Research Paper

Code and Carrier Tracking Loops for GPS C/A Code

Rishija Misra[1],* and Shubham Palod[2]

[1]Indian Institute of Space Science and Technology, Valiyamala, Thiruvananthapuram, ISRO Satellite Centre, Bangalore
[2]Indian Institute of Space Science and Technology,Valiyamala, Thiruvananthapuram, Semi-Conductor Laboratory, Chandigarh

*Corresponding author, e-mail: (rishijamisra5@gmail.com)

Abstract: *The DSSS modulation is studied and its application in satellite based navigation system is presented. Brief details of GPS signal structure is presented. The need of code and carrier tracking loops in GPS is highlighted. Various architectures for code and carrier tracking loops are also presented. The derivations for transfer function of Costas loop for carrier tracking and Delay lock loop for code tracking is presented. The loop coefficients are determined by simulating various conditions. The loop functionality is tested with simulated test signal generated from AWR system simulation tool. Further the loop is tested.*

Keywords: BPSK, PLL, Costas loop, demodulation, Code, Carrier Tracking Loops, GPS, modulation, satellite, navigation systems.

Introduction

Abbreiations

A/D- Analog to Digital Converter
BPSK-Binary Phase Shift Keying
C/A- Coarse Acquisition
CDMA- Code Division Multiple Access
DLL-Delay Lock Loop
DSSS- Direct Sequence Spread Spectrum
FDMA- Frequency Division Multiple Access
FLL- Frequency Lock Loop
GPS- Global Positioning Satellite
IF- Intermediate Frequency
INSAT-Indian National Satellite

IRNSS - Indian Regional Navigation Satellite System
IRS -Indian Remote Sensing
LNA- Low Noise Amplifiers
MAI- Multiple Access Interference
MEO- Middle Earth Orbit
MLS- Maximum Length Sequences
NCO- Numerically Controlled Oscillator
PLL-Phase Lock Loop
PPS- Precise Positioning Service
PRN- Pseudo-Random Noise
RF- Radio Frequency
SPS- Standard Positioning Service
TDMA-Time Division Multiple Access
VCO-Voltage Controlled Oscillator

Satellite based navigation systems provide real time position, velocity and timing information of receiver (Borre et al. 2007). The most commonly used navigation system in today's world is Global Positioning System (GPS) (Borre et al. 2007). Other existing/developing navigation systems are USSR's GLONASS, European GALILEO, China's COMPASS and India's IRNSS (Indian Regional Navigation Satellite System).
All of these navigation systems are similar in their signal structure by employing DSSS signals for ranging, and CDMA for multiple access (except GLONASS which uses FDMA). The navigation data from the satellites comprises of the orbital information of the satellite and precise timings. GPS system provides two categories of services: Precise positioning service used for military purposes and Standard positioning service used by general public. CDMA is a spread spectrum multiple access technique. The direct sequence spread spectrum technique is one which spreads the bandwidth of the data uniformly for the same transmitted power. For this PN codes are used. In DSSS, a locally generated PN code runs at a much higher rate (chip rate) than the data to be transmitted (data rate). Data for transmission is spreaded with the PN code. The figure-1 shows how spread spectrum signal is generated. The data signal with pulse duration of T_b is XOR added with the code signal with pulse duration of T_c. Therefore, the bandwidth of the data signal is $1 / T_b$ and the bandwidth of the spread spectrum signal is $1/ T_c$. Since T_c is much smaller than T_b, the bandwidth of the spread spectrum signal is much larger than the bandwidth of the original signal. The ratio T_b / T_c is called spreading factor or processing gain which is an important parameter in DSSS system design.
Choosing the codes used to modulate the signal is very important in the performance of CDMA systems. The best performance will occur when there is good separation between the signal of a desired user and the signals of other users. The separation of the signals is made by correlating the received signal with the locally generated code of the desired user. If the signal matches the desired user's code then the correlation function will be high and the system can extract that signal. If the desired user's code has nothing in common with the signal the correlation should be as close to zero as possible (thus eliminating the signal). If the code is correlated with the signal at any time offset other than zero, the correlation should be as close to zero as possible. This is referred to as auto-correlation.
This property forms the basis for the ranging of the satellites by user receiver in GPS, where the receiver correlates the user generated code with incoming signal. It adjusts the code phase to get a correlation output. The code phase offset gives an estimate of the pseudorange of that particular satellite.

Code division multiple access is a scheme in which a number of users can occupy all of the channel bandwidth all of the time. By contrast, time division multiple access (TDMA) divides access by time, while frequency-division multiple access (FDMA) divides it by frequency. The CDMA is implemented by spreading user signal with different PN codes. The selected PN codes satisfy certain orthogonality properties to maximize the system .
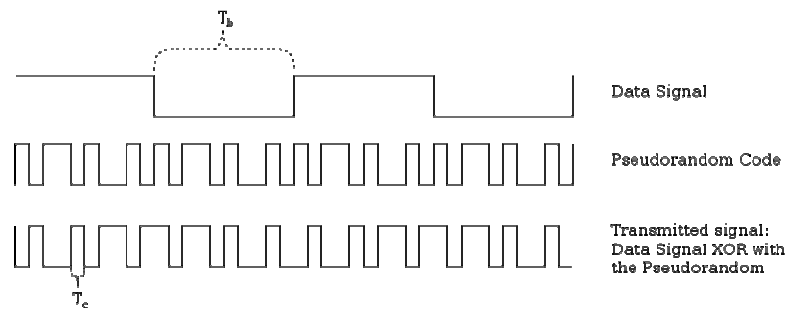
Figure-1 DSSS signal generation

Satellite based navigation systems have three segments: the Space Segment, Control Segment and User Segment. In space segment, satellites are placed in a constellation at MEO/GEO, which transmit navigation data and ranging signal. In control segment the ground stations synchronize the satellites clock and update the navigation message. In user segment, the receiver is used to demodulate and track the signal for obtaining the position, velocity and time solution.

Pseudo Noise Code (PN Code)

A pseudo noise code (PN code) is one that has a spectrum similar to a random sequence of bits but is deterministically generated. The most commonly used sequences in direct-sequence spread spectrum systems are maximum length sequences (MLS). The N-order MLS sequence is periodic with period 2^N-1. Different MLS sequences of the same order can exist. They can be easily obtained by using a shift-register with different feedback taps.
Properties of maximum length sequences are Balance property, Run property, Auto-Correlation property and Modulo 2 addition.
The GPS signals use Gold codes. Gold codes are generated by modulo-2 addition of the MLS. Contrary to MLS, they have three valued correlation function. [1]

2.3 Modulation of GPS Signal

GPS signals, which are CDMA signals, are modulated using BPSK modulation. Two radio frequencies of L band are used to transmit the GPS signals. Carrier frequencies L1 and L2 are derived from the same carrier frequency, $f_0 = 10.23$MHz. The navigation data at 50 bps is spreaded by multiplying it with the uniquely generated PN code (C/A code). Now this spreaded data is polarised and multiplied with the carrier wave to produce BPSK signal to form the DS-CDMA signal, which is transmitted as the GPS signal. The C/A code is a unique spreading sequence of 1023 chips, with 1.023 Mcps chip rate giving

a period of 1ms. There is another unique spreading sequence known as the encrypted precision code (P). The P code is a longer code ($\approx$2.35· 10^4 chips) with a chipping rate of 10.23 MHz. It is more precise but is unavailable to unauthorized users. [2]

2.4 Demodulation of GPS Signal

For demodulation of DSSS signal the basic steps are *Acquisition,* and *Tracking.*

2.4.1 Acqusition

The purpose of acquisition is to identify all satellites visible to the user. If a satellite is visible, the acquisition must determine the following two properties of the signal:

***Frequency*:** The frequency of the signal from a specific satellite can differ from its nominal value. In case of downconversion, the nominal frequency of the GPS signal on receiver corresponds to the IF. However, the signals are affected by the relative motion of the satellite and receiver, causing Doppler Effect. The Doppler frequency shift can be as high as 10 kHz.

***Code Phase*:** The code phase denotes the point in the current data block where the C/A code begins. If a data block of 1ms is examined, the data include an entire C/A code and thus one beginning of C/A code.
In correlation, a received signal is multiplied with a locally stored replica of the signal, and the resulting output is integrated. In correlation when a perfectly synchronised code is multiplied with the incoming signal the correlation output is maximised, as shown in figure-2.
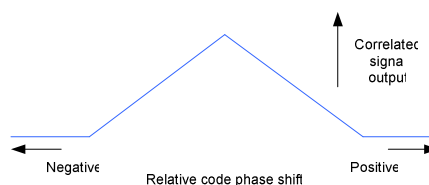

Figure-2 Autocorrelation for binary code sequence in a region of perfect synchronisation

The improved real time accuracy and clock timings are being exploited for many other applications besides the air, marine and land navigations. Some of the applications are: Surveying, Spacecraft orbit determination, scientific research such as ionosphere study, Monitoring agriculture, Time synchronization and Defense application.

2.4.2 Tracking

The main purpose of tracking is to refine the coarse values of code phase and frequency and to keep track of these as the signal properties change over time. There are two tracking system required: Code and Carrier tracking. After a spread spectrum receiver has synchronised with a received signal it must continue to operate in such a way that it remains locked with its code reference exactly tracking the coded incoming signal; that is once the code sequences are matched in time the receiver must cause its own code chip rate to match the incoming code chip rate as precisely as possible. In carrier tracking we can track carrier frequency using FLL or PLL. In code tracking we can use either "Tau-

Dither" or "Delay Lock" tracking loops. The accuracy of the final value of the code phase is connected to the accuracy of the pseudorange computed later on. Tracking runs continuously to follow the changes in frequency as a function of time. If the receiver loses track of a satellite, a new acquisition must be performed for that particular satellite.

2.4.3 Demodulation

The net signal transmitted from GPS satellite can be represented as:

$$s^k(t) = \sqrt{2P_C}\, C^k(t) D^k(t) \cos(2\pi f_{L1} t) + \sqrt{2P_{PL1}}\, P^k(t) D^k(t) \sin(2\pi f_{L1} t) + \sqrt{2P_{PL2}}\, P^k(t) D^k(t) \sin(2\pi f_{L2} t),$$

where $D^k(t)$ is the navigation data sequence, f_{L1} and f_{L2} are the carrier frequencies of L1 and L2 of the signal transmitted from satellite "k" with powers P_C, P_{PL1} and P_{PL2} for C/A code, $C^k(t)$ is the C/A code sequence and $P^k(t)$ is the P(Y) code sequence. A basic demodulation scheme is shown in figure-3. After filtering and downconversion, we get:

$$s^k(t) = \sqrt{2P_C}\, C^k(t) D^k(t) \cos(\omega_{IF} t) + \sqrt{2P_{PL1}}\, P^k(t) D^k(t) \sin(\omega_{IF} t),$$

where ω_{IF} is the IF to which the front end has downconverted the carrier frequency. Since we are interested in receiving C/A signal the P code signal can be neglected from the above equation.

$$s^k(t) = \sqrt{2P_C}\, C^k(t) D^k(t) \cos(\omega_{IF} t),$$
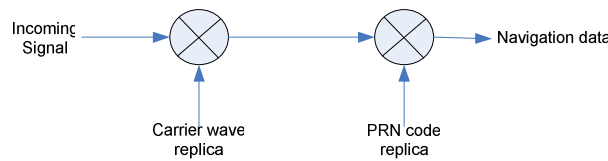
Figure-3 Basic demodulation scheme to demodulate the navigation data

With having an exact carrier wave replica from tracking, signal is converted to baseband by multiplying with carrier and filtering.

$$s^k(t) = \frac{\sqrt{2P_C}}{2} C^k(t) D^k(t)$$

Code is then removed from the signal by multiplying it with exact code replica produced by code tracking loop.

$$s^k(t) = \frac{\sqrt{2P_C}}{2} D^k(t)$$

The feedback loop to produce the carrier replica is referred to as the carrier tracking loop and the feedback loop to produce the exact code replica is referred to as the code tracking loop. [1]

3. Code and Carrier Tracking for GPS C/A Code

The tracking contains two parts, code tracking and carrier frequency/phase tracking:

Code tracking: The code tracking is most often implemented as a delay lock loop (DLL) where three local codes (replicas) are generated and correlated with the incoming signal. These three replicas are referred to as the early, prompt and late replica, respectively. The three codes are often separated by a half chip length.

Carrier frequency/phase tracking: The other part of the tracking is the carrier wave tracking. This tracking can be done in two ways: either by tracking the phase of the signal or by tracking the frequency. [2]

3.1 Carrier Tracking

An exact carrier wave replica is generated for the data demodulation using a PLL or Frequency Lock Loop (FLL). The first two multiplications, shown in Figure-5, wipe off the carrier and the PRN code of the input signal. The prompt output of the early-late code tracking loop is used to wipe off the PRN code. The frequency of the local carrier wave generated is adjusted as per the feedback given by the change in the phase error. The PLL implemented for the carrier tracking has a second order transfer function.

3.1.1 Second Order PLL

The analytic linear phase lock loop (PLL) model is an extremely powerful tool to predict the performance of both the tracking loops. Hence we use a linearised digital second order PLL model which contains a First Order Filter and Voltage Controlled Oscillator (VCO).

The transfer function of the analog loop filter and VCO are respectively,

$$F(s) = \frac{1}{s} \frac{\tau_2 s + 1}{\tau_1}$$
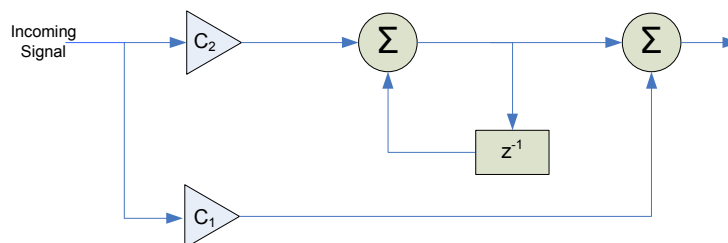
$$N(s) = \frac{K_0}{s}$$

where K_0 is the VCO gain.

The transfer function of linearised analog PLL is

$$H(s) = \frac{K_d F(s) N(s)}{1 + K_d F(s) N(s)};$$

$$H(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where K_d is the gain of the phase discriminator, the natural frequency $\omega_n = \sqrt{(k_0 k_d)/\tau_1}$, and the damping ratio $\zeta = (\tau_2 \omega_n)/2$. Figure-4 shows a Second order PLL.

Figure-4 Second order phase lock loop filter

By applying bilinear transformation we convert the analog versions of the transfer function to digital form.

$$s = \frac{2}{t_s}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

We get,

$$F(z) = \frac{\dfrac{2\tau_2}{t_s}\left(\dfrac{1-z^{-1}}{1+z^{-1}}\right)+1}{\dfrac{2\tau_1}{t_s}\left(\dfrac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$\Rightarrow F(z) = \frac{2\tau_2 + 2\tau_2 z^{-1} + t_s + t_s z^{-1}}{2\tau_1 - 2\tau_1 z^{-1}}$$

$$\Rightarrow F(z) = \frac{\left(\dfrac{2\tau_2 + t_s}{2\tau_1}\right) - \dfrac{(2\tau_2 - t_s)z^{-1}}{2\tau_1}}{1 - z^{-1}}$$

$$\Rightarrow F(z) = \frac{(C_1 + C_2) - C_1 z^{-1}}{1 - z^{-1}}$$

$$N(z) = \frac{K_0 z^{-1}}{1 - z^{-1}}$$

where $C_1 = (\tau_2/\tau_1) - (t_s/(2\tau_1))$, $C_2 = (t_s/\tau_1)$, and t_s is the sampling interval.

F(z) and N(z) are transfer function of the digitised PLL filter and Numerically controlled oscillator (NCO).

The natural frequency can be found as:

$$\omega_n = \frac{8\zeta B_L}{4\zeta^2 + 1},$$

where B_L is the noise bandwidth in the loop. [3]

3.1.2 Costas Loop

The problem with using an ordinary PLL is that it is sensitive to 180° phase shifts. Due to navigation bit transitions, a PLL used in a GPS receiver has to be insensitive to 180° phase shifts.
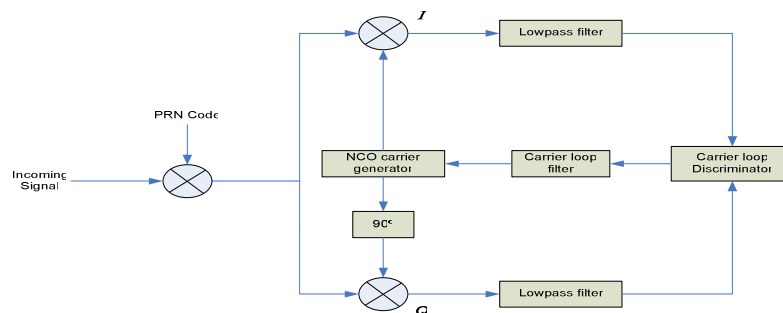
Figure-5 Block diagram of Costas loop used for carrier tracking

The Costas loop (Figure-5) contains two multiplications. The first multiplication is the input signal and the local carrier wave and the second multiplication is between the 90° phase shifted carrier wave and the input signal. The goal of Costas loop is to try to keep all energy in the I (in-phase) arm.

The multiplications in the I and Q arm provides the following results respectively:

$$D^k(n)\cos(\omega_{IF}n)\cos(\omega_{IF}n+\phi) = \frac{1}{2}D^k(n)\cos(\phi) + \frac{1}{2}D^k(n)\cos(2\omega_{IF}n+\phi)$$

$$D^k(n)\cos(\omega_{IF}n)\sin(\omega_{IF}n+\phi) = \frac{1}{2}D^k(n)\sin(\phi) + \frac{1}{2}D^k(n)\sin(2\omega_{IF}n+\phi)$$

If the two signals are lowpass filtered after the multiplication, the two terms with the double intermediate frequency are eliminated and the following two signals remain:

$$I^k = \frac{1}{2}D^k(n)\cos(\phi)$$

$$Q^k = \frac{1}{2}D^k(n)\sin(\phi)$$

The phase error of the local replica can be found as:

$$\frac{Q^k}{I^k} = \frac{\frac{1}{2}D^k(n)\sin(\phi)}{\frac{1}{2}D^k(n)\cos(\phi)} = \tan(\phi)$$

$$\phi = \tan^{-1}\left(\frac{Q^k}{I^k}\right)$$

Thus it can be seen that the phase error can be minimised when the correlation in the quadrature-phase arm is zero and the correlation value in the in-phase arm is maximum. This phase discriminator output is fed to the local oscillator so as to keep all the energy in the in-phase I arm. The phase discriminator outputs are zero when the real phase error is 0 and ±180°. This is why Costas loop is insensitive to the 180° phase shifts in case of a navigation bit transition. [2]

## 3.2 Code Tracking

To obtain a perfectly aligned replica of the code, code tracking loop is implemented. The goal of the code tracking loop is to keep track of the code phase of a specific code in the signal. The code tracking loop in the GPS receiver is a delay lock loop (DLL) called an early-late tracking loop. After multiplying the incoming signal with a perfectly aligned local replica of the carrier wave, the signal is multiplied with three code replicas, namely early, prompt and late, with a spacing of ½ chip. After the second multiplication, the three outputs are integrated and dumped. The output of these integrations is numerical value indicating how much the specific code replica correlates with the incoming signal. The prompt replica of the code has a phase shift obtained from the acquisition. The early and late have additional shift of -½ and +½ chip from the prompt. As explained in figure-6. The DLL with three correlators is optimal when the local carrier wave is locked in phase and frequency, which is the case of coherent tracking loop but when there is a phase error on the local carrier wave, the signal will be more noisy, making it more difficult for the DLL to keep lock on the code. So instead a non coherent DLL is designed such that it is independent of the phase of the local carrier wave. If the local carrier wave is in phase with the input signal all the energy will be in the in-phase arm but if it drifts compare to

the input signal the energy will switch between the in-phase and quadrature arm, thus giving rise to six correlators, three (I_E, I_P, I_L) in the I arm and rest (Q_E, Q_P, Q_L) in the Q arm as shown in figure-7.
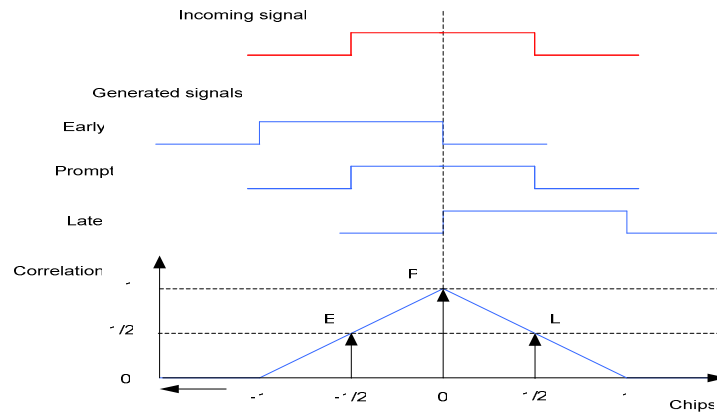
Figure-6 A perfectly tuned code loop with highest correlation at prompt output
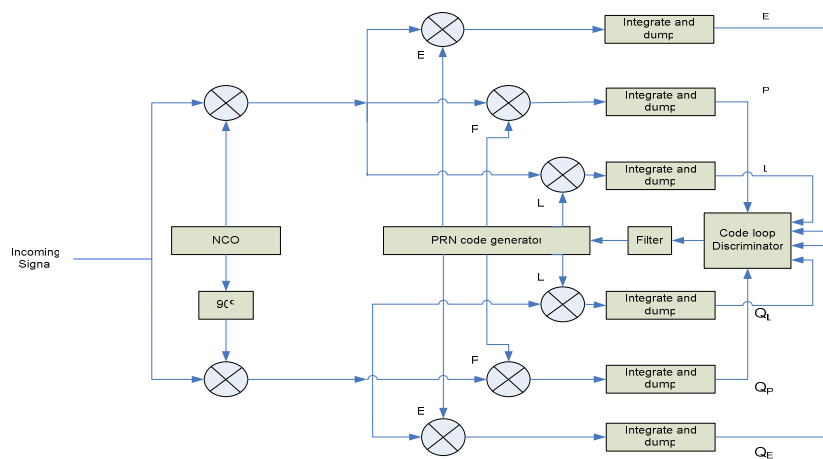
Figure-7 DLL block diagram with six correlators

Here we need to provide maximum correlation in the prompt arm of the code loop.
Some common DLL discriminators are:
Coherent: I_E-I_L : Requires a good tracking loop for optimal functionality.

Noncoherent:

➢ $(I_E{}^2 + Q_E{}^2) - (I_L{}^2 + Q_L{}^2)$: Early minus Late power, response is nearly the same as the coherent discriminator.

➢ $\dfrac{(I_E{}^2 + Q_E{}^2) - (I_L{}^2 + Q_L{}^2)}{(I_E{}^2 + Q_E{}^2) + (I_L{}^2 + Q_L{}^2)}$: Normalised Early minus Late power, helps keep track

in noisy signals when the chip error is larger than ½ chip.

➢ $I_P(I_E-I_L)+Q_P(Q_E-Q_L)$: Dot product, it is the only DLL discriminator that uses all six correlators.
We implemented Normalised Early minus Late power discriminator.[2]

3.3 Combined Code and Carrier Tracking Loop

We have implemented both Code and Carrier tracking loops in a single tracking loop as described in the following block diagram (figure-13):
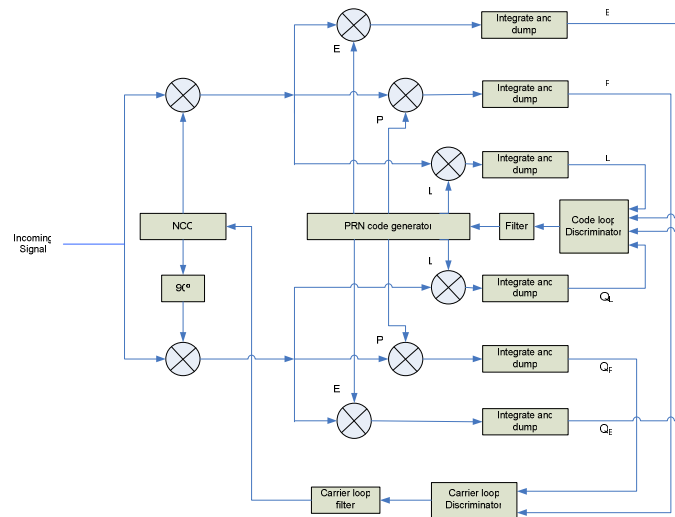
Figure-8 Block diagram of a complete tracking channel on a GPS receiver

In the combined code and carrier tracking loop the carrier tracking, Costas loop (PLL), is followed by code tracking, early-late tracking loop (DLL). Here the early and late outputs of the in-phase and quadrature arm are used for the code loop discriminator while the prompt outputs of both In-phase and quadrature arm are used in the carrier loop discriminator. The prompt output of the In-phase arm also provides us the navigation data.

The natural measurement which is being made by satellite receiver is pseudorange measurement. This is done in tracking loops inside the GPS receiver. The study of these loops is the aim of our project work.

Our objectives are to: (i) learn the spread spectrum techniques for modulation and demodulation of GPS signals, (ii) get a conceptual idea of code and carrier tracking loops and (iii) implement the code and carrier tracking loops for GPS C/A code in MATLAB.

Methods:

The combined code and carrier tracking loop is implemented in MATLAB as a single function. While implementing in software many problems were encountered. Most important of them were:

Neglecting remainder phase for code and carrier: GPS signal is a continuous signal, so while digitizing it, the remainder phase of the signal should be kept in mind as the number of points in use may not make the phase zero at the end.

Using inaccurate Doppler estimate: The initial Doppler estimate was offset by several hertz, and the loop was not able to lock to correct frequency.

Selecting wrong loop parameters and coefficients: For getting the loop parameters, firstly an analytical process was being carried out which gave us some values but the results were not satisfactory with them. Thus afterwards new parameters were calculated experimentally and also their effect on the loop was studied.

A simple logical sequence of tracking loop is shown in Figure-9. We started with initializing all the required parameters. Then we multiplied two orthogonal components of carrier signal (0 and 90º phase shifts) with 1ms of the signal. This gives us two channels – I and Q. The I and Q channels, both are multiplied simultaneously with the early, late and prompt channels of the PN code generated earlier. Now, we get six correlators, out of which the two prompt outputs are used for the carrier loop discriminator which updates the carrier frequency and remainder phase. This is again fed back to the carrier signal generator for obtaining carrier wave with the updated carrier frequency and phase. While the early and late correlators are used in the code loop discriminator. This discriminator is implemented to update the code frequency and the residual phase. Similar to the carrier signal, these two parameters are used to generate the PN code.

The correlator prompt output from the in-phase arm is also plotted. This process is carried out for 1ms and then a new set of 1ms of signal is used to again find out the correlator prompt output in the in-phase arm
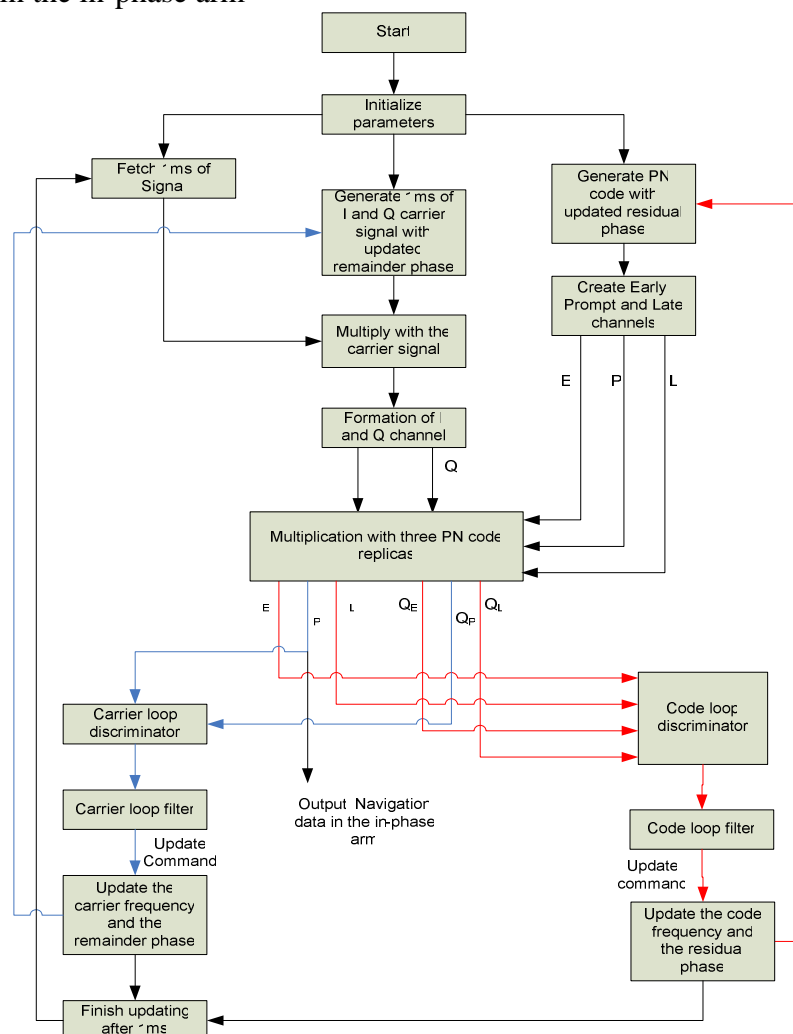
Figure-9 A systematic approach to a complete tracking loop
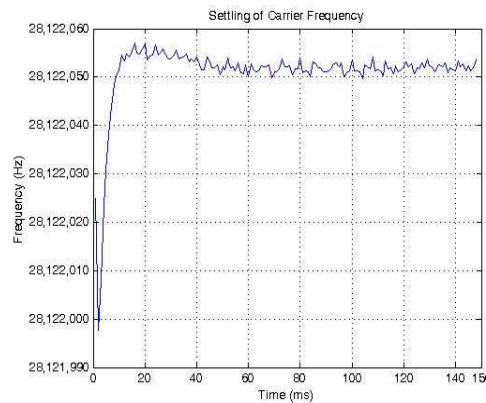
Calculation of Loop Parameters

a) Carrier frequency:

For a given noise bandwidth (B_L) and damping factor (ζ), we calculate the natural frequency (ω_n). We know the values of VCO gain (K_o) and the gain of the phase discriminator (K_d). Using these parameters, we find the loop coefficients C_1 and C_2.

With $K_o K_d = 0.25$ it was found out experimentally that $B_L = 25$Hz and $\zeta = 0.707$ best suits our loop which yields us the loop coefficients as:

$\omega_n = 47.1428$ Hz, $C_1 = 262.1949$ $C_2 = 8.8899$.

Carrier frequency output

The output of a carrier tracking loop based on PLL with above mentioned specifications for a test signal comes as in figure-10:

Figure-10 Settling of Carrier Frequency

b) Code frequency:

As we know that code loop is implemented using a DLL, which is a linear PLL. All the calculations are same as that of carrier loop parameters. With $B_L = 2$Hz, $\zeta = 0.707$ and $K_o K_d = 1.0$, the code loop parameters implemented have the following values:

$\omega_n = 3.7714$, $C_1 = 5.3257$, $C_2 = 0.0142$

Code frequency output

The output of a code tracking loop based on Early-Late DLL with above mentioned specifications for a test signal comes as:
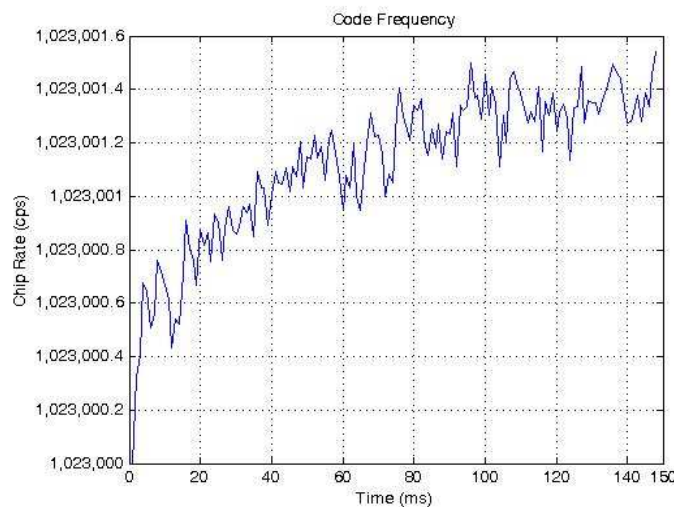
Figure-11 Code Frequency

Results and Discussions

The Code and Carrier of the GPS C/A code are successfully tracked. The effects of various parameters such as Noise Bandwidth, Damping Ratio over the tracking of the signal were studied. The prompt output of the In-phase arm provided the data where as the Quadrature arm prompt output is minimised at steady state.
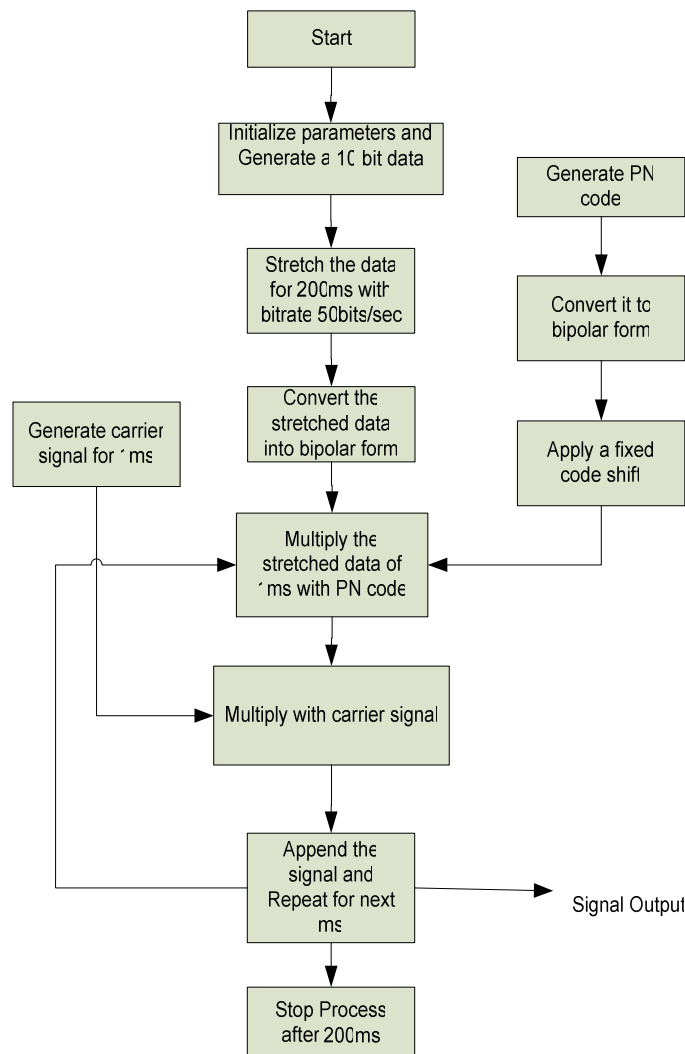
4.2.1 Test Signal

a) MATLAB generated test signal
For our testing in initial phase we generated a test signal in MATLAB using the steps shown in figure-12, with specifications as:
Sampling frequency, fs=112MHz, Carrier frequency, f= 28.12 MHz,
Code frequency, fc= 1.023 MHz.
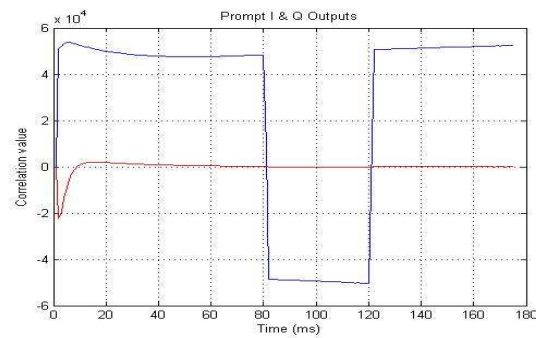We implemented a code shift of 45672 in it with changing code frequency.

Figure 12 Logical sequence for test signal generation

The spectrum of our test signal is shown in figure-13.
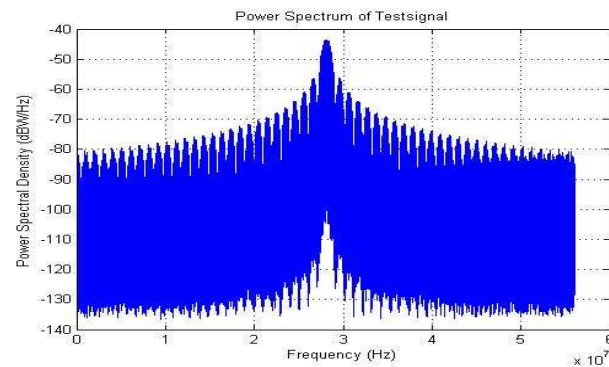
Figure-13 Spectrum of MATLAB generated test signal

The I and Q output of our test signal are as follows:

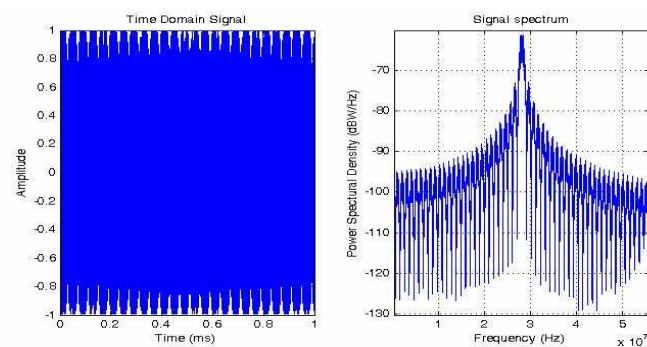**Figure-14** I & Q output of test signal

a) AWR generated test signal
The signal is generated from AWR. This application is a system simulator used to simulate the whole system.

Specifications:
Sampling frequency, fs=112.53MHz, Carrier frequency, f= 28.12 MHz,
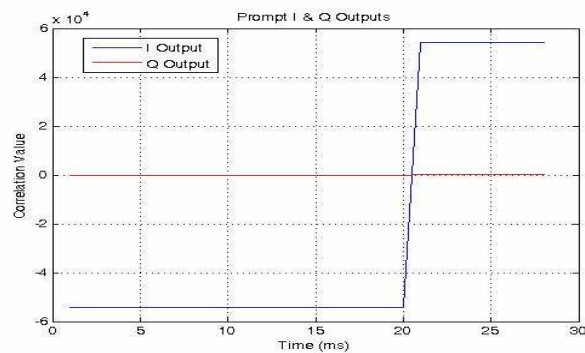Code frequency, fc= 1.023 MHz.
The original signal and its corresponding frequency domain graph is given below:

Figure-15 Signal in time and frequency domain

Implementing the code over a test signal we get the data from the prompt output of the In-phase arm and at steady state the prompt output of the Quadrature arm can be seen approaching Zero.
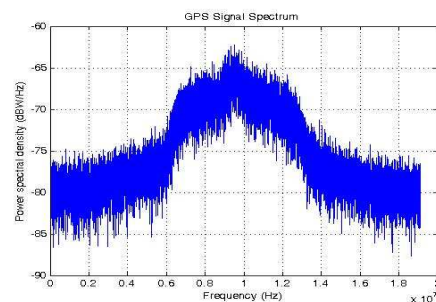

Figure-16 Prompt I & Q Output

4.2.2 GPS Signal

Specifications:
Sampling frequency, fs=38.192MHz, Carrier frequency, f= 9.549949 MHz,
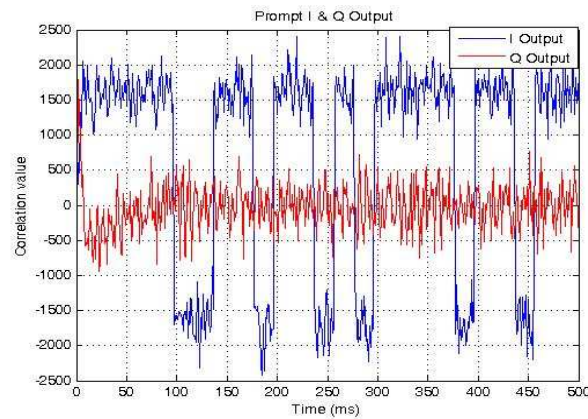Code frequency, fc= 1.023 MHz
The spectrum of GPS signal:

Figure-17 GPS Signal Spectrum

This is an 8 bit quantized signal of more than 50s duration. We are showing the output for first 500ms. Figure 17 and figure 18 shows tracking output for C/A code - 3 and C/A code – 9, respectively. The signal is available from the [2].
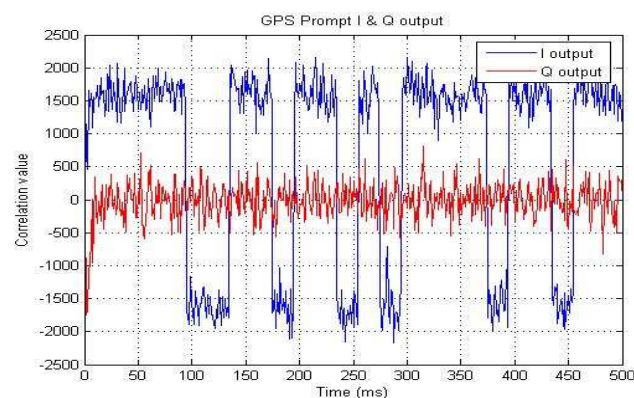
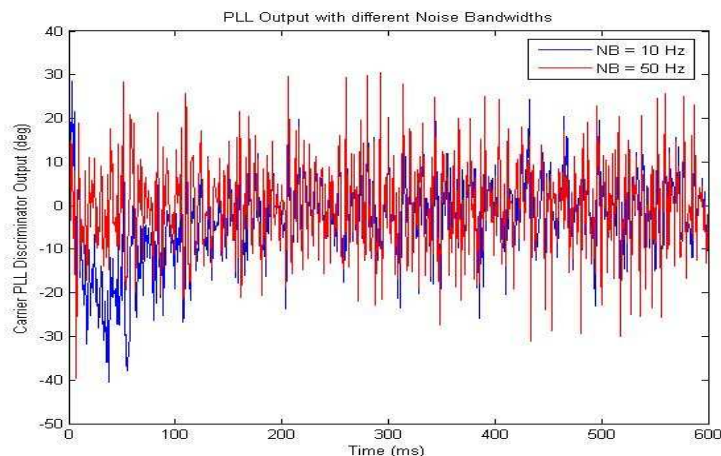**Figure-18** Output of prompt arms for C/A code-3

Figure-19 Output of prompt arms for C/A code-9

4.2.3 Impact of Loop Parameters

i) Different Noise Bandwidths:

    a) **Carrier Frequency:** The comparison of the carrier frequency over noise bandwidths (10 & 50 Hz) for $\zeta=0.707$.
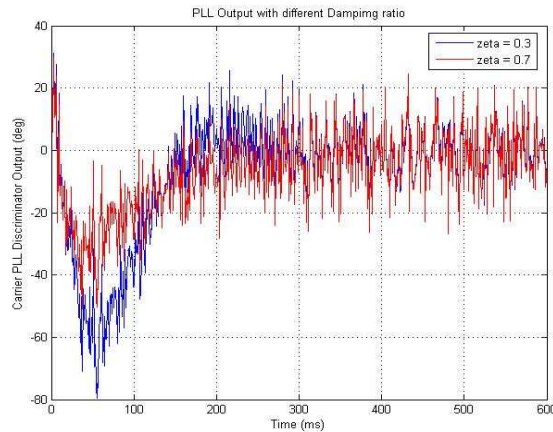

Figure-20 PLL Output for different Noise Bandwidths

It was planned to take output for 5 different values of noise bandwidth, but due to high noise coming in the signal, they could not be identified separately. Hence, only two values of noise bandwidth are used for the comparison. This parameter controls the amount of noise allowed in the filter as well as the settling time. A large noise bandwidth, the tracking loop quickly locks to the real frequency but has a relatively large frequency noise in the locked state. A smaller noise bandwidth implies that it can take some time before the tracking loop can be locked to the frequency, but after the lock the frequency is stable. [1]

ii) Different Damping Ratio:
a) Carrier Frequency: The variation in carrier frequency for damping ratio, ζ= 0.3 & 0.7 was observed for B_L = 25 Hz.

Figure-21 PLL Output for different Damping Ratio

It is found that the damping ratio is a compromise between settling time and overshoot of the filter. The damping is chosen to $\zeta = 0.7$ resulting in a filter that converges reasonably fast and does not make a high over shoot.

4.3 Conclusion

We used the concept of second order PLLs and derived the required coefficients to appropriately generate the MATLAB codes for carrier and code tracking loops separately. We implemented the MATLAB codes for the tracking of the GPS signal to recover the navigation data based on our study on the tracking loops. All the parameters were analytically set. The correlator prompt output in the in-phase arm was observed to match the navigation data and the quadrature output was minimised at steady state. The carrier frequency was updated according to the phase shift. As expected, the carrier frequency Vs time plot had an initial overshoot and then attained almost a constant value after the settling time is reached. Hence the complete study of the demodulation process was verified in this project.

Appendix

Specifications:
For GPS Signal:
Sampling frequency, fs=38.192MHz, Carrier frequency, f= 9.549949 MHz,
Code frequency, fc= 1.023 MHz
For test signal:
Sampling frequency, fs=112MHz, Carrier frequency, f= 28.122018 MHz,
Code frequency, fc= 1.023 MHz

The MATLAB code for signal generation:
%generation of test data width pn code shift of 45672 and no Doppler phase
clear all;

```matlab
fs=112e6;  x=0;    f=28.12e6;
ms=200;   cf=1.023e6;
data=[1,1,1,1,0,0,1,1,1,0];
sch=ceil((10/(ms*f))*(1:(ms*f)));
dsch=data(sch);
dpol=2*dsch-1;
pn=SPScode;

for i=2:ms
   af=ceil((cf(i-1)/fs)*(1:floor((fs*1023)/(cf(i-1)))));
   adf=pn(af);    adfg=2*adf-1;
   ad=[adfg(end-45672+1:end) adfg(1:end-45672)];

   dpn=dpol(x+1:x+floor((fs*1023)/(cf(i-1)))).*ad;

   n=(1:floor((fs*1023)/(cf(i-1))))*1/fs;
   ang=(2*pi*f*n)+remphs;
   remphs=rem(ang(end),2*pi);
   car=sin(ang);

   sig(x+1:x+floor((fs*1023)/(cf(i-1))))=car.*dpn;

   cf(i)=cf(i-1)+0.01;
   x=x+floor((fs*1023)/(cf(i-1)));
end
```

The MATLAB code of tracking loop:
```matlab
clear all;
f(1)=9.549949e6; fs=38.192e6;   cf(1)=1.023e6; remphs=0; h=0;   cs=0;
Ip=0;  Il=0;  Ie=0;  Qe=0;  Qp=0;  Ql=0;  D=0;   th=0;
fid = fopen('GPSdata-DiscreteComponents-fs38_192-if9_55.bin','rb');
pnc=generateCAcode(3);
pn=[pnc(1023) pnc pnc(1)];

%carrier loop parameters
bl=25; ts=0.001;  zeta=0.7;
wn=8*zeta*bl/(4*zeta^2+1);
t1=1/(wn^2); t2=2*zeta/wn;
c1=(2*t2-ts)/2/t1;
c2=ts/t1;

%code loop parameters
blc=2;  zeta1=0.7;
wnc=8*zeta1*blc/(4*zeta1^2+1);
t3=1/(wnc^2);   t4=2*zeta1/wnc;
c3=(2*t4-ts)/2/t3;
c4=ts/t3;

sta=fseek(fid,34211,'bof');
resph=0;
```

```
for i=2:500
   %adding the code phase error
   cps=cf(i-1)/fs;
   blksize=ceil((1023-resph(i-1))/cps);
   ecp=-0.5+resph(i-1):cps:(blksize-1)*cps+resph(i-1)-.5;
   lcp=0.5+resph(i-1):cps:(blksize-1)*cps+resph(i-1)+.5;
   pcp=0+resph(i-1):cps:(blksize-1)*cps+resph(i-1);
   resph(i)=pcp(end)+cps-1023;
   eci=ceil(ecp)+1;
   lci=ceil(lcp)+1;
   pci=ceil(pcp)+1;
   ec=pn(eci);
   lc=pn(lci);
   pc=pn(pci);

 %data input
   dt=fread(fid,blksize);
   t=dt';
   s=t-mean(t);    r=s/max(s);

   %adding carrier phase error
   n=(1:blksize)*1/fs;
   ang=(2*pi*f(i-1)*n)+remphs;
   remphs=rem(ang(end),2*pi);
   car=sin(ang);
   ort=cos(ang);
   %multiplying with the carrier
   I=r.*car;
   Q=r.*ort;

  %generating different channels and multiplying with the pn code
  %e channel
   ie=I.*ec;
   qe=Q.*ec;
  %p channel
   ip=I.*pc;
   qp=Q.*pc;
  %l channel
   il=I.*lc;
   ql=Q.*lc;

   Qp(i)=sum(qp);    Ip(i)=sum(ip);
   Ie(i)=sum(ie);    Il(i)=sum(il);
   Qe(i)=sum(qe);    Ql(i)=sum(ql);
   a=(Qp(i)/Ip(i));    th(i+1)=atan(a)/(2*pi);
%carrier loop
   temp(i)=(c1+c2)*th(i+1)-c1*(th(i));
   f(i)=f(i-1)+temp(i);
%code loop
   D(i)=((Ie(i)^2+Qe(i)^2)-(Il(i)^2+Ql(i)^2))/((Ie(i)^2+Qe(i)^2)+(Il(i)^2+Ql(i)^2));
```

```
   cf(i)=cf(i-1)-((c3+c4)*D(i)-c3*(D(i-1)));
end
fclose(fid);
```

Acknowledgement

We are thankful to SAC and ISRO for the successful completion of this project. Thanks are due to Shri J.K. Hota , Sci./Engr. 'SG', Head, NSED, Mr. Pravin Patidar, Sci./Engr. 'SD', Ms. Kriti Khatri, Sci./Engr. 'SD', Mr. Rahul Datta, Sci./Engr. 'SC' and Mrs. Soniya Nibhani Sci./Engr. 'SC' of Navigations Systems Engineering Division for their guidance and help during the work.

References

[1]   R.E. Best, Phase Locked Loops: Theory Design and Applications, McGraw Hill, 2007.
[2]   K. Borre, D. M. Akos, N. Bertelsen, P Rinder and S.H. Jensen, A software defined GPS and galileo receiver: a single frequency approach, Birkhauser, Boston, *Applied and Numerical Harmonic Analysis Series:* XXI, 2007.
[3]   B.Y. Chung, C. Chien, H. Samueli and R. Jain, Performance analysis of an all digital BPSK direct-sequence spread-spectrum IF receiver architecture, *IEEE Journal on Selected Areas in Communications*, 11(7) (1997), 1096-1107.
[4]   R.C. Dixon, Spread Spectrum Systems: With Commercial Applications, 3rd ed., Wiley-Interscience, New York, Chichester, 1994.
[5]   D. Frederick and J. Chow, Feedback Control Problem Using MATLAB, PWS Pub. Co., 1995.
[6]   V.F. Kroupa, Phase Lock Loops and Frequency Synthesis, Wiley, 2003.
[7]   T. Pratt, C.W. Bostian and J.E. Allnut, Satellite Communications, 2nd Edition, Wiley, 2003.
[8]   D.R. Stephens, Phase Locked Loops for Wireless Communications: Digital and Analog Implementations, Kluwer, 1998.
[9]   www.googlebooks.com
[10]   www.wikipedia.com