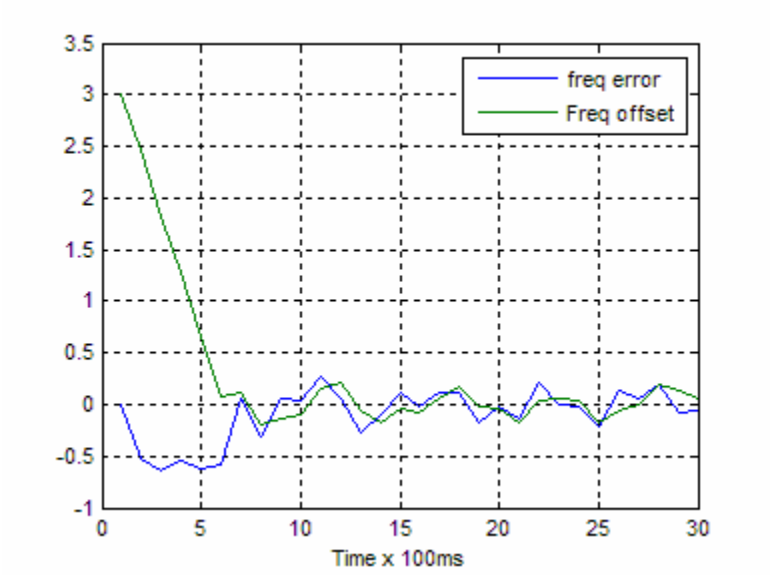

Figure 4.3/17: Time Response of Galileo L1c Frequency Locked Loop

Figure 4.3/18 shows the time response of the phase locked loop with data demodulation of L1b with *S/No*=29 dB Hz. The exclusive-or of the transmitted data and the received data shows that there are 7 data errors giving a bit error rate of .008. This error rate performance is about 1 dB worse than theoretical. This degradation is due to the filtering and sampling loss.

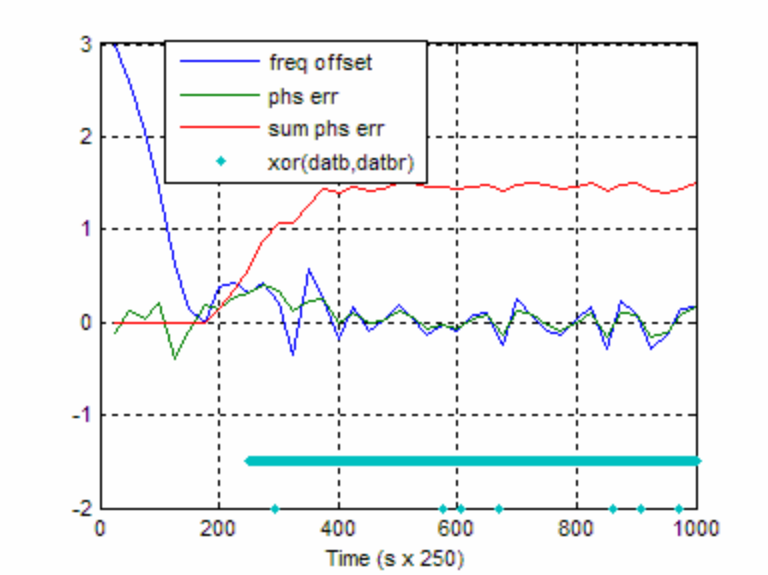

Figure 4.3/18: Time Response of Galileo L1c Phase Locked Loop with *S/No*=29 dB Hz

Figure 4.3/19 shows the response of the same loop with an *S/No*=14 dB Hz. Although the frequency and phase excursions are larger, the loop does acquire the signal and remains locked. The bit error rate is close to 0.5 as expected as the *Es/No* is -10 dB.

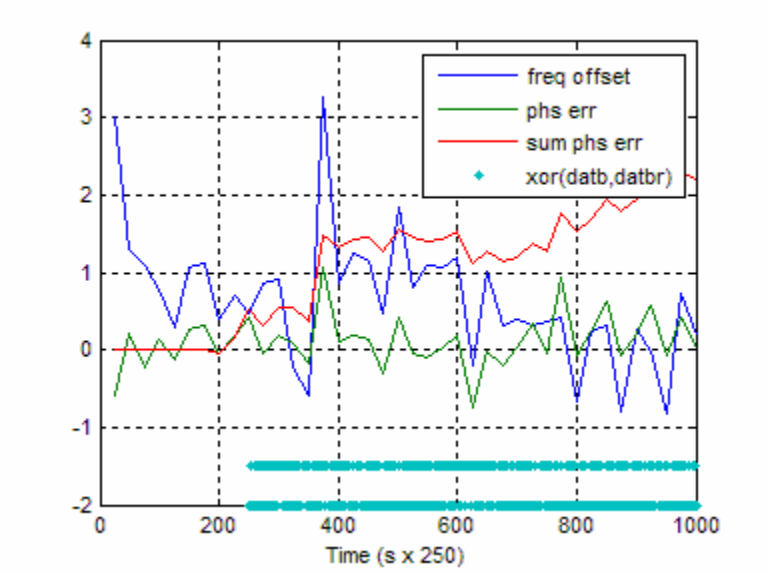

Figure 4.3/19: Time Response of Galileo L1c Phase Locked Loop with *S/No*=14 dB Hz

5.0 SUMMARY and CONCLUSIONS

We have shown that weak GNSS signals can be acquired and tracked. With some assistance, the acquisition can be performed within a few minutes. Further assistance can increase the sensitivity and shorten the acquisition time. Given this ability to acquire and track weak signals, GNSS is suitable for cellular phone location in a large percentage of situations and is the only viable option in many rural locations.

Section 1.4 defines the expected minimum received signal levels for cellular phones with an unobstructed path to the satellites. This section also includes a discussion, with references, of the expected attenuation in various environments. It is concluded that signals with *S/No* =14 dB Hz or greater are expected in most locations except in large buildings or multi-level parking garages.

Chapter 2 explains the theory of correlation and averaging that is required for acquisition of weak GNSS signals. A comparison of averaging techniques, showing the required averaging for acquisition given an input signal to noise ratio, is developed and presented.

Section 2.7 analyzes the acquisition time for weak and strong signals. The importance of assistance is shown. The acquisition times discussed in this section would apply if the phone was just turned on prior to the emergency call. However, most users leave their phone on except when it is not desired to receive calls. In this case, satellites can be acquired as they appear and tracked until no longer visible.

Section 2.8 discusses the assistance required for acquisition of weak signals. It is shown that the minimum level of assistance is not excessive. Further assistance is shown to decrease the acquisition time. The minimum level of assistance is the position and range rate of the visible satellites. Navigation data to derive the precise satellite position in relation to the earth is required to locate the mobile. This data can be kept at the central location where the location of the mobile is determined. Further assistance for GPS L1 could include data and data timing.

Chapter 3 presents optimum correlation and averaging techniques that enable the acquisition of weak signals. This section uses the theory developed in Section 2 to show that signals as weak as *S/No* = 14 dB Hz can be acquired in an acceptable time. Procedures for acquisition of GPS L1 C/A code, GPS L5 and Galileo L1 are presented. It is shown in Section 3.1 that with minimal assistance GPS L1 C/A code can be acquired using a combination of coherent and non-coherent averaging. With the availability of data and data timing, the acquisition time can be significantly decreased by employing Fourier transforms in the averaging process. Galileo L1 and GPS L5 signals include a data-less or pilot channel. It is shown that the weak signals can be acquired with a combination of coherent and non-coherent averaging. Section 3.4 shows methods of acquiring a combination of the GPS L5a and L5b codes. It is shown that an increase in sensitivity of about 1 dB can be achieved.

Chapter 4 shows that the weak signals can be tracked once they have been acquired. The tracking techniques may not be optimum but are adequate for the intended purpose.

Figure 5.0/1 summarizes the environments and shows the likely *S/No* for each signal in these environments. This data is provided with references in Section 1.4. Table 1.4/3 shows the expected *S/No* for the three signals of interest under clear sky conditions. As explained, the specified signal strengths are used for Galileo and GPS L5. The measured signal strength for GPS L1 is somewhat higher and is used in the analysis. There is reason to believe that the measured GPS L5 signals will be as strong as GPS L1.

Stand alone or unassisted location is limited by the ability to demodulate the navigation data. For signals weaker than required for stand alone operation, assistance is required.

The final rows of the figure show the range of environments for which the three signals can be acquired and tracked.

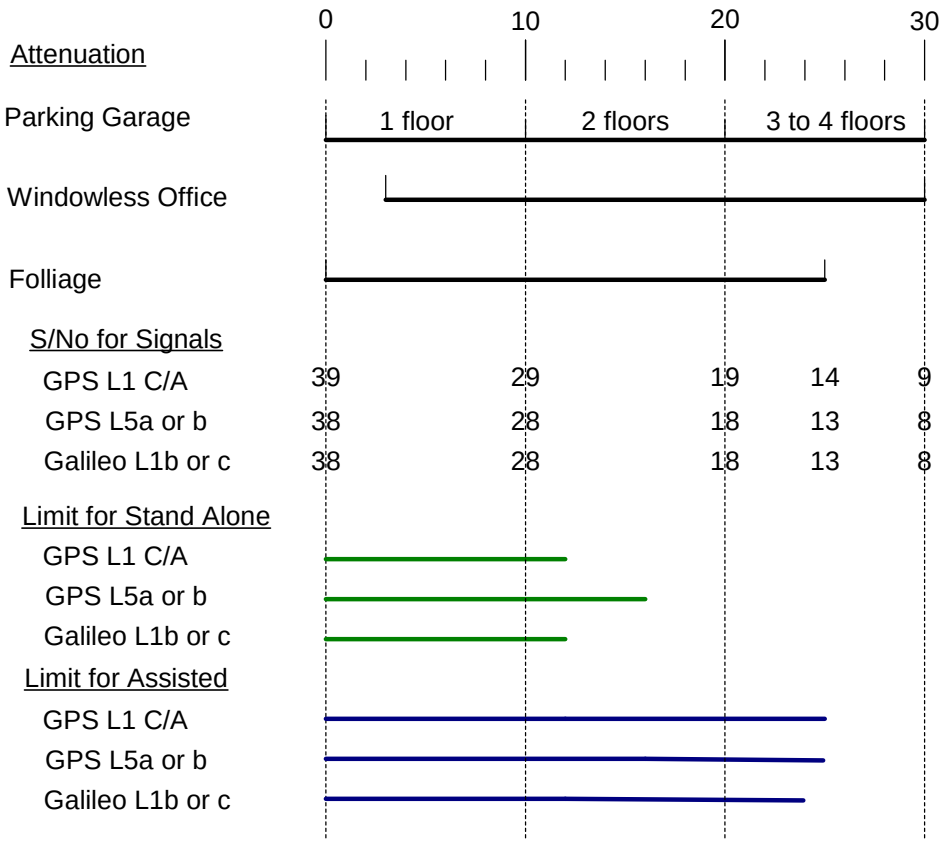

Figure 5.0/1: Capability of GNSS for Cell Phone Location in Various Environments

Signals that are too weak to be acquired and tracked by cellular phones have been detected. An increase in sensitivity of the phone would increase the range of environments where location can be determined.

In summary, GNSS is suitable for cellular phone locations in almost all rural locations and most urban locations. In some urban locations the signal will be too weak. In these urban locations there is likely to be adequate cellular coverage and techniques such as uplink time difference of arrival (UTDOA) can be used.

It is clear that an increase in receiver sensitivity would expand the use of GNSS for cellular phone location. There is already a lot of research in methods for weak signal acquisition and tracking although most of it has been aimed at hand held or mobile GPS receivers rather than cellular phones receiving GPS signals. Little improvement in this field is expected to be gained.

One area that could yield significant improvement is research in multi purpose antennas. What is needed is a phone antenna that has adequate performance for cellular signals and improved performance for GNSS signals. A gain of close to 3 dB could be obtained if the antenna was optimized for right hand circular polarization. Further gains can be obtained by shaping the antenna pattern to reduce the gain below the horizontal and increasing above. The potential gain can be seen in the relative performance of a GNSS receiver vs. a mobile phone receiver as shown in Table 1.4/3.

Another area that could yield some benefits is research into the other signals that are transmitted by the satellites. GPS has a third signal denoted L2. There are several alternate forms of L2 that will be available. One or more of these may be advantageous for weak signal acquisition.

Galileo will have two other carriers, E5 and E6. E6 will not be available to the general public. E5 carries four signals. Two of these signals are pilots and the other two have data. Potentially one or more of these signals could prove advantageous. Combinations of E5 signals or E5 and L1 signals could yield improvements in sensitivity or speed of acquisition.

REFERENCES

Books

[1] Parkinson, B. W. & J. J. Spilker eds. (1996) "Global Positioning System: Theory and Applications, Volume 1", American Institute of Astronautics and Aeronautics, Inc., Washington DC.

[2] A. Papoulis and S. Unnikrishna Pillai, "Probability, Random Variables and Stochastic Processes, Fourth Edition", McGraw-Hill, 2002

[3] A. V. Oppenheim, R. W. Schafer with J. R. Buck," Discrete-Time Signal Processing, Second Edition", Prentice Hall, 1998

[4] James Bao-Yen Tsui, "Fundamentals of Global Positioning System Receivers, A Software Approach", John Wiley & Sons, 2000

[5] M.K. Simon, J.K Omura, R.A Scholtz, B.K Levitt, "Spread Spectrum Communications Handbook", McGraw-Hill Inc.

Official GNSS Documents (Available on-line from multiple sources)

[6] "Navstar Global Positioning System, Interface Specification", IS-GPS-200D, Navstar GPS Joint Program Office, 2004

[7] "Navstar GPS Space Segment/User Segment L5 Interfaces", IS-GPS-705, ARINC Engineering Services, 2005

[8] "Signal-In Space Interface Control Document, SIS-ICD", D. Flachs and Signal Team, Galileo Industries, 2006

[9] "Galileo SIS Updates (MBOC, Component Power Sharing in E1 and E6, EIRP)", Th. Burger, Directorate Galileo Programme and Navigation related Activities, ESA, 11/09/08

Papers

[10] D. M Akos and J. B. Y. Tsui, "Design and Implementation of a Direct Digitization Front End", IEEE Transactions on Microwave Theory and Techniques, vol. 44, no. 12, December 1966, pp 2334-2339

[11] D. Borio, L. Camoriano, L. Lo Presti, “Impact of Acquisition Searching Strategy on the Detection and False Alarm Probabilities in a CDMA Receiver”, IEEE/ION PLANS 2006, San Diego, CA

[12] D. Borio, C. Gernot, F. Machi , G. Lachapelle, “The Output SNR and its Role in Quantifying Signal Acquisition Performance” , Proceedings of the European Navigation Conference, Toulouse, 23-25 April, 2008

[13] M. S. Braasch, A. J. van Dierendonk, “GPS Receiver Architectures and Measurements”, Proceedings of the IEEE, vol. 87, No 1, January 1999

[14] R. Bryant, “subATTO™ Indoor GPS – Pitfalls, Solutions and Performance Using A Conventional Correlator”, GPS Solutions, Vol. 6 No. 3, December 2002

[15] R. Challamel, P. Tome, D. Harmer, S. Beauregard, “Performance Assesment of Indoor Location Technologies”, ION PLANS 2008, session A4-Indoor/Personal Navigation

[16] F. van Diggelen, “Indoor GPS theory and implementation”, IEEE Position & Navigation Symposium, 2002

[17] F. van Diggelen, “Global Locate Indoor GPS Chipset & Services”, ION GPS 2001, Salt Lake City, USA

[18] H. A. El-Natour, A.-C. Escher, C. Macabiau, O. Juien, M. Monnerat, “GPS Acquisition Solution For Use Cases in all Types of Environments”, ION GNSS, International Technical Meeting of the Satellite Division, Fort Worth, TX, 2006

[19] S. C. Fisher and K. Ghassemi, “GPS IIF – The Next Generation”, Proceedings of the IEEE, vol. 87, no 1, January 1999

[20] H. Grant and D. Dodds, “A New Comparison of Averaging Techniques Used for Weak Signal Acquisition with Application to GPS L5 Signals, ION GNSS, Savannah, GA, September, 2009

[21] G. Gunnarsson, M. Alanen, T. Rantelainen, V. Ruutu and V-M. Teittinen, “Location Trial System for Mobile Phones”, 1998 IEEE

[22] M. Hafizi, S. Feng, T. Fu, K. Schultz, R. Ruth , R. Swab, P. Karlsen, D Simonds and Q. Gu, “RF Front End of Direct Conversion Receiver RFIC for CDMA-2000”, IEEE Journal of Solid State Circuits, vol. 39, no. 10, October 2004, pp 1622-1632

[23] J.H.J Iinatti, “On the Threshold Setting Principles in Code Acquisition of DS-SS Signals”, IEEE Journal on Selected Areas in Communications, vol. 18, no. 1, January 2000

[24] M. W. Ingalls, D. Smith, “Microstrip Antennas for GPS Applications”, IEEE 2002

[25] P. L. Kazemi, C. O’Driscol, “Comparison of Assisted and Stand-Alone Methods for Increasing Coherent Integration Time for Weak GPS Signal Tracking”, ION GNSS 2008, Session C4, Savannah, GA

[26] O. Kivekas, J. Ollikainen, T. Lehtiniemi, and P. Vainikainen, “Bandwidth, SAR and Efficiency of Internal Mobile Phone Antennas”, IEEE Transactions on Electromagnetic Compatibility, vol. 46, no. 1, February 2004, pp 71-86

[27] F. Macchi, D. Borio, M. Petovello, G. Lachapelle, “New Galileo L1 Acquisition Algorithms: Real Data Analysis and Statistical Characterization”, European Navigation Conference, Toulouse France, April 2008

[28] C. O’Driscol, M. G. Petovello, G. Lachapelle, “Software Receiver Strategies for the Acquisition and R-Acquisition of Weak GPS Signals”, ION NTM 2008, Session B4, San Diego, CA

[29] Si-Ming Pan, H. A. Grant, D. E. Dodds, S. Kumar, “An Offset-Z Search Strategy for Spread Spectrum Systems”, IEEE Transactions on Communications, vol. 43, No. 12, December 1995

[30] Chan-Woo Park, Sun Choi, J. Yoon, “FFT Based High Sensitivity Indoor GPS Receiver Technologies Using CDMA Cellular Network”, ION GNSS 17th International Technical Meeting of the Satellite Division, Sept. 2004, Long Beach, CA

[31] B. Persson, D.E. Dodds and R.J. Bolton, “A Segmented Matched Filter for CDMA Code Synchronization in Systems with Doppler Frequency Offset”, Telecommunications Conference, 2001. Proceedings of IEEE Globecom 01, vol 1, pp. 648-653.

[32] J. T. Rowley, and R. B. Waterhouse, “Performance of Shorted Microstrip Patch Antenna for Mobile Communications Handsets at 1800 MHz”, IEEE Transactions on Antennas and Propagation, vol.47, no. 5, May 1999, pp 815-822

[33] S.K. Shanmugam, R. Watson, J. Nielson, G. Lachapelle, “Differential Signal Processing Schemes for Enhanced GPS Acquisition”, GNSS05 Conference, Institute of Navigation, Long Beach, CA, 13-16 sep 2005

[34] P. K. Sagiraju, D. Akopian, G. V. S. Raju, “Performance Study of Reduced Search FFT-Based Acquisition”, ION NTM January 2007, San Diego, CA

[35] A. Soloviev, D. Bruckner, F. van Grass, “Assesment of GPS Signal Quality in Urban Environments Using Deeply Integrated GPS/IMU”, ION NTM 2007, San Diego CA

[36] M. Srinivassan, D. V. Sarwate, “Simple Schemes for Parallel Acquisition of Spreading Sequences in DS/SS Systems”, IEEE Transactions on Vehicular Technology, vol. 45, No. 3, August 1996

[37] C. Strassle, D. Megnet, H. Mathis, “The Squaring-Loss Paradox”, ION GNSS 20th International Technical Meeting of the Satellite Division, Sept. 2007, Fort Worth, TX

[38] T. Taga and K. Tsunekawa, “ Performance analysis of a built-in planar inverted F antenna for 800 MHz band portable radio units”, IEEE Journal on Selected Areas of Communications, vol SAC-5, pp 921-929, June 1987

[39] Y. Zhao, “Standardization of Mobile Phone Positioning for 3G Systems”, IEEE Communications Magazine, July 2002

[40] Y. Zhao, “Mobile Phone Location and Its Impact on Intelligent Transportation Systems”, IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 1, March 2000

[41] B. Zheng, G. Lachapelle, “Acquisition Schemes for a GPS L5 Software Receiver”, Proceedings of ION GNSS 2004 (Session A3), Long Beach, CA, Sept. 2004

[42] B. Zheng, G. Lachapelle, “GPS Software Receiver Enhancements for Indoor Use”, Proceedings of ION GNSS 2005 (Session C3), Long Beach, CA, September 13-16, 2005

Web Documents

[43] C. Abraham, F. van Diggelen, “Indoor GPS: The No-Chip Chalenge”, www.gpsworld.com/gpsworld/article/articleDetail.jsp?id=3053

[44] “Atmel And u-Blox Provide New Indoor-Capable GPS Signal Tracking Technology”, SPACE DAILY, www.spacedaily.com, 11/06/09

[45] N. Hayes, “Locating Your Location Based Service Provider”, www.wireless.com/channels/lbs/features/newsbite11.html

[46] NAVSYNC GPS Technologies, www.navsync.com/notes1.html, 11/06/2009

[47] SARANTEL, “GeoHelix-H™ GPS Antenna, www.sarantel.com, version 1 /May 03

[48]TOKO, “Dielectric Patch Antenna”, www.tokoam.com, 1996 Toko,inc.

A. APPENDICES

A1: CHI-SQUARED DISTRIBUTION AND MARCUM Q_FUNCTION

It is clear that the signal and noise distributions encountered in GNSS signal processing are chi-squared or non-central chi-squared. The simplest is the distribution of the total noise power combining the in-phase and quadrature components. After power averaging for *M* correlations, the signal plus noise distribution is non-central chi-squared with 2*M* degrees of freedom.

The chi-squared distribution results from the sum of squares of two or more normally distributed samples. The distribution is given for a normalized distribution with zero mean and variance equal to 1. The distribution is characterized by the number of degrees of freedom, *r* equal to the number of normal distributions that are summed.

The probability distribution is:

$$f(x) = \frac{x^{\frac{r}{2}-1}\exp(-\frac{x}{2})}{\Gamma\left(\frac{r}{2}\right)2^{\frac{r}{2}}}$$

The distributions with different values of *r* are given by Figure A1/1.
Of particular interest is the distribution with 2 degrees of freedom. With *r* = 2 the distribution reduces to;

$$p(x) = \frac{1}{2}\exp(-x/2)$$

This is the distribution of the noise power when summing the in-phase and quadrature components.
The mean of the distribution is *r* and the second central moment (variance) is 2*r*. Of interest in signal acquisition is the cumulative sum of the tail of the distribution:

$$P(Th) = \left(1 - F(Th)\right) = \int_{Th}^{\infty} f(x)dx$$

In this thesis the cumulative chi-distribution in EXCEL was used. This function is limited to values of r less than 200. For values of *r* greater than 200 a normal approximation was used.

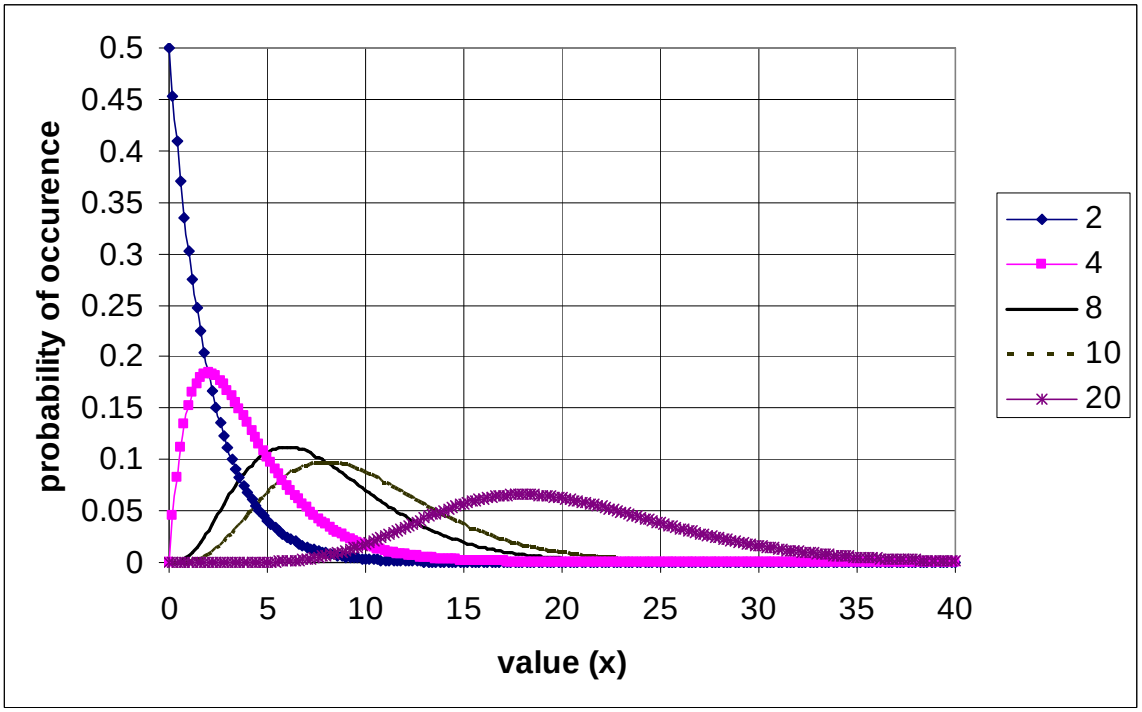

Figure A1/1: Chi-squared Distribution

The non-central chi-squared distribution results from summing two or more random variables (X_i) that are normally distributed with a non-zero mean. This is the case when summing signals plus Gaussian noise. The distribution is characterized by the variance (σ^2), the non-centrality factor (λ) and the number of random variables (k).

$$\lambda = \sum_{1}^{k} \left(\frac{\mu_i}{\sigma_i} \right)^2$$

The variable μ is the mean of X_i.

The probability density function is:

$$f(x) = \frac{1}{2} \exp\left(-\frac{x+\lambda}{2} \right) \left(\frac{x}{\lambda} \right)^{\left(\frac{k}{4} - \frac{1}{2} \right)} I_{\left(\frac{k}{2} - 1 \right)} \left(\sqrt{\lambda x} \right)$$

The function $I_k(z)$ is the modified Bessel function of the first kind.

The mean of the distribution is $k+\lambda$ and the variance is $2(k+2\lambda)$.

It is interesting to note that the difference in the mean of the chi-squared distribution and the non central chi-squared distribution is λ.

Generally we are interested in the probability of exceeding a power value known as the threshold (*Th*).

$$P(x) = \int_{Th}^{\infty} f(x)dx$$

We could compute the integral of f(*x*) but it is more common to use the Marcum Q function.

$$Q_M(\alpha, \beta) = \frac{1}{\alpha^{M-1}} \int_{\beta}^{\infty} x^M \exp(-\frac{x^2 + \alpha^2}{2}) I_{M-1}(\alpha x)dx$$

At first this may seem to be a new function but it is simply the cumulative non-central chi-squared function with a change of variables.

$$\alpha = \sqrt{\lambda}$$

$$\beta = \sqrt{Th}$$

$$M = \frac{k}{2}$$

$$x = \sqrt{x_{ch}}$$

The independent variable in the chi-squared distribution is the power of the ith sample. In the Marcum Q function the independent variable is the voltage. When computing the sum of the squares of *M* complex results, there are $2M = k$ values to sum. Thus the *k* in the chi-squared distribution is twice the number of samples that are summed.

A2 REVISED GALILEO L1 SIGNAL

The body of this thesis analyzes the Galileo L1 signal as described in the Signal-In-Space ICD dated 24/10/2006. Recently, ESA released a change notice modifying the L1 signal. This DCN is dated 11/09/08.

The changes include modification of the L1b and c BOC and the sharing of power between the three signal components.

The BOC(1) signal is replaced by a BOCC signal where

$$BOCC = \sqrt{\frac{10}{11}}BOC(1) + \sqrt{\frac{1}{11}}BOC(6) \text{ for L1b and}$$

$$BOCC = \sqrt{\frac{10}{11}}BOC(1) - \sqrt{\frac{1}{11}}BOC(6) \text{ for L1c.}$$

The result of this change is to produce signal sidebands at ±6 x 1.023 MHz as shown by Figure A2/1.

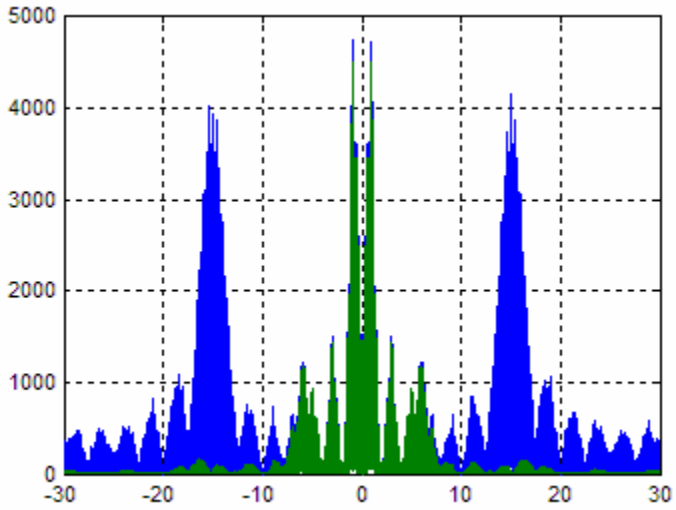

Figure A2/1: New Galileo L1 Spectrum

The blue shows the full 60 MHz spectrum without filtering. The strong 15 MHz lobes from the L1a BOC(15) signal are evident. The green shows the spectrum filtered to ±8 MHz. Figure A2/2 is the filtered spectrum after down sampling to 15 x 1.023 Msps. This sample rate is a 3 fold increase from that required for the previous version of L1b and c. The number of samples for a complete code is now 61440. The increased sample rate will require additional resources for acquisition and tracking. It is possible to filter

and down sample to 5 samples per chip but this narrow band results in a loss of sensitivity.

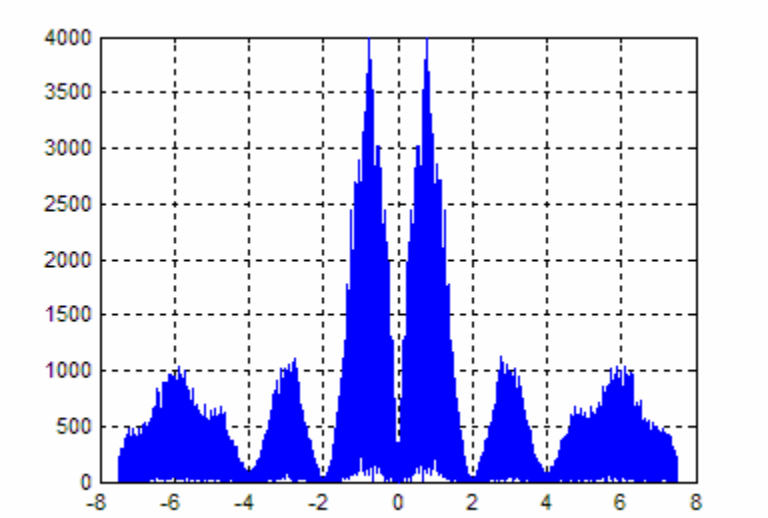

Figure A2/2: New Galileo L1 Spectrum, Filtered and Down Sampled

The new signal results in a lower percentage of the total power in L1b and c than with the original. The signal is filtered to 45 MHz before transmission from the satellite. After filtering 18.4 % of the power is in each of L1b and L1c. Considering the total signal strength of -152 dBW, the signal strength for each of L1b and c is -159.2 dBW. This signal strength is 1 dB lower than for the previous version. The lower signal strength coupled with the increased sampling rate results in the new signal being less attractive for location of cellular phones.

A3 MATLAB™ SIMULATIONS

A3.1 General

In this thesis, verification of the theoretical results was performed using MATLAB™ simulations and analysis.
The following is a list of the main types of programs:

- Acquisition of code timing
- Tracking
- Acquisition time
- Various distributions of noise and or signal products

A3.2 Acquisition Programs General

The acquisition routines included the following sections.

1. Define constants and initialize variables
2. Create the spreading code
3. Create samples of the received signal plus noise
4. Filter the received signal and down sample as necessary
5. Correlate with the spreading code
6. Average the result
7. Determine detection
8. Plot the results

In order to determine the probability of detection, sections 3 to 7 were repeated a number of times.

A3.2.1 GPS L1 C/A code Acquisition

The following is an example of a GPS L1 code acquisition program. This program determines the probability of detection for coherent averaging. Detection is determined in two ways. The parameter Pd is the probability that the correlation output for the correct delay exceeds the threshold. The parameter Pdet is the probability that the maximum value corresponds to the correct delay.

```
%gold code generator
%offset frequency in kHz
%Coherent averaging M
close all;clear all;
%Define constants and initialize variables
%b=[.00025,.06366,-.00025,-.1061,.00025,.31831,.49975,...
   %.31831,.00025,-.1061,-.00025,.06366,.00025];
   % bw=4, cutoff= pi/2
%b=[.00006,-.05516,-.06888,.00006,.13786,.27563,.333207,...
    %.27563,.13786,.00006,-.06888,-.05516,-.00006];
```

```
    %BW = 3 cutoff = pi/3
b=[-.05305,-.044926,.000126,.07511,.159155,.224989,.249873,...
     .224989,.159155,.07511,.000126,-.044926,-.05305];
 % BW=2MHz
 %b=[.00306,.014857,-.01091,-.066219,.02087,.30285,.47464,...
    % .30285,.02087,-.066219,-.01091,.014857,.00306];
df=0;%kHz
M=10;
NT=20;
Th=3533;
dth=df*2*pi/8184;
dthm=dth*8184;
nd=0:1:8183;
ndeto=0;
SNd=input('signal to noise ratio (dB) ');
SN=10^(SNd/10);
CNo=SNd+69;
A=sqrt(SN);
%A=0;
NP=1;
% Create Code
vo=[1 1 1 1 1 1 1 1 1 1];
v8x=[1 1 1 1 1 1 1 1];
pno = 0;
v1=vo;
v2=vo;
cgv=[];
cgmv=[];
for k=0:1:1022;
    ps=mod(k,10);
    if 2-ps > 0
        p2=2-ps;
    else
        p2=2-ps+10;
    end
    if 3-ps > 0
        p3=3-ps;
    else
        p3=3-ps+10;
    end
    if 6-ps > 0
        p6=6-ps;
    else
        p6=6-ps+10;
    end
    if 8-ps > 0
        p8=8-ps;
    else
        p8=8-ps+10;
    end
    if 9-ps > 0
        p9=9-ps;
    else
        p9=9-ps+10;
    end
    if 10-ps > 0
        p10=10-ps;
```

```
    else
        p10=10-ps+10;
    end
    v1out=v1(p10);
    v2out=xor(v2(p3),v2(p8));
    %create gold code, xor c1 and c2
    cgout=xor(v1out,v2out);
    cgv=[cgv cgout*v8x];
    cgmv=[cgmv (2*cgout-1)*v8x];
    %xor 3, 10 for code 1
    v6n=xor(v1(p3),v1(p10));
    %xor 2,3,6,8,9,10 for code 2
    v1n=xor(v2(p2),v2(p3));
    v2n=xor(v2(p6),v2(p8));
    v3n=xor(v2(p9),v2(p10));
    v4n=xor(v1n,v2n);
    v5n=xor(v4n,v3n);

    if pno-ps > 0
        pn=pno-ps;
    else
        pn=pno-ps+10;
    end
    v1(pn)=v6n;
    v2(pn)=v5n;

end

ndet=0;
for nt=1:NT;
for mt=1:M
% Create samples of the received signal
cxv=[];
cgmvn=A*cgmv.*(cos(dth.*nd+(mt-1)*dthm)+j*sin(dth.*nd+(mt-1)*dthm))+...
    NP/sqrt(2)*(randn(1,8184)+j*randn(1,8184));
%Filter
for m= 1:13;
    if m==1;
        cgmvs=circshift(cgmvn,[0,1]);
        cgmvlp=cgmvs*b(m);
    else
        cgmvs=circshift(cgmvs,[0,1]);
        cgmvlp=cgmvlp+cgmvs*b(m);
    end
end
% Determine filtered and unfiltered spectra for plotting
cgmfn=fft(cgmvn);
cgmfns=circshift(cgmfn,[0,4096]);
cgmf=fft(cgmvlp);
cgmfs=circshift(cgmf,[0,4096]);
% Delay the received signal
cgmvr=circshift(cgmvlp,[0,4097]);
% Downsample
cgmdsr=cgmvr(4:4:8184);
cgmds=cgmv(4:4:8184);
% Correlate using FFT
```

```
fcgmrds=fft(cgmdsr);
fcgmds=fft(cgmds);
fxcgm=fcgmrds.*conj(fcgmds);
cxv=ifft(fxcgm);
% Sum M correlation results
if mt==1
    cxvs=cxv;
else
    cxvs=cxvs+cxv;
end
end
cxvavg=cxvs/sqrt(M);
cxpwr=cxvavg.*conj(cxvavg);
[C,I]=max(cxpwr)
nt
% Determine detection
if abs(I-1027)<=1
    ndeto=ndeto+1;
end
if cxpwr(1027)>Th
    ndet=ndet+1;
end
end
Pd=ndet/NT
Pdet=ndeto/NT
Figure(1)
tsv=0:2045;

plot(tsv,cxpwr,[0,2045],[Th,Th],'--g')
grid on
title(['GPS Correlation, C/No = ',num2str(CNo),' dB Hz'])
xlabel('code offset')
ylabel('amplitude')
Af=max(600*A,250);
Figure(2)
tsf=1:8184;
plot(tsf,abs(cgmfns),tsf,abs(cgmfs),[3072,3072,5120,5120],...
    [0,Af,Af,0],'r')
grid on
```

The following lines of code replace the correlation and averaging sections of the previous program to perform power averaging.

```
fcgmrds=fft(cgmdsr);
fcgmds=fft(cgmds);
fxcgm=fcgmrds.*conj(fcgmds);
cxv=ifft(fxcgm);

if mt==1
    cxvs=cxv.*conj(cxv);
else
    cxvs=cxvs+cxv.*conj(cxv);
end
```

```
end
```

The following lines of code perform differential averaging.

```
fcgmrds=fft(cgmdsr);
fcgmds=fft(cgmds);
fxcgm=fcgmrds.*conj(fcgmds);
cxv=ifft(fxcgm);

if mt==1
    cxvp=conj(cxv);
elseif mt==2
    cxvs=cxv.*cxvp;
    cxvp=conj(cxv);
else
    cxvs=cxvs+cxv.*cxvp;
    cxvp=conj(cxv);
end
end
S2=cxvs(1027);
cxvavg=cxvs/sqrt(M);
cxpwr=abs(cxvavg);
```

The following lines of code save MF correlations then perform FFT on the columns. Power averaging for M follows. It is assumed that the received data is known.

```
for mt=1:M
   for mf=1:MF
    db=ceil((mf+(mt-1)*MF)/20);% data bit

cxv=[];
%create 1ms of data
cgmvn=A*dati(db)*cgmv.*(cos(dth.*nd+(mf-1)*dthm)+...
    j*sin(dth.*nd+(mf-1)*dthm))+...
NP/sqrt(2)*(randn(1,8184)+j*randn(1,8184));
%Filter
for m= 1:13;
    if m==1;
        cgmvs=circshift(cgmvn,[0,1]);
        cgmvlp=cgmvs*b(m);
    else
        cgmvs=circshift(cgmvs,[0,1]);
        cgmvlp=cgmvlp+cgmvs*b(m);
    end
end
cgmfn=fft(cgmvn);
cgmfns=circshift(cgmfn,[0,4096]);
cgmf=fft(cgmvlp);
cgmfs=circshift(cgmf,[0,4096]);
cgmvr=circshift(cgmvlp,[0,4097]);
cgmdsr=cgmvr(2:4:8184);
cgmds=cgmv(4:4:8184);

fcgmrds=fft(cgmdsr);
fcgmds=fft(cgmds*dati(db));% assumes data is known
```

```
fxcgm=fcgmrds.*conj(fcgmds);
cxv=ifft(fxcgm);
%Wn=0.5-0.5*cos(2*pi*mf/MF); %Hanning window coefficient
Wn=1;
    cxvA(mf,:)=cxv*Wn;
end
cxvF=fft(cxvA); %computes the FFT of the columns of cxvA

if mt==1
    cxvFM=cxvF.*conj(cxvF);
else
    cxvFM=cxvFM+cxvF.*conj(cxvF);
end
    end
    maxv=max(cxvFM);
    [maxd,I]=max(maxv);
if (abs(I-1027)<=1)
    ndeto=ndeto+1;
end
```

A3.2.2 GPS L5 Acquisition Programs

For the L5 program the signal samples were created at a rate of twice the code rate. No filtering or down sampling was needed.

The following program performs coherent averaging with a long code created from the spreading code plus NH code. Further coherent averaging for M long codes is performed.

```
%GPS L5
%Correlation for long code spread code + NH code,
%coherent averaging for M
%Noise power  = 1, signal amp=sqrt(C/N), half I half Q
close all; clear all
M=5;
Ntry=20;
SNd=input('signal to noise ratio (dB)' )
st=fix(clock)
SN=10^(SNd/10);
CNo=SNd+73;
A=sqrt(SN);
%Df=.015625;
Df=0.0005;
%Df=.0125;
dth=Df*2*pi/10230/2;
s2=[1,1];
%strt=clock
%GPS L5 signal generator
NHQ=[0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 1 1 1 0];
Nqm=NHQ.*2-1;
NHI=[0 0 0 0 1 1 0 1 0 1];
Nim=NHI.*2-1;
XAo=[1 1 1 1 1 1 1 1 1 1 1 1 1];
```

```
XAr=[1 1 1 1 1 1 1 1 1 1 1 0 1];
%start codes for signal no. 2
XBIo=[1 1 0 0 0 0 0 1 1 0 1 0 1];
XBQo=[0 1 0 0 0 1 1 1 1 0 1 1 0];
XA=XAo;
XI=XBIo;
XQ=XBQo;
C5I=[];
C5Q=[];
nhb=1
for k=1:10230;
    cxa=XA(13);
    cxI=XI(13);
    cxQ=XQ(13);
    cIout=2*xor(cxa,cxI)-1;
    cQout=2*xor(cxa,cxQ)-1;
    C5I=[C5I cIout*s2];
    C5Q=[C5Q cQout*s2];
    xo1=xor(XA(12),XA(13));
    xo2=xor(XA(9),XA(10));
    xan=xor(xo1,xo2);
    xi1=xor(XI(12),XI(13));
    xi2=xor(XI(7),XI(8));
    xi3=xor(XI(4),XI(6));
    xi4=xor(XI(1),XI(3));
    xi12=xor(xi1,xi2);
    xi34=xor(xi3,xi4);
    xin=xor(xi12,xi34);
    xq1=xor(XQ(12),XQ(13));
    xq2=xor(XQ(7),XQ(8));
    xq3=xor(XQ(4),XQ(6));
    xq4=xor(XQ(1),XQ(3));
    xq12=xor(xq1,xq2);
    xq34=xor(xq3,xq4);
    xqn=xor(xq12,xq34);
    if XA==XAr;
        rst=1;
    else
        rst=0;
    end
    XA=circshift(XA,[0,1]);
    XA(1)=xan;
    if rst==1;
        XA=XAo;
    end
    XI=circshift(XI,[0,1]);
    XI(1)=xin;
    XQ=circshift(XQ,[0,1]);
    XQ(1)=xqn;
    if mod(k+1,10230)==0;
        XA=XAo;
        XI=XBIo;
        XQ=XBQo;

    end
end
%End of code generator
```

```
%Creat transmitted waveform
ndet=0;
for ntry=1:Ntry

%Correlation

stcorr=fix(clock)
for in=1:M
    %Wn=.5-.5*cos(2*pi*in/M);
    Wn=1;
 for is=1:20;
    ii=mod(is,10);
    if ii==0
        ii=10;
    end
    %generate received code
 dthn=(is-1)*dth*20460+dth*(in-1)*20460*20;
id=0:20459;
cgrq=Nqm(is)*A/sqrt(2)*C5Q.*(-sin(dth.*id+dthn)+j*cos(dth.*id+dthn))...
    +Nim(ii)*A/sqrt(2)*C5I.*(cos(dth.*id+dthn)+j*sin(dth.*id+dthn))...
   +1/sqrt(2)*(randn(1,20460)+j*randn(1,20460));
wxq=C5Q*Nqm(is);

%FFTPq=FFTq.*FFTcq';
%cxvq=ifft(FFTPq);
    if is==1
        cgqm=cgrq;
        wxqr=wxq;
    else
        cgqm=[cgqm cgrq];
        wxqr=[wxqr wxq];
    end
 end
 cgqms=circshift(cgqm,[0,10230]);
 FFTq=fft(cgqms);
 FFTCQ=fft(wxqr);
 cgqx=FFTq.*conj(FFTCQ);
 cgqCs=ifft(cgqx);

if in==1
    cgqMs=cgqCs;
else
    cgqMs=cgqMs+cgqCs;
end
end

cxvMqp=cgqMs.*conj(cgqMs);
[Pmax Imax]=max(cxvMqp)
if abs(Imax-10231)<2
    ndet=ndet+1;
end
endcorr=fix(clock)
ntry
end; %end of ntry loop
pdet=ndet/Ntry
tsv=1:409200;
```

```
ends=fix(clock)


Figure(1);
plot(tsv,cxvMqp)
grid on
xlabel('Code Offset (chips)')
ylabel('amplitude')
title(['GPS L5Q; C/No = ',num2str(CNo),',coh avg M = ',num2str(M)]);
%Figure(2);
```

The following lines of code correlate with the spreading code 20 times and save the results in a matrix. The timing of the NH code is determined. Further differential averaging for M is performed.

```
wxq=[];
wxi=[];

%Create Padded codes
C5Qp=[C5Q zeros(1,20460)];
C5Ip=[C5I zeros(1,20460)];
wxqr=C5Q;
wxir=C5I;
%Create transmitted waveform
FFTcq=fft(C5Qp);
FFTci=fft(C5Ip);
%stcorr=fix(clock)
Ndet=0;
Bdet=0;
for ntry=1:Ntry
for in=1:M;
    cgrqN=[];
 for is=1:21;
    ii=mod(is,10);
    iq=mod(is,20);
    if ii==0
        ii=10;
    end
    if iq==0
       iq=20;
    end
 dthn=(is-1)*dth*20460+dth*(in-1)*20460*20;
id=0:20459;
cgrq=Nqm(iq)*A/sqrt(2)*wxqr.*(-
sin(dth.*id+dthn)+j*cos(dth.*id+dthn))...
    +Nim(ii)*A/sqrt(2)*wxir.*(cos(dth.*id+dthn)+j*sin(dth.*id+dthn))...
   +1/sqrt(2)*(randn(1,20460)+j*randn(1,20460));
cgrqN=[cgrqN cgrq];
 end; %end of is loop, created rx signal 21 codes long
 %delay received code by shift
 cgrqs=circshift(cgrqN,[1,shift]);
 %select 20 sets of samples 2 codes long
 for kr=1:20
```

```
     cgrq2=cgrqs((kr-1)*20460+1:kr*20460+20460);
FFTq=fft(cgrq2);
FFTPq=FFTq.*conj(FFTcq);
cxvq=ifft(FFTPq);
    cxvqk(kr,:)=cxvq;
 end; %end of kr loop
 %Test 20 NH code alignments
 for ka=1:20
     for kra=1:20
         knh=ka+kra-1;
         if knh>20
             knh=knh-20;
         end
         if kra==1
             cxvSq(ka,:)=cxvqk(kra,:)*Nqm(knh);
         else
             cxvSq(ka,:)=cxvSq(ka,:)+cxvqk(kra,:)*Nqm(knh);
         end
     end
 end
if in==1
    cxvSqp=cxvSq;% Note that cxvSq is a matrix 20 x 2046
elseif in==2
    cxvMq=cxvSq.*conj(cxvSqp);
    cxvSqp=cxvSq;
else
    cxvMq=cxvMq+cxvSq.*conj(cxvSqp);
    cxvSqp=cxvSq;
end
end; %end of in loop (average M)
%Pwr=cgrq*cgrq'
cxvMqa=abs(cxvMq);
%endcorr=fix(clock)
tsv=1:40920;
cxvMqm=max(cxvMqa);
[pmax Imax]=max(cxvMqm);
[ntry Imax]
[pmaxa Ial]=max(cxvMqa(:,Imax));
Ial
if abs(Imax-shift-1)<2
    Ndet=Ndet+1;
else
    cxvMqb=cxvMqm;
    Bdet=1;
end
end
Pdet=Ndet/Ntry
ends=fix(clock)
Figure(1);
plot(tsv,cxvMqm,'b')
grid on
xlabel('Code Offset (chips)')
ylabel('amplitude')
title(['GPS L5Q; S/No = ',num2str(CNo),', M = ',num2str(M)]);
Figure(2);
plot(0:20,cxvMqm(10220:10240))
grid on
```

```
if Bdet==1
Figure(3)
plot(tsv,cxvMqb)
grid on
end
```

A3.2.3 Galileo L1 Acquisition Programs

The following program performs coherent averaging for the length of the short code. Further power averaging for M is performed. The correct short code alignment is searched for.

```
%Revised for noise power =1, signal A=sqrt(S/N)
%correllate with spreading code then align short code
%Power averaging after correlation & SC
close all; clear all;
SNd=input('signal to Noise Ratio dB ')
SN=10^(SNd/10)
SNo=SNd+74.8-6; % 1/4 of the power in L1c;
A=sqrt(SN);
%shift=0;
shift=10230;
M=10;
Pfa=.00005;
sigLsq=5*4096/12;
Th=-2*log(Pfa)*sigLsq
%Df=.031738;
%Df=.03125;
Df=0.005;
dth=Df*2*pi/30690;
nd=0:122759;
dthn=Df*2*pi*4;
strt=fix(clock)
Cdb=floor(2*rand(1,4092));
Cmb=-2*Cdb+1;
Cdc=floor(2*rand(1,4092));
Cmc=-2*Cdc+1;
CmcF=fft(Cmc);
CmcxF=CmcF.*conj(CmcF);
ACmc=ifft(CmcxF);
%Filter Coefficients
b=[-.0161 .0168 .0593 .1048 .1453 .1731 .1830 .1731 .1453 ...
.1048 .0593 .0168 -.0161]; %cutoff = .575 radians/sample
%b=[.00019 .032 .069 .1061 .1377 .159 .1665 .159 .1377 .1061...
   %.069 .032 .00019];% pi/6 rad/sample

boc15=[1 -1 1 -1 1 -1 1 -1 1 -1 1 -1];
c25=[1 1 1 1 1 1 1 1 1 1 1 1];
boc1=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 ...
    -1 -1 -1 -1 -1 -1 -1];
c1=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
scc=[0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 0 1 0];
scm=-2*scc+1;
scmF=fft(scm);
scmx=scmF.*conj(scmF);
```

```
scmcorr=ifft(scmx);
ecv=[];
ebv=[];
eav=[];
sq2=sqrt(2);
stloop=fix(clock)
kb=1;
%build basic codes with BOC, 4 ms worth
for k=1:122760;
    if mod(k,12)==1
        bta=-2*floor(2*rand)+1;
        eav=[eav boc15*bta];
    end
    if mod(k,30)==1
        ebv=[ebv Cmb(kb)*boc1];
        ecv=[ecv Cmc(kb)*boc1];
        kb=kb+1;
    end
end
%compute d/s C code
ecvd=ecv(1:6:122760);
%add SC to C

strtRxcd=fix(clock)
for im=1:M
ecvscL=[];
SL1sc=[];
for n=1:26
    scb=mod(n,25);
    if scb==0
        scb=25;
    end
    ecvsc=ecv*scm(scb);
    ecvscd=ecvsc(1:6:122760);
    ecvscp=[ecvscd zeros(1,20460)];
    ecFsc=fft(ecvscp);
%Create composite modulation signal
SL1v=(sq2*(ebv-ecvsc)+j*(2*eav+eav.*ebv.*ecvsc))/3;
%modulate carrier and add noise
SL1r=A*SL1v.*(cos(dth.*nd+(n-1)*dthn+(im-1)*dthn*25)+...
            j*sin(dth.*nd+(n-1)*dthn+(im-1)*dthn*25))...
    +1/sq2*randn(1,122760)...
    +j*1/sq2*randn(1,122760);
%filter
for m=1:13
    if m==1
        SL1s=circshift(SL1r,[0,1]);
        SL1lp=SL1s*b(m);
    else

        SL1s=circshift(SL1r,[0,m]);
       SL1lp=SL1lp+SL1s*b(m);
    end
end
SL1ds=SL1lp(1:6:122760);
SL1sc=[SL1sc SL1ds];
```

```
end
%delay received code by shift
SL1sh=circshift(SL1sc,[0,-shift]);
%create matrix of 2 correllations length by 25
strtcorr=fix(clock)
im
for n2=1:25
   SL1n=SL1sh(((n2-1)*20460+1):(n2*20460+20460));
   SL1F=fft(SL1n);
   SL1x=SL1F.*conj(ecFsc);
   if n2==1
       SL1m=ifft(SL1x);
   else
       SL1m=[SL1m;ifft(SL1x)];
   end
end
%align short code
strtAlign=fix(clock)
Cn3=0;
for n3=1:25
    for k=1:25
        nk=k+n3-1;
        if nk > 25
            nk=nk-25;
        end
        SL1mx(k,:)=SL1m(k,:)*scm(nk);
    end
  Sn3=sum(SL1mx);
  Sn3p=Sn3.*conj(Sn3);
  if im==1
      Sn3pm(n3,:)=Sn3p;
  else
      Sn3pm(n3,:)=Sn3pm(n3,:)+Sn3p;
  end
end
end; %end of M loop
 Pmax=max(Sn3pm); %vector with 40920 offsets
 [Dmax Imax]=max(Pmax)
 [Rmax n3max]=max(Sn3pm(:,Imax))
plot(1:40920,Pmax)
grid on
title(['Galileo L1c, Align sc, avP, C/No = ',num2str(SNo)])
```

A3.3 Tracking Programs

A3.3.1 GPS L1 C/A Code

```
%gold code generator
%offset frequency in kHz
%reduce SNd by dSNd each step from nt=50 to 150
close all;clear all;
%filter coefficients
b=[.0249,-.0001,-.0322,-.0531,-.0449,.0001,.0751,.1592,...
    .2250,.2499,.2250,.1592,.0751,.0001,-.0449,-.0531,...
    -.0322,-.0001,.0249];
 % BW=2.046MHz
```

```
shift=1023;%delay in rx signal
dT=4;%offset for early late
dfo=0.010;%kHz
tho=pi*1/4;
dero=4;
dSNd=.1;
dps=0;
NT=200;
Idat=2*floor(2*rand(1,NT))-1;
%Idat=zeros(1,NT)+1;
nd=0:1:8183;
SNdo=input('signal to noise ratio (dB) ');
SN=10^(SNdo/10);
CNo=SNdo+69;
ddev=0;
if ddev==0
    fsh=0;
else
fsh=input('sh freq ;')
end
b1=.2;
b2=0;
b3=0;
bo=1;
bf=.0015;
bf1=.0007;
bph=6;
bsph=3;
bph1=.2;
thph=pi*1/12;
th2=pi/12;
A=sqrt(SN);
emax=(8184*A*20)^2;
%A=0;
NP=1;
%Code Generator
vo=int8([1 1 1 1 1 1 1 1 1 1]);
%8 samples per chip
v8x=int8([1 1 1 1 1 1 1 1]);
pno = 0;
v1=vo;
v2=vo;
cgv=int8([]);
cgmv=single([]);
%create code
for k=0:1:1022;
    ps=int8(mod(k,10));
    if 2-ps > 0
        p2=2-ps;
    else
        p2=2-ps+10;
    end
    if 3-ps > 0
        p3=3-ps;
    else
        p3=3-ps+10;
    end
```

```
    if 6-ps > 0
        p6=6-ps;
    else
        p6=6-ps+10;
    end
    if 8-ps > 0
        p8=8-ps;
    else
        p8=8-ps+10;
    end
    if 9-ps > 0
        p9=9-ps;
    else
        p9=9-ps+10;
    end
    if 10-ps > 0
        p10=10-ps;
    else
        p10=10-ps+10;
    end
    v1out=int8(v1(p10));
    v2out=int8(xor(v2(p3),v2(p8)));
    %create gold code, xor c1 and c2
    cgout=int8(xor(v1out,v2out));
    cgv=[cgv cgout*v8x];
    cgm1=single((2*cgout-1)*v8x);
    cgmv=single([cgmv cgm1]);
    %xor 3, 10 for code 1
    v6n=xor(v1(p3),v1(p10));
    %xor 2,3,6,8,9,10 for code 2
    v1n=xor(v2(p2),v2(p3));
    v2n=xor(v2(p6),v2(p8));
    v3n=xor(v2(p9),v2(p10));
    v4n=xor(v1n,v2n);
    v5n=xor(v4n,v3n);

    if pno-ps > 0
        pn=pno-ps;
    else
        pn=pno-ps+10;
    end
    v1(pn)=v6n;
    v2(pn)=v5n;

end; %end code generator
stloop=fix(clock)
    cgmp=circshift(cgmv,[0,shift+10-dero]);
    cgme=circshift(cgmp,[0,-dT]);
    cgml=circshift(cgmp,[0,dT]);

%cxv=[];%error vector
cxv=zeros(1,NT);
tso=0;
%tsov=[];
tsov=zeros(1,NT);
cxpa=0;
```

```
cxera=0;
cxpv=zeros(1,NT);
dero=4;
dpho=8184;
dph=dpho;
sdph=0;
vrp=0;
sumerr=0;
cxer1p=0;
cxer1p2=0;
%rsigv=[];
%isigv=[];
rsigp=0;
isigp=0;
fsigv=[];
phsv=[];
sumfsig=0;
sumfphs=0;
df=dfo;
sphso=tho;
fph1p=0;
fph2p=0;
SNd=SNdo;
for nt=1:NT; %repeat for NT 20 ms samples
    if (nt>50)&&(nt<200)
        SNd=SNdo-(nt-50)*dSNd;
        SN=10^(SNd/10);
        A=sqrt(SN);
        emax=(8184*A*20)^2;
    end

    %df=dfo-dfo*nt/100;
    dth=df*2*pi/8184;
   dthn=dth*8184;
    for kc=1:20
    sphase=dth.*nd+(kc-1)*dthn+sphso;
    if kc==20
        sphso=sphase(8184)+dth;
    end
cgmvn=A*cgmv.*((cos(sphase)...
    +j*sin(sphase))*j*Idat(nt))+...
    NP/sqrt(2)*(randn(1,8184)+j*randn(1,8184));

%Filter
for m= 1:15;
    if m==1;
        cgmvs=circshift(cgmvn,[0,1]);
        cgmvlp=cgmvs*b(m);
    else
        cgmvs=circshift(cgmvs,[0,1]);
        cgmvlp=cgmvlp+cgmvs*b(m);
    end
end
%sum 20 samples
if kc==1
    cgmvk=cgmvlp;
```

```
else
    cgmvk=cgmvk+cgmvlp;
end
    end;%end of kc loop
    cgmF=fft(cgmvk);
    dsh=fix(ddev*sin(2*pi*fsh*nt/50));
    %received signal
rxvp=circshift(cgmvk,[0,shift+dsh]);
rxF=fft(rxvp);

%start correlation
vr=0;
np=0;
cxe=0;
cxl=0;
cxp=0;
%NCO
  for nc=1:32736
      ncc=mod(nc,4);
          vr=vrp+dph;
    if vr > 32735
        vr=vr-32736;
        vo=1;
    else
        vo=0;
    end
    vrp=vr;
    if vo==1
        np=np+1;
         ts=np+tso;
      if ts>8184
        ts=ts-8184;
      elseif ts<1
          ts=ts+8184;
      end
    end
    if ncc==0
       ns=nc/4;
        cxe=cxe+rxvp(ns)*cgme(ts);
        cxl=cxl+rxvp(ns)*cgml(ts);
        cxp=cxp+rxvp(ns)*cgmp(ts);
    end
  end; %end of nc loop

  tso=tso+(np-8184);
  %Compute delay error
  cxer=cxl*conj(cxl)-cxe*conj(cxe);
  cxer1=cxer/emax;
  sumerr=sumerr+cxer1;
  cxpp=cxp*conj(cxp);
  cxpa=.95*cxpa+.05*cxp;
  cxpv(nt)=cxp*conj(cxp)/emax;
  lpxer=bo*(cxer1-2*b2*cos(th2)*cxer1p+b2^2*cxer1p2)/cxer1;
    cxer1p2=cxer1p;
    cxer1p=cxer1;
    cxer2=lpxer*(b1*cxer1+b3*sumerr);
```

```
if cxer2>2
    cxer2=2;
elseif cxer2<-2
    cxer2=-2;
end
cxera=.95*cxera+.05*cxer2;
cxv(nt)=cxera;
tsov(nt)=tso;
%Compute new phase rate for NCO
dph=dpho-cxer2;
dphv(nt)=dph;
sdph=sdph+dph;
%frequency and phase detection
rsig=real(cxp);
isig=imag(cxp);
%Frequency error
fsig=(rsig*isigp-rsigp*isig)/emax*...
    sign(isig*isigp+rsig*rsigp);

rsigv(nt)=rsig;
isigv(nt)=isig;
 fsigv=[fsigv fsig];

 rsigp=rsig;
 isigp=isig;
 %Compute phase error
  fphs=(rsig*isig)/emax;

  if fphs > 1
     fphs=1;
  elseif fphs <-1
         fphs=-1;
  end
  fphs1=fphs/1000;
 if nt<25;%frequency correction only
    sumfsig=sumfsig+fsig;
  sumfsigv(nt)=sumfsig;
  df=dfo+(sumfsig*bf+fsig*bf1);
  %df=dfo;
 %df1=df;
 elseif nt<40;% Frequency and phase correction
      sumfsig=sumfsig+fsig;
  sumfsigv(nt)=sumfsig;
  lpgn=(fphs1-2*bph1*cos(thph)*fph1p+bph1^2*fph2p)/fphs1;
   %lpgn=1;

     lpgnv(nt)=lpgn;
  sumfphs=sumfphs+fphs1*lpgn;
  sumphv(nt)=sumfphs;
  df=dfo+fphs1*bph*lpgn+bsph*sumfphs+(sumfsig*bf+fsig*bf1);
 else;% Reduce frequency and phase correction gains
     sumfsig=sumfsig+fsig/4;
  sumfsigv(nt)=sumfsig;
  lpgn=(fphs1-2*bph1*cos(thph)*fph1p+bph1^2*fph2p)/fphs1;
   %lpgn=1;
```

```
       lpgnv(nt)=lpgn;
    sumfphs=sumfphs+fphs1*lpgn/4;
    sumphv(nt)=sumfphs;
    df=dfo+fphs1*bph*lpgn/2+bsph*sumfphs+(sumfsig*bf+fsig*bf1/4);
   end
   fph1p=fphs1;
   fph2p=fph1p;
   dfv(nt)=df;
   sumfsigv(nt)=sumfsig/100;

   fphsv(nt)=fphs;
end; %end of NT loop
tsomn=sum(tsov(40:NT))/(NT-39)
vartso=sum(tsov(40:NT).^2)/(NT-39)-tsomn^2
vpp=sqrt(2*vartso)*2
RMSerr=sqrt(cxv(21:NT)*cxv(21:NT)'/(NT-20))
cxF=cgmF.*conj(cgmF);
Rcgr=ifft(cxF);
Rcgrs=circshift(Rcgr,[0,4092]);
sdph/NT
endloop=fix(clock)
drmp=-dps/50*(1:NT)-dero;
datev=xor(Idat(36:NT)+1,sign(-isigv(36:NT))+1);
dater=sum(datev)
BER=dater/(NT-35);
if BER>0.5
    BER=1-BER;
end
BER
RMSfreq=sqrt(sum(dfv(31:NT).^2)/(NT-30))*1000
Figure(1)
%plot(1:NT,abs(cxF),[0,NT/25,NT/25],[100,100,0],'r')
plot(1:NT,cxv,1:NT,cxpv)
grid on

%legend('real','imag')
Figure(2)
plot(1:NT,tsov,1:NT,drmp)
grid on
%axis([1,NT,-10,0])
Figure(3)
%plot(1:NT,cxpv,[1,NT],[emax,emax])
%plot(1:NT,rsigv,1:NT,isigv)
plot(1:NT,fsigv,':',1:NT,dfv*1000,1:NT,sumphv*1000,...
    1:NT,fphsv,'k',36:NT,datev-5,'.')
 %plot(1:NT,fsigv,1:NT,dfv*1000);
 %legend('freq error','freq offset (Hz)')
 xlabel('Time x 20ms')
legend('errorf','df*1000','sumpherr','pher','datev')
%plot(1:NT,rsigv.*isigv)
grid on
Figure(4)
NC=NT-20;
nc=0:NC-1;
xc=cos(2*pi*nc/NC);
yc=sin(2*pi*nc/NC);
```

```
%plot(1:NT,-isigv/sqrt(emax),1:NT,Idat/2,'r')
plot(rsigv(21:NT)/sqrt(emax),isigv(21:NT)/sqrt(emax),'.',...
    [-1,1],[1,-1],[-1,1],[-1,1],xc,yc)
grid on
```

A3.3.2 GPS L5 Tracking

The following program performs frequency and phase tracking for GPSL5. The phase of the quadrature channel is tracked and the data on the in-phase channel is demodulated.

```
%GPS L5
%Frequency and Phase Tracking
%Phase Lock to L5Q and demodulate L5I data

%coherent averaging for M=20 (Length of NHQ code

%Noise power  = 1, signal amp=sqrt(C/N), half I half Q
close all; clear all
M=5;
Ntry=200;
Idat=2*floor(2*rand(1,Ntry*2))-1;
%Idat=zeros(1,Ntry*2)-1;
SNd=input('signal to noise ratio (dB)' )
st=fix(clock)
SN=10^(SNd/10);
CNo=SNd+73;
SNo=CNo-3
A=sqrt(SN);
emax=(20460*20*A)^2/2;
emaxi=(20460*10*A)^2/2;
%Df=.015625;
Dfo=0.015;
%Df=.0125;
bf=.0015;
bf1=.0007;
bph=3;
bsph=1.5;
bph1=.2;
thph=pi*1/12;

s2=[1,1];
%strt=clock
%GPS L5 signal generator
NHQ=[0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 1 1 1 0];
Nqm=NHQ.*2-1;
NHI=[0 0 0 0 1 1 0 1 0 1];
Nim=NHI.*2-1;
XAo=[1 1 1 1 1 1 1 1 1 1 1 1 1];
XAr=[1 1 1 1 1 1 1 1 1 1 1 0 1];
%start codes for signal no. 2
XBIo=[1 1 0 0 0 0 0 1 1 0 1 0 1];
XBQo=[0 1 0 0 0 1 1 1 1 0 1 1 0];
XA=XAo;
XI=XBIo;
XQ=XBQo;
```

```
C5I=[];
C5Q=[];
nhb=1
for k=1:10230;
    cxa=XA(13);
    cxI=XI(13);
    cxQ=XQ(13);
    cIout=2*xor(cxa,cxI)-1;
    cQout=2*xor(cxa,cxQ)-1;
    C5I=[C5I cIout*s2];
    C5Q=[C5Q cQout*s2];
    xo1=xor(XA(12),XA(13));
    xo2=xor(XA(9),XA(10));
    xan=xor(xo1,xo2);
    xi1=xor(XI(12),XI(13));
    xi2=xor(XI(7),XI(8));
    xi3=xor(XI(4),XI(6));
    xi4=xor(XI(1),XI(3));
    xi12=xor(xi1,xi2);
    xi34=xor(xi3,xi4);
    xin=xor(xi12,xi34);
    xq1=xor(XQ(12),XQ(13));
    xq2=xor(XQ(7),XQ(8));
    xq3=xor(XQ(4),XQ(6));
    xq4=xor(XQ(1),XQ(3));
    xq12=xor(xq1,xq2);
    xq34=xor(xq3,xq4);
    xqn=xor(xq12,xq34);
    if XA==XAr;
        rst=1;
    else
        rst=0;
    end
    XA=circshift(XA,[0,1]);
    XA(1)=xan;
    if rst==1;
        XA=XAo;
    end
    XI=circshift(XI,[0,1]);
    XI(1)=xin;
    XQ=circshift(XQ,[0,1]);
    XQ(1)=xqn;
    if mod(k+1,10230)==0;
        XA=XAo;
        XI=XBIo;
        XQ=XBQo;

    end
end
%End of code generator
%Creat transmitted waveform
ndet=0;
Df=Dfo;
sphso=0;
rsigp=1;
isigp=0;
fph1p=0;
```

```
fph2p=0;
sumfsig=0;
sumfphs=0;
datr=[];
riv=[];
iiv=[];
for ntry=1:Ntry
%Correlation

%stcorr=fix(clock)
dth=Df*2*pi/20460;
dthn=dth*20460;
nd=0:20459;
 for is=1:20;
    ii=mod(is,10);
    if ii==0
        ii=10;
    end
  sphase=dth.*nd+(is-1)*dthn+sphso;
  if is==20;
    sphso=sphase(20460)+dth;
  end
  %generate received signal (1 ms worth)
ib=2*ntry-1+floor((is-1)/10);

cgrq=Nqm(is)*A/sqrt(2)*C5Q.*(-sin(sphase)+j*cos(sphase))...
    +Nim(ii)*Idat(ib)*A/sqrt(2)*C5I.*(cos(sphase)+j*sin(sphase))...
   +1/sqrt(2)*(randn(1,20460)+j*randn(1,20460));
%wxq=C5Q*Nqm(is);
%wxi=C5I*Nim(ii);
%FFTPq=FFTq.*FFTcq';
%cxvq=ifft(FFTPq);
    if is==1
        cgqm=cgrq*Nqm(is);
        cgim1=cgrq*Nim(ii);
    elseif is<11
        cgqm=cgqm+cgrq*Nqm(is);
        cgim1=cgim1+cgrq*Nim(ii);
    elseif is==11
       cgqm=cgqm+cgrq*Nqm(is);
       cgim2=cgrq*Nim(ii);
    else
        cgqm=cgqm+cgrq*Nqm(is);
        cgim2=cgim2+cgrq*Nim(ii);
    end

 end; %end of is loop
 Rxq=C5Q*cgqm';
 Rxi1=C5I*cgim1';
 Rxi2=C5I*cgim2';

 datr=[datr sign(imag(Rxi1)) sign(imag(Rxi2))];
 riv=[riv real(Rxi1) real(Rxi2)];
 iiv=[iiv imag(Rxi1) imag(Rxi2)];
  rsig=real(Rxq);
  isig=imag(Rxq);
```

```
  rsigv(ntry)=rsig;
  isigv(ntry)=isig;
  fsig=(rsig*isigp-rsigp*isig)/emax;
  fsigv(ntry)=fsig;
  rsigp=rsig;
  isigp=isig;
  fphs=rsig*isig/emax;
  fphsv(ntry)=fphs;
  fphs1=fphs/1000;
  if ntry<25
    sumfsig=sumfsig+fsig;
    sumfsigv(ntry)=sumfsig;
    Df=Dfo-(sumfsig*bf+fsig*bf1);
   elseif ntry<40
     sumfsig=sumfsig+fsig;
     sumfsigv(ntry)=sumfsig;
     lpgn=(fphs1-2*bph1*cos(thph)*fph1p+bph1^2*fph2p)/fphs1;
     lpgnv(ntry)=lpgn;
    sumfphs=sumfphs+fphs1*lpgn;
    sumphv(ntry)=sumfphs;
    Df=Dfo+(fphs1*bph*lpgn+bsph*sumfphs)-(sumfsig*bf+fsig*bf1);
    else
       sumfsig=sumfsig+fsig/4;
    sumfsigv(ntry)=sumfsig;
    lpgn=(fphs1-2*bph1*cos(thph)*fph1p+bph1^2*fph2p)/fphs1;
     %lpgn=1;

       lpgnv(ntry)=lpgn;
    sumfphs=sumfphs+fphs1*lpgn/4;
    sumphv(ntry)=sumfphs;
    Df=Dfo+(fphs1*bph*lpgn/2+bsph*sumfphs)-(sumfsig*bf+fsig*bf1/4);
   end
   fph1p=fphs1;
   fph2p=fph1p;
   Dfv(ntry)=Df;
   sumfsigv(ntry)=sumfsig/100;
   fphsv(ntry)=fphs;
end
datev=xor(Idat(51:Ntry*2)+1,datr(51:Ntry*2)+1);
dater=sum(datev)
BER=dater/(2*Ntry-50);
if BER>0.5
    BER=1-BER;
end
BER
sm=sqrt(emax);
RMSfreq=sqrt(sum(Dfv(51:Ntry).^2)/(Ntry-50))*1000
Figure(1)
plot(rsigv(1:50)/sm,isigv(1:50)/sm,'.r',...
    rsigv(51:Ntry)/sm,isigv(51:Ntry)/sm,'.b',...
    riv(51:2*Ntry)/sm,iiv(51:2*Ntry)/sm, '.c')
grid on
Figure(2)
plot(2:2:2*Ntry,Dfv*1000,2:2:2*Ntry,sumphv*1000,'r',...
    2:2:2*Ntry,fphsv,'k',51:2*Ntry,datev+6,'.c')
```

```
 xlabel('Time x 10ms')
legend('df*1000','sumpherr','pher','xor(Idat,datr)')

grid on
%Figure(3)
%plot(36:2*Ntry,datev,'.')
%grid on
%axis([1,2*Ntry,-2,2])
```

A3.3.3 Galileo L1 Tracking

The following program tracks the code delay and frequency and phase for Galileo L1. The phase of L1c is tracked and the data on L1b is demodulated.

```
%Galileo L1 signal generator
%average power for iN codes
%Track after correlation with quad NH code
%delay and phase tracking
close all; clear all;
SNd=input('signal to Noise Ratio dB ')
SN=10^(SNd/10)
SNo=SNd+74.8-6.2; % 1/4 of the power in L1c;
A=sqrt(SN);
NP=1;
M=25;
NT=40;
dT=7;
datb=2*floor(2*rand(1,25*NT))-1;
dero=4;
shift=1023;
b1=2;
bf=.001;
bf1=.0005;
bph=1.2;
bsph=0.6;
bph1=.2;
thph=pi/12;
Dfo=0.003;
%Df=.03125;

nd=0:122759;

emax=(122760*A*M)^2*2/9;
strt=fix(clock)
Cdb=floor(2*rand(1,4092));
Cmb=-2*Cdb+1;
Cdc=floor(2*rand(1,4092));
Cmc=-2*Cdc+1;
b=[-.0395 -.0351 -.0161 .0168 .0593 .1048 .1453 .1731...
    .1830 .1731 .1453 .1048 .0593 .0168 -.0161 -.0351 -.0395];
%cutoff = .575 radians/sample
%b=[.00019 .032 .069 .1061 .1377 .159 .1665 .159 .1377 .1061...
   %.069 .032 .00019];% pi/6 rad/sample
```

```
boc15=[1 -1 1 -1 1 -1 1 -1 1 -1 1 -1];
c25=[1 1 1 1 1 1 1 1 1 1 1 1];
boc1=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 ...
    -1 -1 -1 -1 -1 -1 -1];
c1=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
scc=[0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0 1 1 0 0 1 0];
scm=-2*scc+1;
ecv=[];
ebv=[];
eav=[];
sq2=sqrt(2);
stloop=fix(clock)
kb=1;
for k=1:122760;
    if mod(k,12)==1
        bta=-2*floor(2*rand)+1;
        eav=[eav boc15*bta];
    end
    if mod(k,30)==1
        ebv=[ebv Cmb(kb)*boc1];
        ecv=[ecv Cmc(kb)*boc1];
        kb=kb+1;
    end
end

ecvd=ecv(3:6:122760);
ebvs=circshift(ebv,[0,8]);
ebvd=ebvs(3:6:122760);
ebvde=circshift(ebvd,[0,-1]);
ebvdl=circshift(ebvd,[0,1]);
ecvp=circshift(ecv,[0,shift]);
ecve=circshift(ecvp,[0,-dT]);
ecvl=circshift(ecvp,[0,dT]);
%ebF=fft(ebvd);
%ecF=fft(ecvd);
ecvsc=[];
%ndet=0;
nbxc=0;
der=dero;
errv=[];
expv=[];
sumerr=0;
rxpp=1;
ixpp=1;
strtcorr=fix(clock)
Df=Dfo;
Df1=Dfo;
sumfrq=0;
sumphs=0;
phro=0;
phs1p=0;
phs1pp=0;
exbv=[];
flg=0;
for nt=1:NT
```

```
    dth=Df*2*pi/30690;
    dthn=Df*2*pi*4;
    for me=1:M

SL1v=(sq2*(ebv*datb(me+(nt-1)*M)-ecv*scm(me))+...
    j*(2*eav+eav.*ebv*datb(me+(nt-1)*M).*ecv*scm(me)))/3;
phr=dth.*nd+(me-1)*dthn+phro;
if me==M
    phro=phr(122760)+dth;
end
SL1r=A*SL1v.*(exp(j*phr))+NP/sq2*(randn(1,122760)...
    +j*randn(1,122760));
%filter
for m=1:17
    if m==1
        SL1lp=SL1r*b(m);
    else
     SL1s=[SL1r(122760-(m-2):122760) SL1r(1:122760-(m-1))];
       SL1lp=SL1lp+SL1s*b(m);
    end

end
%SL1lp=filter(b,1,SL1r);
SL1ds=SL1lp(3:6:122760);
exb=ebvd*SL1ds';
exbv=[exbv exb];
if me==1
    SL1M=SL1lp*scm(me);
else
    SL1M=SL1M+SL1lp*scm(me);
end
    end; % end of M loop
  nt
SL1Ms=circshift(SL1M,[0,shift-8+der]);
exe=SL1Ms*ecve';
exl=SL1Ms*ecvl';
expp=SL1Ms*ecvp';
rxp=real(expp);
ixp=imag(expp)
expv=[expv expp*conj(expp)];
errs=(exl*conj(exl)-exe*conj(exe))/emax;
errv=[errv errs];
sumerr=sumerr+errs;
sumerv(nt)=sumerr;
derv(nt)=der;
der=dero-fix(sumerr*b1);

phs=-(real(expp)*imag(expp))/emax;
%phs=-real(expp)/sqrt(emax);
[der Df phs]
phs1=phs/1000;
phv(nt)=phs;
frqe=(rxp*ixpp-rxpp*ixp)/emax;

frqv(nt)=frqe;
rxpp=rxp;
```

```
ixpp=ixp;
rxpv(nt)=rxp;
ixpv(nt)=ixp;
if nt<8
   sumfrq=sumfrq+frqe;
   Df=Dfo+sumfrq*bf+frqe*bf1;

   Df1=Df;
elseif nt<16
    %if ixp<0
    sumfrq=sumfrq+frqe
    lpgn=(phs1-2*cos(thph)*bph1*phs1p+bph1^2*phs1pp)/phs1
    sumphs=sumphs+phs1*lpgn;
    Df=Dfo+sumphs*bsph+phs1*bph*lpgn+sumfrq*bf+frqe*bf1;
    phs1pp=phs1p;
    phs1p=phs1;
    sumphv(nt)=sumphs;
 else
   sumfrq=sumfrq+frqe/2
   lpgn=(phs1-2*cos(thph)*bph1*phs1p+bph1^2*phs1pp)/phs1
   sumphs=sumphs+phs1*lpgn/2;
   Df=Dfo+sumphs*bsph+phs1*bph*lpgn/2+sumfrq*bf+frqe*bf1/2;
   phs1pp=phs1p;
   phs1p=phs1;
   sumphv(nt)=sumphs;

end
Dfv(nt)=Df;
end; %end of NT loop
mnder=sum(derv(11:NT))/(NT-10)
vard=sum(derv(11:NT).^2)/(NT-10)
endp=fix(clock)
datbr=sign(real(exbv));
datv=xor(datb(251:25*NT)+1,datbr(251:25*NT)+1);
errors=sum(datv)
BER=errors/(NT*25-250)
if BER>.5
    BER=1-BER
end
RMSfreq=sqrt(sum(Dfv(11:NT).^2)/(NT-10))*1000
sN=sqrt(5*4092/2)*1.0;
Ab=A*sqrt(2/9);
nn=0:500;
Figure(1)
%plot(1:NT,errv,1:NT,sumerv)
plot(real(exbv(200:250))/A,imag(exbv(200:250))/A,'.r',...
    real(exbv(251:25*NT))/A,imag(exbv(251:25*NT))/A,'.g',...
   cos(2*pi*nn/500)*1*sN/A+9000,sin(2*pi*nn/500)*1*sN/A,...
   cos(2*pi*nn/500)*1*sN/A-9000,sin(2*pi*nn/500)*1*sN/A);
grid on
axis([-15000,15000,-15000,15000])
Figure(2)
plot(1:NT,expv,[1,NT],[emax,emax])
grid on
Figure(3)
plot(1:NT,derv)
```

```
grid on
xlabel('Time x 100ms')
ylabel('Tracking Error (Chips x 30)')
Figure(4)
plot(25:25:25*NT,Dfv*1000,25:25:25*NT,phv,25:25:25*NT,...
    sumphv*1000,251:25*NT,datv/2-2,'.')
legend('freq offset','phs err','sum phs err','xor(datb,datbr)')
grid on
xlabel('Time (s x 250)')
```