

Acquisition and Tracking of Coarse/Acquisition code with relevance to Global Positioning System

Ravindra.S.Kashyap (06307923), Shehbaz Thakur (03D07016)

Abstract—We are demonstrating the acquisition and tracking process for a Pseudo noise(PN) sequence with particular relevance to Global Positioning system(GPS). A Time domain coarse acquisition of the so called "Coarse/Acquisition (C/A)" codes used for ranging in GPS has been demonstrated on MATLAB®. Tracking has been achieved by simultaneous synchronisation of carrier frequency and code phase using a Phase Lock Loop (PLL).The acquisition and tracking has been simulated on real time samples of GPS signals.

Index Terms—PN sequences, Acquisition of PN sequences, GPS

GPS is an example of Code Division Multiple Access (CDMA) technique, in which all the 32 Satellite vehicles (SV) share the same bandwidth. One of the main applications of Spread Spectrum Communication has been in field of GPS. All the 32 SV's have a unique Pseudo Random Noise (PRN) number assigned to them which is used by the receiver to sift the required signal from the background noise. In this document,we will provide a brief review of PN theory, spread spectrum communication followed by the GPS signal structure and the general methodology for acquisition. We will finally delve into the details of our MATLAB C/A code acquisitions along with the relevant plots.

I. PN SEQUENCES

[1] A PN sequence is defined as a string of 0s and 1s with the required autocorrelation properties. These sequences are periodic in nature as opposed to the aperiodic random phenomena observed in nature¹. PN sequences are generated using the maximum-length sequence generator which are implemented in the form of a linear feedback shift register as shown in Fig 1.

A shift register of length m consists of m flip-flops (or registers) with all the registers being clocked by a single source. At each clock pulse all the bits in the shift register moves down the chain. A new bit is then feedback which depends on the particular feedback logic used. This logic generates a feedback bit, which depends

¹E.g., random phenomena like the temperature of a resistor excited with a specific voltage at its ends

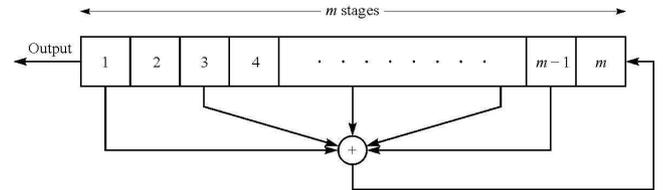


Fig. 1. Maximum Length Shift Register with m stages (SOURCE: J.G.Proakis, *Digital Communications*, 4th edition, McGraw-Hill)

on the state of the shift register. The PN sequence generated from a m stage shift register will be periodic with a period of

$$N = 2^m - 1$$

During the design process of a maximum-length sequence generator, care should be taken to ensure that the shift register will never enter the all 0s state. Such a condition will result in a *catastrophic cyclic code* (i.e., the shift register will enter a dead-lock condition).

A. Properties of PN sequences

Here we shall put down the three important properties of a PN sequence

Property 1: In each period of a PN sequence, the number of 1s is one more than the number of 0s.

Property 2: In every period of a PN sequence, one half the runs of each kind are of length one, one-fourth of length two, one-eighth are of length three. The property is called as run property.

Property 3: PN sequences have periodic and binary valued autocorrelation function. This property is called as correlation property.

By definition the autocorrelation of a PN sequence $\{p_n\}$, is given by

$$R_p(k) = \frac{1}{N} \sum_{n=1}^N p_n p_{n-k}$$

where N is the length of the PN sequence and k is the shift parameter of the autocorrelation function. The autocorrelation is binary valued as shown below

$$R_p(k) = \begin{cases} 1 & k = lN \\ -\frac{1}{N} & k \neq lN \end{cases}$$

II. SPREAD SPECTRUM COMMUNICATION

[1] This method of communication involves the spreading of the signal spectrum to make its "appearance" *noise-like*. The signal will thus be hidden and any intentional jammer would not be able to detect it. The method usually employed for spreading the signal spectrum is to multiply it with PN sequences. These PN sequences, with their noise-like structure, has a wide spectrum and hence will spread the required signal. This spreading of the signal using PN sequences has a two-fold effect; the first being the immunity against *evesdropping* which will be demonstrated next and the second being a form of multiple access technique called Code Division Multiple Access (CDMA), which is exactly how GPS operates.

For demonstrating the immunity against jamming, lets consider a signal to be transmitted $s(t)$. The transmitted signal $m(t)$ will then be the product of the signal and the PN sequence $p(t)$

$$m(t) = s(t)p(t)$$

The received signal, in the presence of a jamming signal $j(t)$ will then be

$$\begin{aligned} r(t) &= m(t) + j(t) \\ &= s(t)p(t) + j(t) \end{aligned}$$

The despreading operation at the receiver is again multiplying the received signal $r(t)$ with the PN sequence $p(t)$. Despread signal is

$$z(t) = p^2(t)s(t) + p(t)j(t)$$

If the PN sequence is in NRZ format then

$$p^2(t) = 1 \quad \forall t$$

The despread signal can then be represented as

$$z(t) = s(t) + j(t)p(t)$$

The despread signal can be viewed as a sum of two terms; the first being the signal itself and the second being the cross correlation between the jamming signal and the PN sequence. If the PN sequence is chosen to minimize the cross correlation then the effect of jamming signal is minimized. The second effect of the spreading, as already mentioned, can be inferred in this way: If two users are assigned two PN codes chosen in such a way so that the cross correlation between them is minimized, then both the users can transmit the data through the same channel using the same band. Both the users will despread the received signal with their PN sequence and detect the signal in the presence of a small cross correlation signal. This CDMA technique is used in GPS to distinguish the different GPS satellites.

The two forms of spread spectrum communication are:

- 1) Direct-sequence spread spectrum (DSSS): The data to be transmitted is first multiplied with PN sequence and then used for Binary Phase Shift Keying (BPSK)
- 2) Frequency hop spread spectrum(FHSS): The data to be transmitted is first M-ary Frequency shift keyed and then multiplied by a randomly hopping carrier.

III. GPS SIGNAL STRUCTURE

The GPS signal for Standard Positioning Service (SPS) consists of the Navigational Data stream at 50bps mixed with the C/A code, modulated using a carrier of 1575.42MHz (L-band). The C/A has a chipping rate of 1.023MHz with a code period of 1.023ms. The bits obtained after spreading the data with the PN sequence is represented in NRZ format, which is then used for BPSK modulation of a sinusoidal carrier of 1575.42MHz. So, the GPS-SPS signal for the i^{th} satellite can be represented as

$$s_{GPS,SPS}^i(\omega_c, t) = A(C_i(t) \oplus D_i(t))\cos(\omega_c, t)$$

where,

A : The carrier amplitude

$C_i(t)$: C/A code for the i^{th} satellite

$D_i(t)$: Navigational data for the i^{th} satellite

ω_c : L1 carrier with frequency of 1575.42MHz

We can readily infer that GPS is a CDMA technique employing the DSSS scheme.

GPS uses C/A code for CDMA which fall under the category of *Gold codes*. C/A code generation consists of two individual MLSRs called G1 and G2 which are used to generate the C/A codes for all the SVs. G1 and G2 registers are both 1023bit ($2^{10} - 1$) long governed by the following polynomials [2]

$$G1 = 1 + X^3 + X^{10}$$

and

$$G2 = 1 + X^2 + X^3 + X^6 + X^8 + X^9 + X^{10}$$

C/A code is generated by modulo-2 addition of G1 shift register and shifted version of G2 register outputs. The delaying of G2 register is accomplished by modulo-2 addition of two stages of the shift register². This phase

²Cyclic property of shift register is used here, which states that if you perform a modulo-2 addition on a PN sequence (with a given initial condition) and the shifted version of it, then it will result in a PN sequence which can be generated by the same MLSR with a different initial condition

shift logic will be different for all the 32 SVs thus generating 32 different C/A codes. C/A codes have the property of minimum cross correlation between two codes as this is very important for acquiring the SV. The autocorrelation peaks of C/A codes are summarised in Table 1 [3]

Code Period	No. of Shift Registers	Normalized Cross Correlation Levels	Prob. of Level
$P = 2^n - 1$	$n = \text{odd}$	$-\frac{2^{\frac{(n+1)}{2}} + 1}{P}$	0.25
		$-\frac{1}{P}$	0.5
		$-\frac{2^{\frac{(n+2)}{2}} + 1}{P}$	0.25
$P = 2^n - 1$	$n = \text{even}$	$-\frac{2^{\frac{(n+2)}{2}} + 1}{P}$	0.125
		$-\frac{1}{P}$	0.75
		$-\frac{2^{\frac{(n+2)}{2}} + 1}{P}$	0.125

Table 1: Cross Correlation properties of C/A Code

Using these formulas for the C/A code used in GPS we can get the cross correlation values as $-\frac{65}{1023}$ (probability of 0.125), $-\frac{1}{1023}$ (probability of 0.75) and $\frac{63}{1023}$ (probability of 0.125). The autocorrelation function is shown in Fig 2

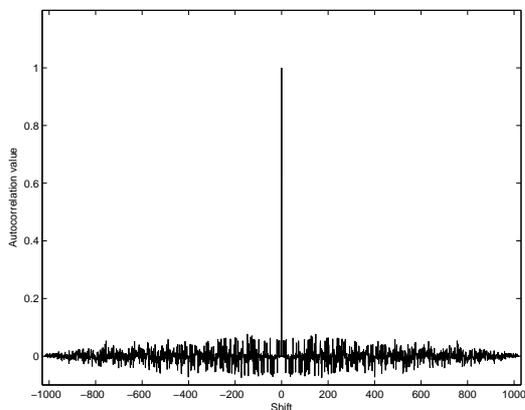


Fig. 2. Auto correlation of CA code

A. Acquisition methodologies

[3] The process of acquisition involves detecting the presence of the GPS signal. This happens in three dimensions namely, the chip shift in C/A codes (due to finite signal transit delay), Doppler shift in the BPSK carrier (due to satellite and user movement), and the PRN code number. The acquisition methodologies can be broadly classified into two types which are

- 1) Time domain correlation
- 2) Frequency domain correlation

In Time domain correlation, the input signal samples are first multiplied with locally generated CA code (sampled at the same frequency) thus despreading the signal. This despread signal is then analysed for determining the doppler frequency shift using Fast Fourier Transform (FFT). The process of multiplying the signal samples with CA code samples is called as *code stripping*. The process is illustrated in the following figures. A BPSK waveform is shown in Fig 3 along with the CA code bit.

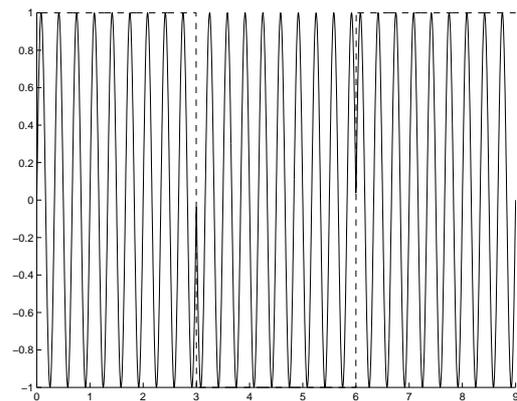


Fig. 3. A BPSK wave

Fig 4 shows three bits of a CA code which is used for stripping and Fig 5 shows the continuous waveform after code stripping.

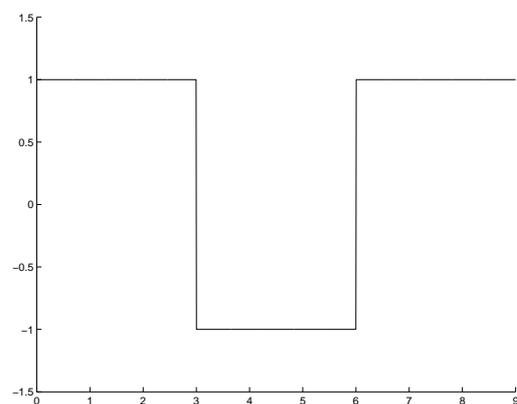


Fig. 4. CA code

Frequency domain correlation performs the circular correlation in the frequency domain by using the following property of the Discrete Fourier Transform(DFT). For $x(n)$ and $h(n)$ as complex valued we have If

$$x(n) \xleftrightarrow{DFT} X(k)$$

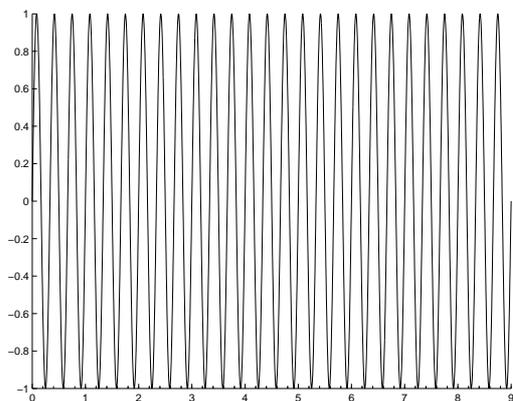


Fig. 5. Continuous waveform after code stripping

and

$$h(n) \stackrel{DFT}{\leftrightarrow} H(k)$$

then

$$r_{xh}(l) \stackrel{DFT}{\leftrightarrow} R_{xh} = X(k)H^*(k)$$

where $r_{xh}(l)$ is the unnormalized circular cross correlation sequence, defined as

$$r_{xh}(l) = \sum_{n=0}^{N-1} x(n)h^*((n-l)_N)$$

The process involves taking the FFT of the input samples and conjugating the FFT of the generated samples. These two FFT values are multiplied point to point and then the inverse FFT is taken to search for the peak. The whole process can be substantiated by considering the received waveforms and the generated waveforms assuming a noise-free environment³. Lets consider the received waveform sampled at instants of (nT_s) for the i^{th} satellite as

$$s_i(t) = CA_i(nT_s)e^{j(\omega_c + \omega_d)nT_s}$$

where,

$CA_i(nT_s)$: C/A code for the i^{th} satellite
 ω_d : Doppler frequency shift for the i^{th} satellite
 ω_c : L1 carrier with frequency of 1575.42MHz

Lets represent the generated waveform as

$$l_i(t) = CA_i(nT_s)e^{j\omega_c nT_s}$$

The ω_d term is not there as it is an unknown to be determined. After performing the circular correlation we can represent the correlation result as

$$r(nT_s) = CA_i^2(nT_s)e^{j(\omega_c + \omega_d)nT_s} \times e^{-j\omega_c nT_s} = e^{j\omega_d nT_s}$$

³The argument can then be extended to the noisy environment by considering the degradation in the autocorrelation peaks

A FFT operation on this will essentially have a peak at a point corresponding to ω_d thus knowing the doppler shift. A distinct peak is obtained in the FFT operation only if the correct CA code was used for code stripping. Thus the process will reveal both the correct chip shift and the Doppler uncertainty.

IV. ACQUISITION SIMULATION

The simulation was carried out on MATLAB® on real time data collected using a GPS *SignalTap*^{TM4}. The data for this simulation was obtained with an IF as 15.42 MHz and sampling frequency as 4.7 MHz. The whole simulation consists of 4 different files doing the following

- 1) CA code generation for a particular SV
- 2) Sampling the CA code with the sampling frequency same as that used for sampling the input
- 3) Frequency domain correlation process
- 4) Manager function which sequentially tries to acquire all the 32 SVs and declares acquisition based on some criterion

A. Coherent and Noncoherent averaging

These two techniques of signal averaging are frequently used in GPS acquisition methodologies.

[4]Coherent averaging involves averaging of multiple sets of signal (corrupted by noise) ensuring that the phase of the signal in each set to be identical. The gain obtained by such an averaging is given in terms of the averaging factor N as

$$SNR_{coh\,gain}(dB) = 20\log_{10}(\sqrt{N})$$

Noncoherent averaging involves averaging of multiple sets of signal without any timing constraints. This type of averaging is usually performed in frequency domain. The FFT output of a set of data are repeatedly added so that the signal peak gets distinct. The gain of Non coherent averaging is also the same as obtained by Coherent averaging.

B. Acquisition algorithm

The acquisition algorithm is identical to the Frequency domain correlation outlined in the previous section. The steps followed are

- 1) Generate the waveform in the form of $CA_i(nT_s)e^{j[(\omega_{IF} - \omega_s) + (k\Delta\omega)]nT_s}$

⁴GPS SignalTap is a GPS downconversion and sampling device developed by Accord Software & Systems, Bangalore. This device can be configured for various Intermediate frequencies(IF) and sampling frequencies

- 2) Perform the FFT operation on this waveform and take the conjugate of it
- 3) Perform the FFT operation on the input waveform and multiply with the output from step 2
- 4) Take the inverse FFT and detect the peak among those values and mean of them and determine (peak/mean)
- 5) Declare the SV as acquired if the (peak/mean) exceeds 3.5^5
- 6) Step 1 through 5 is repeated for different values of k until a SV is acquired or the configured Doppler search range is exhausted
- 7) Step 1 through 6 is repeated for all i from 1 to 32

In the simulation, the duration of Coherent and Noncoherent averaging is configurable. The doppler increment $\Delta\omega$ and the doppler range is also configurable. The doppler range is configured as the upper and lower limits of the doppler search limits. The upper and lower doppler search limits are given in terms of f_{IF} , f_S , and the total doppler search range in Hz Δf as

$$f_{HI} = (f_{IF} - f_S) + \frac{\Delta f}{2}$$

and

$$f_{LO} = (f_{IF} - f_S) - \frac{\Delta f}{2}$$

C. Simulation results

Based on the acquisition criterion mentioned in the acquisition algorithm with Coherent averaging done for 4ms and Noncoherent averaging for 16ms, 5 SVs were acquired namely 1, 7, 14, 21, 25. Fig 6 - 10 show the correlation values obtained for these SVs. As we

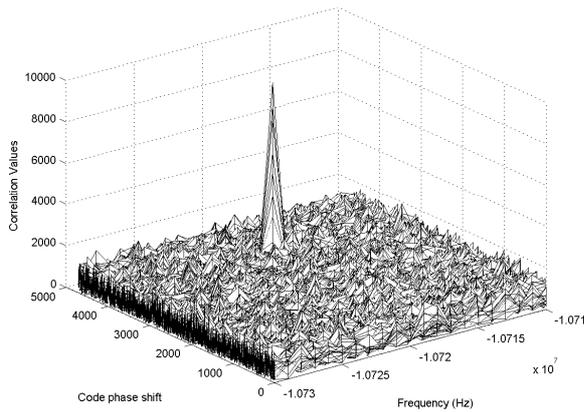


Fig. 6. Acquisition of SV 1

can observe from the plots, that the autocorrelation

⁵The value was decided based on initial simulation runs

peaks starts degrading with decreasing signal strengths. These peaks will be very close to the *noise floor* and their detection needs a different algorithm⁶. The cross correlation between any 2 SVs, as shown in Fig 11, resemble noise and thus will provide immunity against false acquisition.

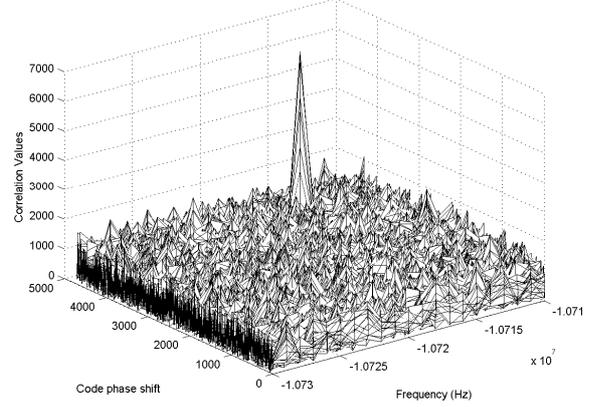


Fig. 7. Acquisition of SV 7

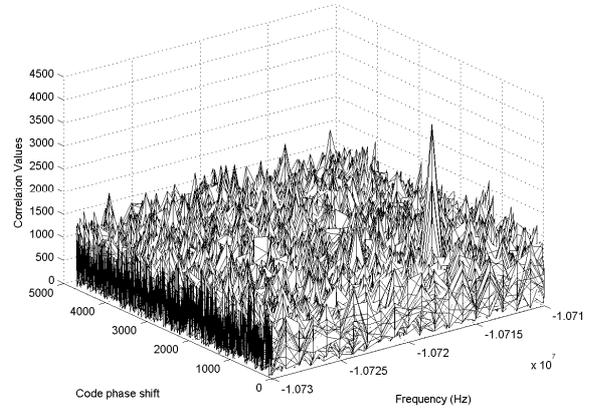


Fig. 8. Acquisition of SV 14

⁶The decision criterion of $\frac{peak}{mean}$ gives good results only at high signal strengths

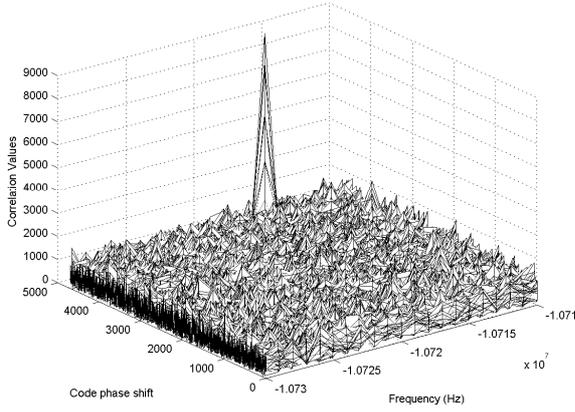


Fig. 9. Acquisition of SV 21

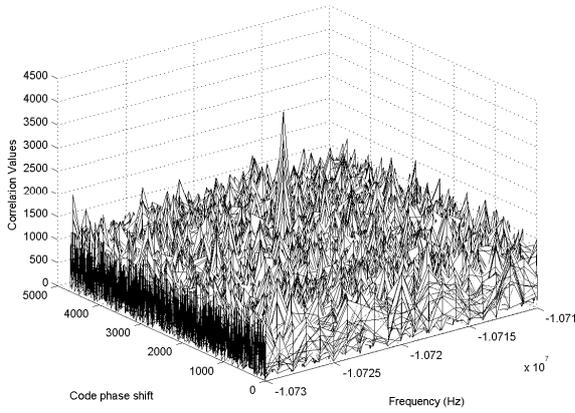


Fig. 10. Acquisition of SV 25

V. TRACKING

Tracking involves the fine estimation of Carrier frequency shift and Code phase using locally generated signals. This is usually done with the aid of a PLL. We shall provide a very brief overview of PLL and then proceed to the details about the simulation of tracking process.

A. Basics of PLL

PLLs are used to obtain a filtered version of the input signal. It is essentially made up of a voltage-controlled oscillator (VCO), a product modulator, loop filter and a phase comparator logic. PLL in time domain is shown in Fig 12 and the corresponding s - domain (Laplace domain) is shown in Fig 13 (next page) [3]

Basic operation of the PLL is to align or synchronise the locally generated carrier to the incoming signal. With reference to the Fig 13 (next page), following equations can be directly written for the various intermediate signals [3]

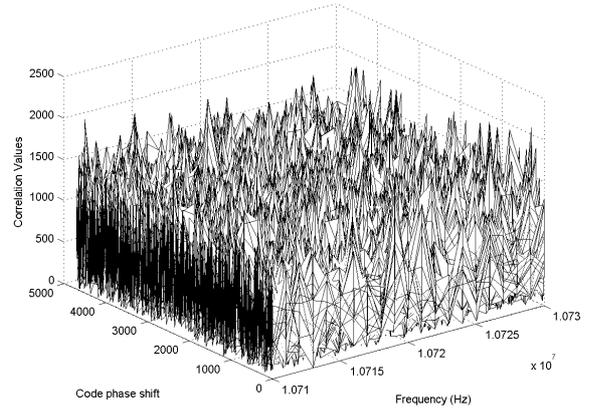


Fig. 11. Cross correlation between 2 PRN sequences

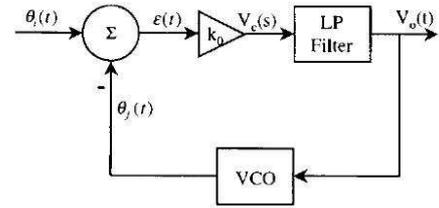


Fig. 12. Block diagram of a PLL in time domain [3]

$$\theta_f(s) = V_o(s) \frac{k_1}{s}$$

and

$$V_c(s) = k_0 \epsilon(s) = k_0 (\theta_i(s) - \theta_f(s))$$

$$V_o(s) = F(s) V_c(s)$$

Using these three equations the following equations can be written

$$\epsilon(s) = (\theta_i(s) - \theta_f(s)) = \frac{V_c(s)}{k_0} = \frac{V_o(s)}{k_0 F(s)} = \frac{s \theta_f(s)}{k_0 k_1 F(s)}$$

$$\theta_i(s) = \theta_f(s) \left(1 + \frac{s}{k_0 k_1 F(s)} \right)$$

The transfer function $H(s)$ is

$$H(s) \equiv \frac{\theta_f(s)}{\theta_i(s)} = \frac{k_0 k_1 F(s)}{s + k_0 k_1 F(s)}$$

Error transfer function $H_e(s)$ is defined as

$$H_e(s) = \frac{\theta_i(s) - \theta_f(s)}{\theta_i(s)} = \frac{s}{s + k_0 k_1 F(s)}$$

and Equivalent noise bandwidth is

$$B_n = \int_0^\infty |H(j\omega)|^2 d\omega$$

fed to the VCO after filtering it through the Loop filter.

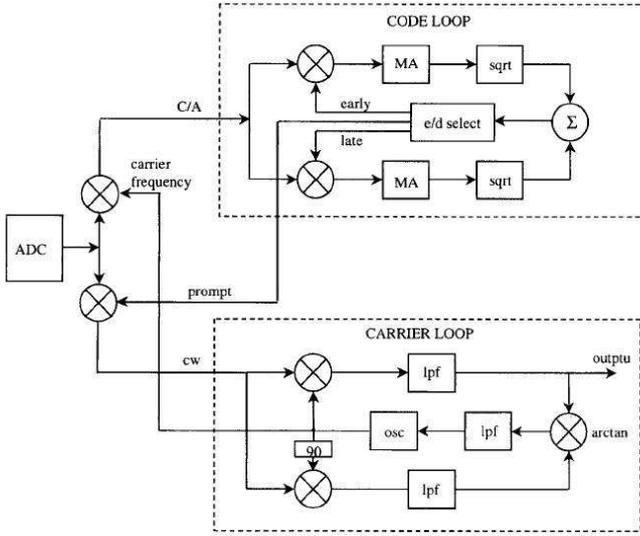


Fig. 16. GPS tracking loop [3]

The final combined code and carrier tracking loops in GPS is formed by combining the two loops discussed above. The final tracking loop is as shown in Fig 16 [3]

C. Tracking simulation

This concluding section details the simulation of the tracking process. The design procedure is

- 1) The bandwidth of the code tracking loop filter is kept narrower than that of the carrier tracking filter as the code tracking operates for longer. Here we have selected a loop bandwidth of 50Hz.
- 2) set ζ as 0.7
- 3) The value of natural frequency is calculated using

$$B_n = \frac{\omega_n}{2} \left(\zeta + \frac{1}{4\zeta} \right)$$

- 4) Choose the loop gain k_0k_1 as 400π and calculate the filter constants C_1 and C_2 as [3]

$$C_1 = \frac{8\zeta\omega_n t_s}{k_0k_1(4 + 4\zeta\omega_n t_s + (\omega_n t_s)^2)}$$

$$C_2 = \frac{4(\zeta\omega_n t_s)^2}{k_0k_1(4 + 4\zeta\omega_n t_s + (\omega_n t_s)^2)}$$

D. Tracking Simulation

The tracking is implemented as a second order PLL for Code and carrier tracking. The loop bandwidth of the PLL is 50Hz. This essentially means that the pull-in (or acquisition) range of the PLL is also 50Hz. The Acquisition, however has a frequency resolution of

1kHz⁷. To ensure that the PLL will pull in the carrier, a fine frequency estimation is performed around the coarse frequency estimate. This process of frequency estimation improves the resolution to 50Hz. The phase discriminator used for the carrier lock loop is an arctangent function and $\frac{EARLY}{LATE}$ for the code lock loop.

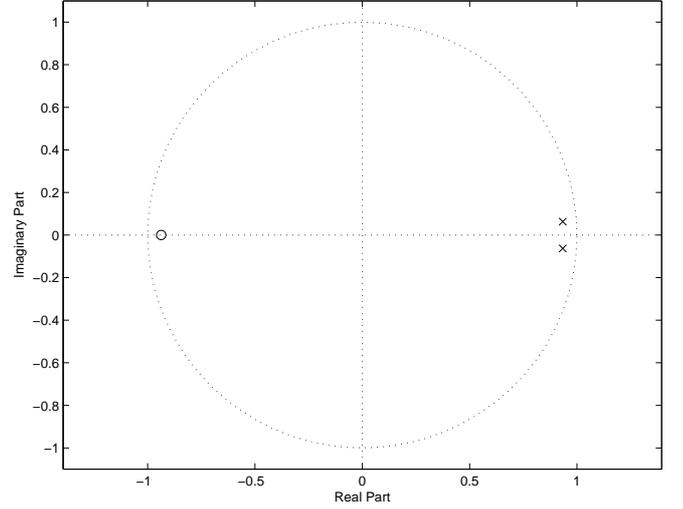


Fig. 17. Pole zero plot for the PLL loop filter with $\zeta = 0.7$

Putting the values for C_1 and C_2 , we have the z-domain transfer function $H(z)$ as

$$H(z) = \frac{k_0k_1(C_1 + C_2)z^{-1} - k_0k_1C_1z^{-2}}{1 + (k_0k_1(C_1 + C_2) - 2)z^{-1} + (1 - k_0k_1C_1)z^{-2}}$$

The corresponding pole-zero plot for $\zeta = 0.7$ is as shown in Fig 17.

As we readily infer from the pole-zero plot, the system is a second order, type 0 system. A type 0 system can track only step changes and as such the PLL used can track constant phase changes. The program was run to track the five acquired SVs (1,7,14,21, and 25). The Filtered phase error for SV 25 is shown in Fig 18.

As we can see from Fig 18, the phase comparator output shows random variations over 40 samples. This random variations are over the duration during which the PLL is still locking to the input carrier. The phase detector output after *lock-in*, then periodically varies between 0 to 2π radians.

⁷This is programmed as the increment in the acquisition algorithm

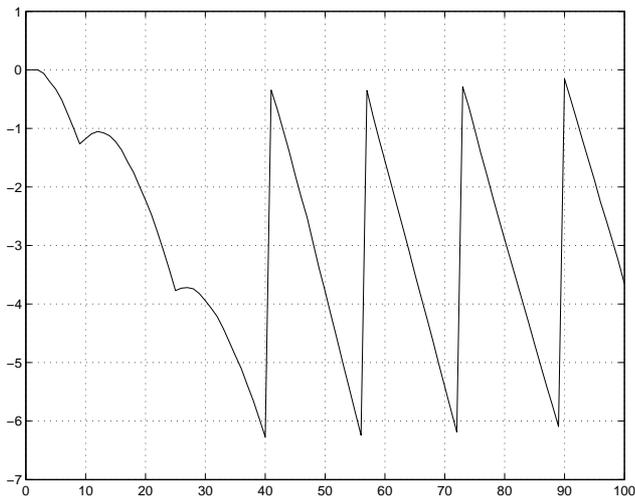


Fig. 18. Filtered Phase Error of Carrier Lock Loop for SV 25

REFERENCES

- [1] S. Haykin, *Digital Communications*. John Wiley & Sons Inc., 2002.
- [2] *GPS Interface Control document*, ARINC Research Corporation, 1993.
- [3] J. Bao and Y. Tsui, *Fundamentals of Global Positioning System Receivers - A Software Approach*. John Wiley & Sons Inc., 2005.
- [4] R. G. Lyons, *Understanding Digital Signal Processing*. Pearson Education Asia, 1997.
- [5] John.G.Proakis, *Digital Communications*. McGraw-Hill, 2001.