

38246007

A SOFTWARE RADIO APPROACH TO
GLOBAL NAVIGATION SATELLITE SYSTEM RECEIVER DESIGN

A Dissertation Presented to

The Faculty of the

Fritz J. and Dolores H. Russ
College of Engineering and Technology

Ohio University

In Partial Fulfillment

of the Requirement for the Degree

Doctor of Philosophy

by

Dennis M. Akos

August, 1997

Thesis
D
1997
AKOS

Ohio University
Columbus, Ohio

ACKNOWLEDGMENTS

The author would like to express his gratitude to Dr. Michael S. Braasch for serving as the advisor for this work. His support, insight, and advice allowed the research to be successful and of equal importance helped the author to develop professionally. Dr. Frank van Graas must also be singled out for his contributions to the effort. His expertise was invaluable and it was he that provided the author with his initial opportunity for study beyond the undergraduate level. The committee as a whole, Dr. Jeffrey C. Dill, Dr. Constantinos Vassiliadis, and Dr. Larry E. Snyder, must be recognized and thanked for their contributions, time, and effort. The help and assistance provided by faculty, staff, and students affiliated with the Avionics Engineering Center will also be not forgotten.

Outside of Ohio University, a significant amount of appreciation needs to be bestowed upon the folks at the Advanced RF Technology Branch of Wright Laboratory, Wright-Patterson AFB. In particular, Dr. James B. Y. Tsui was an outstanding mentor who provided and conveyed tremendous technical knowledge during the course of the research. Michael H. Stockmaster was a great help in the data collection and the interpretation of that data. There were also other government and industry associations that enabled the work to be completed successfully. The Joint University Program which is sponsored by the Federal Aviation Administration and the National Aeronautics and Space Administration provided a tremendous environment for the research. The collaboration with officials from both these agencies as well as faculty and students from the two other participating schools, Princeton University and the Massachusetts Institute of Technology, was extremely beneficial. The advanced hardware supplied by TRW Electronic Systems & Technology Division, Redondo Beach for evaluation purposes enabled the experimental results.

Finally, one cannot forget the encouragement and support provided by family and friends during an endeavor such as this. Without them and a blessing from the one upstairs it would have not have been possible.

The software based signal processing is the second half of the GNSS software radio design. There is a tremendous level of flexibility associated with a software radio design and this should be reflected in the implementation. The signal processing will be based on established proven algorithms to ensure success. It will be the first time the signal processing has been accomplished entirely in software and will establish a framework for the testing of advanced algorithms. In order to have all the signal processing in software, it will be necessary to include signal acquisition, code and carrier tracking, data demodulation and processing routines. These processing routines must provide a position estimate, one of the outputs of a GNSS receiver. Not only will solving for this position estimate require all the signal processing of a GNSS receiver, but it will also provide validation of the coded algorithms through an accurate computation.

Finally, it is important to recognize that this work is not exclusive to the GNSS broadcast. The software radio implementation is applicable to any RF transmission. Thus this research, particularly in the front end design, can be applied to the capture and processing of any signal. The software signal processing is applicable to all CDMA transmissions which require signal acquisition and tracking. GNSSs provide a platform in which the accuracy and integrity of the signal processing is both complex and critical, thus a software radio implementation will offer significant advantages over a traditional design.

2. Global Navigation Satellite Systems

Satellite-based navigation is the latest development in the quest to know, with some degree of accuracy, relative position. It is the most recent of the fully operational radionavigation systems. Satellite-based systems operate on the time-of-transmission or triangulation concept [1]. The time delay of an RF transmission, scaled by the speed of light, provides the distance between a transmitter and receiver. Assuming the transmitter is at a known location, the position of the receiver is then known to be on a sphere with radius equal to the measured distance. Through multiple simultaneous measurements to different transmitters, a number of these spheres can be obtained whose intersection determines the receiver's position. The use of satellites as transmitters allows for uninterrupted, worldwide, passive, and three-dimensional operation.

At the present time there exist two primary GNSSs in operation: GPS and GLONASS. The intent of this chapter is not to attempt to describe satellite-based navigation and operation. This is an extremely complex topic and there are a number of excellent references that provide an exhaustive study of these topics [1, 2]. The goal here is to describe those system parameters, from a theoretical communications perspective, necessary for a software radio implementation.

2.1 Global Positioning System

GPS is a satellite-based radionavigation system deployed by the United States and managed by the U.S. Air Force. It was initially developed as a military specific technology for the U.S. Department of Defense but quickly evolved into a dual use system as a result of the tremendous potential for civilian use. Thus two levels of service are now available: the military-specific Precise Positioning Service (GPS-PPS) and the Standard Position Service (GPS-SPS) for

widespread public use. The target software radio implementation is a GPS-SPS receiver, utilizing the civilian component of GPS [3].

The GPS space segment consists of 24 satellites operating in six orbital planes (four satellites in each plane). This configuration provides, with high probability, at least four visible satellites which are required to compute a user's three-dimensional position and receiver clock offset. The GPS control segment consists of ground stations whose function is to monitor and maintain the integrity of the satellites. Finally, the GPS user segment consists of the base of receivers which extract position and timing information from the broadcast signal.

Extracting the GPS-SPS component from the multiple frequencies in-phase/quadrature GPS broadcast provides the following signal structure.

$$s_i(t) = \sqrt{2P} CA_i(t) D_i(t) \cos(\omega_c t + \phi) \quad (2.1)$$

where s_i : GPS-SPS broadcast from the i^{th} satellite
 i : indicates the satellite number
 P : signal power
 CA_i : C/A, or PRN, code for the i^{th} satellite (1.023 Mbps)
 D_i : navigation data for the i^{th} satellite (50 bps)
 ω_c : $2\pi f_c = 2\pi 1575.42 \times 10^6$
 ϕ : phase offset

Equation 2.1 indicates that each of the satellites is broadcasting on a common carrier frequency of 1575.42 MHz. This is indeed the case as GPS-SPS uses a CDMA spread spectrum modulation format. The spreading or PRN code for the GPS-SPS signal is known as the Coarse/Acquisition (C/A) code.

The C/A code has a chipping rate of 1.023 Mbps and a period of 1023 chips, or 1 ms. The C/A codes are a subset of the Gold code family, a collection of PRN codes which provide good multiple access properties, or low cross correlation, for their period [4]. Each satellite has a unique C/A code, produced from the modulo-2 sum of two 1023 chip PRN codes G_1 and G_2 . G_1

and G_2 are generated using 10 stage maximal length linear shift registers, initialized to all ‘1’s, with tap positions specified by the generator polynomials:

$$\begin{aligned} G_1: & X^{10} + X^3 + 1 \\ G_2: & X^{10} + X^9 + X^8 + X^6 + X^3 + X^2 + 1 \end{aligned} \quad (2.2)$$

The various C/A codes are formed by delaying the G_2 code a certain number of chips prior to performing the modula-2 summation [5].

The navigation data, $D_i(t)$, provides the user the additional parameters necessary to solve for position, velocity, and time. It is generated at 50 bps and is synchronized with the C/A code. The data is formatted into a 1500 bit frame consisting of five 300 bit subframes. Subframe 1 contains clock and health data specific to the broadcasting satellite. Subframes 2 and 3 hold the ephemeris, or precise orbital, parameters for the broadcasting satellite. Subframes 4 and 5 are subcommutated 25 times each and contain almanac data, or approximate orbital parameters, for all satellites in the GPS constellation. In addition, subframes 4 and 5 also contain system support information such as special messages, atmospheric corrections, and timing transformations.

The exact format of the data and the associated algorithms necessary to solve for position, velocity, and/or time are well documented [2, 5]. It is important to recognize that the navigation data, ignoring the subcommutation of subframes 4 and 5, is repeated every 1500 bits or 30 seconds. Therefore a 30 second ‘window’ of GPS-SPS data will provide all the parameters necessary for a position solution.

The minimum received power level of the GPS-SPS signal into a three dB gain linear polarized antenna is specified to be -160.0 dBW for satellites with an elevation angle greater than five degrees [3]. In general, the received signal power spectral density is below that of the thermal noise of the receiver itself. Spread spectrum signal processing is required to obtain the gain necessary for data demodulation and processing.

2.2 Global Orbiting Navigation Satellite System (GLONASS)

GLONASS is the Russian implementation of a GNSS. Its development, both historically and functionally, parallels GPS. There are, however, some significant differences in the underlying signal structure of the two systems.

Initiated as a military system, control of GLONASS falls under the Russian Ministry of Defense. The GLONASS broadcast, like GPS, consists of multiple components in two distinct frequency bands. As a result of its potential benefit to the civilian community, a portion of the broadcast, which is the responsibility of the Russian Space Agency, has been designated for public use. The following description concentrates on the civilian component of GLONASS [6] which is the focus of the software radio implementation.

The GLONASS space segment consists of 24 satellites operating in three orbital planes (eight satellites in each plane). Again, this configuration has been designed to provide at least four visible satellites at any location on the earth with high probability. Monitoring and maintenance (orbital corrections and data uploads) are handled by the GLONASS ground-based control segment. The GLONASS user segment consists of the base of receivers.

The GLONASS signal of interest is described mathematically as follows:

$$s_i(t) = \sqrt{2P} R(t) M(t) D_i(t) \cos(2\pi(f_c + i * 0.5625 * 10^6)t + \phi) \quad (2.3)$$

where	s_i :	GLONASS broadcast on the i^{th} frequency channel
	i :	indicates frequency channel
	P :	signal power
	R :	PRN code (511 kbps)
	M :	meander sequence (100 bps)
	D_i :	navigation data for the i^{th} channel (50 bps)
	f_c :	1602.0 MHz (GLONASS base frequency)
	ϕ :	phase offset

From this equation the major differences from the GPS signal structure are readily apparent. One

of the principle differences is that GLONASS uses the more common FDMA approach, assigning each satellite a unique frequency.

Initially, there were 24 frequency channels, one for each orbital slot, designated for public use. The specific frequencies are determined from Equation 2.3 using an integer value for i between 1 and 24 which corresponds to the specific channel number. As a result of interference with the radio astronomy band (1610.6 to 1613.8 MHz), changes have been proposed and implemented in the frequency plan [6]. Currently, frequency channels 16, 17, 18, 19, and 20 are not in use. In order to compensate for this change, satellites in antipodal positions are assigned the same frequency. The success of this implementation has led to two future frequency plan changes. In the 1998 to 2005 time-frame, GLONASS satellites will use frequency channels 0 through 12. Beyond the year 2005, GLONASS satellites will be assigned to frequency channels -7 through +6, which indicates the center of the GLONASS band will be shifted to the current base frequency of 1602.0 MHz.

Although GLONASS uses FDMA for multiple access, it also incorporates a PRN code for ranging purposes. The GLONASS PRN code has a chipping rate of 511 kbps and a period of 511 chips, or 1 ms. The PRN code is a maximal length sequence generated using a nine position shift register, sampling the output of the seventh digit. The generator polynomial for GLONASS is given by:

$$G: X^9 + X^5 + 1 \quad (2.4)$$

As a result of the FDMA implementation, every GLONASS satellite uses the same PRN code. The navigation data, $D_i(t)$, is generated at 50 bps and combined with a meander sequence, $M(t)$, generated at 100 bps to form the navigation message. Generation of the PRN code, navigation data, and meander sequence is synchronized within the transmitted signal. As is the case with GPS, the GLONASS navigation data provides the user the additional parameters necessary to

solve for position, velocity, and time. The GLONASS data consists of a 7500 bit superframe made up of five 300 bit frames. Each frame is further subdivided into 15 strings of 100 bits. Strings 1-4 are the same in each frame and provide the precise ephemeris and timing information for the broadcasting satellite. String 5 contains additional timing information. Strings 6-15 differ between frames and provide GLONASS almanac and support information. Each navigation string, which is of two second duration, is divided into two distinct parts. The first 1.7 seconds in any string is the navigation message while the last 0.3 seconds of each navigation string consists of a 30 bit time mark. A shortened PRN sequence provides the 30 bit time mark which is given by:

$$G_M: 111110001101110101000010010110 \quad (2.5)$$

Again, the exact format of the GLONASS data and the associated algorithms necessary to solve for position, velocity, and/or time are well documented [6]. It is important to recognize that the critical navigation data, or that required for a PVT solution, is broadcast every 30 seconds. Therefore, as with GPS, a 30 second ‘window’ of GLONASS data will provide all the fundamental parameters necessary for position computation.

3. Software Radio

There are two fundamental design objectives in the development of a software radio: 1) Position the ADC as near as possible to the antenna in the front end design. 2) Process the resulting samples using a programmable microprocessor. If it is possible to adhere to these objectives, there are a number of potential benefits that can be realized. Each of the objectives will be examined in detail, investigating the advantages of such an implementation as well as the obstacles involved.

3.1 Front End Configuration

The first objective is to place the ADC as near as possible to the antenna in the front end implementation. A traditional front end implementation is depicted in Figure 3.1. This configuration consists of multiple stages of frequency translation and amplification. The benefit here is that the demand on each of the individual components is lessened. However, these multiple stages introduce additional analog components with some potential negative consequences. For example, oscillator performance is a function of both age and temperature. Mixers, and to a lesser extent amplifiers, exhibit nonlinear operating characteristics, resulting in intermodulation and spurious performance. Even filters can introduce unexpected problems. Suppose an LC filter is used as an intermediate element. If the inductor is not properly isolated, it can act as an antenna, injecting undesired noise into the system. All of these effects can be extremely difficult to model, and therefore simulate, making performance estimates troublesome.

It is impossible to place the ADC directly next to the antenna, any RF transmission will require some degree of amplification and filtering. In a software radio, the goal is to minimize the number of analog components, and ideally sample the signal directly at RF. An ideal

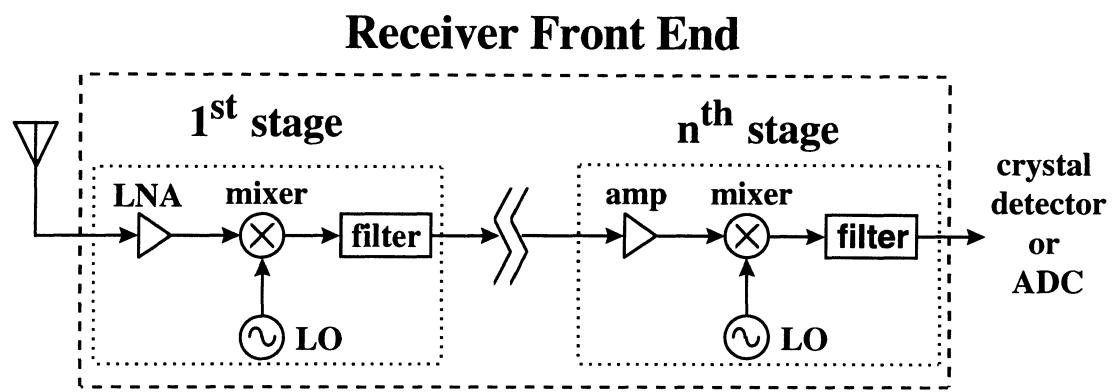


Figure 3.1 Typical Front End Implementation

software radio front end is depicted in Figure 3.2. This implementation samples the signal directly at RF, and greatly simplifies the front end design in terms of the number of components.

Using the front end design depicted in Figure 3.2, the signal must be sampled directly at the carrier frequency. There are two ways to consider the required sampling rate of the ADC involved. One is based on the center frequency and the other is based on the information bandwidth. Both will be briefly summarized and examined in more detail in later chapters.

If one desires the entire band unambiguously, a sampling rate greater than twice the highest signal frequency is required. At the present time, there are only a handful of devices capable of operating at the required frequencies for GNSS transmissions [7]. They are extremely expensive and even more challenging is the subsequent discrete processing which must occur at this same frequency. The practicality of this approach for signals broadcast in the UHF and higher bands is extremely limited by present technology.

However, there is an alternative to the traditional sampling technique. The Nyquist sampling theorem requires that the absolute minimum sampling frequency must be greater than twice the information bandwidth. This suggests the ADC in Figure 3.2 could operate at a fraction of the rate necessary previously. It is important to recognize that this technique, known as bandpass sampling, requires the use of an appropriate amplifier, filter and ADC. It is possible then to sample an RF signal based solely on its information bandwidth [8, 9]. The various stages of local oscillators, mixers, and image reject filters are no longer necessary. Frequency translation is accomplished by intentionally aliasing the signal of interest.

3.2 Software Signal Processing

In a true software radio the samples from the ADC depicted in Figure 3.2 are to be processed on a programmable microprocessor. This will provide the ultimate in receiver

Receiver Front End

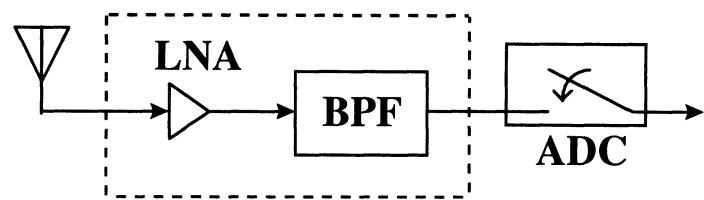


Figure 3.2 Software Radio Direct Digitization Front End Implementation

flexibility [10]. No time-consuming and costly hardware prototyping of different signal processing algorithms is necessary. If a different receiver architecture is desired, the appropriate programming is downloaded to the target processor and executed. Figure 3.3 illustrates the level of flexibility and performance available using various radio architectures.

Assuming the microprocessor of the software radio is replacing an ASIC, flexibility is the primary benefit. As an example, a majority of commercial GPS receivers utilize an ASIC for the initial processing, digital downconversion, correlation, and accumulation of the GPS-SPS signal. As a result of this configuration, the type of acquisition algorithm, which is the first step in processing the signal, is extremely limited. In order to incorporate various acquisition algorithms a new ASIC would have to be designed.

If the ADC/microprocessor software radio platform is replacing analog-based signal processing, the procedure is much more deterministic, and therefore predictable, in nature. This is closely related to the simulation advantage of a software radio. The identical code being used for signal processing can be applied in the actual receiver. With the reduction of front end components, their influence is minimized and effort can be redirected into developing higher accuracy models of the remaining components. Thus, with a software radio there should be little, if any, unexpected results in moving from simulation to receiver implementation.

The flexibility of the software radio allows a single hardware configuration to serve as multiple radios. For example, a broadband antenna, amplifier, and filter could be used to capture a wide span of frequency spectrum, consisting of multiple transmissions. The microprocessor could selectively filter and decimate the desired frequency band, then recall the appropriate program to provide the desired processing. The implementation could process any number of analog and/or digital modulation formats.

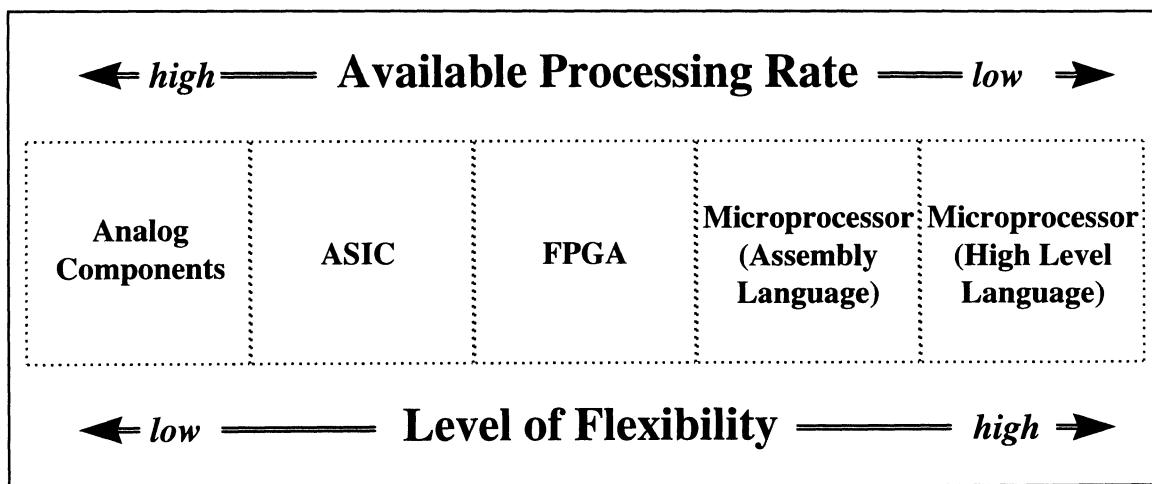


Figure 3.3 Possible Radio Architectures

The disadvantage to processing the resulting samples exclusively in software is the availability and cost of the required programmable computational power, as illustrated in Figure 3.3. A real time GPS-SPS software radio will require a microprocessor capable of providing the entire required signal processing on samples streaming at a 5 MHz rate. This is a challenging task given the current generation of programmable microprocessors [11]. However, available programmable processing power is exponentially increasing. According to Moore's Law microprocessor performance doubles every 18 months [12]. This statement has held true since the inception of the microprocessor. Therefore, the necessary computation power is likely to be available for any software radio implementation given the passage of sufficient time.

As a closing note, it is important to recognize that GNSS is not the only potential software radio implementation. However, the software radio and its associated advantages are of particular importance to the navigation community. This is a result of the critical need for accurate navigation/position information, especially in aviation. A GNSS software radio, with direct RF sampling, eliminates potential IF interference and minimizes problems resulting from RF front end components. Even if direct RF interference were to occur, the programmable signal processing allows the receiver to utilize various architectures, even adaptively given the necessary computational power, to obtain optimal performance. Also GNSS and its associated complex signal processing are relatively new. As a result, there are no "standard" receiver architecture definitions and there are variations in the designs of commercially available receivers, each of which may perform differently in diverse environments. The software radio, with its high level of flexibility, could be used to evaluate each of the competing designs. With this level of understanding, future GNSS software radios could adaptively reconfigure their own architecture to the design most appropriate for the immediate environment and conditions for maximum efficiency.

4. GNSS Front End Design

The front end design and implementation constitutes one half of a software radio development. This chapter will cover front end design for a GNSS receiver within the software radio guidelines. First, the basic front end design equations applicable to this type of implementation are reviewed. Next, the feasibility of sampling the GNSS signal directly at RF is investigated. The theory behind bandpass sampling, the preferred method of directly sampling at RF, is presented. Experimental data is presented which quantitatively validates the applicability of bandpass sampling for GNSS front end design. A novel approach to direct RF digitization of multiple signals is then proposed. The theory, which provides the basis for the implementation, is presented along with experimental results.

In general, sensitivity and dynamic range are the most important factors in the hardware implementation of a front end design. There are well-known equations to calculate both the sensitivity and the dynamic range of a receiver, if the RF front end design is selected [13]. These equations, with slight modification, can be used to calculate the performance of an all digital receiver as well [14].

A generic RF front end consists of amplifiers, filters, and mixers, as depicted in Figure 3.1. Each component has three fundamental parameters: gain, noise figure and third order intercept point [15]. These three component parameters, along with the component order determine the gain, sensitivity and dynamic range of the receiver. Three equations are used to determine the overall gain, noise figure and the resulting third order intermodulation point. The noise figure and third order intermodulation point in turn determine the sensitivity and dynamic range. First, system gain is given by:

$$G_t = G_1 G_2 \cdots G_n \quad (4.1)$$

where G_t is the overall gain of the front end and G_i is the gain of the i^{th} component. The overall

noise figure F_t can be written as:

$$F_t = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \dots + \frac{F_n - 1}{G_1 G_2 \dots G_{n-1}} \quad (4.2)$$

where F_i is the noise figure of the i^{th} component. The overall third order intermodulation product Q_{3t} is:

$$Q_{3t} = \frac{G_t}{\frac{G_1}{Q_{31}} + \frac{G_1 G_2}{Q_{32}} + \dots + \frac{G_1 G_2 \dots G_n}{Q_{3n}}} \quad (4.3)$$

where Q_{3i} is the third order intermodulation product of each individual component.

The criteria used to determine the gain, noise figure and third order intermodulation product are: 1) The gain must be matched, by adding attenuators, to a desired value, 2) The noise figure should be as low as possible, and 3) The third order intermodulation should be as high as possible. However, the noise figure and the third order intermodulation product are opposing parameters: the lower the noise figure, the lower the third order intermodulation product. In general, a compromise must be achieved between the noise figure and the third order intermodulation and this arrangement determines the desired gain value.

As an example, consider the implementation of a front end to receive the GPS-SPS signal. This example assumes that all necessary amplification and noise reduction filtering occurs directly at the RF carrier frequency, as would occur in the direct digitization front end depicted in Figure 3.2. In order to digitize the signal using an ADC, the power level should be at least -45 dBm. A total of 90 dB gain is chosen to amplify the signal to the desired power level. Since a single 90 dB gain amplifier is impractical due to non-linearities, the necessary gain can be obtained by cascading three identical commercially available amplifiers, each with the following specifications; frequency range: 1-2 GHz, gain: 30 dB, noise figure: 2 dB, third order

intermodulation point: 23 dBm. A bandpass filter centered at 1575.42 MHz with a 3 dB bandwidth of 3.4 MHz and an insertion loss of 5.0 dB is available to limit the out of band noise.

The above four components can be cascaded in different ways to obtain different designs. The general rule can be stated as follows. The closer the filter is placed to the antenna, the higher the noise figure and dynamic range. If the filter is placed far away from the antenna, the opposite is true. All the received GPS-SPS signals are close in amplitude, thus the dynamic range requirement on the receiver is low, neglecting interference. All have the same carrier frequency, with any deviation strictly as a result of the Doppler effect and the satellite frequency reference offset. Since each amplifier has 30 dB gain and a 2 dB noise figure, the filter can be placed at any position after the first amplifier, with a negligible effect on the overall noise figure.

However, in a practical implementation it is prudent to place the filter after the first amplifier in order to limit other signals from generating spurious responses. This arrangement will limit out-of-band signals from getting into the second and third amplifiers. It is also possible to place the filter before the first amplifier to limit undesired signals. This arrangement will degrade the sensitivity by 5.0 dB, the insertion loss of the filter and significantly increase the noise figure of the system.

In the practical implementation of a direct digitization front end, there is another factor that must be considered. If the final component prior to the ADC is an amplifier rather than a filter, additional noise will be folded, or aliased, into the resulting information band (assuming bandpass sampling is used). The amount of noise folded into this band will be proportional to the amount of amplification, in terms of both gain and bandwidth, between the last filter and the ADC. This will be described and illustrated in the discussion on bandpass sampling.

4.1 Direct RF Digitization

If the proposed front end design (depicted in Figure 3.2) is to be utilized, then the ADC will be required to sample the desired signal directly at RF. There are two vastly different ways to view the required sampling frequency for the ADC in a direct digitization approach. In this section the theory will first be presented and the GPS-SPS broadcast will be used as an example. Recall that GPS-SPS operates on a RF carrier of 1575.42 MHz and a first null bandwidth of approximately 2 MHz. These parameters provide the necessary information to determine the sampling frequency requirement. It is important to note the advantage in that any sampling directly at RF eliminates various front end components and their associated error contributions.

First, one can consider the minimum sampling frequency based on the RF carrier frequency, f_c , and information bandwidth, BW_I . The minimum allowable sampling frequency, f_s , in this case is given in Equation 4.4.

$$f_s > 2 * \left(\frac{f_c + BW_I}{2} \right) \quad (4.4)$$

The advantage to using this basis for the choice of sampling frequency is that the information in the range $[0, f_s/2]$ is uniquely identified. Therefore, this sampling frequency provides the potential to recover any RF transmission in that frequency range. The disadvantages at the present time to this approach are numerous. In the case of GPS-SPS, this method would require a state-of-the-art ADC operating at a sampling frequency greater than 3 GHz. A related, and more difficult, problem is the processing of the resulting samples. There is no single processor solution available to this problem and any parallel processing implementation would be extremely expensive if it were even possible. Therefore, this type of implementation is impractical at the present time.

Second, the minimum sampling frequency of the information band is expressed in Equation 4.5:

$$f_s \geq 2 * BW_I \quad (4.5)$$

If the signal is bandlimited to its information bandwidth, then f_s represents the Nyquist rate. In the case of GPS-SPS, the information bandwidth can be approximated by the first null bandwidth, thus a sampling frequency requirement of about 4 MHz is imposed. This is a fairly novel idea: sample an 1575.42 MHz RF carrier at about a 4 MHz rate and extract all the desired information. The technique itself is referred to as bandpass sampling or intentional aliasing. As with the previous approach, there are trade-offs in its implementation that are discussed in detail in the next section. However, the primary advantage to this technique is obvious. Sampling occurs at a much lower rate and thus processing the resulting samples is a feasible operation.

4.2 Bandpass Sampling

Bandpass sampling is the technique of undersampling a modulated signal to achieve frequency translation via intentional aliasing [8, 9]. A high level frequency domain depiction of this process is presented in four stages in Figure 4.1 and is based on the direct digitization front end of Figure 3.2.

The signal enters through the antenna and is amplified by the low-noise amplifier (LNA) along with all frequencies within the bandwidth of the LNA (stage 1). In a bandpass sampled system, the amplified signal would then pass through a narrow bandpass filter centered about the carrier frequency. This filter would attenuate all frequencies outside of the information band (stage 2). Next a sampling frequency, f_s , is chosen which defines the resulting sampled bandwidth, $[0, f_s/2]$, as well as the arrangement of the aliasing triangles depicted in stage 3. After sampling, the information band along with the noise from each aliasing triangle is folded

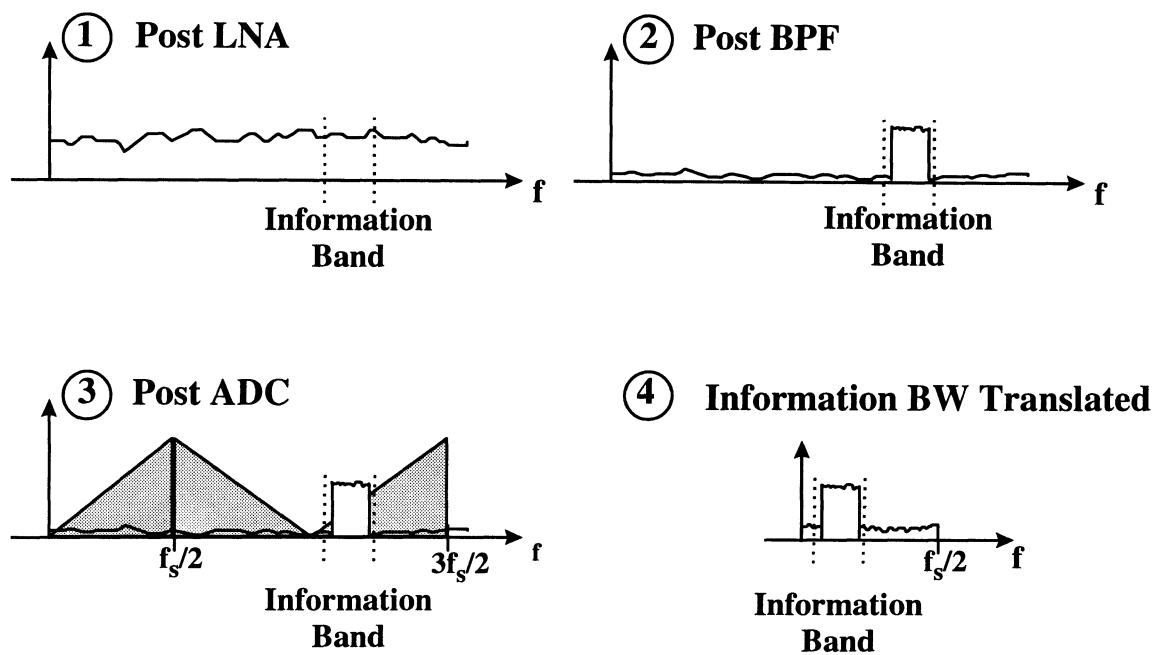


Figure 4.1 Frequency Domain Depiction of the Various Output Stages of a Bandpass Sampling Front End

into the resulting sampled bandwidth (stage 4). Thus the information band is translated without any analog downconversion stages.

4.2.1 Theoretical Background

The process needs to be described mathematically to obtain a more detailed understanding. First, the translation of the original carrier frequency, f_c , to the resulting intermediate frequency, f_{IF} , as a function of the sampling frequency, f_s , must be defined mathematically. This is presented in Equation 4.6 [16].

$$\text{if } \text{fix}\left(\frac{f_c}{\frac{f_s}{2}}\right) \text{ is } \begin{cases} \text{even, } f_{IF} = \text{rem}(f_c, f_s) \\ \text{odd, } f_{IF} = f_s - \text{rem}(f_c, f_s) \end{cases} \quad (4.6)$$

where: $\text{fix}(a)$ is the truncated integer portion of argument a

$\text{rem}(a, b)$ is the remainder after division of a by b

It is important to recognize that Equation 4.6 provides a means of calculating the resulting intermediate frequency (IF) position. Associated with this IF are the corresponding modulation sidelobes that designate the information bandwidth. It is important that f_s be chosen such that the entire information bandwidth is translated within the resulting sampled bandwidth. This can be ensured if the constraint in Equation 4.7 is met.

$$\frac{\text{BW}_I}{2} < f_{IF} < \frac{f_s - \text{BW}_I}{2} \quad (4.7)$$

If this constraint is not met, a portion of the information band of the signal can fold on top of itself, creating destructive interference.

Assuming an appropriate sampling frequency has been selected, the trade-offs in using bandpass sampling, as opposed to traditional sampling, can be discussed. Again, the primary

advantage is that sampling frequency and consequent processing rate are proportional to the information bandwidth rather than the carrier frequency. However, bandpass sampling has some fairly unique hardware requirements that may be considered its disadvantage. One critical requirement is that the analog input bandwidth of the ADC must accommodate the RF carrier, although its sampling frequency can be much less as it is based on the information bandwidth. A narrow bandpass filter centered about the RF carrier is a second requirement. Ideally, this filter must attenuate all energy outside the information bandwidth. This is important as all frequencies, not only the information band, from 0 Hz to the input analog bandwidth of the ADC will fold into the resulting passband, thus affecting the SNR of the information band.

4.2.2 Experimental Results

The direct digitization bandpass sampling theory has been introduced in the previous section. This section will describe experiments and the subsequent results which provide quantitative measurements indicating the optimal hardware configuration. In addition, it will be shown that the direct digitization bandpass sampling front end provides nearly identical performance, in terms of SNR, to the traditional design. Finally, this section will present the design of a direct digitization bandpass sampling front end for the GPS-SPS broadcast and validate its operation.

4.2.2.1 Component Configuration

Previously the idea had been presented that the front end component directly preceding sampling should be a filter in a direct digitization bandpass sampling approach. The hypothesis, illustrated by Figure 4.1, is that this final filter is necessary to minimize the noise that is aliased

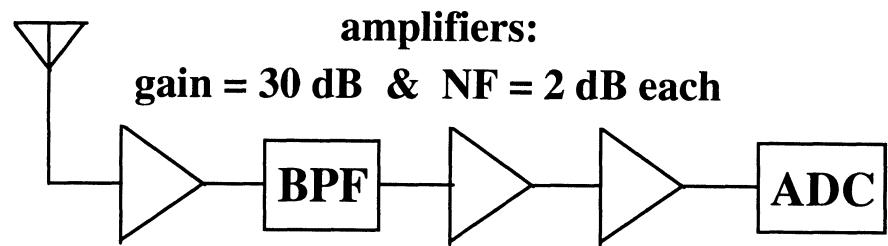
into the resulting sampled information bandwidth. The following experiment verifies the assumption.

In this experiment, two GPS-SPS bandpass sampling front end implementations are presented. The signal-to-noise ratio (SNR) is measured using a CW signal as input. In the first design, the arrangement is the equivalent to that used in analog receiver design. The filter is placed after the first amplifier as shown in Figure 4.2a. Since the amplifier has a gain of 30 dB, the noise figure of the system is approximately 2 dB, the noise figure of the first amplifier. In this case, the filter only limits the noise generated from the first amplifier, but the filter does not limit the noise generated from the second and third amplifiers.

The second design places the filter after the last amplifier as shown in Figure 4.2b. The noise figure is still about 2 dB, but the third order intermodulation will be lower than the first arrangement. Since typical GPS receivers do not have a stringent dynamic range requirement, barring RF interference, this arrangement will not create an adverse effect on the receiver performance. In this configuration, the filter limits the noise from all three amplifiers. Note that both arrangements use the same components: three 30 dB amplifiers and a bandpass cavity filter centered at 1575.42 MHz with a 3 dB bandwidth of 3.2 MHz.

The frequency responses, measured via a spectrum analyzer, from these two arrangements are shown in Figure 4.3. The displayed frequency range is from 1.5 to 1.6 GHz. Curve A and curve B show the results of the first and second arrangements respectively. In the passband of the filter the two curves have the same amplitude. Since only noise from the first amplifier is limited in the configuration A, the noise floor is higher. The noise floor of curve A should be approximately 60 dB higher than curve B, however the limitations of the spectrum analyzer result in only a 20 dB difference being illustrated.

4.2a: Configuration A



4.2b: Configuration B

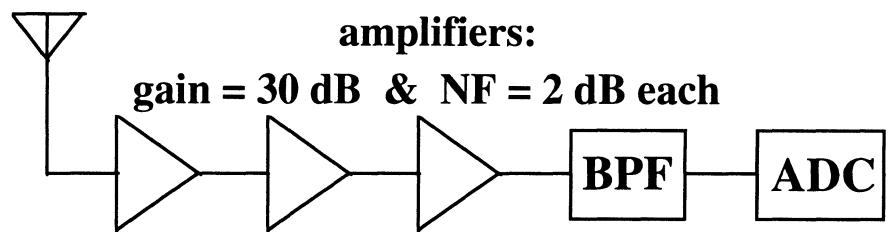


Figure 4.2 Two Bandpass Sampling Front End Implementations

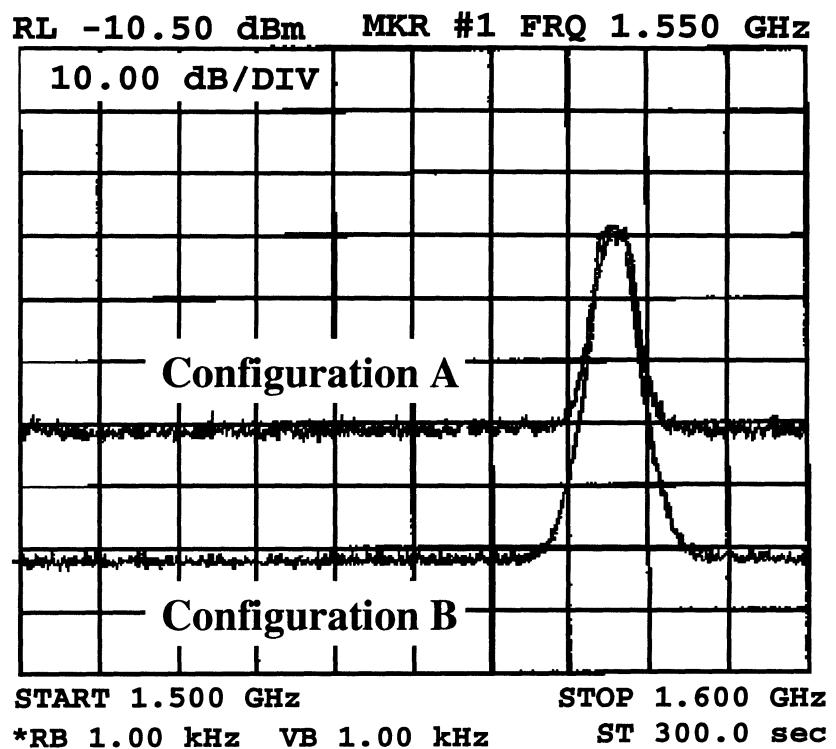


Figure 4.3 Frequency Response of Bandpass Sampling Front End Implementations

In order to evaluate both arrangements for a direct digitization implementation, a CW signal was used. Although both configurations were designed to perform as a GPS- SPS receiver front end, it is difficult to obtain quantitative results with actual GPS signals due to the CDMA spread spectrum modulation. Therefore, a signal generator was used as input with a center frequency of 1575.42 MHz and output power setting of -110 dBm. The settings were based on the actual parameters of the GPS-SPS signal, with a 20 dB stronger power level used to ensure an adequate measurement.

The expected SNR can be found from the following procedure. The noise floor of a 500 Hz bandwidth system with 3 dB noise (including 1 dB insertion loss of the input cable) is calculated in Equation 4.8.

$$\begin{aligned} & 10\log_{10}(1000 \cdot T \cdot B) + 3 \\ & = 10\log_{10}(1000 \cdot 1.3807e-23 \cdot 290 \cdot 500) + 3 \\ & = -144 \text{ dBm} \end{aligned} \quad (4.8)$$

The corresponding SNR is:

$$\text{SNR} = -110 - (-144) = 34 \text{ dB} \quad (4.9)$$

The CW signal is applied to the input of the first amplifier in both configurations in order to measure the SNR. This ratio is calculated by using the FFT of the digitized signal. The Tektronix TDS 684A digital oscilloscope is used as the ADC with a sampling rate of 5.0 MHz, thus the CW signal will be aliased to a 420 kHz IF, calculated using Equation 4.6. A 10,000 point FFT, which spans 2 ms and corresponds to a frequency resolution of 500 Hz, is performed on the sampled data. The input frequency was perturbed slightly from the 1575.42 MHz setting so that the energy of the CW signal was contained within a single frequency bin. Figure 4.4 shows the results of the 10,000 point FFT and an enlarged view of only 21 points about the center frequency bin of the CW signal. It appears that the signal only occupies one frequency bin.

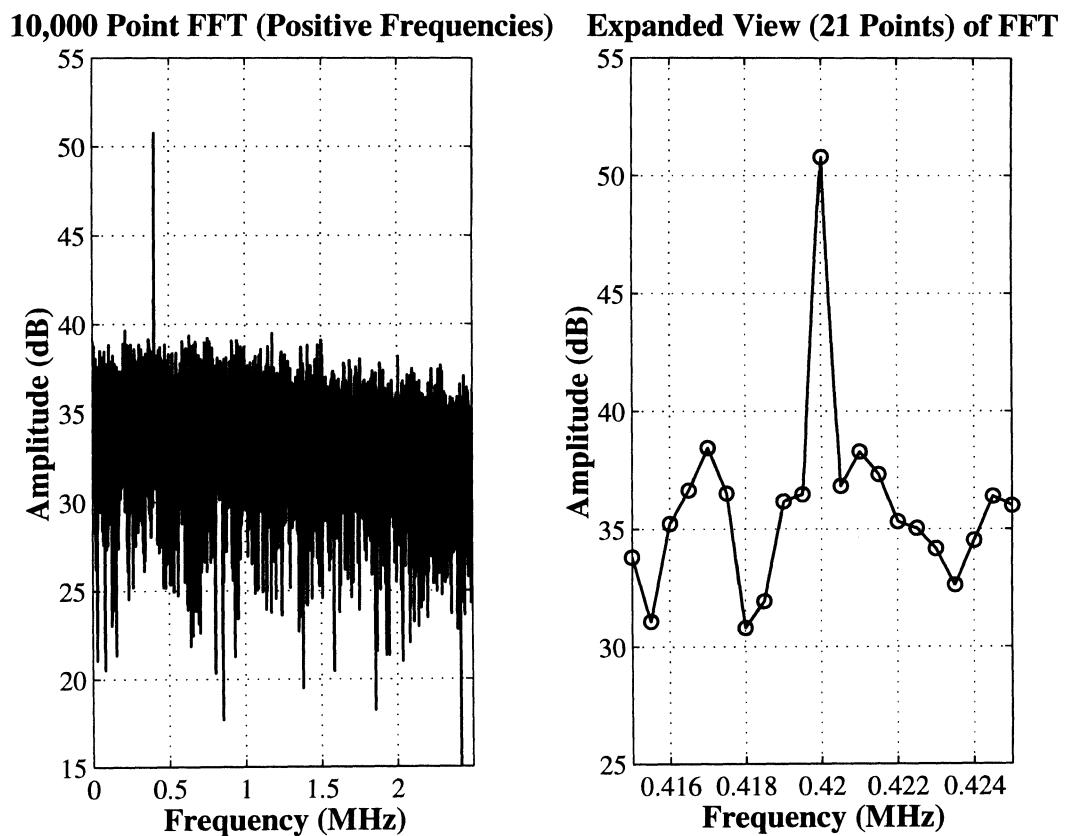


Figure 4.4 FFT Magnitude Response of Sampled Signal

The signal power is calculated from the square of the signal amplitude and the noise power is calculated from averaging the square of the remaining data points. Recall that the second arrangement (Figure 4.2b) is suspected to be the preferred method in the implementation of a bandpass sampling front end. The measured SNR for this configuration is 31.8, about 2.2 dB lower than that predicted. This error could be attributed to three possible sources. First, it might be a result of the measurement equipment, as there is no known equipment readily available to measure the input power directly at -110 dBm. A second possibility is that the filter 3 dB bandwidth of 3.4 MHz is greater than the Nyquist sampling frequency (2.5 MHz), thus additional noise is folded into the desired band. Finally, the degradation may be the result of the attenuated noise folding back into the resulting information bandwidth. Most likely the result is a combination of the three possibilities.

The SNR of the first arrangement (Figure 4.2a) is 24.8 dB which is 7.0 dB less than the second arrangement and 9.2 dB less than predicted result. Although the absolute SNR measured could contain some error, the relative values between the two measurements should be accurate and is valid for comparison.

From this result one may conclude that the filter should be placed at the end of the amplifier chain to obtain the best SNR. This novel sampling technique precludes the use of an anti-aliasing filter with the ADC as the intent is to purposely alias the information band. In order to do so effectively, a bandpass filter, centered about the carrier frequency with passband equal to the desired signal bandwidth, should directly precede the ADC. Also, it is important that this filter has an ultimate reject stopband as low as possible from DC to input bandwidth of the ADC. Any noise within this region will be cumulatively aliased, or folded, into the resulting sampled bandwidth as a result of the direct digitization and degrade the final SNR.

4.2.2.2 Direct Digitization Compared with Downconvert-and-Digitize

Now that the ideal arrangement of front end components has been verified, a similar experiment can be performed to compare the SNR obtained using this direct digitization bandpass sampling technique and the traditional downconvert-and-digitize approach. Again, in order to obtain a quantitative metric, a CW signal is used as the test input.

The experiment is fundamentally identical to that used in testing the two previous configurations. A signal generator, with a 1575.42 MHz frequency and -110 dBm power setting, is used as an input into two different front end designs. The output of each of the front ends is sampled at 5 MHz using the Tektronix TDS 684A digital oscilloscope. A 10,000 point FFT is performed on the data and used in the calculation of the SNR.

The expected SNR has already been determined to be 34 dB from Equation 4.9. Further, the SNR for the preferred direct digitization bandpass sampled front end configuration, depicted in Figure 4.2b, has been calculated to be 31.8 dB. What remains is to perform the experiment on a downconvert-and-digitize implementation, which is depicted in Figure 4.5.

The downconvert-and-digitize arrangement uses a single stage of frequency translation prior to sampling. In this design 30 dB of gain is placed in front of the first filter. This filter has parameters more typical of a first stage filter: a center frequency of 1575.42 MHz and a 3 dB bandwidth of 86 MHz. The signal is mixed with a LO at 1554.17 MHz so that the resulting IF is at 21.25 MHz. After mixing, the signal is further amplified then filtered to remove the double-frequency term. The second filter has a center frequency of 21.4 MHz and a 3 dB bandwidth of 2.25 MHz. The 21.25 MHz signal sampled at 5 MHz will alias to 1.25 MHz, the center of the resulting information bandwidth. Bandpass sampling is also used in this case, however the input bandwidth of the ADC required to digitize this IF signal is only in the 25 MHz range rather than

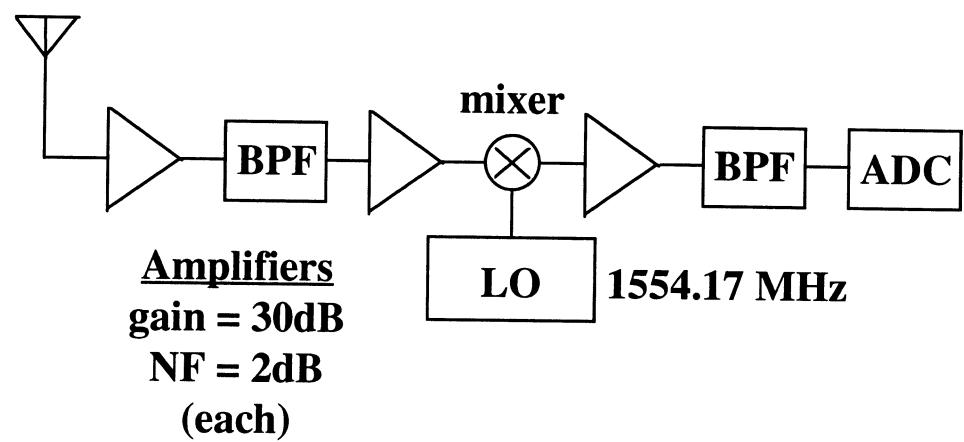


Figure 4.5 Downconvert-and-Digitize Front End Design

the GHz range. This configuration is an attempt to implement the traditional design for SNR comparison.

The SNR measured under this condition is 32.6 dB. Again, the expected SNR is 34 dB. The measured values are slightly less than that expected, but the relative values provide the desired insight. The SNR of the downconverted approach is 0.8 dB (32.6 - 31.8) better than the direct digitization. This difference can be ascribed to the different filter bandwidth directly prior to sampling. The direct digitization approach had a final filter 3 dB bandwidth of 3.4 MHz, which is wider than the resulting sampled bandwidth of 2.5 MHz, thus more noise folded into the resulting band. The downconvert and digitized final filter had a 3 dB bandwidth of 2.25 MHz which is narrower than the resulting sampled bandwidth. If a filter with narrower bandwidth can be utilized in the direct digitization implementation, the SNR of both approaches should be equal.

4.2.2.3 Direct Digitization of the GPS-SPS Signal

These two experiments are important as they provide a quantitative result which verifies the direct digitization bandpass sampling theory presented in Section 4.2.1. The first experiment presented an additional design consideration in constructing a digital receiver front end, critical to achieving the calculated performance. More importantly, the second experiment verified the results obtained using the direct digitization bandpass sampling are equivalent to that obtained using the traditional approach. This is significant as it demonstrates the feasibility of a software radio front end implementation and all the associated benefits described in Chapter 3. Both experiments, thus far, have attempted to mimic the processing of the GPS-SPS transmission, but instead used a CW signal to provide quantitative metrics. The final experiment will involve the

GPS-SPS transmission and confirm that the direct digitization bandpass sampling technique can be applied effectively with the actual signal.

The first published GPS-SPS direct digitization bandpass sampling data directly resulted from this research [16]. This demonstrated the feasibility of the software radio front end for a GPS-SPS receiver. However, the results presented in that initial work were limited by the ADC. A Tektronix TDS 684A digital oscilloscope was used as the ADC and this offered limited control of the sampling frequency. As a result, a 5 MHz sampling frequency was used which aliased the GPS-SPS center frequency to 420 kHz. Although the data indicated the GPS-SPS could be processed, this sampling frequency does not meet the constraint imposed by Equation 4.7. Ideally, a GPS-SPS direct digitization bandpass sampled front end design would begin with the calculation of an appropriate sampling frequency, then continue by assembling the required components, and conclude by collecting data and verifying the result. This is the goal of the next experiment.

The important signal parameters for GPS-SPS are the 1575.42 MHz carrier frequency and the 2.0 MHz first null bandwidth. These parameters, when used with Equations 4.6 and 4.7, will determine an appropriate sampling frequency. Figure 4.6 shows the results of direct RF sampling of the GPS-SPS signal at rates between 3.975 MHz and 4.025 MHz. Superimposed on each plot are dashed lines representing the constraints of Equation 4.7. The lower bound is a constant set at 0 while the upper bound is monotonically increasing, although it may not appear to be over such a short span, and equal to the sampling frequency divided by two. The top figure indicates the IF of only the carrier after bandpass sampling. As shown in the plot, the carrier itself always falls within the constraints. The middle plot of Figure 4.6 indicates the frequency of the carrier and associated sidebands after sampling. It is clear from the figure that a great majority of the sampling frequencies evaluated do not meet the required constraints. The bottom

plot shows only those bands in which all constraints are met. It should be obvious from this plot that no sampling frequencies less than 4 MHz are adequate, which is theoretically correct as it is less than the Nyquist rate for the 2 MHz information bandwidth. This plots shows four narrow bands of acceptable sampling frequencies. In order to clarify what occurs in these acceptable bands, Figure 4.7 shows the results for a slightly higher range of sampling frequencies. The higher sampling frequencies provide a wider band of acceptable frequencies, better illustrating the process.

It is important to consider not only the 2 MHz first null bandwidth of the GPS-SPS signal, but also the bandwidth of the filter directly prior to sampling. The narrowest cavity bandpass filter available has a 3 dB bandwidth of 0.2% of the carrier frequency or 3.2 MHz for GPS-SPS. Now a decision must be made as to use the first null bandwidth or the filter 3 dB bandwidth to define the constraints of Equation 4.7. It is important to recognize that there are many definitions for bandwidth: absolute, 3 dB, first null, equivalent noise bandwidth. As a compromise, we will assume an information bandwidth of approximately 2.5 MHz for the GPS-SPS signal for defining our constraints. Using this metric with Equations 4.6 and 4.7, which provide the data for Figure 4.7, a sampling frequency of 5.161 MHz, which will alias the GPS-SPS center frequency to 1.315 MHz, is adequate.

The actual front end implementation has been depicted in Figure 4.8. The signal enters through an active GPS-SPS antenna, which provides approximately 26 dB of gain, then is split into two paths. The first goes to the input of a GEC Plessey GPS receiver, which is used to determine the visible GPS satellites. The second path goes into the direct digitization front end. It is filtered using a ceramic filter with center frequency of 1575.42 MHz, 3 dB bandwidth of 10 MHz, then further amplified by cascading two amplifiers, each providing 30 dB of gain. The signal is filtered again using a cavity filter centered at 1575.42 MHz with a 3 dB bandwidth of

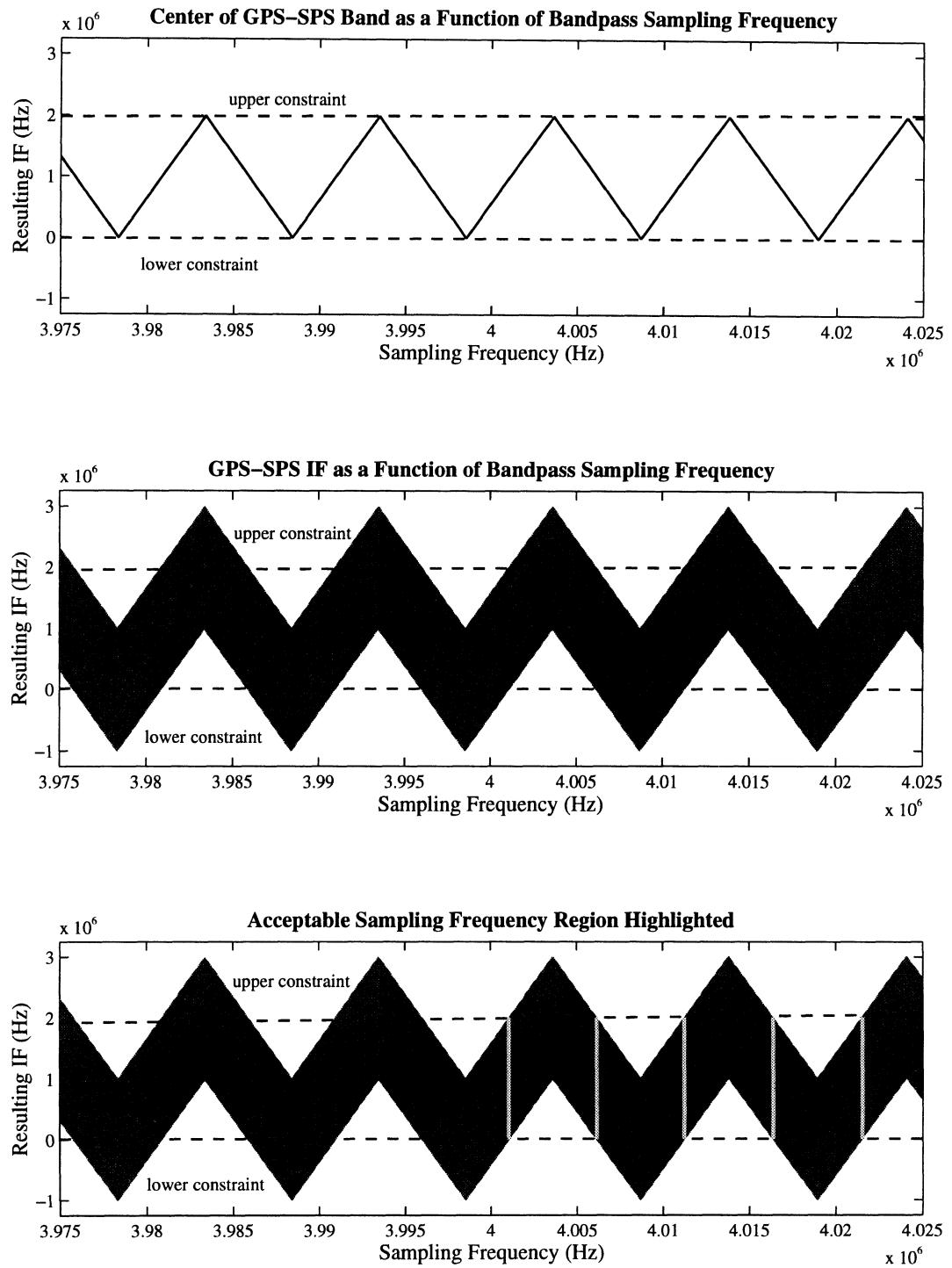


Figure 4.6 Results of Direct RF Sampling the GPS-SPS Signal

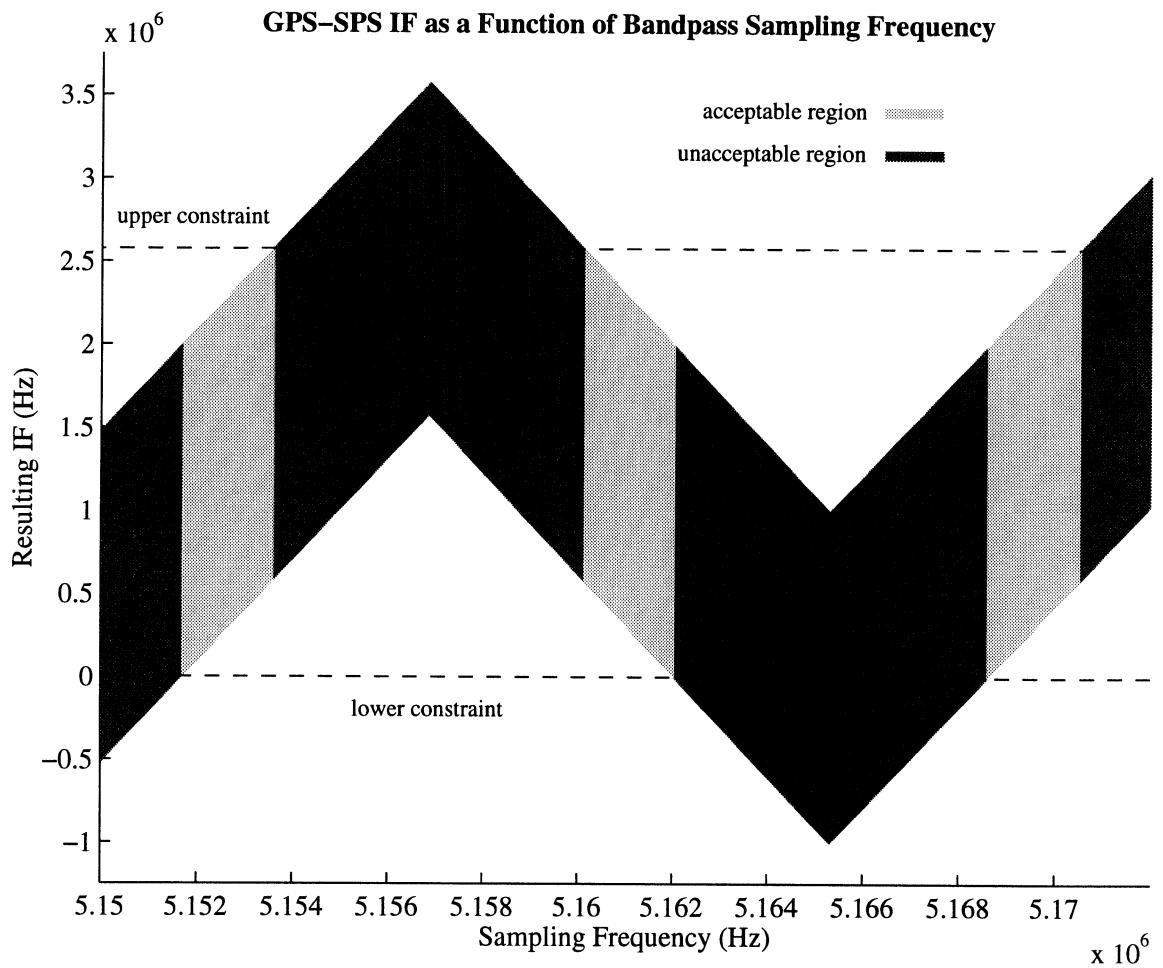


Figure 4.7 Results of Direct RF Sampling the GPS-SPS Signal - Enhanced View

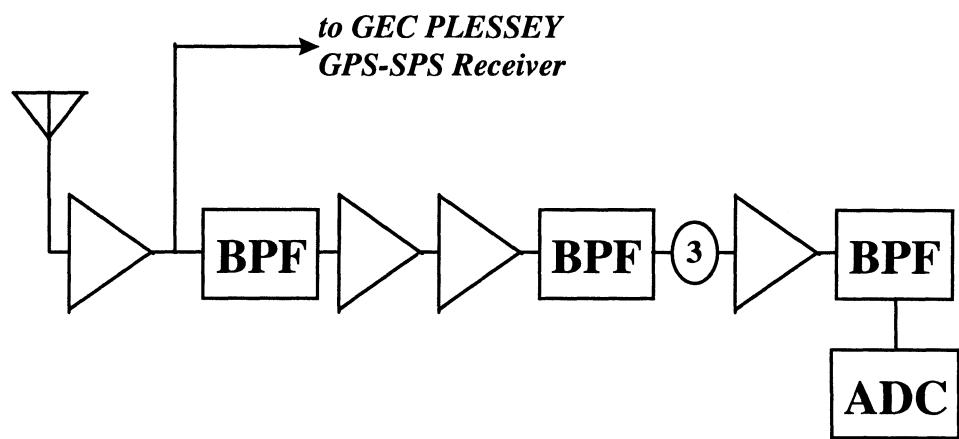


Figure 4.8 Direct Digitization GPS-SPS Front End Implementation

3.2 MHz, then attenuated. The signal is amplified a final time and passed through a second cavity filter centered at 1575.42 MHz and 3 dB bandwidth of 3.2 MHz prior to sampling. This implementation uses additional components when compared with the ideal software radio front end design from Figure 3.2. However, the more problematic components related to frequency translation have been excluded from the design. Rather, multiple amplifiers were used as was the case in previous experiments to achieve the necessary gain. Additional filters have also been incorporated into the design to minimize the potential spurious response of the required amplifiers.

The ADC is a key component in the design. It is the AMAD-7 High Speed Monolithic GaAs HBT ADC from TRW Electronics Systems. This component provides 4-bit samples at rates up to 1.25 GHz with a 2.5 GHz analog input bandwidth. The input bandwidth can accommodate all GNSS signals currently in use. It is necessary to clock the ADC at least at 100 MHz to ensure the sample and hold circuit does not droop. In order to achieve the desired sampling rate, or any rate below 100 MHz, the output of the ADC is further subsampled to obtain the correct rate.

The attenuator is set as to adjust the overall gain of the system in order to exercise the entire dynamic range of the ADC, ± 64 mV, without clipping the signal. Experimental results show a setting of 3 dB provides the ideal amount of gain in the design.

At the time of data collection, the GEC Plessey indicated it was tracking five satellites (PRN numbers: 2, 7, 15, 27, and 31). Therefore, these broadcasts should be present in the collected data. Figure 4.9 shows the time domain, magnitude of the frequency domain, and histogram plots of the raw data.

As should be expected, there is no discernable signal visible in any of the plots of Figure 4.9 even though the GEC Plessey receiver was tracking five different satellites at the time of data

collection. The time domain plot does not resemble any type of sinusoidal waveform, the frequency domain plot appears to be flat with no indication of any signal at the GPS-SPS IF of 1.315 MHz, and accordingly a histogram of the collected samples approximates a Gaussian distribution. This is a result of the low strength of the GPS signals.

In order to verify the signals are present in the data captured using the novel front end design, it is necessary to take the first step in processing the collected data. The initial step in processing any CDMA transmission is acquisition. Acquisition is used to determine the specific PRN code, the phase of the PRN code (typically within $\pm\frac{1}{2}$ of a chip), and an estimate of the carrier frequency shift [17]. A more detailed description of the algorithm is provided in Chapter 5. If the specific PRN code and phase can be identified, the spread spectrum modulation can be removed by correlating the collected data with a synchronized version of the identified code. The result will be the GPS-SPS IF carrier modulated only with the navigation data in noise, which should be definitely visible in the frequency domain. The acquisition algorithm search space could be reduced to the five satellite PRN codes that the Plessey receiver was tracking at the time of data collection, however, all possible 32 GPS-SPS PRN codes were tested. The results, in Table 4.1, show six satellites that were identified in the collected data, five the GEC Plessey receiver was tracking plus an additional visible satellite. In addition to the satellite PRN numbers, the table list the code phase, in terms of samples, and the carrier frequency, resolved to within 50 Hz, returned from the acquisition algorithm. The code phase indicates at which sample number the PRN code for that particular satellite actually starts.

Now it is possible to view the post-correlation FFT and verify that the signal is indeed present in the data set by the appearance of a spike at the corresponding frequency. First, in order to verify that the appearance of a spike in the post-correlation FFT is not an arbitrary result, two tests will be conducted. The first will be to correlate the collected data with a PRN code of a

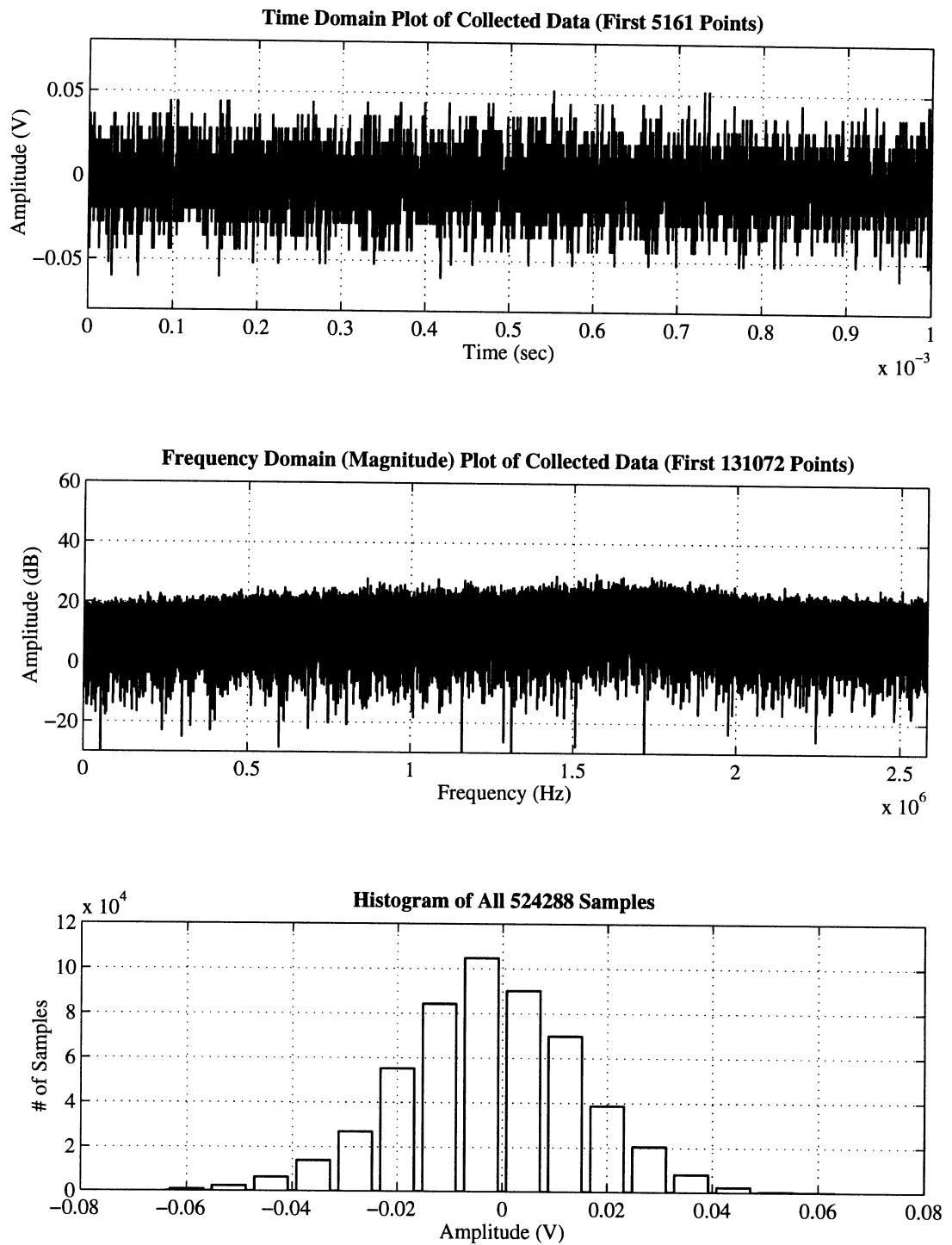


Figure 4.9 Time, Frequency, and Histogram Plots of the Collected Data

Table 4.1 Satellites Identified in Collected Data via Acquisition

Satellite PRN #	Code Phase (samples)	Frequency (Hz)
2	4646	1317050
7	4955	1317650
15	2750	1316000
19	5074	1312100
27	4488	1314700
31	1983	1311800

satellite that was not visible at the time of data collection. The second will be to correlate the collected data with the PRN code of an satellite that is present in the data set, but the code phase alignment is off by more than $\frac{1}{2}$ of a chip. The results of test are presented in Figure 4.10. The top plot shows the FFT of the uncorrelated data, equivalent to middle plot of Figure 4.9, and is used as a reference. After correlation with the PRN code from a satellite on the opposite end of the earth at the time of data collection, any change in the magnitude of the post-correlation FFT is insignificant. The bottom plot is the magnitude of the FFT after correlating the collected data with the PRN code of SV#2 with an arbitrary code phase. Thus it can be concluded that there is little change in the magnitude of the FFT if the data is correlated with the PRN code of a satellite not in view at the time of data collection or the PRN code of a satellite that is in view with the incorrect code phase.

The collected data can now be correlated with the PRN codes of the visible satellites using the correct code phases listed in Table 4.1. A magnitude plot of each of the six post-correlation FFTs is presented in Figure 4.11. When the correct code/code phase is applied to the collected data, the spreading code is removed and a definite spike is present in the plot at the frequency indicated in Table 4.1. This is true for all five satellites the GEC Plessey receiver was tracking and also a sixth. Thus the data indicates that a direct digitization bandpass sampling front end can implemented successfully for GPS-SPS.

4.3 Multiple Signal Bandpass Sampling

Now that a direct digitization bandpass sampling front end for GPS-SPS has been implemented successfully, it is now necessary to consider the simultaneous processing of multiple GNSS signals. Of course, it is important to recognize that the theory about to be presented is applicable to any RF transmission. However the focus of this work is on the various

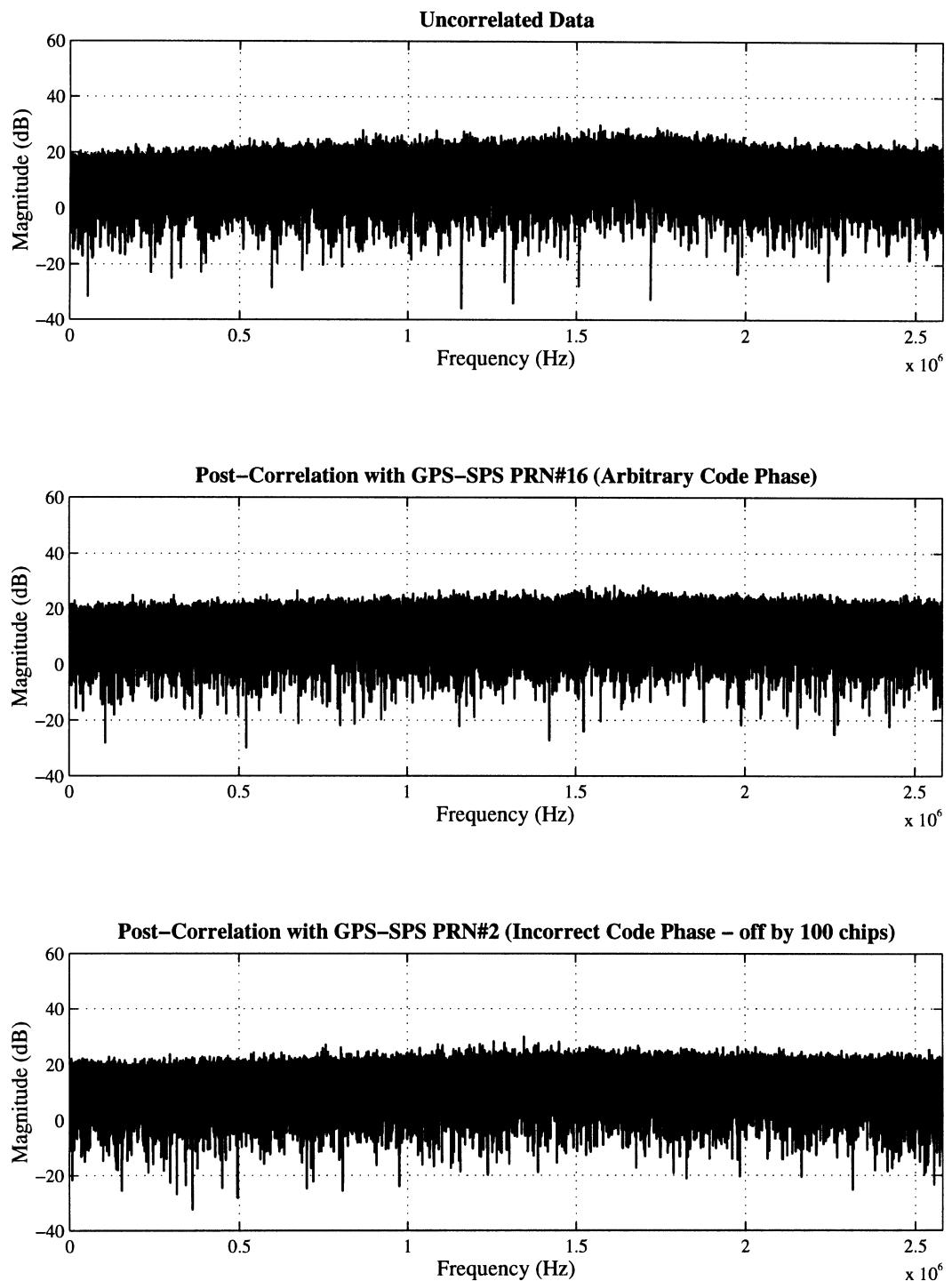


Figure 4.10 Frequency Domain Plots of Collected and Post-Correlated Data

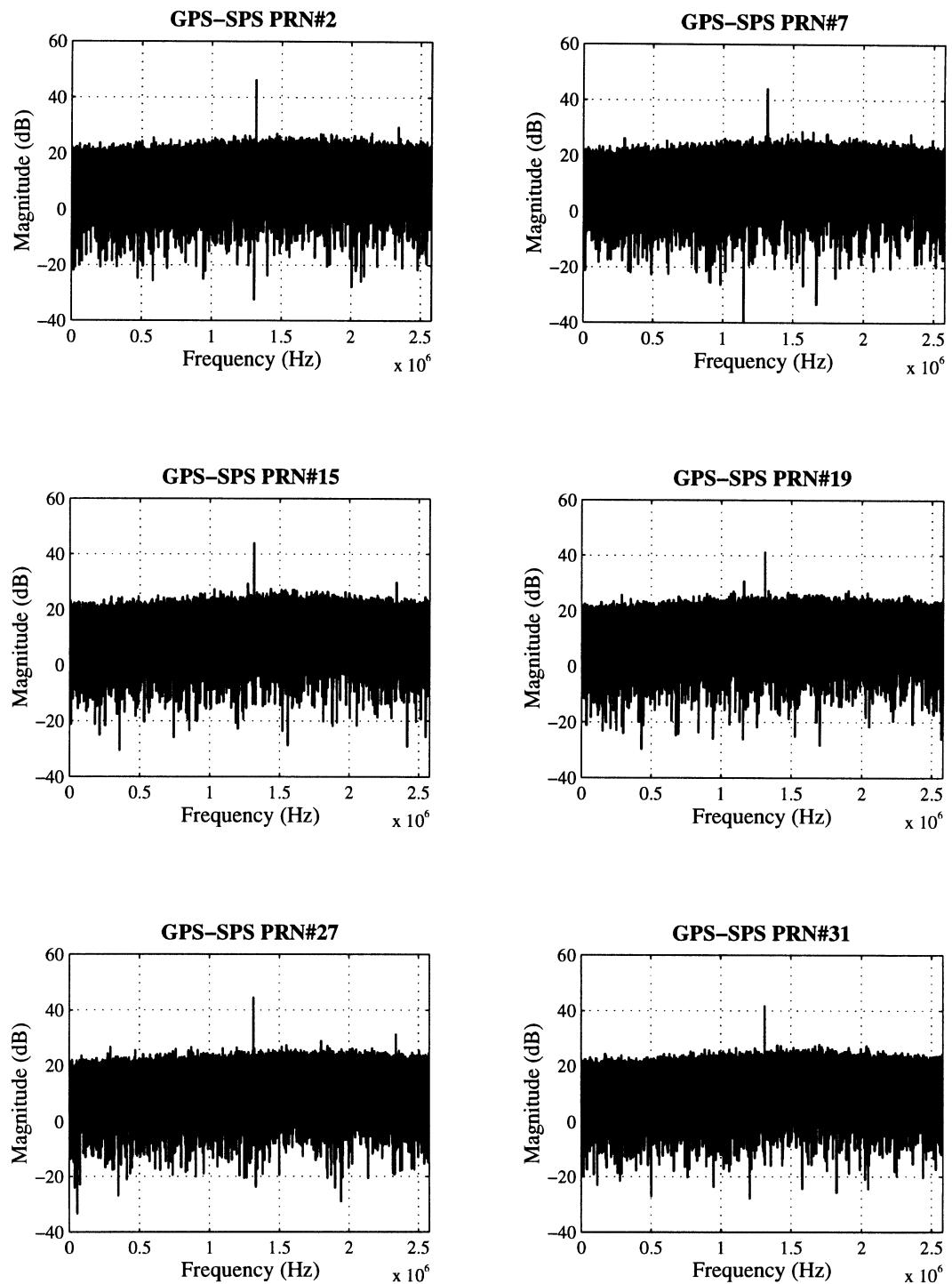


Figure 4.11 Magnitude of the Post-Correlation FFT for each of the 6 GPS-SPS satellites

GNSS transmissions and they are well suited for concurrent processing as their signal structure is similar and the additional broadcasts provide additional measurements adding integrity to the navigation/position solution.

Previous attempts at sampling multiple RF transmission directly have required extremely high sampling rates, even when bandpass sampling is employed. A receiver proposed by Brown and Wolt used direct digitization with bandpass sampling to process the complete GPS-PPS transmission [18]. Although not fully documented in the open literature, the fundamentals of the signal structure of the GPS-PPS broadcast have been disclosed. The parameters of interest here are the two RF carrier frequencies at 1227.6 MHz and 1575.42 MHz with first null bandwidths of 20 MHz each. The Brown and Wolt implementation uses an 800 MHz sampling frequency to bandpass sample the frequency span from 1200.0 MHz to 1600.0 MHz. After sampling, the two frequency bands of interest are digitally filtered and the data is decimated. This design is an archetype of a true software radio design. A wide span of frequency spectrum is directly digitized and bandpass sampled, then the specific signals of interest are digitally filtered. However, the sampling and digital filtering which must occur at the rate of 800 MHz prior to decimation makes such an implementation extremely difficult using present technology. This may be one reason as to why little has been published outside of the concept and preliminary results.

Although the design followed the software radio philosophy and eventually the computation power will be available at a reasonable cost, the required sampling rate appears excessive, particularly when the Nyquist rate for the two bandwidths of interest is on the order of 80 MHz. In this section a bandpass sampling technique will be proposed for the direct RF digitization of an arbitrary number of signals which minimizes the required sampling frequency. This technique is applicable to any RF transmission. Further, knowledge of the specific signal

structure in combination with the proposed technique provides the potential to reduce the required sampling rate even further.

Figure 4.12 presents a high level description of the proposed technique and the associated frequency domain depiction for sampling two distinct information bands [19]. A wideband antenna is used to detect a range of frequencies containing the signals of interest, the entire span is then amplified, split into two channels where each information band is bandpass filtered, the results are recombined and sampled. The sampled bandwidth contain the two signal spectrums of interests, independent of one another.

The hardware design is simple, as depicted in the figure. Conceptually, the technique can be expanded to include any number of signals of interest simply by splitting the output of the amplifier into a number of bandpass filters, recombining all the output, then sampling at an appropriate frequency [20]. The only difficulty in such an implementation is the determination of an appropriate sampling frequency. The mathematical basis for selecting an adequate sampling frequency will be discussed in the next section.

4.3.1 Theoretical Background

The procedure used to determine an adequate sampling frequency is very similar to that used for a single signal with added constraints and is outlined in the following steps:

- 1) A candidate sampling frequency, f_s , is selected.
- 2) The candidate f_s is used with the center frequency, f_c , of each of signals of interest in Equation 4.6 to calculate the resulting IF, f_{IF} .
- 3) Each f_{IF} and its associated information BW_I are checked to ensure that they meet the constraint of Equation 4.7.
- 4) The added constraint for multiple signal bandpass sampling, given in Equation 4.10, is applied to ensure that each signal spectrum in the resulting sampled bandwidth does not interfere with any other.

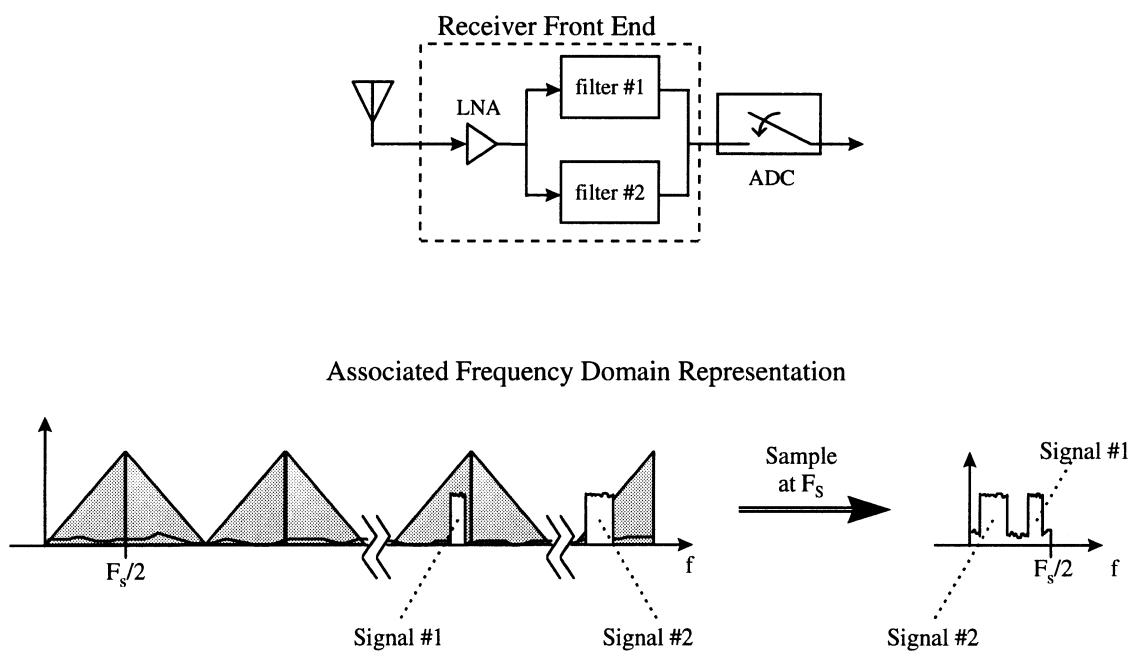


Figure 4.12 Multiple Signal Bandpass Sampling Block Diagram and Associated Frequency Representation

- 5) If the constraints in steps 3 & 4 are satisfied, the candidate sampling frequency is acceptable. Otherwise, a new candidate sampling frequency is selected (step 1) and the process is repeated.

$$\left| f_{if_b} - f_{if_a} \right| > \frac{BW_{I_b} + BW_{I_a}}{2} \quad \begin{cases} \text{for } a = 2 \dots n, b = 1 \dots (a-1) \\ n = \text{total \# of desired signals} \end{cases} \quad (4.10)$$

Equation 4.10 is the only additional constraint to transition to multiple signal bandpass sampling and eliminates the possibility of spectral overlap of signals in the resulting sampled bandwidth. The proposed technique does not violate the Nyquist rate as the minimum possible sampling frequency must be greater than twice the sum of all the information bandwidths of interest.

The difficulty in the determination of an appropriate sampling frequency becomes increasingly complex as n , the number of signals of interest, increases. An iterative approach has been programmed in software to determine sampling frequencies which meet all the necessary constraints.

4.3.2 Experimental Results

With the procedure in place to calculate acceptable sampling frequencies for this novel multiple signal technique, it is possible to examine its potential for a GPS-PPS front end implementation. Since GPS-PPS is a military exclusive broadcast, the specific PRN codes are classified. Therefore although GPS-PPS data collection is possible, the necessary processing of that data to identify the signal is more complex with the limited knowledge available. However, that does not preclude an investigation into appropriate sampling frequencies since the two carrier frequencies, 1227.6 MHz and 1575.42 MHz, and the associated 20 MHz bandwidth of each is known. Using Equations 4.6, 4.7, and 4.10 in the iterative procedure outlined above, it

was determined that the minimum acceptable sampling frequency is approximately 91.956 MHz. Using 91.956 MHz for direct sampling of the GPS-PPS signal, the 1227.6 MHz carrier is aliased to an IF of 32.172 MHz and the 1575.42 MHz carrier is aliased to an IF 12.168 MHz, both of which meet all the required constraints. Compare this sampling frequency with the others mentioned earlier. The absolute minimum sampling frequency would be the Nyquist rate of 80 MHz, the calculation of 91.956 MHz is less than a 20% increase over this absolute minimum. The other direct digitization approach, bandpass sampling an entire span of frequencies containing both signals, required a sampling frequency of 800 MHz. The 91.956 MHz calculation is a little more than 1/10 of the approach proposed in [18]. Thus this example illustrates that the multiple signal direct digitization approach described in Section 4.3.1 has potential to provide a tremendous reduction in the required sampling rate.

The calculation of an appropriate sampling frequency can be extended to accommodate three GNSS signals. These consist of the GPS-SPS broadcast at L1 ($f_c=1575.42$ MHz, BW=2.0 MHz), the GPS-PPS broadcast at L2 ($f_c=1227.6$ MHz, BW=20.0 MHz), and all the GLONASS channels after the frequency plan transition in 2005 ($f_c=1601.7$ MHz, BW=8.5 MHz). An acceptable direct digitization sampling frequency is calculated to be 80.4005 MHz which will alias the GPS-SPS L1 component to an IF of 32.59 MHz, the GPS-PPS L2 component to 21.59 MHz, and GLONASS to 6.31 MHz [20].

The approach has tremendous potential as it can be applied to any number of RF transmissions. Currently, an additional civilian signal is being considered for broadcast in the L5 frequency band [21]. Although the band has been selected, the exact frequency has yet to be determined. Since the technique proposed here appears to have significant potential for a multibroadcast receiver implementation, it would be prudent to consider a frequency that could be easily accommodated into the design.

4.3.2.1 GPS-SPS and GLONASS Implementation I

As a complete investigation into the proposed technique, consider a direct digitization front end designed to process the GPS-SPS and GLONASS broadcasts. For this implementation the interest is in a subset of the GLONASS frequencies, specifically channels #1-12. The use of these channels will exclude a few of the GLONASS broadcasts currently available, but will receive all GLONASS channels after the frequency plan modification in the near future. Since both GPS-SPS and GLONASS signal specifics are readily available, it is possible to process collected data to identify the visible satellites and verify the successful implementation of the multiple signal direct digitization technique.

The first step in the design is the calculation of the required sampling frequency. Again, GPS-SPS has a carrier frequency of 1575.42 MHz and a first null bandwidth of 2.0 MHz. Since GLONASS channels #1-12 are contiguous in the frequency domain, they can be treated as a single information band centered at 1605.656 MHz with a bandwidth of 7.5 MHz. If an entire span of frequency spectrum from 1574 MHz through 1610 MHz, containing both signals of interest, is to be bandpass sampled the minimum sampling frequency would be greater than 70 MHz.

Since the goal is a GPS-SPS/GLONASS front end implementation, Equations 4.6, 4.7, and 4.10 will be used with the bandwidth parameters of the actual filter 3 dB bandwidth to minimize the folding of noise into the opposite band. In this case, the bandwidth of GPS-SPS will be considered to be 3.2 MHz while the GLONASS bandwidth will remain at 7.5 MHz. Using these bandwidth figures the Nyquist rate for the two signal spectra will be 21.4 MHz.

Figure 4.13 provides some detailed insight into the acceptable range of sampling frequencies. This plot depicts the resulting IF position of the GPS-SPS and GLONASS bands for acceptable sampling frequencies, plotted over a range of sampling frequencies from 22 MHz to

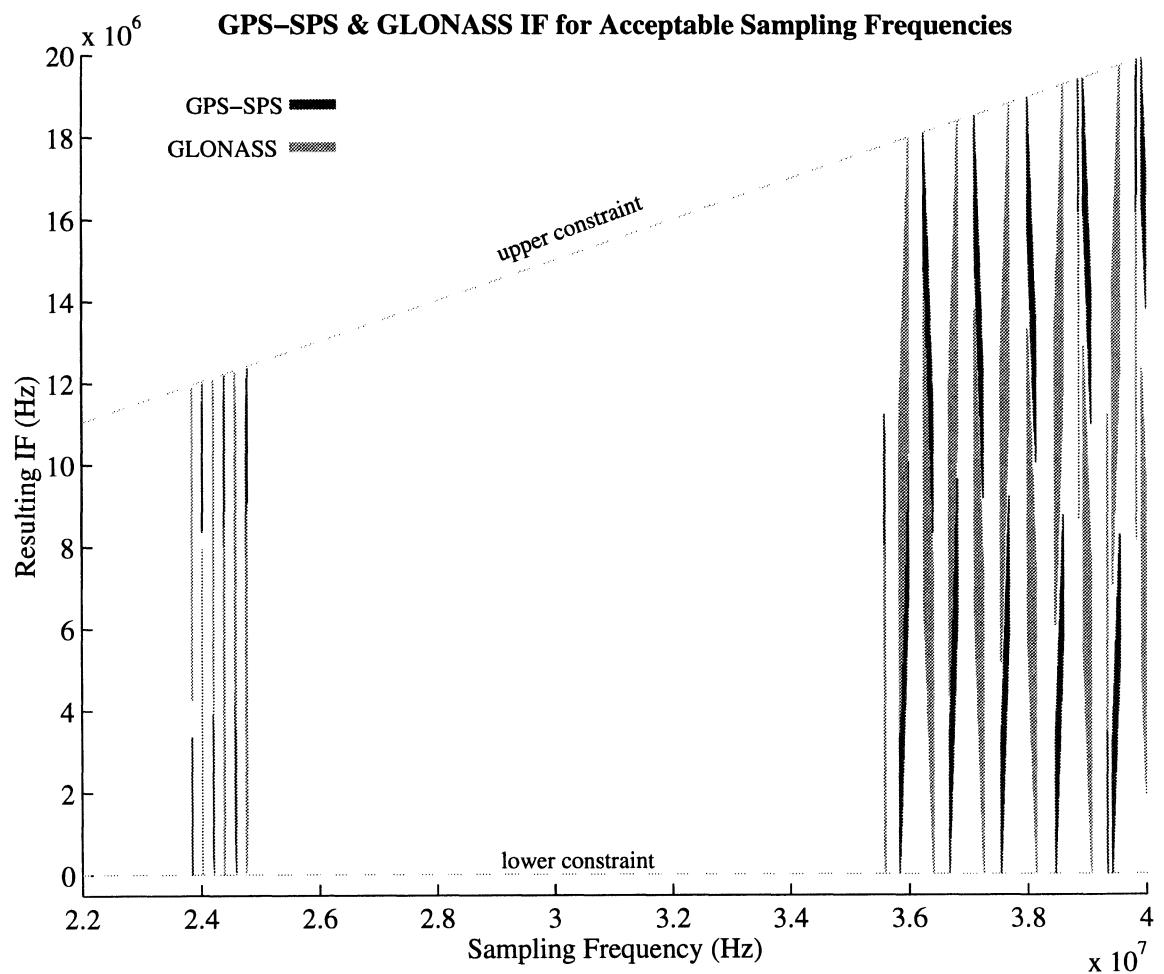


Figure 4.13 Resulting GPS-SPS and GLONASS IFs for a Range of Direct Digitization Sampling Frequencies

40 MHz. What is unusual about this figure is that there is a large span, from approximately 25 MHz to 35 MHz, where the constraints are not satisfied. Thus there are no sampling frequencies in that region that can directly digitize the combination of GPS and GLONASS. This is particularly unexpected for three reason. First, it is a relatively large span of sampling frequencies. Second, the entire span is well above the Nyquist rate for the bandwidths of interest. Third, and most surprising, is there are sampling frequencies below the span which are acceptable and meet all the required constraints. This figure illustrates the highly nonlinear nature of determining possible sampling frequencies.

From Figure 4.13, it is evident that sampling frequencies around 24 MHz can be used to sample both GPS-SPS and GLONASS and meet all the required constraints. Figure 4.14 is a zoomed view about an acceptable sampling frequency. The plot shows not only the resulting IFs for the small range of acceptable sampling frequencies, but also where the spectra falls when the various constraints are not met. The figure shows all possible cases: violation of the upper constraint, violation of the lower constraint, interference between the two spectra, and a small region where all constraints are met. The acceptable sampling frequency region includes 24.205 MHz, which will alias GPS-SPS to an IF of 2.095 MHz and the base GLONASS frequency of 1602 MHz will alias to 4.47 MHz. Thus GLONASS channel #1 will be at:

$$4.47 + (1)(0.5625) = 5.0325 \text{ MHz} \quad (4.11)$$

and channel #12 will be at:

$$4.47 + (12)(0.5625) = 11.22 \text{ MHz} \quad (4.12)$$

from Equation 2.3.

A front end was implemented in hardware based on this sampling frequency and is illustrated in Figure 4.15. The design is based on the ideal configuration of Figure 4.12, dividing the necessary gain across multiple amplifiers separated by filters to limit the spurious response

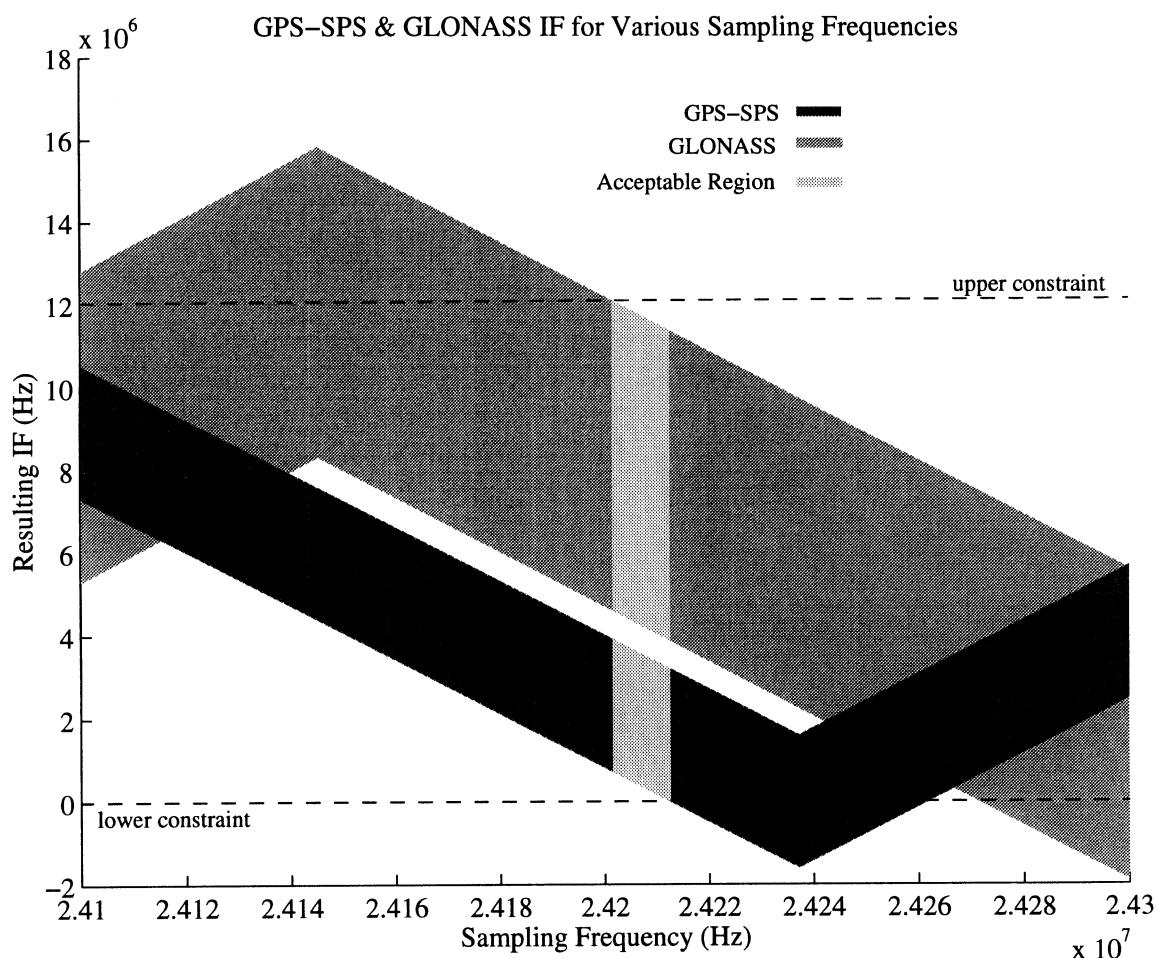


Figure 4.14 Zoomed View of Resulting GPS-SPS and GLONASS IFs

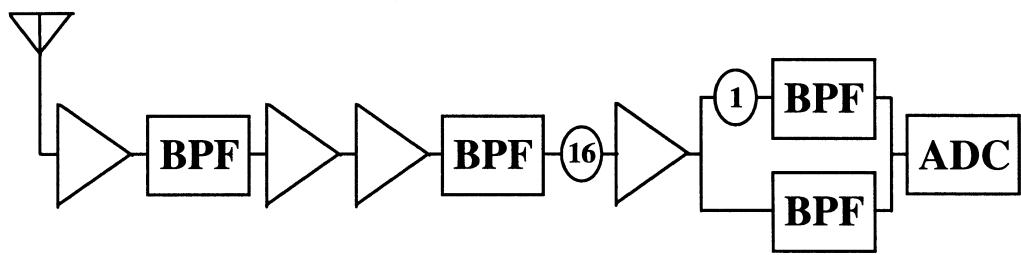


Figure 4.15 GPS-SPS/GLONASS Direct Digitization Front End Implementation

[22]. The signal enters through a wideband passive helix antenna and is immediately amplified using approximately 30 dB of gain. It is filtered using a tunable filter with center frequency of 1590 MHz, 3 dB bandwidth of 80 MHz, then further increased by cascading two amplifiers, each providing 30 dB of gain. The signal is filtered again using a second tunable filter centered at 1590 MHz with a 3 dB bandwidth of 80 MHz, passed through the first attenuator. The signal is amplified a final time and split into two paths. The first path passes through a second attenuator and then a cavity filter centered at 1605.656 MHz and 3 dB bandwidth of 7.5 MHz to extract the GLONASS signal spectrum. In the second path the GPS-SPS spectrum is filtered using a cavity filter centered at 1575.42 MHz and a 3 dB bandwidth of 3.2 MHz. The two paths are then recombined and sampled at 24.205 MHz using the TRW AMAD-7 ADC.

The two attenuators are used to adjust the gain. The first provides a 16 dB pad, again, to adjust the overall gain of the system to fully exercise all four bits of the TRW ADC without clipping. The second provides a 1 dB pad in line with the GLONASS filter as an attempt to balance the insertion losses of the GPS-SPS and GLONASS cavity filters.

A data collection effort was performed using this front end implementation and the unprocessed results are shown in Figure 4.16. The top plot shows the first millisecond of data collected in the time domain, the second plot is the magnitude of the FFT of the first 13072 points, and the bottom plot is a histogram of all the collected samples. As was the case with the raw GPS-SPS data, there are no signals apparent in any of the plots of Figure 4.16. From all initial indications it would appear the data collected is white noise. Again, it is important to recall that both GPS-SPS and GLONASS used CDMA spread spectrum modulation and pre-correlation signal power levels are below that of the noise floor. The data, which constitutes the plots of Figure 4.16, must be processed prior to any visual detection of the signal in the frequency domain. An acquisition algorithm (which will be discussed in Chapter 5) was applied

to the collected data to search for each of the 32 possible GPS-SPS PRN codes and across all 12 GLONASS frequency channels. The results are listed in Table 4.2. In this data nine satellites were identified, five GPS-SPS and four GLONASS. The post-correlation FFTs for two of the five visible GPS-SPS satellites and all four of the GLONASS satellites visible in the data set are shown in Figure 4.17. Only two of the GPS-SPS post-correlation FFTs are shown although the signal is visible in all five. The reason for their exclusion is that all are very similar in appearance, as was shown in Figure 4.11. Those two that are depicted represent the strongest, PRN#15, and the weakest, PRN#31, of the GPS-SPS signals visible. Even the weakest signal is still visible in the post-correlation FFT. Rather than repeat similar plots here, the focus is on the visible GLONASS signals. The FDMA structure of GLONASS is clearly apparent in the post-correlation FFTs. Each visible GLONASS satellite utilizes a different frequency for its broadcast.

This is a significant increase in the number of signals that could be used in a position/navigation receiver. The front end design is simple yet elegant, enabling direct digitization of both GPS-SPS and GLONASS with a sampling frequency only slightly higher than that required by the minimum Nyquist rate.

4.3.2.1 GPS-SPS and GLONASS Implementation II

The combination of GPS-SPS and GLONASS offer a unique opportunity for further reduction in the required sampling frequency. The reason for this is that both broadcasts use CDMA spread spectrum with unique PRN codes. First, the GPS-SPS PRN codes have been chosen for their good cross correlation properties enabling all SVs to use a common carrier frequency. The maximum cross correlation between any of the two GPS-SPS PRN codes is -22 dB [23]. The maximum cross correlation between any of the GPS-SPS PRN codes and the

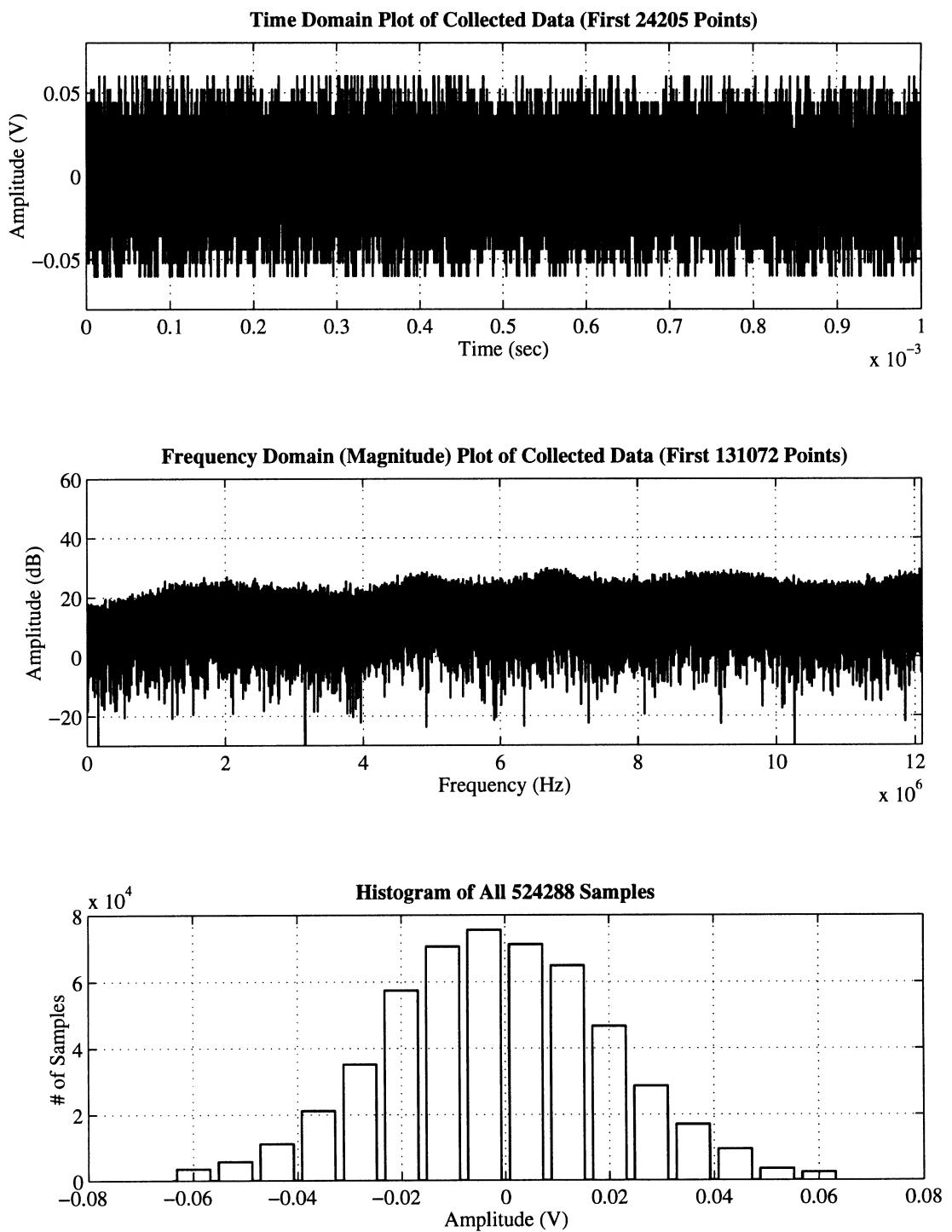


Figure 4.16 Time, Frequency, and Histogram of Collected Data

Table 4.2 Satellites Identified in Collected Data via Acquisition

Signal Type	SV PRN # (GPS-SPS) or Freq Channel (GLONASS)	Code Phase (samples)	Frequency (Hz)
GPS-SPS	2	21857	2096200
GPS-SPS	7	20359	2096550
GPS-SPS	15	11110	2093350
GPS-SPS	27	22983	2092250
GPS-SPS	31	11368	2091800
GLONASS	2	7268	5592200
GLONASS	4	8181	6719200
GLONASS	5	6948	7286500
GLONASS	9	3343	9528500

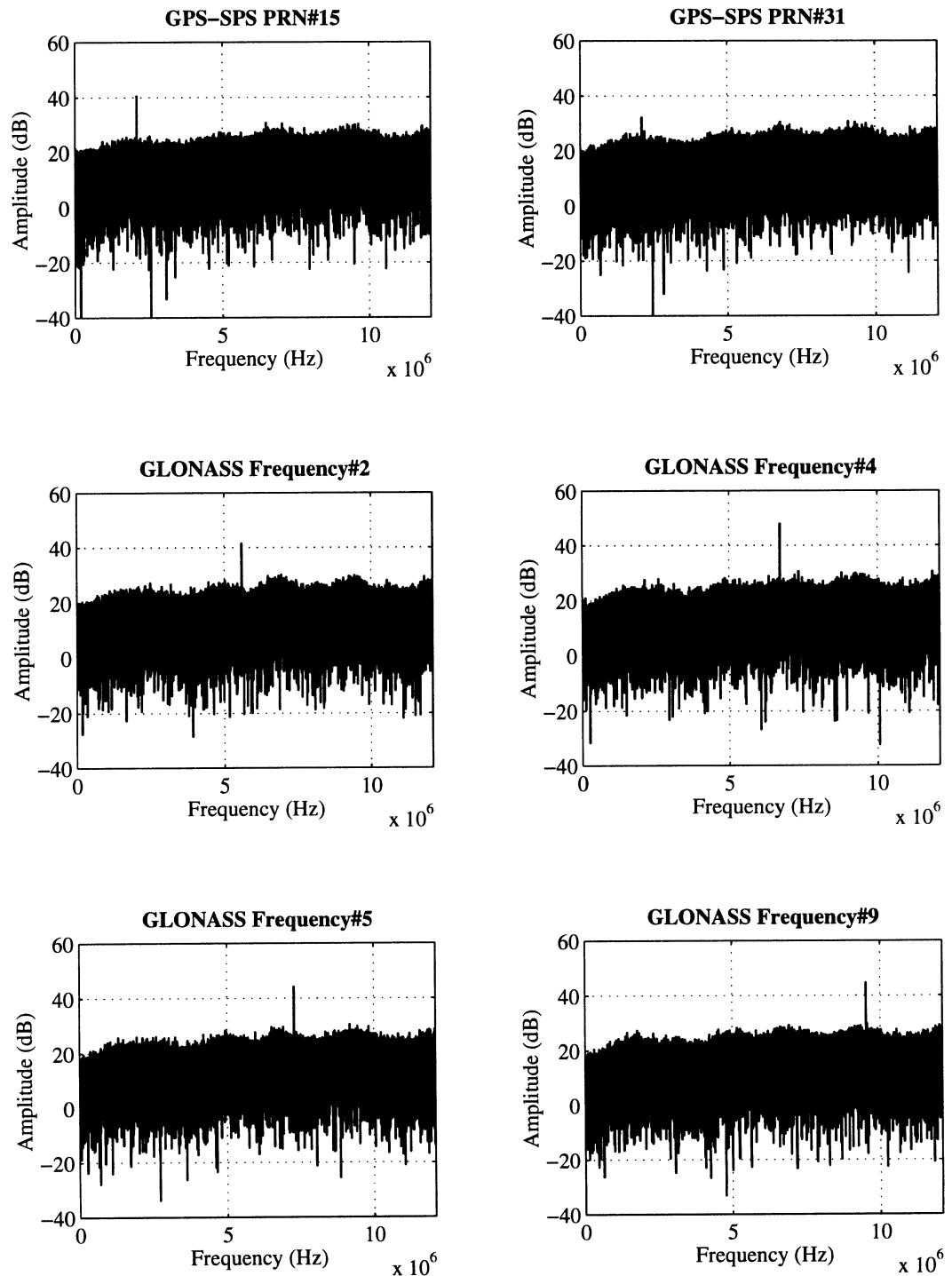


Figure 4.17 Magnitude of the Post-Correlation FFT for 2 GPS-SPS and 4 GLONASS Satellites

GLONASS PRN code is slightly higher at -16 dB, yet still a sufficient margin given the similar received power levels of both systems. Thus it should be possible to fold, or alias, both signal spectra to the same position in the resulting sampled bandwidth. This would remove the constraint of Equation 4.10 imposed for multiple signal bandpass sampling. The minimum sampling frequency would be determined by the information bandwidth of the GLONASS spectrum, the wider of the two signals.

A plot of resulting IF positions as a function of sampling frequency is depicted in Figure 4.18 center about an acceptable region. From the region, a sampling frequency of 15.402 MHz was selected. This frequency is significantly less than that used previously to capture both signal spectra. It is important to recognize that the front end implementation is identical to that used previously, shown in Figure 4.15. The only change necessary is in the sampling frequency used.

There is an additional factor that must be considered. The region in the resulting sampled bandwidth where the two signal spectra overlap will experience a 3 dB noise penalty. This is ignoring the cross correlation of the PRN codes and assuming the folded spectra appears as noise, which is a valid assumption given the worst case cross correlation figure. Further, the potential cross correlation can be reduced further in the choice of a sampling frequency. The maximum interference would occur when the spectral lines of GPS-SPS and GLONASS fall directly on top of each other. The sampling frequency can be chosen as to fold the center of the GPS-SPS band directly between any two of the GLONASS frequency channels thus minimizing the cross correlation problem. The choice of 15.402 MHz will fold the GPS-SPS center frequency to an IF of 4.416 MHz between GLONASS frequencies 7 and 8, which will alias to an IF of 4.1295 MHz and 4.692 MHz, respectively. This 3 dB penalty, or doubling of the noise, in the folded region can be adjusted. Recall that the front end design for this implementation is the same as that depicted in Figure 4.15. The second attenuation in line with the GLONASS filter

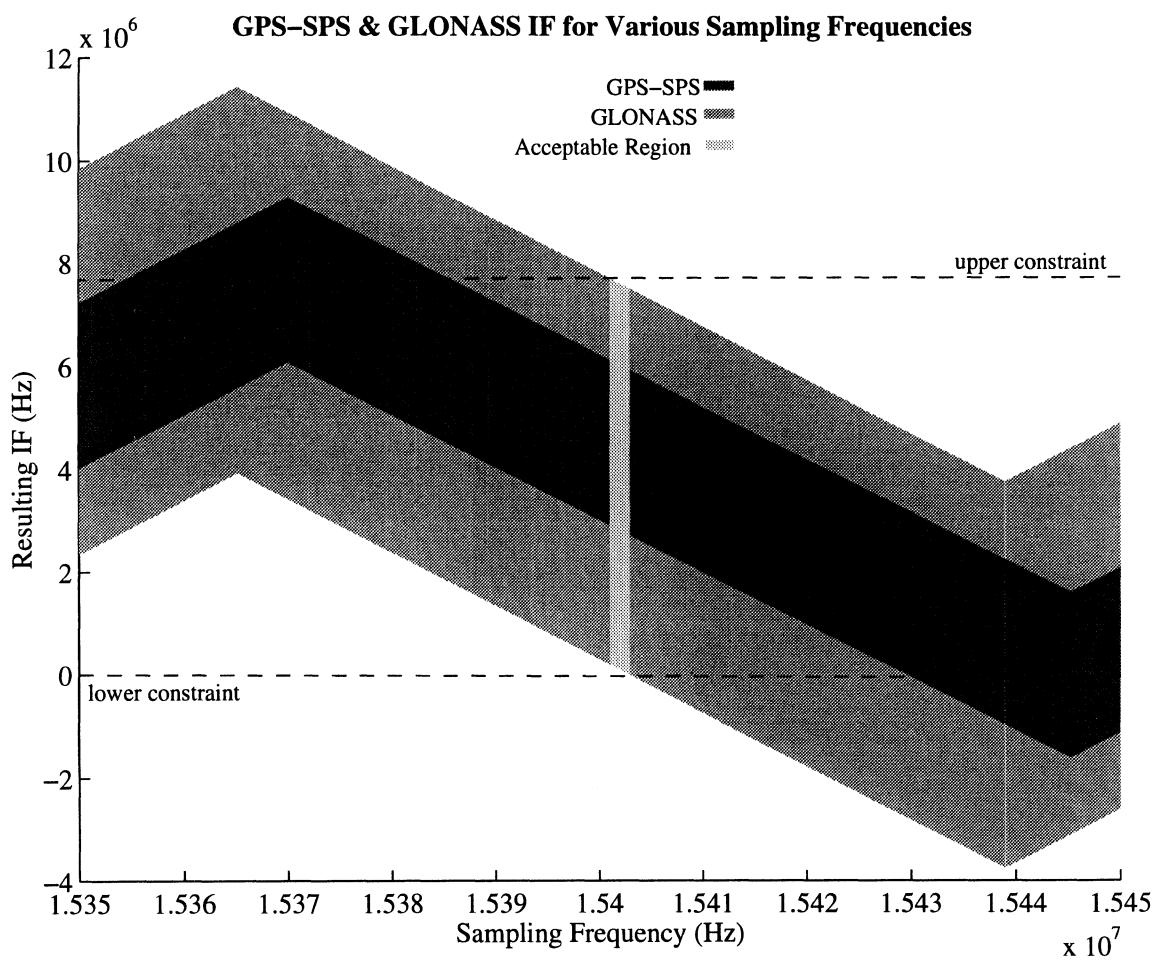


Figure 4.18 Acceptable Direct Bandpass Sampling Frequencies for Folding GPS-SPS and GLONASS to the Same Resulted Sampled Bandwidth

was used to balance the insertion loss of the two final filters prior to their recombination for digitization. Both filtering paths are passive thus implying that only losses can occur. If one path has a higher loss, the opposite path will experience less interference in processing. Since the GPS-SPS spectrum falls across only a portion of the GLONASS spectrum, if the attenuation in the GLONASS path is increased, the interference on the GPS-SPS signal should be reduced. At the same time GLONASS will experience a higher level of interference in those frequency channels which overlap the GPS-SPS spectrum, yet those outside the range will literally be unaffected. This hypothesis is considered further in the resulting implementation and data collection/processing.

As a result of the negative consequences associated with folding the two signal spectra on top of each other, it is obvious that folding the two band next to each other is the preferred choice in terms of noise performance. However, sampling frequency is an important consideration, particularly given the available computational power. The sampling frequency of 15.402 MHz is approximately 60% of the rate required to fold each band separately. The next step in evaluating the implementation is to perform a data collection, process that data, and evaluate the results.

The front end implementation of Figure 4.15 is used with a sampling frequency of 15.402 MHz. Directly digitizing the GPS-SPS carrier will alias it to an IF of 4.416 MHz and the base GLONASS frequency of 1602 MHz will alias to 0.192 MHz. The subsequent #1-12 GLONASS frequency channels will alias to 0.192 MHz plus the channel multiplied by 0.5625 MHz. The value of the second attenuator was varied with a first data set taken using a 1 dB value to balance the insertion losses of the GPS-SPS and GLONASS filters and a second data set taken with a 5 dB attenuator to increase the GPS-SPS signal power. The result of the data

collection for both 1 dB and 5 dB configuration are shown in Figures 4.19 and 4.20, respectively, showing time, frequency, and histogram plots.

As was the case with all previous data collection sets, the plots appear to depict white noise, with the exception of the frequency plot where the 5 dB attenuator was used. This can be expected as the spectrum from the GPS-SPS path is somewhat stronger than that of the GLONASS path and thus the bandwidth of its 3.2 MHz final filter appears to be slightly elevated in the resulting frequency spectrum.

Once again in order to detect if the front end design was successful, it is necessary to apply the acquisition algorithm to identify the satellite present in the data set. Since the data collection occurred within a few hours of each other as opposed to a few minutes, a comparison between which satellites were received in each is not valid as the constellation can change significantly in that time period. The satellites identified in the first data set, with the 1 dB attenuator, are listed in Table 4.3. The satellites identified in the second data set, with the 5 dB attenuator, are listed in Table 4.4. Correspondingly, the post-correlation FFTs for all satellites identified in the first and second data sets are shown in Figures 4.21 and 4.22, respectively. The figures clearly show the predicted spike at the correct frequency for each of the satellites identified via acquisition.

The results of folding the two signal spectrum to the same location in the resulting sampled bandwidth demonstrate the success of the technique. GNSS measurements are available from two different systems in a fraction of the bandwidth previously required. In this configuration, the RF transmissions have been digitized directly without the need for any frequency translation, thus minimizing the components required. Unfortunately, this approach does decrease the available SNR of the received signals. However, proper balancing of the individual filters prior to sampling should help to minimize the effect.

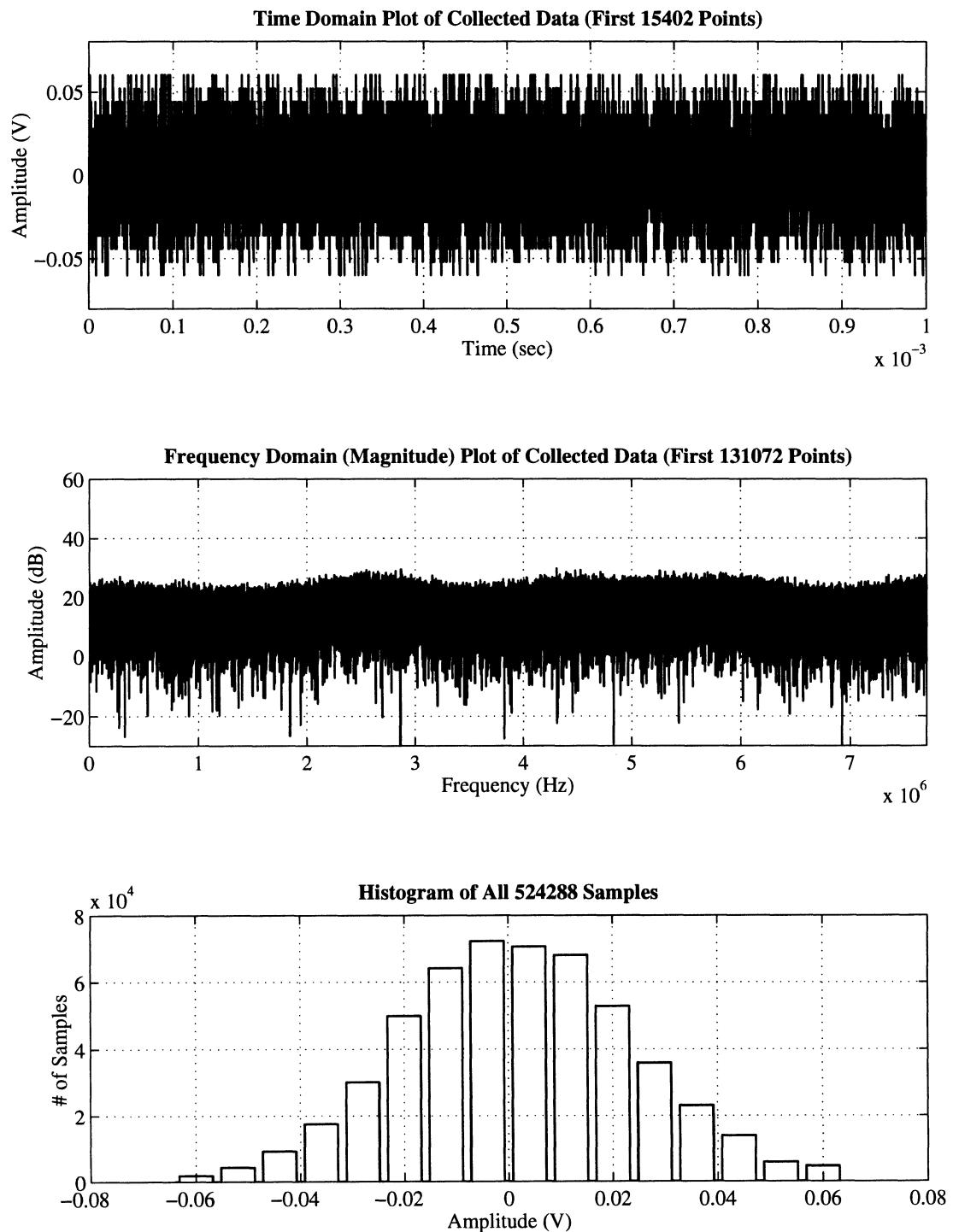


Figure 4.19 Time, Frequency, and Histogram of Collected Data with 1 dB Attenuator

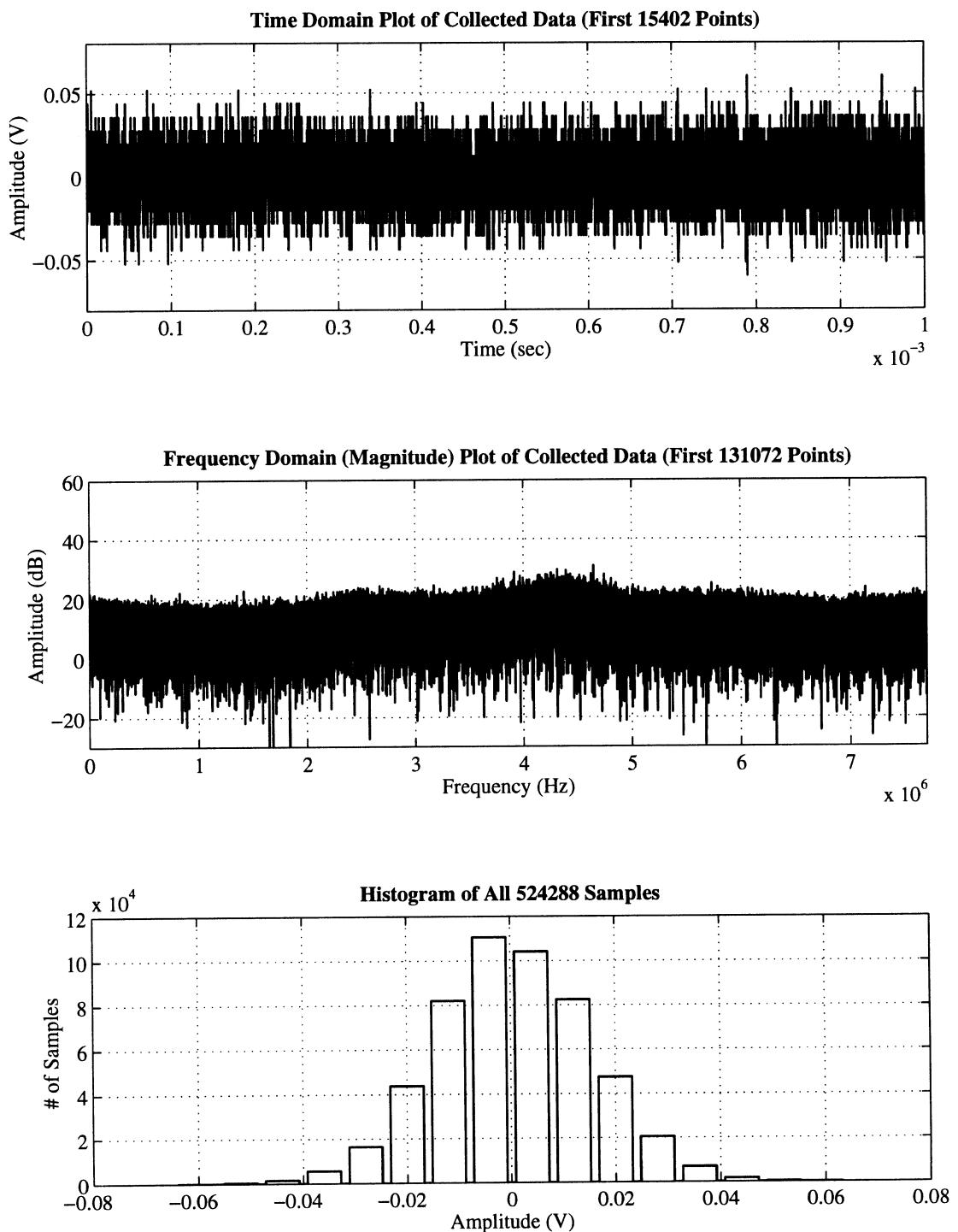


Figure 4.20 Time, Frequency, and Histogram of Collected Data with 5 dB Attenuator

Table 4.3 Satellites Identified in Collected Data via Acquisition (1 dB Attenuator)

Signal Type	SV PRN # (GPS-SPS) or Freq Channel (GLONASS)	Code Phase (samples)	Frequency (Hz)
GPS-SPS	3	3345	4415400
GPS-SPS	18	10111	4415400
GPS-SPS	19	1981	4418150
GPS-SPS	22	9361	4412950
GPS-SPS	31	2944	4418000
GLONASS	2	2137	1317100
GLONASS	4	8082	2440600
GLONASS	9	2382	5250700
GLONASS	10	5871	5818800

Table 4.4 Satellites Identified in Collected Data via Acquisition (5 dB Attenuator)

Signal Type	SV PRN # (GPS-SPS) or Freq Channel (GLONASS)	Code Phase (samples)	Frequency (Hz)
GPS-SPS	3	13197	4415550
GPS-SPS	18	4609	4415600
GPS-SPS	19	2497	4418150
GPS-SPS	22	13458	4412950
GPS-SPS	31	3293	4418150
GLONASS	4	664	2440700
GLONASS	9	5091	5250800
GLONASS	10	5640	5818900

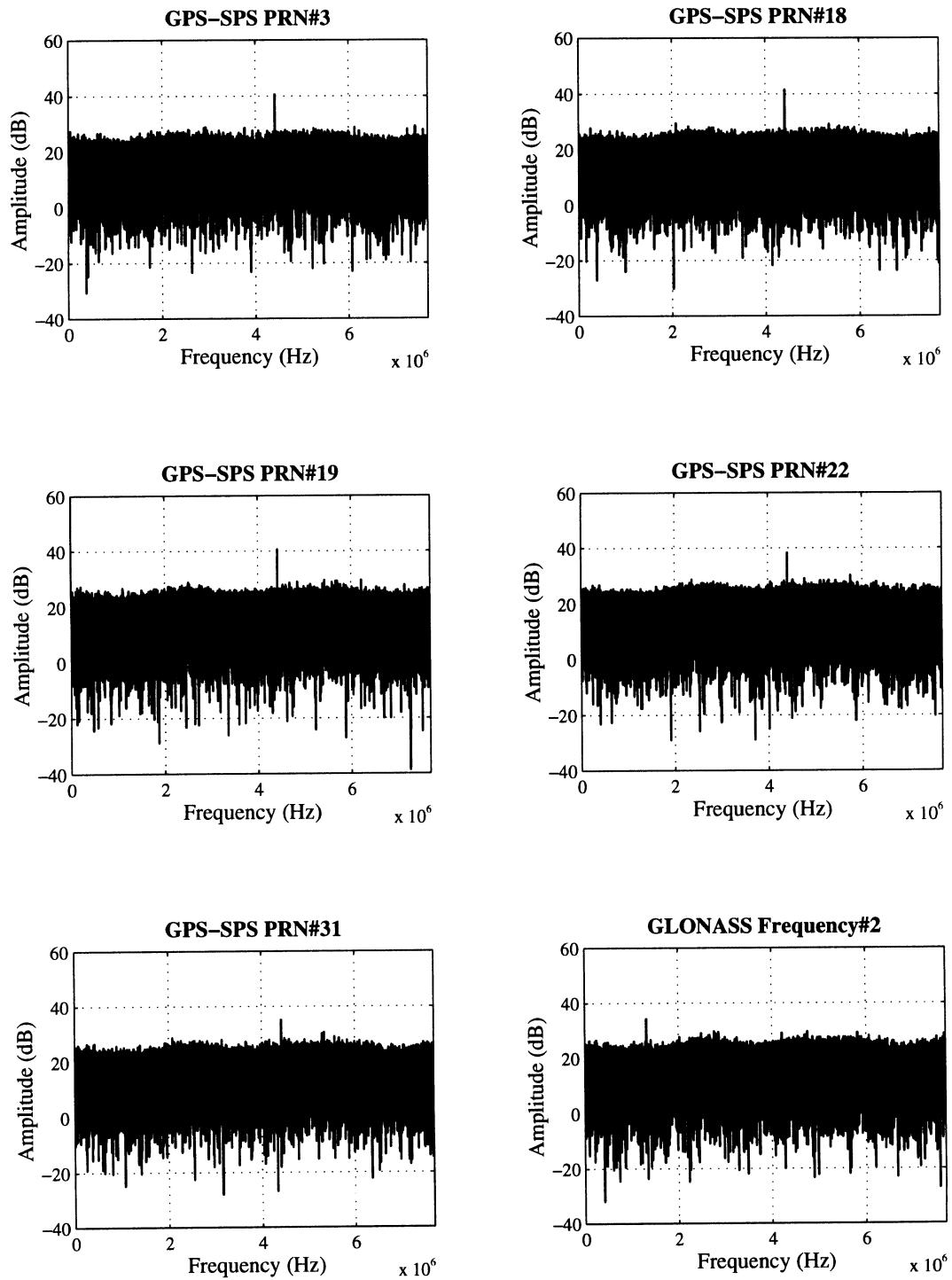


Figure 4.21a Magnitude of the Post-Correlation FFTs (1 dB Attenuator)

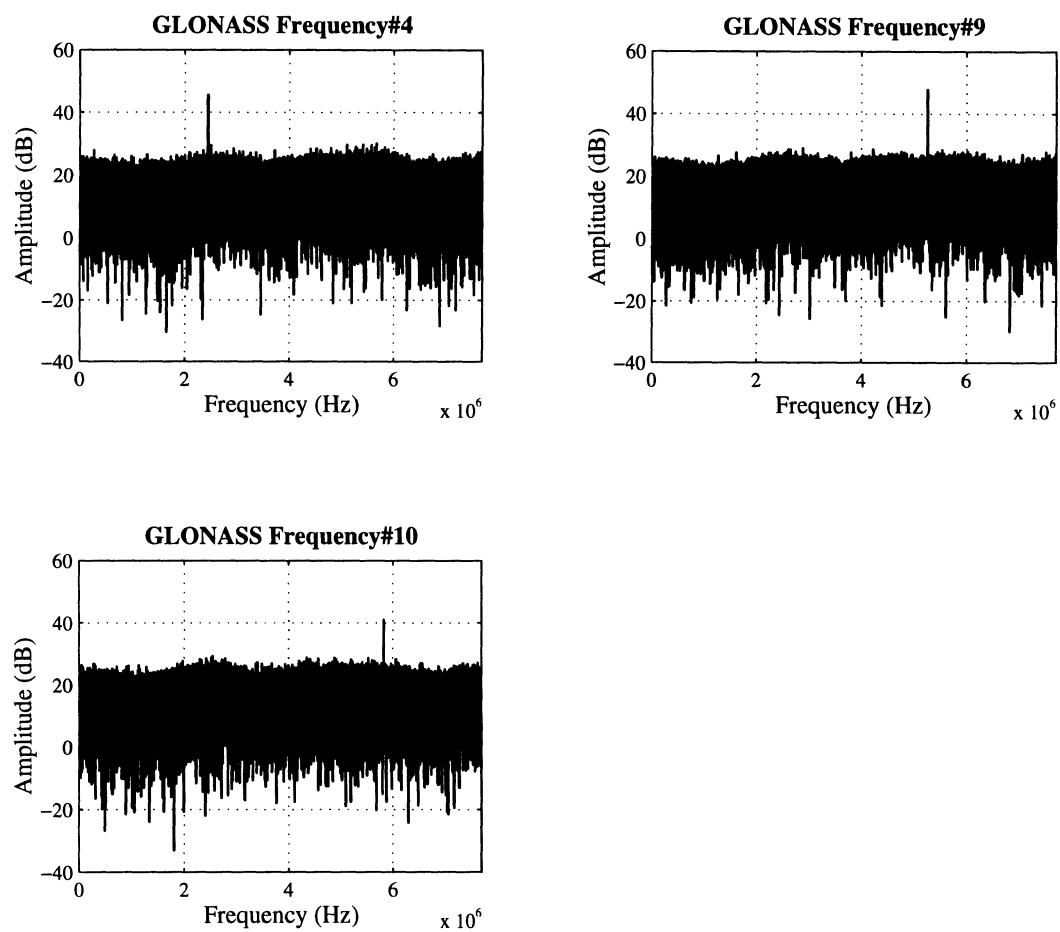


Figure 4.21b Magnitude of the Post-Correlation FFTs (1 dB Attenuator)

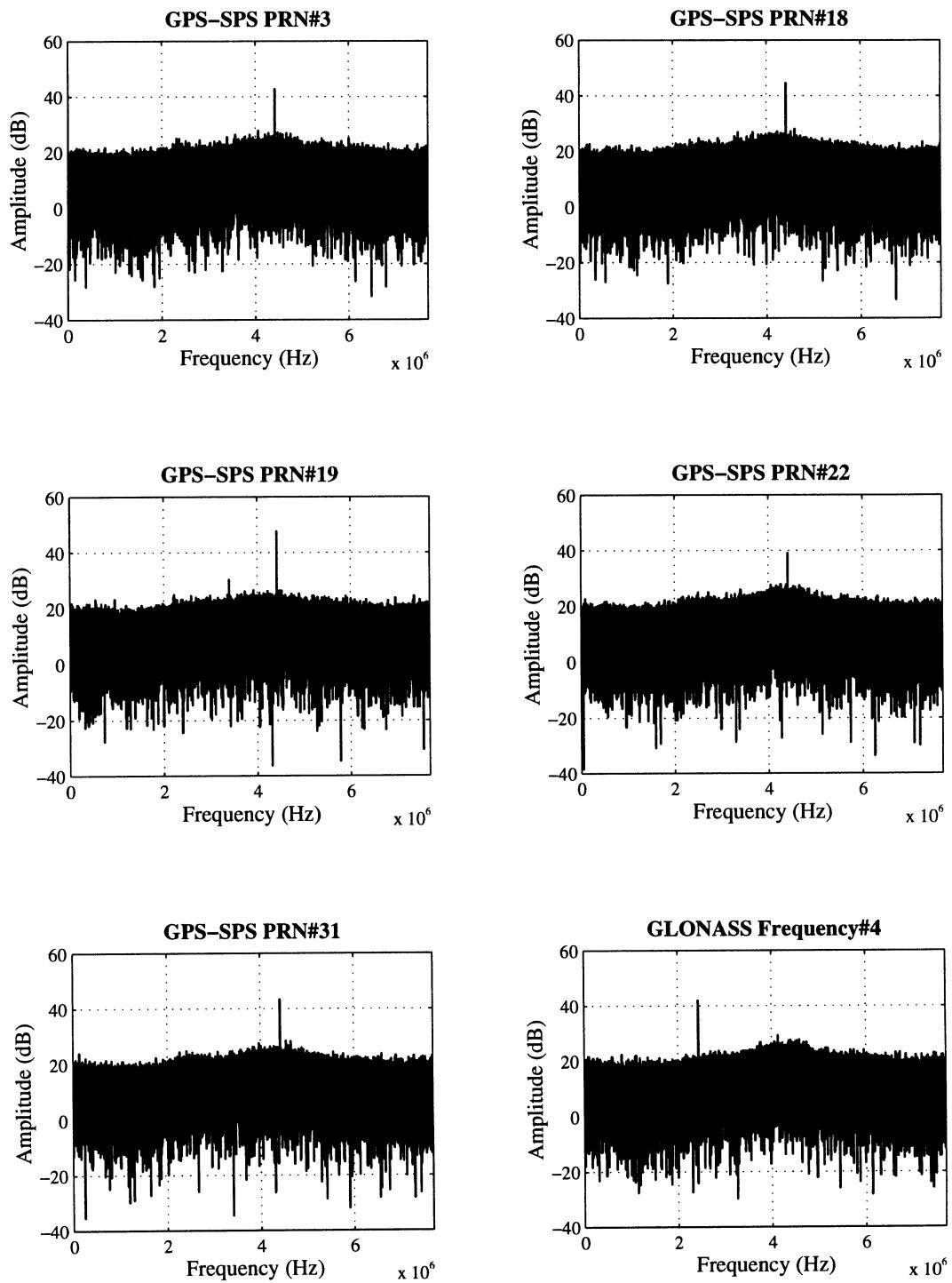


Figure 4.22a Magnitude of the Post-Correlation FFTs (5 dB Attenuator)

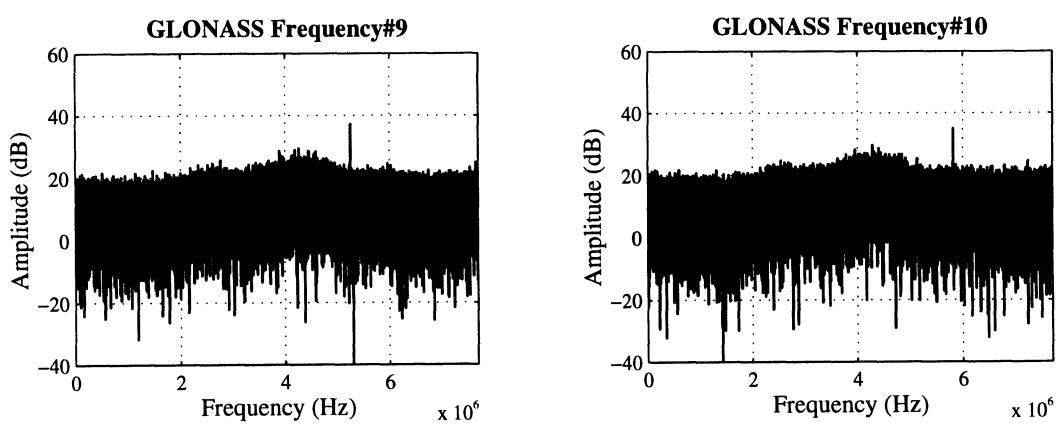


Figure 4.22b Magnitude of the Post-Correlation FFTs (5 dB Attenuator)

5. GNSS Software Signal Processing

The software signal processing can be considered the second half of a GNSS software radio implementation. The software algorithms can be divided into three main areas: acquisition, code tracking, and carrier tracking. The algorithms chosen for software implementation to accomplish these tasks are fundamentally proven algorithms, particularly those used for code and carrier tracking. Their performance is well predicted using linear models. These established processing techniques provide a framework from which advanced GNSS signal processing implementation and evaluation can occur.

The most comprehensive software simulation of GPS signal acquisition and tracking was published by Zhuang [24]. In this work both theoretical and software simulations were accomplished to determine performance of various signal processing algorithms. None of these involved the true GNSS signals, but do provide a valuable reference for the software signal processing.

This chapter will review the underlying theory in each of the algorithms implemented and provide a list of reference for further study into their operation. Examples and experimental results of the algorithms will be provided in the next chapter.

5.1 Spread Spectrum Signal Acquisition

Signal acquisition is the first step in processing the GPS-SPS signal and the GLONASS signals. Acquisition here describes a three parameter search required prior to the inception of code and carrier tracking [17].

The three parameters under investigation are: PRN code, PRN code phase, and carrier frequency. The PRN code is the spreading code used for a particular satellite. In the case of GPS-SPS the PRN code is the specific C/A code. This search space consists of any of the

possible 32 GPS-SPS PRN codes. For GLONASS, the same PRN code is used on each of the frequency channels thus eliminating this parameter from the search space and trading it for a more complex carrier frequency search. The second parameter is PRN code phase. This is the search for the specific code alignment in the received data. For example, when a GPS-SPS receiver takes its first sample of a GPS broadcast, that sample could occur at any location in the 1023 chip sequence. Determination of the PRN code phase, typically within $\pm\frac{1}{2}$ of a chip, allows a local synchronized version of the PRN code to be generated which is then correlated with the incoming signal to remove the spread spectrum modulation. The final parameter in the search is the carrier frequency, which can be considered the IF if downconversion or bandpass sampling is applied. For GPS-SPS this is equivalent to the 1575.42 MHz carrier or its resulting IF, which may initially appear to be deterministic given the front end implementation. However, there are many factors which contribute to a deviation from the expected value. The line-of-sight velocity, which results from dynamics of the satellite and receiver, create a Doppler shift in the received carrier frequency. In addition, if a LO is used to downconvert the incoming RF signal, frequency drift proportional to the type of oscillator used, will also influence the effective frequency. Typically, a search of ± 10 kHz, in 500 Hz steps, about the nominal IF is adequate. This search space assumes the line-of-sight dynamics are not extreme, such as in the case of a missile or rocket, and the LO is of fairly high quality, such as a temperature compensated or oven controlled oscillator. The frequency search for GLONASS consists of this ± 10 kHz about each of its frequency channels.

If no external information is available to the user, all parameters in the search space must be explored completely. This is known as a “cold start” in acquiring and tracking GNSS signals. However, a priori knowledge can greatly reduce the search space, minimizing the time required for acquisition. For example, both GPS-SPS and GLONASS transmit almanac data which

provides the approximate positions of all satellites in the constellation as a function of time along with each satellite's C/A code, in the case of GPS-SPS, or the satellite's channel frequency, in the case of GLONASS. Ideally, a receiver would store this information before it was powered down. Then when the receiver was to be used again, it could utilize the almanac data, its internal clock for a time reference, and its last location to identify which satellites, and the corresponding C/A code or carrier frequency, should be visible along with an estimate of the Doppler frequency. This technique is known as a "warm start" and provides an intelligent starting point from which to base the search, thus minimizing acquisition times. However, if the receiver has been moved a significant distance since last used or it has not been used for such a long time that the almanac data is no longer valid, the initial guess of visible satellites is most likely incorrect and the receiver will revert to a cold start. There are also other techniques, such as knowledge of the probability density function of the unknown parameters, which can be used to search intelligently as an attempt to minimize search times [25, 26].

The bulk of commercial GNSS receivers utilize an ASIC for code correlation, mixing in-phase and quadrature components to baseband, and accumulating the result over a given interval. The advantage here is that the same hardware can be used for acquisition as well as tracking. However, this configuration restricts the type of acquisition algorithm. In order to demonstrate the flexibility of the software radio approach, three different acquisition algorithms have been implemented.

5.1.1 Serial Search Acquisition

The serial search algorithm was one of the first techniques used for CDMA signal acquisition. As such, the associated probabilities and mean synchronization time have been defined [25]. A block diagram of the serial search algorithm is shown in Figure 5.1. In this

approach each unknown parameter is evaluated serially until all possibilities are exhausted or the correct combination of parameters has been detected.

The signal is initially multiplied by a PRN code with a specific code phase. The result is then mixed with a local oscillator to get in-phase and quadrature components at baseband. This composite result is typically integrated over a PRN code period (1 ms is the case of GPS-SPS or GLONASS) and could be integrated over an entire data bit period. A longer integration period results in a lower probability of a missed or false detection. The output of the integrator is then tested for the presence of the signal. If all parameters are correct, the result will be a maximum and should exceed a predefined threshold, in which case tracking will be attempted. Otherwise, a new combination of parameters will be tested.

The disadvantage of a serial search is that it is a time consuming process to evaluate every possible combination of PRN code, PRN code phase, and carrier frequency sequentially. In the case of a single GPS-SPS satellite this approach will typically require a search testing:

$$\underbrace{\left(\frac{1023}{2}\right)}_{\text{possible PRN Code phases}} \underbrace{\left(2 \left(\frac{10000}{500}\right)\right)}_{\text{possible frequencies}} \approx (512)(40) = 20480 \text{ combinations} \quad (5.1)$$

A variant of the serial search acquisition algorithm known as dual dwell serial search has been designed to improve search times [28]. The idea is that two integrators are used with two different time constants. The first integrator has a period approximately $1/10^{\text{th}}$ that of the single integrator. The first integrator will have higher probabilities of false acquisition, but it will also quickly eliminate many candidates which are not correct. The second integrator extends the integration period for that set of parameters which meet the criteria for the first integrator, thus reducing the probability of false lock. This approach is still considered a serial search, but can improve acquisition times through the increased complexity.

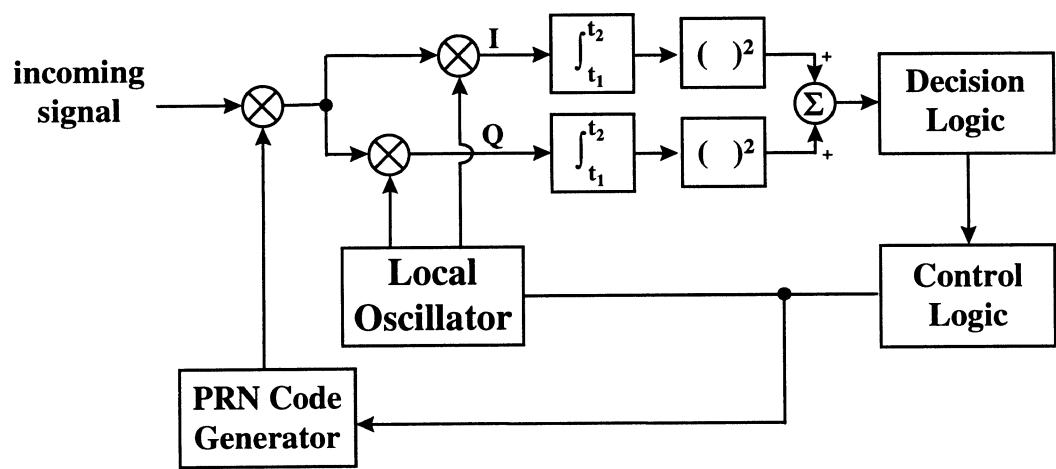


Figure 5.1 Serial Search Acquisition Block Diagram

5.1.2 Acquisition Via Parallel Frequency Space Search

The acquisition process can be improved significantly if any of the three parameters, PRN code, PRN code phase, and intermediate frequency, can be eliminated from the search or conducted in parallel as opposed to being tested sequentially. The Fourier Transform or any frequency domain transformation can be incorporated into the acquisition process, as shown in Figure 5.2, to parallelize the frequency space search. This approach has also been investigated in significant detail and the associated probabilities have been calculated [29].

The operation can be described as follows. The incoming signal is multiplied by a locally generated version of the PRN code with a specific code phase. The resulting sequence of data, equivalent to one code period, is then transformed to a frequency domain representation with the interest here being the magnitude of the result. Assuming the correct code phase has been utilized, the spreading PRN code will be removed from the carrier. Thus the signal should be readily apparent in the frequency domain, as was the case in the post-correlation plots of Figures 4.11, 4.17, 4.21, 4.22. If the code phase was not correct, the plot should resemble that of Figure 4.8, or simply white noise. In which case the control logic will change the locally generated code phase and repeat the process.

In this configuration it is only necessary to step through the possible code phases for each PRN code. Identifying the maximum frequency bin of the magnitude of the Fourier Transform provides the frequency information. This comes at the cost of a frequency domain transformation with each code phase tested. This approach achieves significant performance increases when used with GLONASS where the same PRN code is used on a number of different frequency channels.

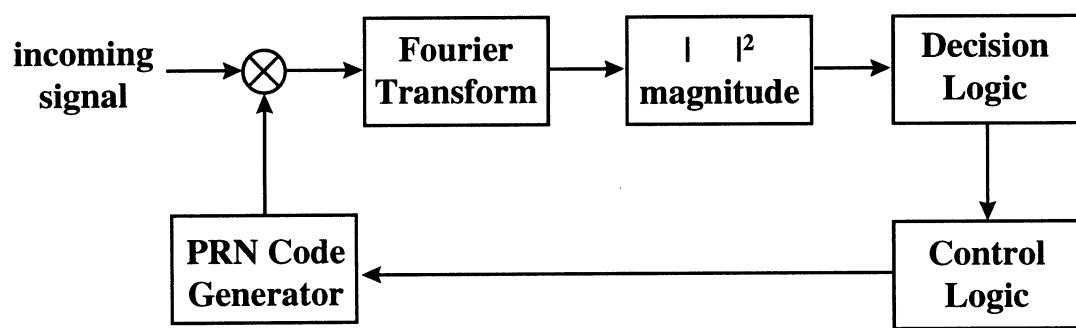


Figure 5.2 Parallel Frequency Space Search Acquisition Block Diagram

5.1.3 Acquisition Via Parallel Code Phase Search

The most recent development in GNSS signal acquisition is the use of the frequency domain circular convolution [30]. The goal of this technique is to parallelize the code phase search. In defining the acquisition search space, Equation 5.1 indicates that the code phase contributes a much larger ambiguity (factor of 10) than that of the intermediate frequency. Thus a parallelization of the code phase search will greatly reduce acquisition times, even beyond what can be accomplished using the previous method which parallelizes the frequency search. A block diagram of this technique is depicted in Figure 5.3.

The mathematical basis for the technique is illustrated in Equation 5.2. The IDFT of the

$$\underbrace{x_1[n] \circ x_2[n]}_{\text{circular convolution}} \leftrightarrow X_1[k] X_2[k] \quad (5.2)$$

product of two finite-duration n -point sequences corresponds to the circular convolution of their respective time domain sequences [31]. The incoming signal is mixed to baseband, generating the in-phase and quadrature components to be used as the real and imaginary inputs, respectively, in the calculation of the DFT. The result is multiplied by the DFT of the complex conjugate of a complex number whose real and imaginary components are equated to the complete PRN code. The magnitude of the IDFT is taken of the product and the resulting sequence is the circular convolution of the two sequences.

This result is very similar to that obtained using the serial search. The maximum value of the resulting sequence corresponds to the best estimate of the code phase of that PRN sequence in the data set for the frequency tested. If that maximum value does not exceed the predetermined threshold, either the collected data does not contain a signal utilizing that PRN

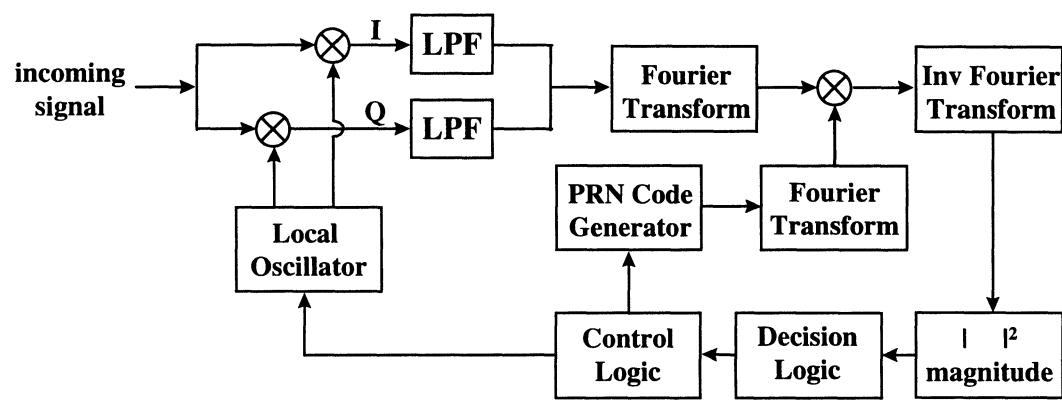


Figure 5.3 Parallel PRN Code Phase Search Acquisition Block Diagram

code and/or the frequency evaluated is incorrect. The remaining potential parameters, PRN codes and frequencies, can be cycled through until acquisition is successful.

Note that all possible code phases are evaluated in a single operation using this technique. This approach reduces the search space for a single PRN code to only the frequency uncertainty. Thus the total search space for a single PRN code defined in Equation 5.1 of more than 20000 different combinations has been reduced to only 40 candidates.

There has been little published concerning the probabilities associated with this acquisition method. However, in principle the technique is very similar to the serial search acquisition algorithm barring a data bit transition in the incoming sequence. Thus, the probabilities derived for the serial search approach should serve as an adequate approximation for this frequency domain circular convolution technique.

5.2 Spread Spectrum Signal Tracking

Once acquisition has successfully identified a signal and its associated parameters in the data, control is transferred to two coupled tracking loops. The carrier tracking loop is the first of the two tracking loops. Its purpose is to track either the phase or frequency of the incoming carrier, depending on the implementation. It is also responsible for demodulation of data encoded on the carrier, or the navigation data in the case of GNSS. The second of these two loops is designated the code tracking loop. Its function is to generate a synchronized version of the PRN code on the signal used primarily to remove the spread spectrum modulation. In conventional GNSS signal processing, this local PRN reference also provides critical timing information used in the position solution. These two loops are coupled in that the code tracking loops needs an accurate phase or frequency (depending on the code loop implementation) estimate of the carrier.

The area of feedback tracking loops has been and continues to be an active area of research. Mathematical analysis of tracking loops is extremely complex, particularly in the presence of noise. Exact solutions for noise performance have been derived only for first order tracking loops [32]. Simulation is often used as an alternative to estimate tracking loop parameters. Again, the most comprehensive analysis/simulation of GPS tracking loops was published by Zhuang where emphasis was placed on investigating various order carrier tracking loops [21]. These results could be further incorporated into the GNSS software radio implementation to maximize performance.

The tracking loops implemented for use with the GNSS software radio are structures that have been studied in depth. Their performance are well understood and predictable. Their software-based implementation allows maximum flexibility in their design. Their implementation will establish a functional GNSS software radio and provide the framework from which to incorporate advanced signal processing into GNSS receiver design.

5.2.1 Costas (Carrier Tracking) Loop

Almost any tracking loop that utilizes a feedback-type structure can be generalized to the block diagram depicted in Figure 5.4. In the first stage an input signal is compared with a locally generated reference to obtain the phase discrepancy between the two inputs. This difference is then filtered and applied as input to a voltage or numerically controlled oscillator. Based on the filtered phase error signal, the oscillator adjusts the frequency of its reference signal in an attempt to match the phase of the input.

A block diagram for the Costas loop implementation is in Figure 5.5. Although it initially appears to be somewhat more complex, it is very similar to the structure shown in Figure 5.4. The main difference is in the phase detector function. The Costas loop depicted uses an

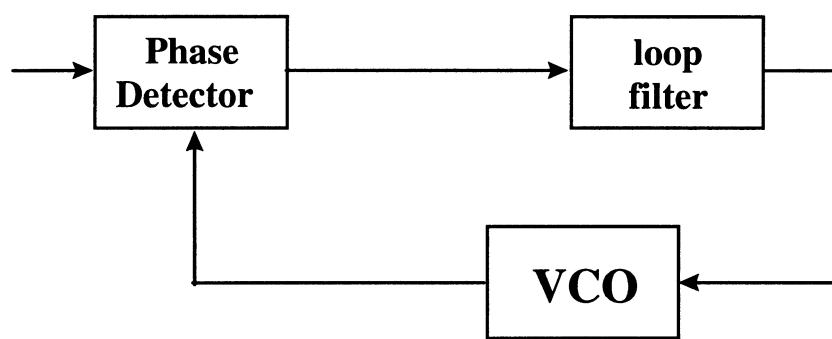


Figure 5.4 Simplified Feedback Tracking Loop

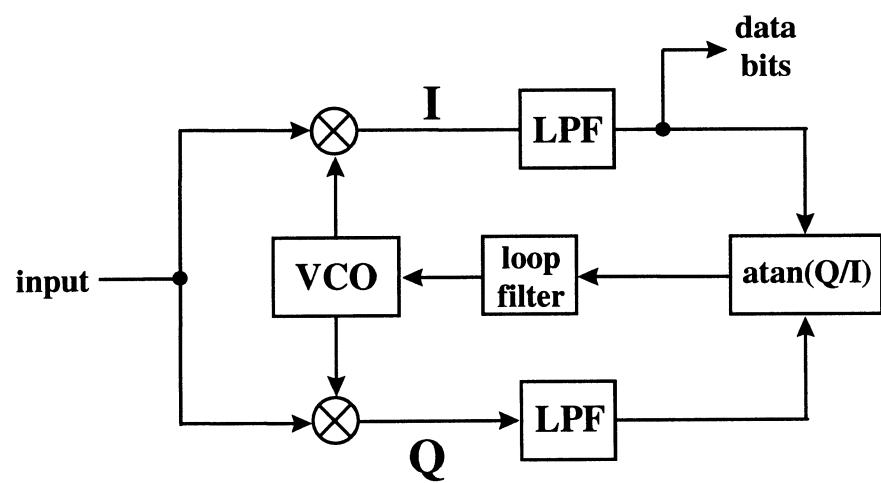


Figure 5.5 Costas Loop Block Diagram

arctangent discriminator function. This is the optimal (maximum likelihood estimator) phase detector, but also the most computationally expensive. The operation can be described as follows. The input signal is the despread carrier modulated only with navigation data. This data modulated carrier is initially mixed with the reference oscillator, generating in-phase and quadrature components which are then lowpass filtered. The two-quadrature arctangent function takes the quotient of the baseband in-phase and quadrature components to compute the error signal. This error signal is filtered and input to the oscillator to adjust the frequency of the reference signal to match that of the input. Assuming the loop is operating in lock operation, the modulated data bits are present on the in-phase arm of the Costas loop, directly after the lowpass filter.

Again, the performance analysis of the Costas loop is quite complex, particularly during the pull-in process or in the presence of noise. In order to simplify the analysis, a linear model of the feedback tracking loop shown in Figure 5.4 is often used. It has been shown that this linearization can be used to model the Costas loop implementation of Figure 5.5 [33]. This linear model is depicted in Figure 5.6 with the phase transfer function given in Equation 5.3 [32].

$$H(s) = \frac{\Theta_2(s)}{\Theta_1(s)} = \frac{K_1 K_2 F(s)}{s + K_1 K_2 F(s)} \quad (5.3)$$

The $F(s)$ term in the transfer function describes the loop filter. Incorporating the popular active proportional integrator filter and combining the K_1 and K_2 constants into a single constant K results in the transfer function of Equation 5.4

$$H(s) = \frac{\Theta_2(s)}{\Theta_1(s)} = \frac{K F(s)}{s + K F(s)} = \frac{K \frac{1 + s \tau_2}{\tau_1}}{s^2 + s K \frac{\tau_2}{\tau_1} + \frac{K}{\tau_1}} \quad (5.4)$$

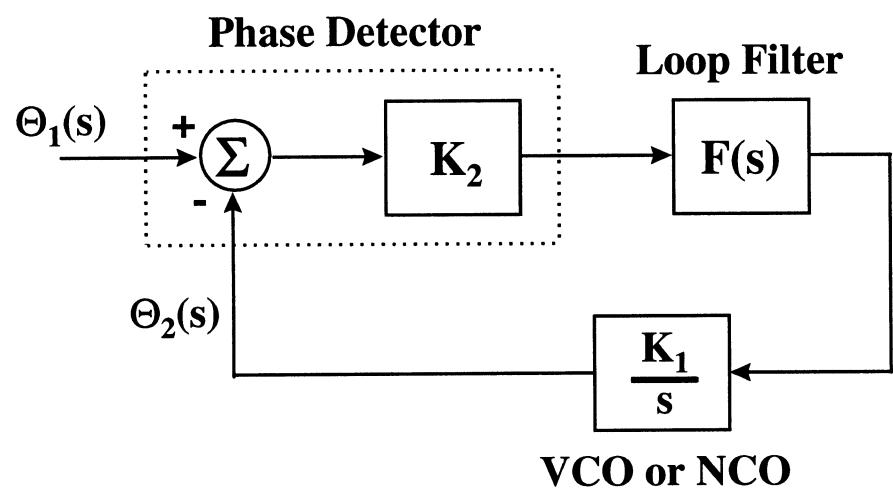


Figure 5.6 Linearized Tracking Loop Model (s domain)

In order to relate Equation 5.4 to the standard in control theory, it is necessary to express the denominator of the transfer function in normalized form, as illustrated by Equation 5.5.

$$\text{denominator of } H(s) \text{ in normalized form} = s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (5.5)$$

Doing so allows the natural frequency, ω_n , and damping factor, ζ , of the system to be defined which in turn permits the control theory for this type of feedback system to be applied. Thus for this configuration, these parameters are given in Equations 5.6 & 5.7.

$$\omega_n = \sqrt{\frac{K}{\tau_1}} \quad (5.6)$$

$$\zeta = \frac{\omega_n \tau_2}{2} \quad (5.7)$$

The final phase transfer function, expressed in control theory notation is in Equation 5.8.

$$H(s) = \frac{\Theta_2(s)}{\Theta_1(s)} = \frac{2s\zeta\omega_n + \omega_n^2}{s^2 + 2s\zeta\omega_n + \omega_n^2} \quad (5.8)$$

From this, the filter parameters, τ_1 and τ_2 , can be picked taking into account the gain, k , to achieve the desired response. These parameters define a number of key properties associated with phase lock loops such as: lock-in time, pull-in range, and pull-out range from which the performance of the implementation can be accurately predicted [32].

However, the derivations described thus far relate to an analog implementation of the linear model of the phase lock loop. The ultimate goal here is a software or discrete implementation, dealing with samples as opposed to a continuous signal. There are three different approaches that have been used to transition from the analog-based theory to a discrete time implementation. The first, and most basic, is based on the derivative of the filter expression and open loop transfer function. This is used by the GEC Plessey GPS-SPS receiver to derive

the tracking loop filter coefficients that, in turn, provide the parameters to describe the operation of the entire loop [34]. The second approach is based on the bilinear transform of $F(s)$, the loop filter. In this case the filter coefficients are specified based on the desired s-domain linear phase lock loop analysis, then the bilinear transform is applied to these coefficients to obtain the z-transform discrete equivalent, as is shown in Equations 5.9 -5.11 where ΔT is the sampling interval [32].

$$F(s) = \frac{\tau_2 s + 1}{\tau_1 s} \xrightarrow{\text{bilinear transform}} F(z) = \frac{(C_1 + C_2) - C_1 z^{-1}}{1 - z^{-1}} \quad (5.9)$$

$$C_1 = \frac{\tau_2}{\tau_1} - \frac{\Delta T}{2 \tau_1} \quad (5.10)$$

$$C_2 = \frac{\Delta T}{\tau_1} \quad (5.11)$$

The third and most accurate approach is based on applying the bilinear transform to each component in the linearized phase lock loop model, then equating coefficients in the resulting transfer function to determine the filter parameters C_1 and C_2 of Equations 5.10 & 5.11. This has been done by Chung with the results presented in Equations 5.12 & 5.13 [35].

$$C_1 = \frac{1}{K} \frac{8 \zeta \omega_n \Delta T}{4 + 4 \zeta \omega_n \Delta T + (\omega_n \Delta T)^2} \quad (5.12)$$

$$C_2 = \frac{1}{K} \frac{4 (\omega_n \Delta T)^2}{4 + 4 \zeta \omega_n \Delta T + (\omega_n \Delta T)^2} \quad (5.13)$$

This provides the most accurate discrete implementation of the linear phase lock loop model and is thus the approach used in the implementation in the Costas carrier tracking loop for GNSS software radio implementation.

5.2.2 Early/Late Noncoherent Code Tracking

Acquisition provides an initial accurate estimate of the code phase but line-of-sight dynamics will perturb the code rate and thus tracking the incoming code is required. This is true of just about any spread spectrum receiver as the locally generated version of the PRN code is used to despread the signal for further processing. In the case of a GNSS receiver, the importance of the code tracking loop is elevated as it is used to despread the incoming signal as well as provide the time-of-transmission measurements critical for range measurements and subsequently a position solution.

The code tracking loop used in the GNSS software radio is the popular early/late noncoherent delay lock loop [36]. The block diagram of this code tracking loop is depicted in Figure 5.7. The input to the loop is the composite signal consisting of the carrier modulated with the navigation data and the spreading code. The input signal is split into two paths and correlated with two versions, an early and late, of the locally generated PRN code. The two version are equally spaced, typically ± 0.5 chip, about the synchronized or prompt PRN code. Each of these two paths are mixed to baseband, generating in-phase and quadrature components. The energy in the early and late paths is differenced and the result is filtered and input to the voltage or numerically controlled oscillator which clocks the PRN code generator. In this case the bias in the error signal indicates which path, early or late, contains more energy and thus whether the VCO needs to speed-up or slow down the locally generated PRN code. Ideally these two paths are balanced and the resulting difference is zero.

Although this implementation appears significantly more complex than the generic feedback structure of Figure 5.4, it is actually very similar. The majority of components in the block diagram of the early/late noncoherent delay lock loop represent the phase detector. The difference of the early and late paths represents the error signal which goes through the

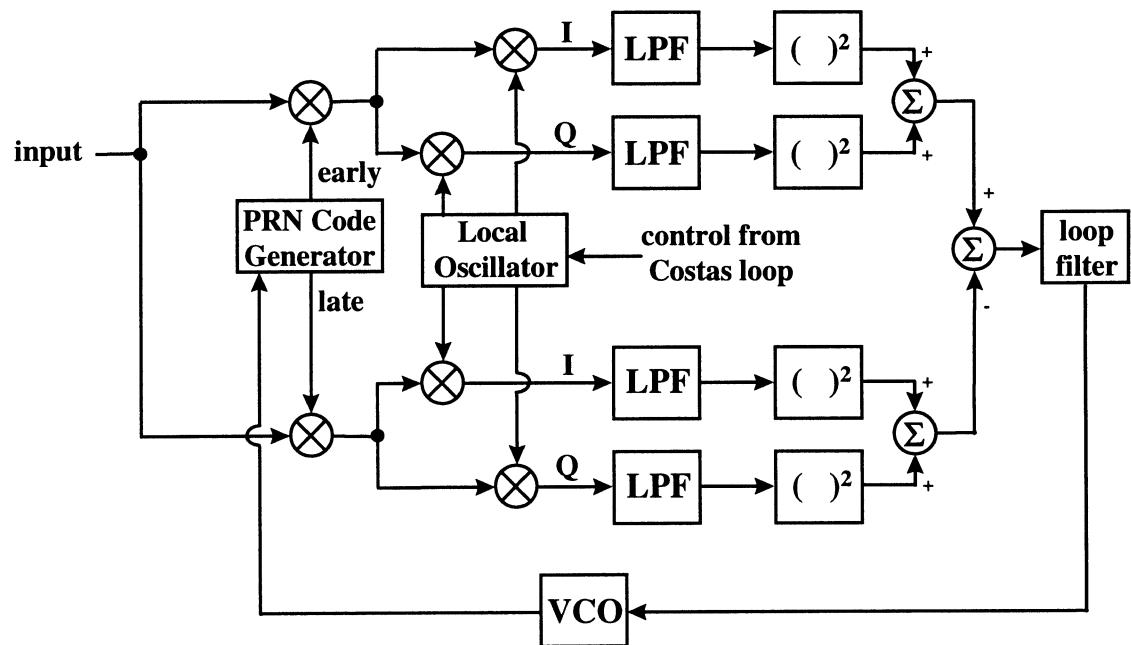


Figure 5.7 Early/Late Noncoherent Delay Lock Code Tracking Loop Block Diagram

traditional loop filter and is used as an input to the voltage/numerically controlled oscillator which clocks the PRN code. Thus the block diagram of Figure 5.7 can be understood to be a lower level depiction of the structure in Figure 5.4.

This comparison can also be done mathematically. The linear phase lock loop model of Figure 5.6 has been shown to be an adequate theoretical approximation of the early/late noncoherent delay lock loop [37]. This result is of utmost importance as it allows equations presented in Section 5.2.1 for the linearized feedback loop to be directly applicable to the design and software implementation of the code tracking loop. This also implies that the performance of the code tracking loop will be well modeled by the linear feedback loop. Thus Equation 5.12 & 5.13 can be used to calculate the appropriate coefficients for the loop filter once the parameters for loop operation have been specified.

6. GPS-SPS Software Radio Implementation

Now that all GNSS and software radio concepts have been presented, novel bandpass sampling front ends have been proposed and successfully demonstrated, and the required software-based signal processing algorithms have been discussed, the next step is the complete design and implementation of a GNSS software radio.

In order to limit the scope of the project, the GPS-SPS broadcast will be the primary signal of interest. Thus the hardware and software implementation will be based on the GPS-SPS broadcast. However, in order to demonstrate the flexibility of the software radio, some GLONASS processing will also be presented. Further, the results presented for GPS-SPS processing will correspond to different receiver architectures obtained simply via a software algorithm change.

The ultimate goal is to collect a window of GPS-SPS data using a front end design consistent with the goals of a software radio and to process it exclusively in software to obtain an accurate position solution. The software algorithms will be coded in the Matlab programming language to provide the highest level of flexibility. All data will be postprocessed and conclusions will be made regarding the feasibility of a real time implementation.

6.1 Front End Design

Ideally the front end design would resemble that of Figure 4.8 where bandpass sampling is applied directly to the GPS-SPS broadcast. The feasibility of this technique has been demonstrated in chapter four and it adheres to the software radio design philosophy of minimizing the number of components in the front end. Recall that the traditional receiver typically consists of multiple stages of amplification and frequency downconversion. The design used for the GPS-SPS software radio is shown in Figure 6.1 and is a compromise between the

ideal and traditional implementation. This design uses a single downconversion stage with bandpass sampling. This single stage of frequency translation adds a minimal number of components and significantly reduces the analog input bandwidth requirement on the ADC. Although data was previously collected for this research using the design of Figure 4.8, the hardware configuration required for the sophisticated ADC did not allow for the collection of the extended 30 second window of GPS-SPS data necessary for a position solution.

In the design of Figure 6.1, the antenna is located at a surveyed point to be used as a truth reference in determining a position solution. The antenna is active and amplifies the signal upon reception. It is connected to a filter and further amplified prior to being mixed with a 1554.174 MHz LO. This LO centers the desired frequency component at 21.246 MHz, slightly off the 21.4 MHz center frequency of the resulting bandpass filter. Yet the bandpass filters will easily accommodate the first null bandwidth of the translated GPS-SPS signal. The downconverted frequency is further amplified and passed through a narrow bandpass filter directly prior to sampling with an Analog Devices 9102 ADC. Sampling of the 21.246 MHz IF occurs at 5.0 MHz and Equation 4.6 can be used to calculate the resulting aliased 1.246 MHz sampled IF which is in the center of the sampled information bandwidth.

The overall gain, Equation 4.1, was specified to trigger 8 bits of the input range of the ADC so that the resulting samples would be 1-byte in size. The noise figure of the system is determined predominantly by the noise figure of the amplifier in the active antenna as is shown in Equation 4.2. This is particularly true in this design as the first amplifier provides a significant amount of initial gain. The dynamic range requirement for the CDMA GPS-SPS transmission is low so the filter is first in the chain of front end components. The front end design was used to collect 30 seconds, or 150 MB, of raw GPS-SPS data. The unprocessed data appears as noise as the CDMA format of the GPS-SPS signal enable the signal power below that

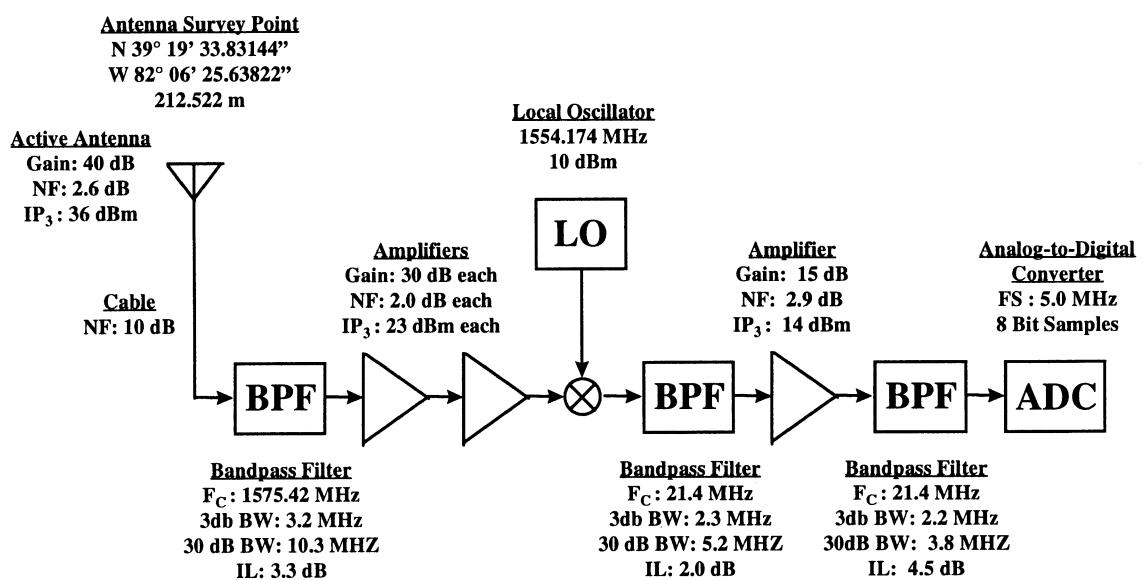


Figure 6.1 GPS-SPS Software Radio Front End Design

of the noise floor. Time, frequency, and histogram plots of the raw data are presented in Figure 6.2. Since the data was collected over an extended period of time, two sets of plots are shown. The first set shows the results from the first data samples collected, while the second set of plots are for data collected 10 seconds into the run. The results in each set appear to be consistent, thus the data can be assumed to be valid although no definite judgment can be made until acquisition processing is complete.

It is important to note that as the hardware becomes more accessible, the single stage of frequency translation could be removed and the ADC substituted for one which would accommodate the analog input bandwidth of the GPS-SPS signal at RF (the TRW AMAD-7, for example). There will be no change to the software algorithms, except for a possible change in sampling frequency.

6.2 Signal Acquisition

The first stage in processing the GPS-SPS signal is acquisition, or identifying the code phase and carrier frequency for each PRN code contained in the data set. The flexibility of the software radio allows for the use of any of the algorithms described in Chapter 5. Thus the most appropriate technique is that which would provide the most rapid acquisition. Experimental results show that the parallel code phase search using the circular convolution, Section 5.1.3, provides the shortest acquisition times.

Once the acquisition parameters are identified, a postcorrelation FFT can be used to identify the resulting carrier. This was demonstrated in Chapter 4 and used to verify the front end design. One advantage in postprocessing the data is that an exhaustive search is possible. This allows testing of all possible acquisition parameters to determine which combination is correct for the satellites contained within the collected data.

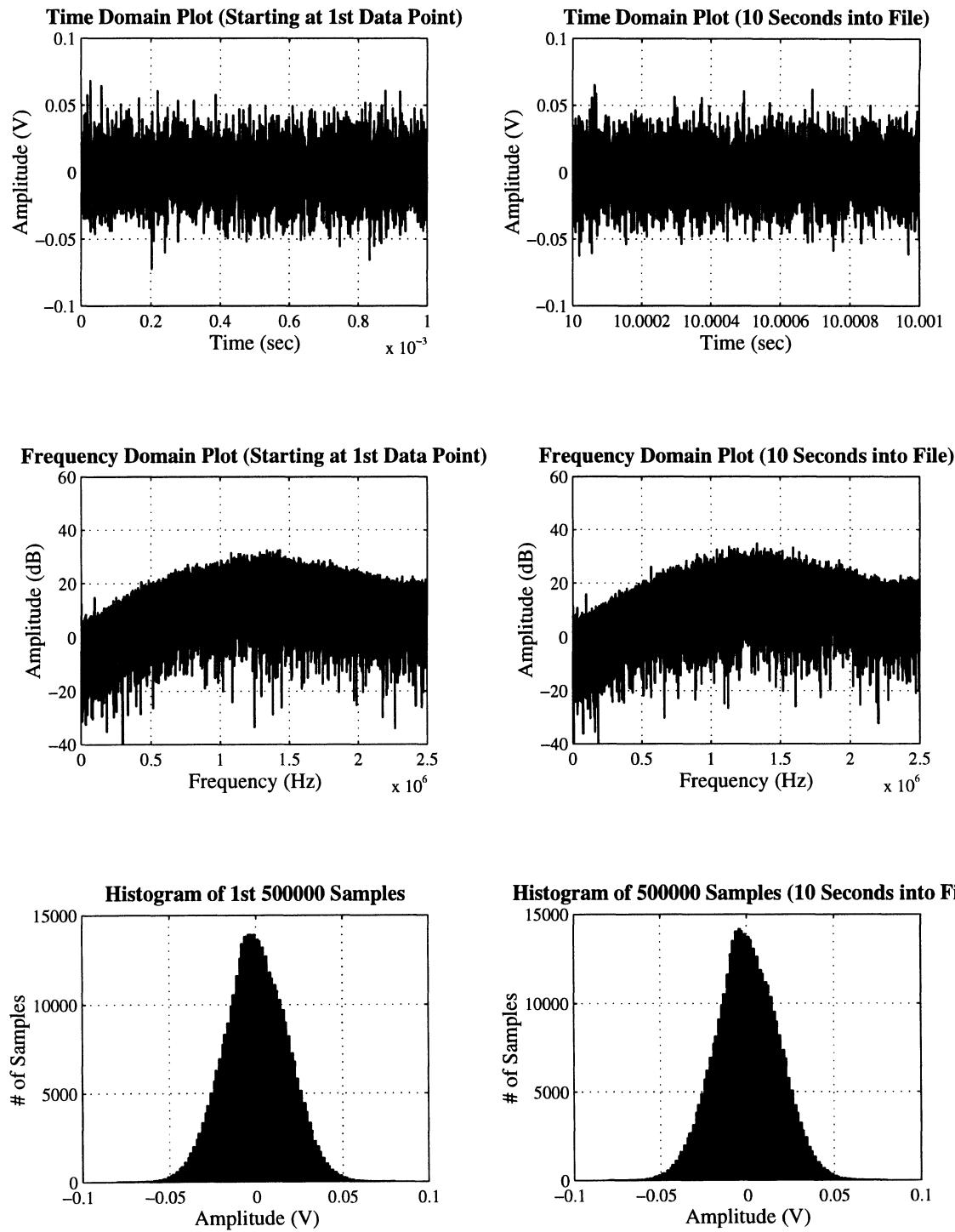


Figure 6.2 Time, Frequency, and Histogram Plots of Collected Data

The results for this exhaustive search for two GPS-SPS PRN codes are presented in Figure 6.3. For each PRN code all possible code phases are evaluated and at each code phase 17 different frequencies are evaluated from 1.238 MHz to 1.254 MHz, in 500 Hz steps. A maximum occurs when the correct combination of parameters are used for that PRN code. The top plot depicts the results for PRN #12, a GPS-SPS PRN code not currently in use by any of the satellites. There is no distinct maximum at any combination of parameters as is expected as the signal is not present in the collected data. The bottom plot shows the results for PRN #17. It is clear that there is a definite maximum which corresponds to the correct acquisition parameters for this satellite. Specifically, this maximum occurs at a code phase of 4079 samples and a frequency of 1.249 MHz. When these parameters are used in conjunction with a post-correlation FFT for PRN #17, the resulting carrier is clearly evident as is shown in Figure 6.4.

The acquisition algorithm is applied to the collected data for all possible 32 GPS-SPS PRN codes. Only the first 10 milliseconds of data are evaluated. Six different GPS-SPS are identified in the collected data with their acquisition parameters presented in Table 6.1. The results presented in Table 6.1 were confirmed using both of the other acquisition algorithms, serial search and parallel frequency space search, described in Chapter 5. In each case, the code phase was within one sample and the carrier frequency was within 250 Hz of the above result. Thus all software-based acquisition algorithms have been successfully validated. The acquisition parameters allow the next step in GNSS signal processing, signal tracking and data demodulation, to occur.

6.3 Signal Tracking

Signal tracking consists of the coupled code and carrier tracking loops, as was described in Chapter 5. Based on the analysis presented in Chapter 5, each tracking loop has two

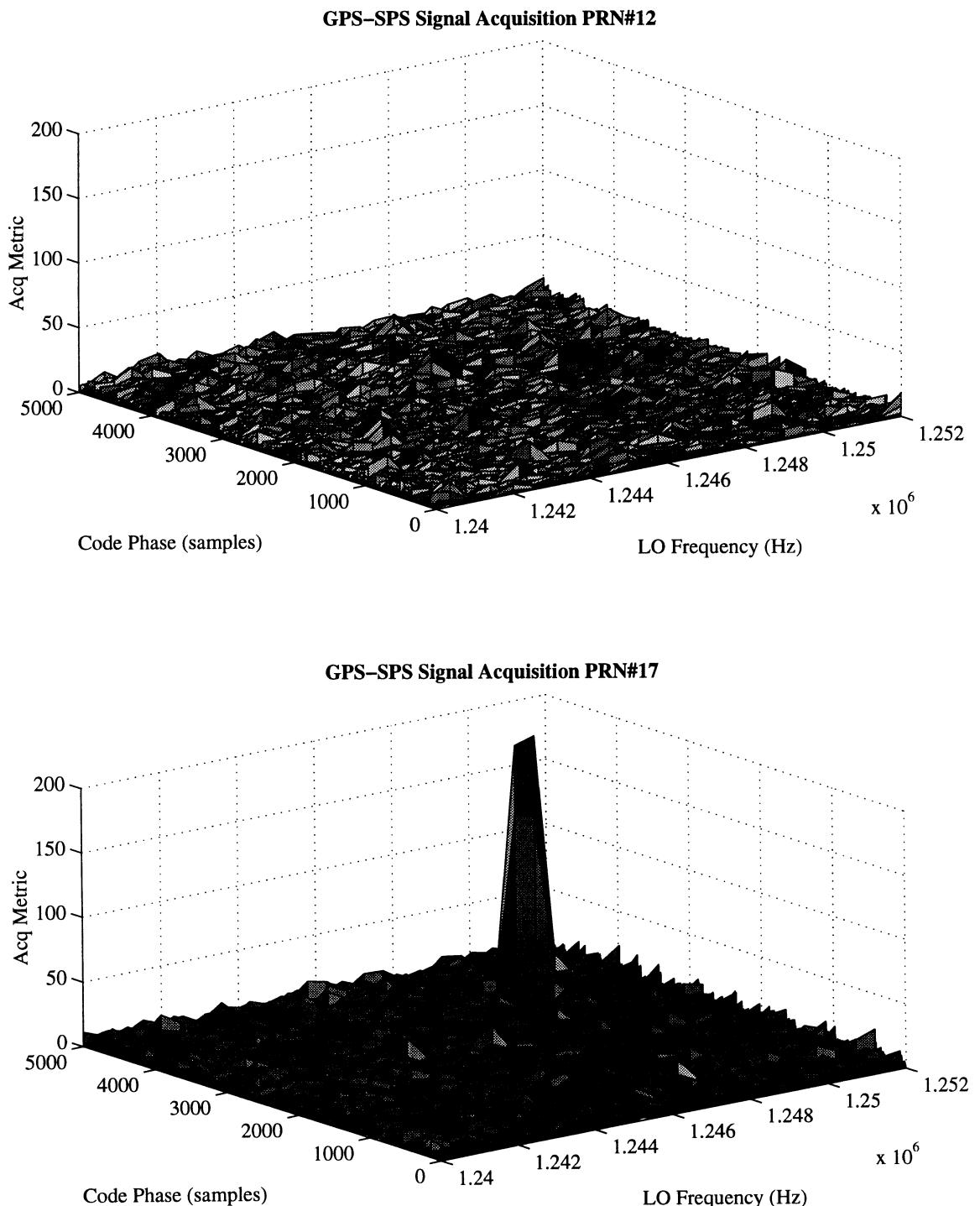


Figure 6.3 GPS-SPS Signal Acquisition Results for Two PRN Codes
(PRN #12 is not present in the data)

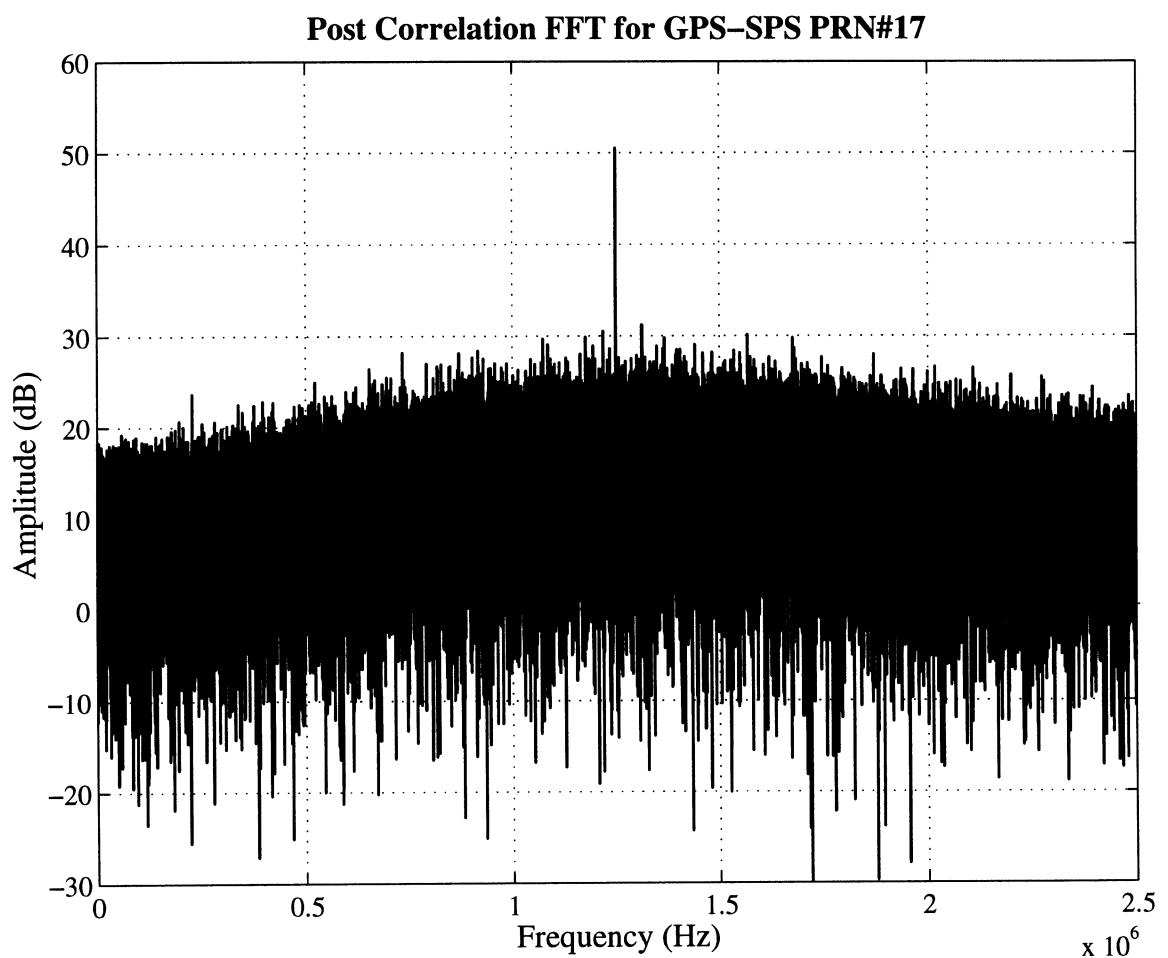


Figure 6.4 Post-Correlation FFT for GPS-SPS PRN#17

Table 6.1 Satellites Identified in Collected Data via Acquisition

satellite PRN #	Code Phase (samples)	Frequency (Hz)
6	2885	1245800
10	3815	1245050
17	4079	1249750
23	2201	1249950
26	2665	1247750
28	3270	1249950

fundamental parameters which, when specified, completely determine all components of the loop. These parameters are the natural frequency, ω_n , and damping factor, ζ .

The damping factor plays an important role in the dynamic performance of the tracking loop. Control theory defines the following conditions for various damping factors [38].

$\zeta = 0$	undamped
$0 < \zeta < 1$	underdamped
$\zeta = 1$	critically damped
$\zeta > 1$	overdamped

When working with an underdamped system, the step response is rapid and will overshoot the desired state and will oscillate until settling to the final desired state. If the system is overdamped, the step response will be slow in achieving the desired state but will do so without any oscillation. The optimally flat response is achieved using $\zeta = (1/\sqrt{2})$, which corresponds to a second order Butterworth lowpass filter. Thus, this will be the choice used for both carrier and code tracking loops.

The natural frequency of the loop is the second parameter that must be specified in order to implement the tracking loops. It, like the damping factor, is a compromise. Assuming that the damping factor is fixed, a relatively small natural frequency will provide excellent noise performance but will be unable to track dynamics induced on the signal. A relatively large natural frequency will be able to track signal dynamics but will have poor noise performance. These conclusions are based strongly on the noise bandwidth (B_L) approximation for the tracking loop that is presented in Equation 6.1

$$B_L \approx \frac{\omega_n}{2} \left(\zeta + \frac{1}{4\zeta} \right) \underset{\zeta=1/\sqrt{2}}{\approx} 0.53 \omega_n \quad (6.1)$$

Based on the dynamics expected, typical noise bandwidths for the code and carrier tracking loops are 1 Hz and 25 Hz, respectively [1]. Thus, the corresponding natural frequencies of 2 Hz and 50 Hz will be used for the code and carrier tracking loops, respectively, in the GPS-SPS software radio implementation.

In order to illustrate and measure the performance of the tracking loops, the collected data is utilized along with the acquisition parameters for GPS-SPS PRN#6. The coupled tracking loops are designed to provide four measurements, although using the software-based implementation, virtually any point in the tracking loops could be monitored. These measurements are the in-phase and quadrature arms of the Costas loop along with the filtered error signals of the delay lock loop and Costas loop.

Assuming the loops are operating in the locked state, the navigation data is present in the in-phase arm of the Costas loop. This is clearly evident from Figure 6.5 which shows a plot of the in-phase and quadrature arms of the Costas loop. This operation is ensured if the acquisition parameters of Table 6.1 are used to initialize the tracking loops. However, one must consider the case where the acquisition parameters are slightly in error. There will be a pull-in process during which the tracking loops will acquire and lock on to the signal. This is shown in Figure 6.6, where the frequency initialization of the tracking loops is biased by 30 Hz. The tracking loop does lock on to the signal and correctly demodulate the data bits, but there is a transient period as a result of the erroneous frequency initialization.

The tracking loop bandwidth can be used to reduce the pull-in time required for an initial frequency bias. This is shown in Figure 6.7, where the same 30 Hz bias is used to initialize the tracking loops. However, in this case lock is achieved much more rapidly. This is a result of increasing the carrier tracking loop bandwidth from the 25 Hz to 100 Hz. However, the increase in the tracking loop bandwidth is a compromise. Although lock is achieved more rapidly, it is

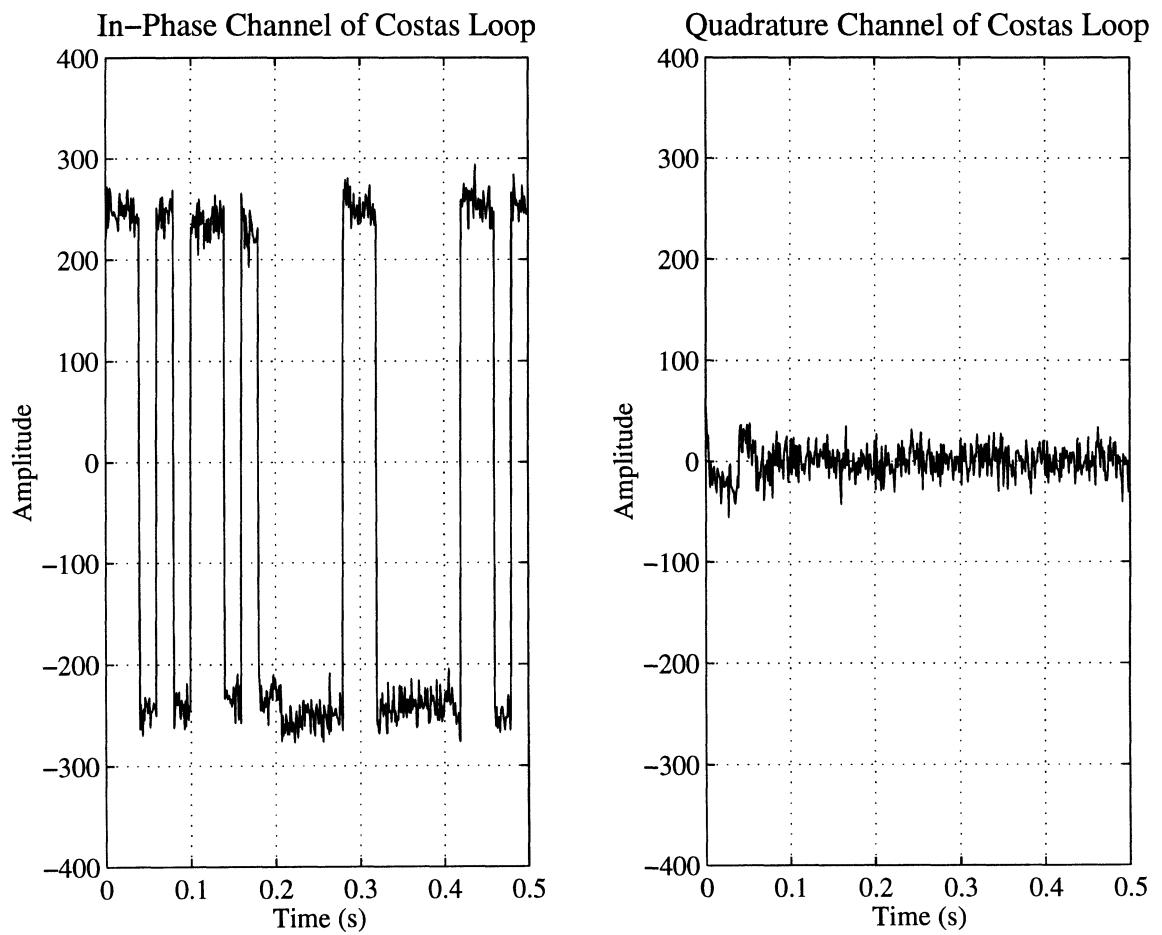


Figure 6.5 In-phase and Quadrature Costas Loop Components

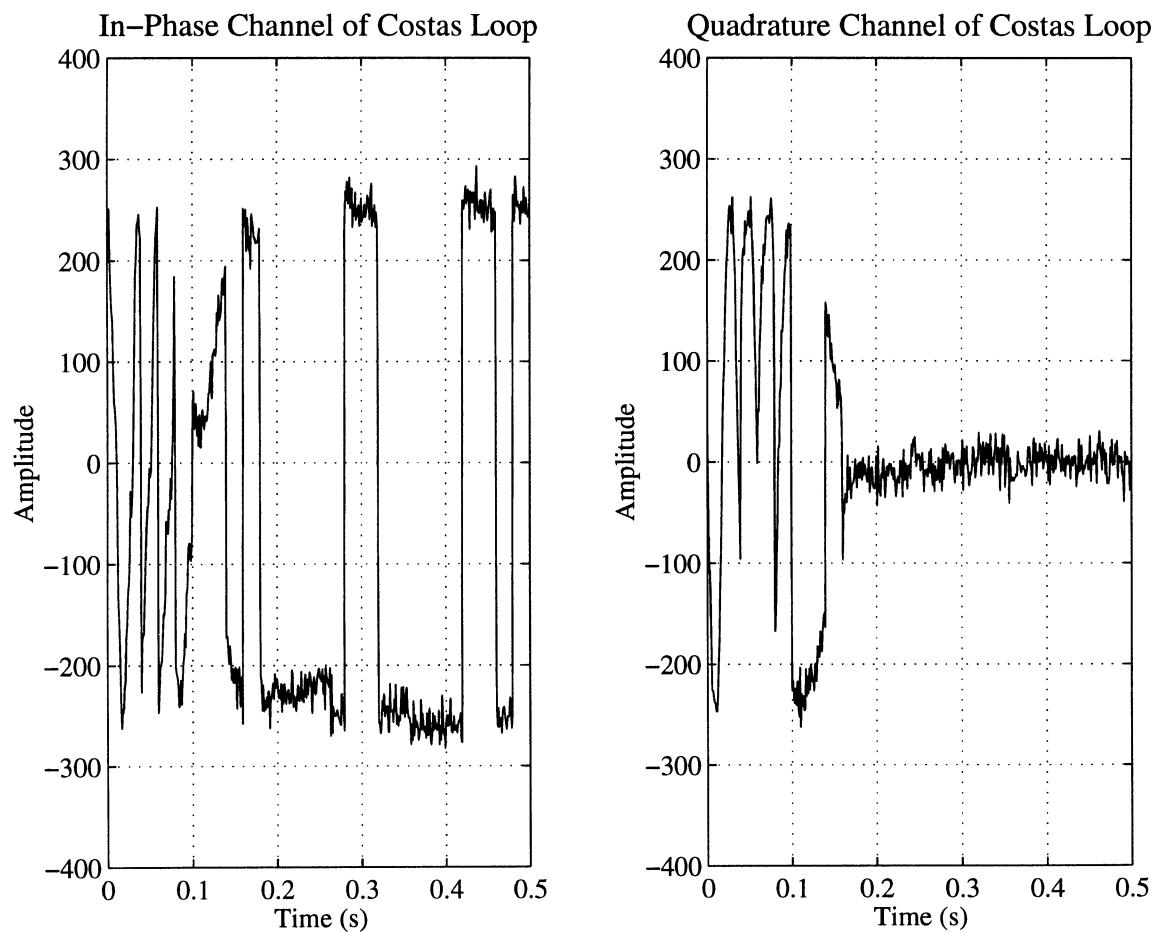


Figure 6.6 In-phase and Quadrature Costas Loop Components (30 Hz Initial Bias)

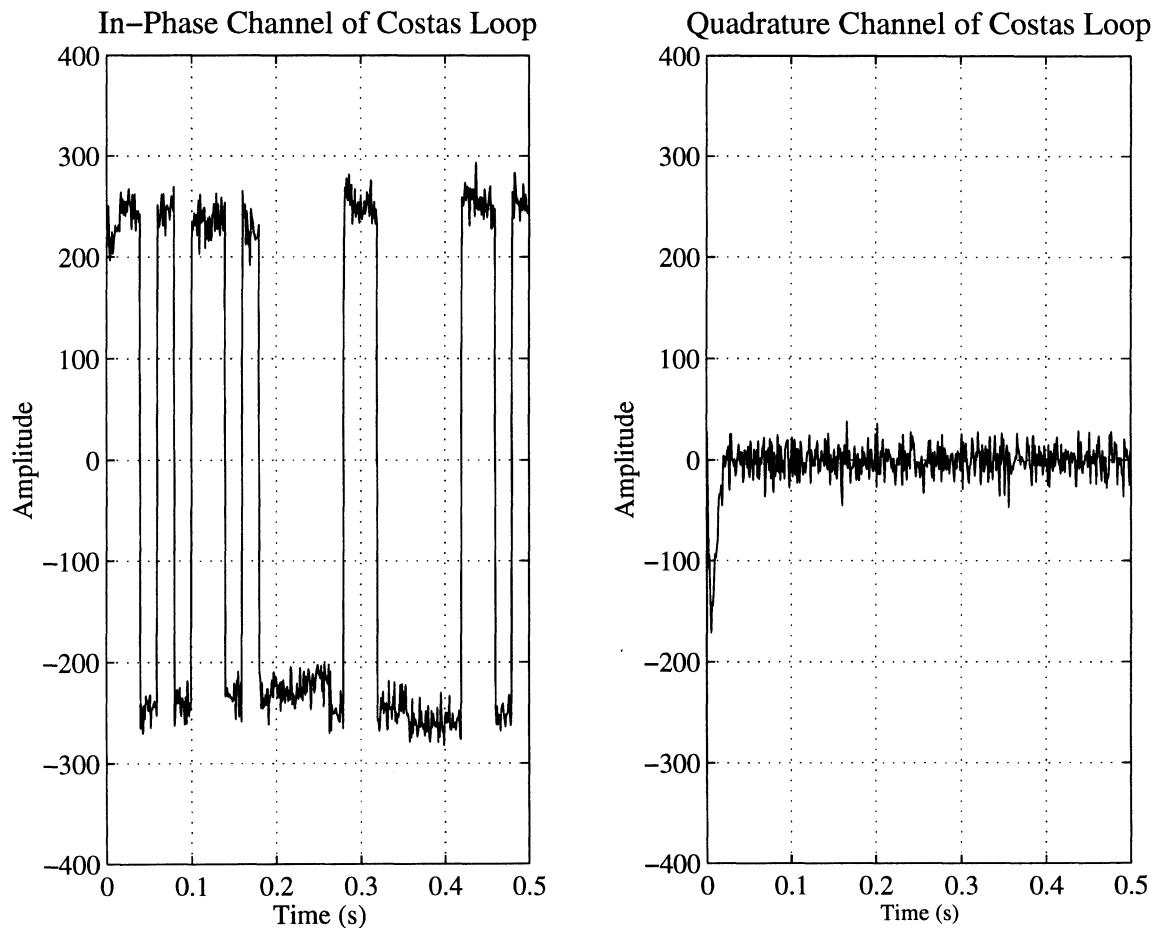


Figure 6.7 In-phase and Quadrature Costas Loop Components (30 Hz Initial Bias & 100 Hz Loop Bandwidth)

considerably noisier. This too is clearly evident after inspecting at the Costas loop filtered error signal. This is shown for the 25 Hz and 100 Hz loop bandwidths, when initialized with a 20 Hz bias, in Figure 6.8. From this plot it is clear the there is a higher level of tracking error in the wider bandwidth system. Also note how the response conforms to the linear second order model with the slight overshoot and settling time in both cases. Also shown in the plot is the pull-in process for the 25 Hz bandwidth loop. Each of these responses, as well as other parameters such as lock-in time and RMS tracking error, are predicted via the linear phase lock loop model for both the carrier and code tracking loops [32].

In a traditional communication receiver, the tracking error would only be considered if it affected the bit error rate of the channel. This is not true for a GNSS receiver, in which the code tracking loop measurements, in conjunction with the carrier tracking loop in some configurations, provide the precise clock synchronization necessary to solve for position. Thus Figure 6.8 illustrates the trade-off in designing the tracking loops for a GNSS receiver. A wide loop bandwidth will pull-in and lock onto the input signal quickly. However, this same wide loop bandwidth will introduce a significantly higher level of tracking error into the loops.

Consider the case in which the acquisition algorithm is designed to provide a very accurate set of parameters to the tracking loops. Initially, it may appear that the tracking loop bandwidth could be significantly reduced. This is true, but the minimum bandwidths must still accommodate the signal dynamics due to the line-of-sight velocity between the receiver and transmitter. Thus the goal is to use the minimum tracking loop bandwidths for the dynamics expected as this design will maintain lock and provide the least amount of tracking error. This is the philosophy behind the choice of the 1 Hz code tracking loop bandwidth and 25 Hz carrier track loop bandwidth.

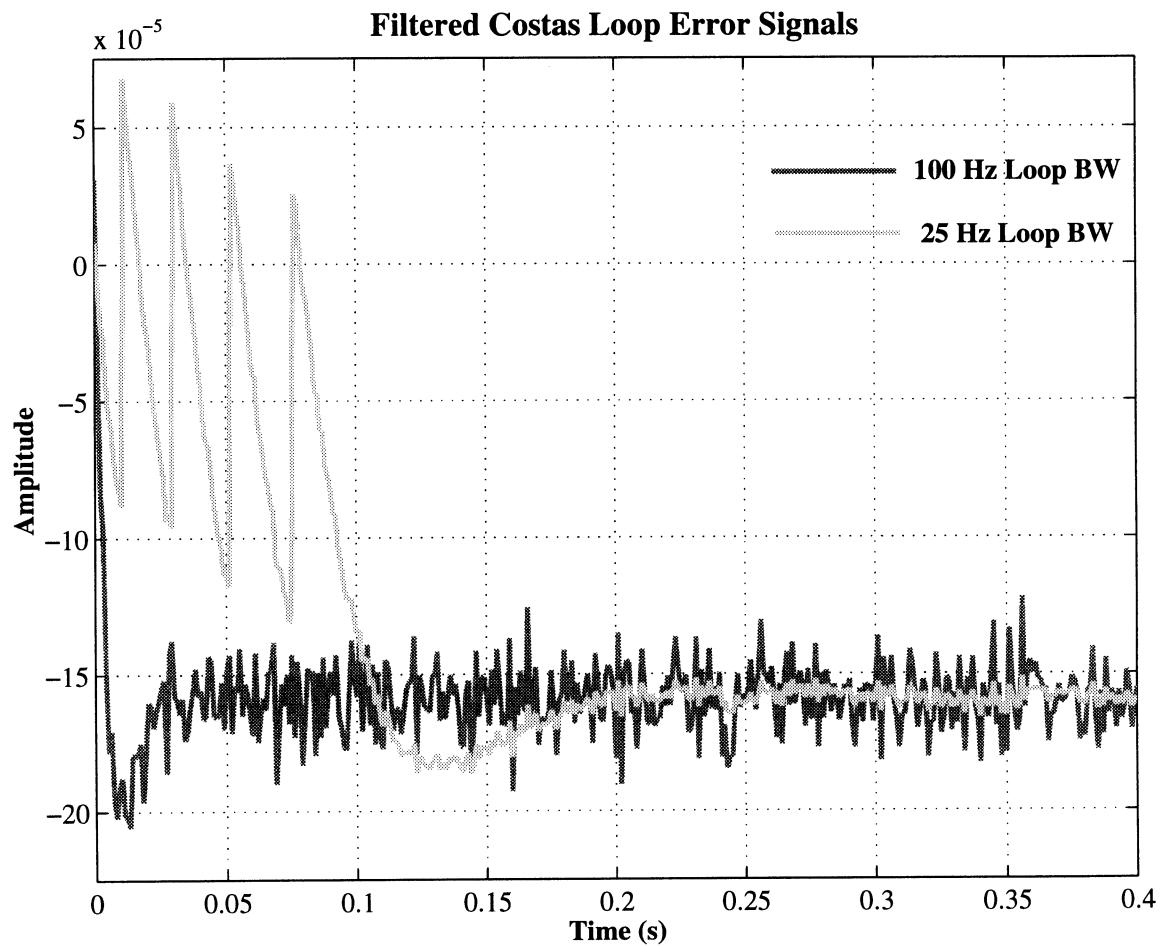


Figure 6.8 Resulting Error Signals for 25 Hz and 100 Hz Loop Bandwidth
(30 Hz Initial Bias)

The software radio implementation offers tremendous flexibility in the tracking loop implementation. For example, the tracking loop bandwidths could be adjusted dynamically. In this case the acquisition algorithm could provide only a rough estimate of the necessary parameters and the tracking loops could be initiated with a wide bandwidth. Once locked, the bandwidths could be reduced to accommodate only signal dynamics. Further, if line-of-sight dynamics could be monitored, the software implementation could utilize this information to further adapt the loop bandwidths.

6.4 Position Solution

The demodulated navigation data combined with the code tracking loop measurements provide the necessary components to compute a position solution. Although there has been little discussion regarding the processing of the navigation data and resulting position calculation, it is by no means trivial. The decoded data bits must be searched for a possible preamble and if successful, a parity check is performed and words in the data are decoded. Once the validity of the words has been confirmed, they are combined with timing measurements from the code tracking loop. The cumulative information is applied mathematically to solve for a position estimate. The reason for the exclusion of a detailed description of this procedure is that it is well documented in other references and need not be repeated here [2, 5].

It is important, however, to present those data words which have been decoded from the collected data necessary for the position solution. Consistent with the GPS-SPS software radio design, the extraction and processing of the data words to obtain a position solution is accomplished entirely in software. Tables 6.2a and 6.2b present the satellite clock and health data extracted from the first subframe. Tables 6.3a and 6.3b lists the ephemeris parameter obtained from the second and third subframes. Finally, Tables 6.4a and 6.4b list the time of

week biases resulting from the code tracking loop, the resulting satellite ECEF coordinate system positions calculated from the data in the previous tables, and the pseudoranges which are based on the time of week biases assuming a transmission time of 70 milliseconds for PRN#6 which was selected as a reference satellite for the computation. Note that all parameters in the table are given in the units in which they are transmitted.

Table 6.2, 6.3 and 6.4 contain all the parameters necessary to obtain a position solution. Recall that the antenna used for data collection was positioned at a surveyed location for reference. If those coordinates, listed in Figure 6.1, are transformed to the ECEF system, the following position results.

$$\begin{aligned} X &= 678454.5 \text{ m} \\ Y &= -4893826.1 \text{ m} \\ Z &= 4020518.8 \text{ m} \end{aligned}$$

Based on the parameters of Tables 6.2, 6.3, and 6.4 obtained from the software radio implementation, the following position estimate was calculated [2, 5].

$$\begin{aligned} \hat{X} &= 678472.6 \text{ m} \\ \hat{Y} &= -4893780.9 \text{ m} \\ \hat{Z} &= 4020484.5 \text{ m} \end{aligned}$$

Therefore, the error in the position estimate can be calculated.

$$\begin{aligned} \Delta X &= |X - \hat{X}| = 18.1 \text{ m} \\ \Delta Y &= |Y - \hat{Y}| = 45.2 \text{ m} \\ \Delta Z &= |Z - \hat{Z}| = 34.3 \text{ m} \\ \sqrt{(X - \hat{X})^2 + (Y - \hat{Y})^2 + (Z - \hat{Z})^2} &= 59.6 \text{ m} \end{aligned}$$

The geometric dilution of precision, a metric which indicates the geometric satellite/user configuration, for the parameters involved was calculated to be: 3.267 [2, 5].

Table 6.2a Satellite Clock & Health Data (First Three Satellites)

Parameter	Satellite		
	PRN#6	PRN#10	PRN#17
Week #	875	875	875
Time of Week	81231	81231	81231
URA User Range Accuracy	7	7	7
SV Health	0	0	0
T _{GD} (s) Group Delay Differential	1.39698E-09	-1.86265E-09	1.39698E-09
t _{OC} (s) Clock Correction Term	489600	489600	489600
a _{f 2} (s/s ²) Clock Correction Term	0.00000E+00	0.00000E+00	0.00000E+00
a _{f 1} (s/s) Clock Correction Term	2.27374E-13	3.75167E-11	-1.13687E-12
a _{f 0} (s) Clock Correction Term	2.80188E-06	2.42089E-04	-1.28600E-04
IODC Issue of Data, Clock (Complete 10 Bits)	222	578	697
IODC Issue of Data, Clock (8 Least Significant Bits)	222	66	185

Table 6.2b Satellite Clock & Health Data (Last Three Satellites)

Parameter	Satellite		
	PRN#23	PRN#26	PRN#28
Week #	875	875	875
Time of Week	81231	81231	81231
URA User Range Accuracy	7	7	2
SV Health	0	0	0
T _{GD} (s) Group Delay Differential	1.39698E-09	1.39698E-09	1.39698E-09
t _{OC} (s) Clock Correction Term	489600	489600	489600
a _f ₂ (s/s ²) Clock Correction Term	0.00000E+00	0.00000E+00	0.00000E+00
a _f ₁ (s/s) Clock Correction Term	4.54747E-13	-6.13909E-12	2.16005E-12
a _f ₀ (s) Clock Correction Term	5.10458E-06	-5.48181E-05	1.12602E-04
IODC Issue of Data, Clock (Complete 10 Bits)	680	38	386
IODC Issue of Data, Clock (8 Least Significant Bits)	168	38	130

Table 6.3a Ephemeris Parameters (First Three Satellites)

Parameter	Satellite		
	PRN#6	PRN#10	PRN#17
M ₀ (sc) Mean Anomaly	-1.22331E-01	9.31305E-01	-1.70162E-01
Δ n (sc/s) Mean Motion Difference	1.46872E-09	1.43052E-09	1.31274E-09
e Eccentricity	6.63736E-03	1.60454E-03	9.09195E-03
√A (m ^{0.5}) Square Root of Semi-Major Axis	5.15367E+03	5.15366E+03	5.15366E+03
Ω ₀ (sc) Longitude of Ascending Node	-2.05056E-01	4.53042E-01	1.47840E-01
i ₀ (sc) Inclination Angle	3.05875E-01	3.05820E-01	3.11550E-01
ω (sc) Argument of Perigee	-8.60292E-01	-1.08436E-01	7.56198E-01
OMEGADOT ($\dot{\Omega}$) (sc/s) Right Ascension Rate	-2.68642E-09	-2.52760E-09	-2.56534E-09
IDOT (\dot{i}) (sc/s) Rate of Inclination Angle	3.91083E-11	1.02318E-10	1.69393E-11
C _{uc} (rad) Cosine Correction (Latitude)	2.10851E-06	-2.96719E-06	-4.35486E-06
C _{us} (rad) Sine Correction (Latitude)	4.28036E-06	1.02613E-05	6.26966E-06
C _{rc} (m) Cosine Correction (Orbit Radius)	2.94781E+02	1.81188E+02	2.68719E+02
C _{rs} (m) Sine Correction (Orbit Radius)	3.88438E+01	-6.02188E+01	-7.91563E+01
C _{ic} (rad) Cosine Correction (Inclination Angle)	5.40167E-08	-3.35276E-08	-9.68575E-08
C _{is} (rad) Sine Correction (Inclination Angle)	9.68575E-08	4.84288E-08	-1.75089E-07
t _{oe} (s) Reference Time Ephemeris	489600	489600	489600
IDOE (Subframe #2) Issue of Data, Ephemeris	222	66	185
IDOE (Subframe #3) Issue of Data, Ephemeris	222	66	185

Table 6.3b Ephemeris Parameters (Last Three Satellites)

Parameter	Satellite		
	PRN#23	PRN#26	PRN#28
M ₀ (sc) Mean Anomaly	9.53197E-01	4.96258E-01	-3.23195E-01
Δ n (sc/s) Mean Motion Difference	1.38868E-09	1.57934E-09	1.36333E-09
e Eccentricity	1.10873E-02	9.28525E-03	2.44119E-03
√A (m ^{0.5}) Square Root of Semi-Major Axis	5.15372E+03	5.15361E+03	5.15378E+03
Ω ₀ (sc) Longitude of Ascending Node	4.66649E-01	7.88221E-01	-2.11516E-01
i ₀ (sc) Inclination Angle	3.06635E-01	3.04739E-01	3.09334E-01
ω (sc) Argument of Perigee	-6.97285E-01	-1.63183E-01	-9.82689E-01
OMEGADOT ($\dot{\Omega}$) (sc/s) Right Ascension Rate	-2.45279E-09	-2.58819E-09	-2.61718E-09
IDOT (\dot{i}) (sc/s) Rate of Inclination Angle	9.17453E-11	9.77707E-12	4.02451E-11
C _{uc} (rad) Cosine Correction (Latitude)	-2.42889E-06	4.19095E-07	1.47335E-06
C _{us} (rad) Sine Correction (Latitude)	9.91859E-06	7.12834E-06	4.01586E-06
C _{rc} (m) Cosine Correction (Orbit Radius)	1.89813E+02	2.34656E+02	3.04719E+02
C _{rs} (m) Sine Correction (Orbit Radius)	-4.68750E+01	4.78125E+00	2.98438E+01
C _{ic} (rad) Cosine Correction (Inclination Angle)	4.09782E-08	1.34110E-07	2.79397E-08
C _{is} (rad) Sine Correction (Inclination Angle)	-1.56462E-07	2.60770E-08	-1.86265E-09
t _{oe} (s) Reference Time Ephemeris	489600	489600	489600
IDOE (Subframe #2) Issue of Data, Ephemeris	168	38	130
IDOE (Subframe #3) Issue of Data, Ephemeris	168	38	130

Table 6.4a Time of Week Bias, Satellite Positions, Pseudorange (First Three Satellites)

Parameter	Satellite		
	PRN#6	PRN#10	PRN#17
Time of Week Bias (s) (from Code Tracking Loop)	4.94600E-04	-6.69500E-03	0.00000E+00
Satellite X Coordinate (m)	-5571316.2	20366521.0	-8812624.8
Satellite Y Coordinate (m)	-25144575.0	-3367013.4	-11897730.0
Satellite Z Coordinate (m)	5924623.8	16770721.0	21861178.0
Pseudorange (m)	20985472.0	23212569.0	21094359.0

Table 6.4b Time of Week Bias, Satellite Positions, Pseudorange (Last Three Satellites)

Parameter	Satellite		
	PRN#23	PRN#26	PRN#28
Time of Week Bias (s) (from Code Tracking Loop)	-5.34760E-03	3.54860E-03	-5.56140E-03
Satellite X Coordinate (m)	-17599910.0	8742707.0	-14900738.0
Satellite Y Coordinate (m)	-17376334.0	-20276563.0	-6265146.4
Satellite Z Coordinate (m)	10384526.0	14617044.0	21057733.0
Pseudorange (m)	22737603.0	20052627.0	22833930.0

The error in the resulting position solution is well within the bounds established by Selective Availability [5], thus the software implementation of the signal processing algorithms for a GPS-SPS software radio implementation can be considered a success.

6.5 GLONASS Processing

The software radio front end implementations of Chapter 4 included designs capable of processing the GLONASS transmissions. In addition, the acquisition algorithms of Chapter 5 have already been successfully applied to the GLONASS signals in verifying the front end implementations. In order to demonstrate further the flexibility and feasibility of a GNSS software radio, GLONASS tracking will be implemented.

In order to capture an extended data record, the hardware configuration of Figure 6.1 was used with slight modifications as is shown in Figure 6.9. Note that the principle difference is the first filter and local oscillator frequency. This design will capture two of the GLONASS frequency channels. Channel 21, originally at 1613.8125 MHz, will be translated to 968.75 kHz and channel 22, originally at 1614.375 MHz will be translated to 1.531250 MHz after sampling.

The GLONASS orbital parameters were utilized to determine when two satellites broadcasting on frequencies 21 & 22 would be in view simultaneously. At the predicted time, a data set was collected. Applying the acquisition algorithm to the collected data provides the results in Table 6.5 that indicates both satellites are present in the data set. The unprocessed data is shown in the time and frequency domains along with the post-correlation FFTs in Figure 6.10. As was the case for the GPS signals, there is no discernable signal in the raw data and after correlation, the signal is clearly evident at the correct frequency.

The same 1 Hz and 25 Hz code and carrier tracking loops, respectively, can be applied to track the GLONASS signal. The resulting in-phase channel and error signal for both GLONASS

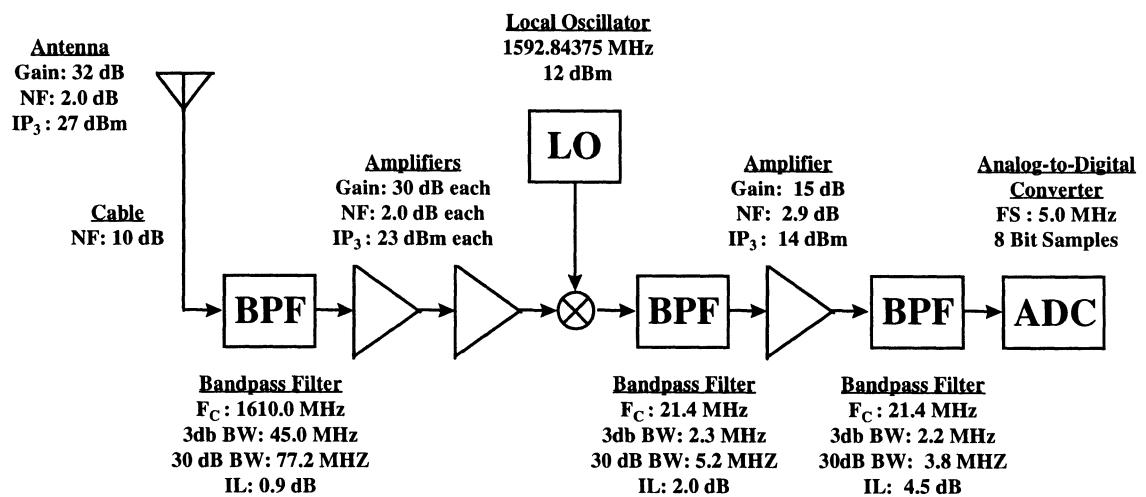


Figure 6.9 Front End Design to Capture an Extended GLONASS Data Set

Table 6.5 GLONASS Satellites Identified in Collected Data via Acquisition

GLONASS Frequency	Code Phase (samples)	Frequency (Hz)
21	431	968250
22	1190	1532150

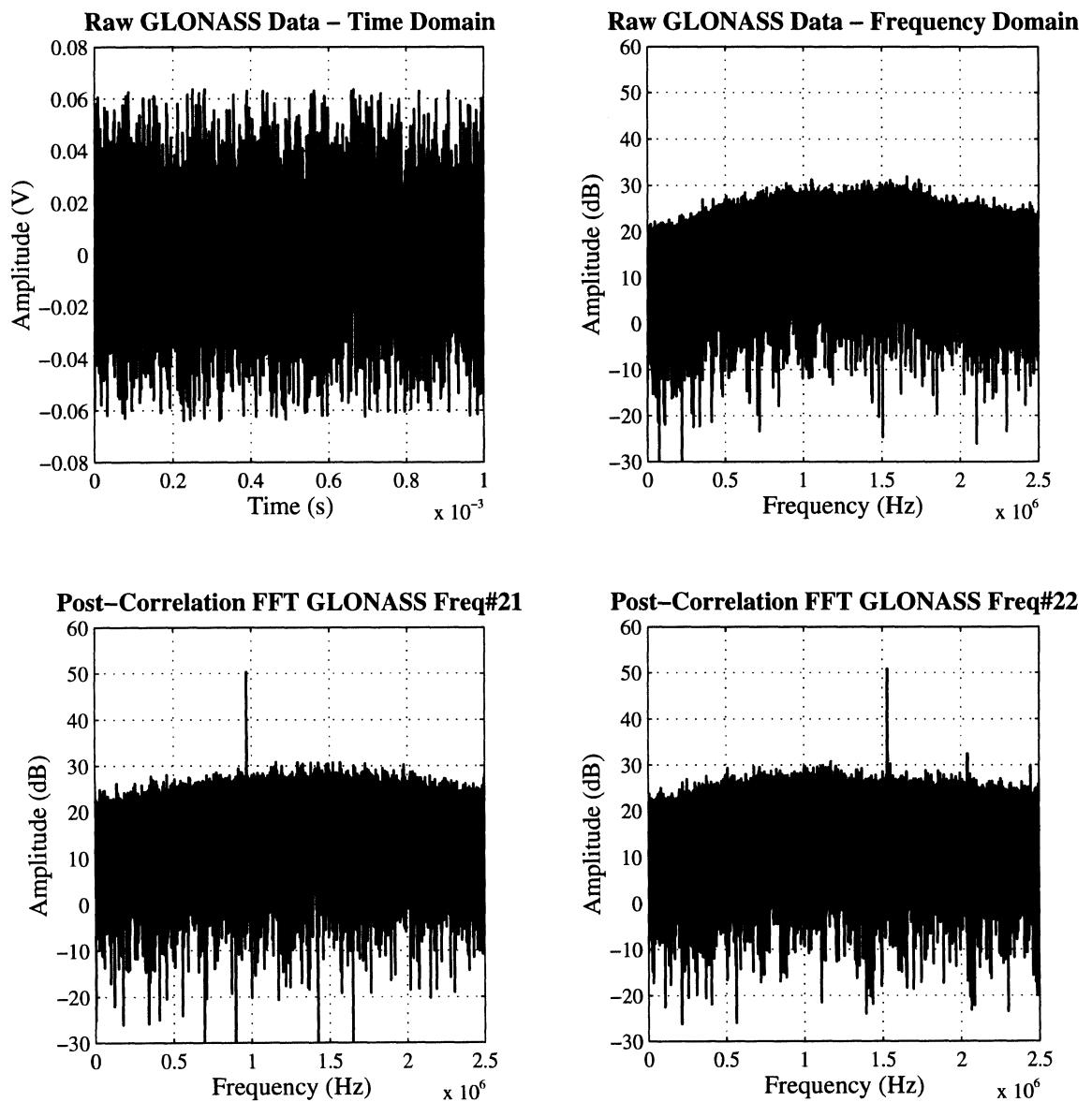


Figure 6.10 Unprocessed GLONASS Data in Time and Frequency Domains Along With the Post-Correlation FFTs

satellites are shown in Figure 6.11 for the first half of a second. After a short pull-in process, the product of meander and navigation data is apparent in the in-phase channel of both satellites. The short pull-in process is not a result of a change in the loop bandwidth, rather it is a result of the accuracy of the parameters in the transition between acquisition and tracking. Previously, it was possible to determine the carrier frequency down \pm 25 Hz. However, the higher bit rate, 100 bits/second, which results from the product of the meander sequence with the navigation data allows resolution of the carrier only to within \pm 50 Hz. Thus an initial pull-in process is more likely in processing the GLONASS data when the GPS-SPS tracking loop bandwidths are utilized.

A position solution is not possible using data from only two satellites. As a result of the FDMA structure of the GLONASS transmission, it is not possible to capture additional satellite transmissions using the existing hardware. However, software algorithms have been developed and have demonstrated acquisition, tracking, and data demodulation for the GLONASS broadcast.

6.6 Real Time Processing Considerations

All of the results presented have been postprocessed using the Matlab programming language. This type of implementation provided the highest level of flexibility at the expense of reduced computational efficiency. One second, or 5,000,000 samples of GPS-SPS data requires approximately 45 seconds of processing time using an Intel Pentium Pro 200 MHz microprocessor. This raises the question as to whether a programmable processor will ever be able to process the resulting samples at a rate equal to or faster than they are generated for a real time implementation. This is of importance when one considers the higher sampling frequencies required of the multiple signal GNSS software radio implementations proposed in Chapter 4.

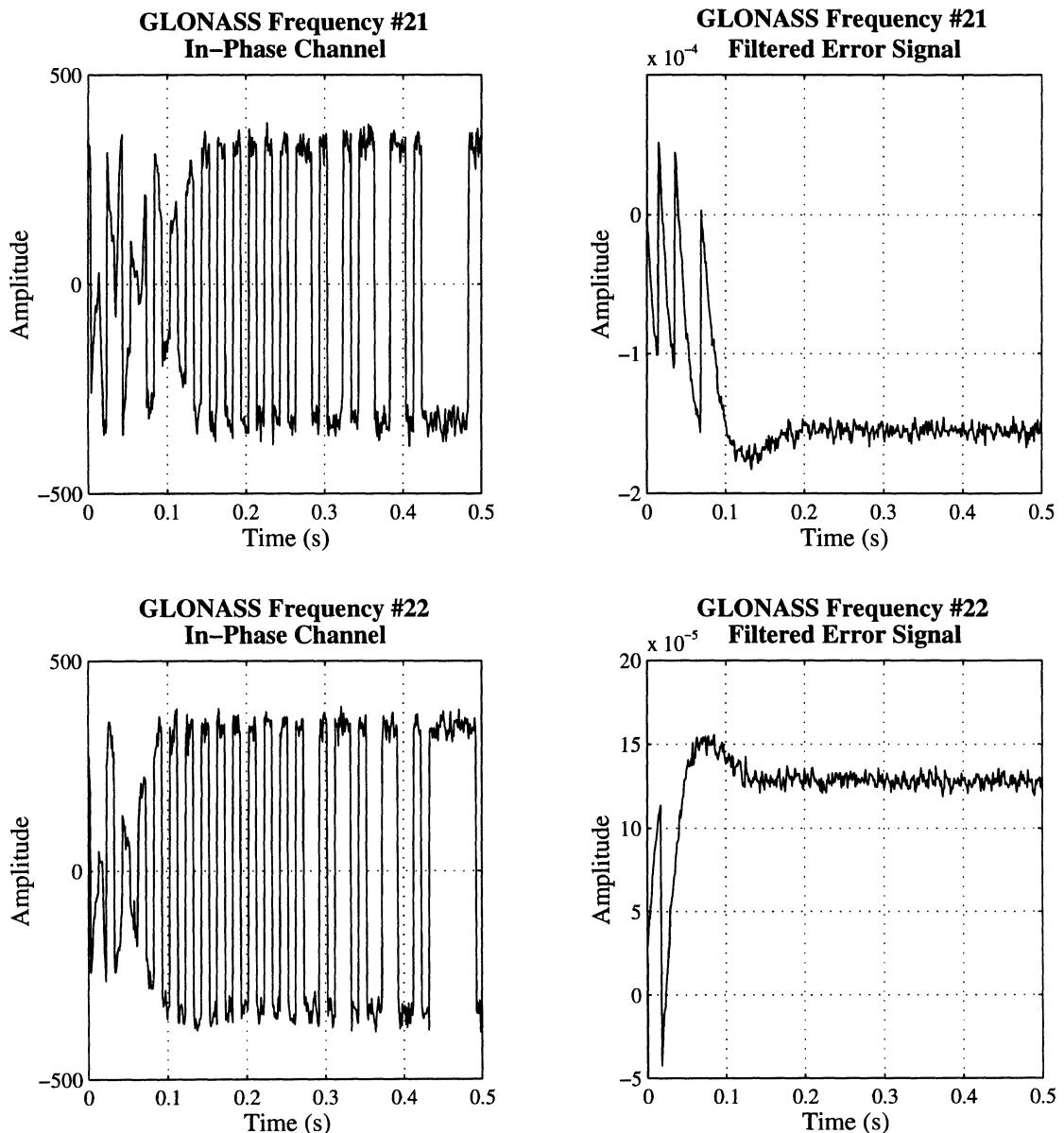


Figure 6.11 GLONASS In-Phase and Quadrature Channels for both Satellites

The obvious answer is yes. Recent history indicates that programmable processing power will be available for a real time implementation, most likely in the near future. The confidence here is based on Moore's Law, a prediction made by Intel Chairman of the Board Gordon E. Moore at the time of inception of the microprocessor. Stated simply, it says that "processing power will double every 18 months" and it has held true since it was made [12]. Further Andrew S. Grove, CEO of Intel, made a prediction at the 1996 COMDEX convention that by the year 2011 the typical PC microprocessor will contain 1 billion transistors, in comparison with the 5.5 million in the Pentium chip, operate at clock speeds of 10 GHz, and execute 100 billion instructions per second [39]. Although real time processing may not be feasible immediately, it will be in the near future with the exponential increase in programmable processing power.

This is important as it will allow the focus of the software radio implementation to be placed on the signal processing algorithms as opposed to implementation of those algorithms. In other words, research can be targeted to find the optimum signal processing for a particular software radio implementation. Developing and testing these routines in a high level language will offer tremendous flexibility and will eventually provide real time operation, if it is not already possible. If efforts are focused on the implementation of basic signal processing on a target microprocessor, it will limit the flexibility. For example, when a next generation processor is released it often contains new instructions to improve performance. Since next generation processors appear on a yearly or bi-yearly basis, much of the effort in targeting the algorithm and learning a specific architecture will be made obsolete very rapidly. Also, if an improvement is proposed to the signal processing algorithm it may require a significant change in the programming if it was initially done in the machine language of the chip. It is understood that some software radio implementations are needed to operate in real time and that it may not be

possible to wait for processing power to catch up. In this case a low level machine code implementation still offers more flexibility than an ASIC or FPGA implementation (see Figure 3.3).

Finally, the performance measured here for the GPS-SPS implementation is not absolute for the programmable microprocessor and high level language technology currently available. The results were achieved using Matlab, one of the highest level languages available whose instructions are processed by an interpreter as opposed to being compiled and executed. An implementation in the C programming language should definitely offer improved performance and yet maintain the cross platform capability of a high level language. Also, Intel and Texas Instruments recently introduced the Pentium II MMX processor and TMS320C6x digital signal processor, respectively. Both of which have the capacity to offer improved performance.

7. Summary & Conclusions

This research has focused on a GNSS software radio implementation and brought about the first GPS-SPS software radio implementation. First, both the software radio and GNSS concepts were introduced describing the benefits of each. GNSS is an ideal choice for a software radio implementation. This is a result of complex signal processing required for GNSS and increasing reliance on the transmission within the national airspace system for navigation and positioning information. The GNSS software radio implementation was then described and the results from a successful GPS-SPS implementation were presented.

In any software radio design, there are two basic components: the front end design and the software signal processing. The front end design and implementation was a highlight of the research. Ideally in the software radio the ADC is placed as near the antenna as possible eliminating many of the components related to frequency translation. The placement of the ADC may initially appear to require an impractical sampling frequency, however, bandpass sampling can be used to provide frequency translation. Although bandpass sampling has been established theoretically, this research resulted in the first direct digitization bandpass sampled GNSS front end design.

The bandpass sampling concept was extended to incorporate direct RF sampling of multiple signals. This is of particular importance in a GNSS receiver as there are multiple frequency bands of interest and use of these different signals adds integrity and reliability to the measurements. The proposed design allows a single front end implementation to capture any number of signals of interest through a careful choice of a single parameter, the sampling frequency. The theory has been successfully demonstrated for the combination of GPS-SPS and GLONASS transmissions. Further, an adequate sampling frequency has been calculated to process GLONASS as well and the GPS L1 and L2 transmissions.

The second half of the software radio is the implementation of the signal processing routines on a programmable microprocessor. The signal processing is quite complex as GNSS signals typically use a CDMA spread spectrum modulation. Signal acquisition, PRN code and carrier tracking, data demodulation and processing are all required. The signal processing, although not as revolutionary as the front end implementation, established a framework for the testing and implementation of advanced signal processing algorithms. A variety of acquisition algorithms were coded to demonstrate the flexibility of the approach. The tracking loops, both code and carrier, are based on a discrete implementation of the linear phase lock loop and have been coded to provide maximum flexibility.

Finally, a GPS-SPS software radio was implemented to postprocess a 30 second window of data to obtain a position solution. The position estimate was well within the accuracy limits imposed by Selective Availability, thus the GPS-SPS software radio implementation was successful. In order to show the flexibility of the software algorithms, they were also successfully applied to acquire and track the GLONASS signal.

A basis for a GNSS software radio has been established. This includes advanced front end designs to process all the GNSS transmissions and the basic software algorithms to utilize the GPS-SPS signal to obtain a position solution. With this, all the benefits of the software radio have been brought into GNSS receiver design. From this framework, advanced signal processing, such as adaptive filtering and nulling of interference sources can be developed and implemented. This research will allow the optimum GNSS receiver structure to be implemented, which is of utmost importance given the tremendous responsibilities being placed on GNSS.

REFERENCES

- [1] Kaplan, Elliott D., Editor, Understanding GPS : Principles and Applications, Artech House, ISBN: 0890067937, March, 1996.
- [2] Parkinson, Bradford W. and Spilker, James J. Jr., Editors, Global Positioning System: Theory and Application, Volume 1, American Institute of Astronautics and Aeronautics, ISBN: 156347106X, June 1996.
- [3] ICD-GPS-200 (GPS Interface Control Document), GPS Joint Program Office (prepared by ARINC Research), July 1991.
- [4] Sarwate, D. V., and Pursley, M. B., "Crosscorrelation Properties of Pseudorandom and Related Sequences," Proceeding of the IEEE, May 1980.
- [5] Global Positioning System Standard Positioning Service Signal Specification, U.S. Department of Transportation, Second Edition, June 2, 1995.
- [6] GLONASS Interface Control Document, (RTCA Paper No. 639-95/SC 159-685), Russian Federation Institute of Space Device Engineering, 1995.
- [7] Wepman Jeffery A., "Analog-to-Digital Converters and Their Application in Radio Receivers," IEEE Communications Magazine, May 1995, Vol. 33, No. 5, pp. 39-45.
- [8] Vaughan, R., Scott, N. and White, D., "The Theory of Bandpass Sampling", IEEE Trans. on Signal Processing, Vol. 39, No. 9, pp. 1973-1984, Sept. 1991.
- [9] Hill, G. "The Benefits of Undersampling", Electronic Design, pp. 69-70, July 11, 1994.
- [10] Mitola, Joe, "The Software Radio Architecture," IEEE Communications Magazine, May 1995, Vol. 33, No. 5, pp. 26-38.
- [11] Baines, Rupert, "The DSP Bottleneck," IEEE Communications Magazine, May 1995, Vol. 33, No. 5, pp. 46-55.
- [12] Lewis, Ted, "The Nethead Gang," IEEE Computer, Dec. 1995, Vol. 28, No. 12, pp. 8-10.
- [13] Tsui, J. B. Y., Digital Techniques for Wideband Receivers. Artech House, ISBN: 0890068089, 1995.
- [14] Sharpin, D. L., and Tsui, J. B. Y., "Analysis of the Linear Amplifier/Analog-Digital Converter Interface in a Digital Microwave Receiver", IEEE Trans. Aerospace and Electronic Systems, Vol. 31, No. 1, pp. 248-256, Jan 1995.
- [15] Vizmuller, Peter, "RF Design Guide - Systems, Circuits, and Equations," Artech house, ISBN: 0890067546, 1995.

- [16] Akos, Dennis M., and Tsui, James B. Y., "Design and Implementation of a Direct Digitization GPS Receiver Front End," *IEEE Transactions on Microwave Theory and Techniques*, Dec. 1996.
- [17] Rappaport, Stephan S., and Grieco, Donald M., "Spread-Spectrum Signal Acquisition: Methods and Technology," *IEEE Communications Magazine*, June 1984, Vol. 22, No. 6, pp. 6-21.
- [18] Brown, Alison and Wolt, Barry, "Digital L-Band Receiver Architecture with Direct RF Sampling," *IEEE 1994 Position Location and Navigation Symposium*, Las Vegas, Nevada, April 11-15, 1994, pp. 209-215.
- [19] Tsui, J. B. Y., and Sanderson, R. B., "Selecting Minimum Sampling Frequency to Digitally Downconvert Two Separate Narrowband Signals," Wright Laboratories, Wright-Patterson AFB, Patent Pending.
- [20] Akos, D. M., and Stockmaster, M. H., "Direct Sampling of Three or More Frequency Separated Bandpass Signals at RF with a Minimum Sampling Rate," Wright Laboratories, Wright-Patterson AFB, Patent Pending.
- [21] Divis, Dee Ann, "L5: The Wait Continues," *GPS World*, Vol. 8, No. 4, April 1997, pp. 12-16.
- [22] Akos, Dennis M., and Braasch, Michael S., "A Software Radio Approach to Global Navigation Satellite System Receiver Design," *1996 Institute of Navigation Annual Meeting*, Cambridge, MA, June 1996.
- [23] Braasch, Michael S., Akos, Dennis M., Caschera, Joe, Stockmaster, Michael H., and Tsui, James B. Y., "Test Results From A Direct L-Band Digitizing GPS/Glonass Receiver," *1997 Institute of Navigation Annual Meeting*, Santa Monica, CA, January 1997.
- [24] Zhuang, Weihua, "Composite GPS Receiver Modeling, Simulations and Applications," Ph.D. Dissertation, University of New Brunswick, Oct. 1992.
- [25] Jovanovic, V. M., "Analysis of Strategies for Serial-Search Spread-Spectrum Code Acquisition—Direct Approach," *IEEE Transactions on Communications*, vol. COM-36, no. 11, Nov. 1988, pp. 1208-1220.
- [26] Soong, Chi-Li, "Fast Time-Domain-Based GPS Acquisition," M.S. Thesis, Ohio University, June 1996.
- [27] Holmes, Jack K., and Chen, Chang C., "Acquisition Time Performance of PN Spread-Spectrum Systems," *IEEE Transactions on Communications*, Vol. COM-25, No. 8, Aug. 1977, pp. 778-783.
- [28] Dicarlo, D. M. and Weber, C. L., "Multiple Dwell Serial Search: Performance and Application to Direct Sequence Code Acquisition," *IEEE Transactions on Communications*, Vol. COM-31, No. 5, May 1983, pp. 650-659.

- [29] Cheng, Unjeng, Hurd, William J. and Statman, Joseph I., "Spread-Spectrum Code Acquisition in the Presence of Doppler Shift and Data Modulation," IEEE Transactions on Communications, Vol. 38, No. 2, February 1990, pp. 241-250.
- [30] van Nee, D. J. R. and Coenen, A. J. R. M., "New Fast GPS Code-Acquisition Technique Using FFT," Electronic Letters, January 17, 1991, Vol. 27, No. 2, pp. 158-160.
- [31] Oppenheim, Alan V. and Schafer, Ronald, Discrete-Time Signal Processing, Prentice-Hall, ISBN: 013216292X, 1989.
- [32] Best, Roland E., Phase Locked Loops : Design, Simulation, and Applications, 3rd Ed., McGraw Hill, ISBN: 0070060517, 1997.
- [33] Makarios, A. H. and Tozer, T. C., "Noise and False-Lock Performance of the PSK-Tanlock Loop," IEEE Transactions on Communications, Vol. COM-30, No. 10, May 1982, pp. 2277-2283.
- [34] GPS BuilderTM Designer's Guide, GEC Plessey Semiconductors, GPS Group, Wiltshire, United Kingdom, Nov. 1994.
- [35] Chung, Bong-Young, Chien, Charles, Samueli, Henry, and Jain, Rajeev, "Performance Analysis of an All-Digital BPSK Direct Sequence Spread-Spectrum IF Receiver Architecture," IEEE Journal on Selected Areas in Communications, Vol. 11, No. 7, Sept. 1993, pp. 1096-1106.
- [36] Spilker, J. J. Jr., "Delay-Lock Tracking of Binary Signals," IEEE Transactions on Space Electronics and Telemetry, Vol. SET-9, March 1963, pp. 1-8.
- [37] Ziemer, Rodger E. and Peterson, Roger L., Digital Communications and Spread Spectrum Systems, MacMillan Publishing, ISBN: 0024316709, 1985.
- [38] Phillips, Charles L. and Harbor, Royce D., Feedback Control Systems, Prentice Hall, ISBN: 0133716910, 1996.
- [39] San Jose Mercury News, "Intel Chief Sounds an Alarm," Nov. 19, 1996, pp. C1-C4.