New Acquisition Method in GPS Software Receiver with Split-Radix FFT Technique

W. L. Mao, W. H. Lin, Y. F. Tseng, H. W. Tsao, and F. R. Chang

Department of Electronic Engineering, National Formosa University
Department of Electrical Engineering, National Taiwan University
Department of Electrical Engineering, University of California, Los Angeles

wlmao@nfu.edu.tw, roger1109@gmail.com, yifen1224@yahoo.com.tw, frchang@cc.ee.ntu.edu.tw, tsaohw@cc.ee.ntu.edu.tw

1. Introduction

Global positioning system (GPS) receivers must detect the presence of GPS signal to track and decode the information which enables position computation. Tracking of the signal is possible only after they have been acquired, so acquisition is the first step in GPS signal processing scheme. The acquisition must ensure that the signal is acquired at the correct code phase and carrier frequency.

Software receivers provide flexibility, reconfigurability and high fidelity among many other advantages. There has been a variety of software based receiver architectures for GPS and other navigational system.

The fast algorithm for computing Discrete Fourier Transform (DFT) has been studied for many years. The split-radix FFT approach is proposed by Duhaamel and Hollmann in [9] differs from that of Cooley-Tukey in decomposition process. For example, in the Cooley-Tukey radix-2 FFT, a radix-2 index map is used to factor both the even and odd index term.

The split-radix fast Fourier transform FFT algorithm [10] is known as the FFT algorithm that uses the lowest total number of operation. The split-radix idea can be extended to other radix pairs. When we want to enhance the speed of GPS software receiver, we use the algorithm of FFT that is radix-2, split-radix-2/4, and split-radix-2/8. We will compare their stores, loads, additions, and multiplications.

The rest of paper is organized as follows. Section II presents the architecture of GPS System. In Section III, we will introduce Split-radix 2/8 fast Fourier transform and other algorithms about FFT. In the section IV, we will show the simulation results and some table that compare each algorithms of FFT. Final is our conclusions about the GPS software receiver using FFT algorithm.

2. The architecture of GPS system

The basic GPS receiver discussed in this paper is show in Figure 1. The signals transmitted from the antenna. Through the radio frequency (RF) chain the input signal is amplified to proper amplitude and the frequency is converted to desired output frequency. An analog-to-digital converter (ADC) is utilized to digitize the output signal. The antenna, RF chain and ADC are the hardware used in receiver.

There are basically two types of signals: the coarse/acquisition (C/A) and precision (P) codes. The actual P code is not directly transmitted by the satellite, but it is modified by a Y code, which is often referred to as the P(Y) code. The P(Y) code is not available to civilian user and primarily used by the military. In general, in order to acquire the P(Y) code, the C/A code is usually acquired first.

The signal structure of the satellite may be modified in the future. However, at the present time, the L1 frequency contains the C/A and P(Y) signals, while the L2 frequency contains only the P(Y) signal. The C/A and P(Y) signals in the L1 frequency are quadratic phase of each other and they can be written as:

$$S_{L1}(t) = A_c P(t) \cos(2\pi f_c t + \phi) + A_C C(t) \cos(2\pi f_c t + \phi)$$

where $S_{L1}$ is the signal at L1 frequency, $A_c$ is the amplitude of the P code, $C(t) = \pm 1$ represents the phase of the P code, $D(t) = \pm 1$ represents the data code, $f_c$ is the L1 frequency, $\phi$ is the initial phase, $A_c$ is the amplitude of the C/A code, $C(t) = \pm 1$ represents the phase of the C/A code. In this equation the P code is used instead of the P(Y) code. The P(Y), C/A, and the carrier frequencies are all phase locked together. The role of acquisition is to provide a coarse estimate of the code phase...
and the Doppler to the tracking loops.

![Fig 1: The architecture of GPS system](image1)

**Fig 2: Parallel Code Phase Search Acquisition**

### A. Signal acquisition

The satellites are differentiated by 32 different PRN sequences. The second parameter, code phase, is the time alignment of PRN code in current block data. The code phase is necessary to be able to generate a local PRN code that is perfectly aligned with incoming data. Only when this is the case, the incoming code can be removed from the signal. The third parameter is the carrier frequency, which in case of down conversion corresponds to the IF. The IF should be known from the L1 carrier frequency of 1575.42MHz and from the expected value. The line of sight velocity of the satellite causes a Doppler effect resulting in a higher or lower frequency. In the worst case, the frequency can deviate up to ±5kHz. It is important to know the frequency of the signal to be able to generate a local carrier signal carrier signal. The signal is used to remove the incoming carrier from the signal. In most case it is sufficient to search the frequency in steps of 250Hz.

There are three acquisition methods. First, serial search acquisition, parallel frequency space search acquisition, and parallel code phase search acquisition. Comparison with three methods, we will use the third method. It is to provide a better estimate of frequency to the carrier loop, and the parallel code phase search algorithm has better execution time and parameter estimation.

### B. Parallel Code Phase Search Acquisition

In the same way as with the order acquisition methods, the implementation of this one is very straight-forward, as it can be implemented directly based on the block diagram in Fig. 2.

One difference in this method is that only one PRN code should be generated for each acquisition. That is, it is not necessary to take the 1023 different code phase into account.

The first step of the method is multiply the incoming signal with locally generated cosine and sine carrier wave respectively, giving an I and a Q signal component. These two are combined as a complex input to the Fourier transform. The result of the Fourier transform is multiplied with the result of lower branch of the block diagram in Figure 2. This signal is created as follow. The PRN generator generates a code with no code phase. As in the implementation of the other acquisition algorithm, the code generation is performs a Fourier transform of the PRN code, and the result is complex conjugated.

The goal of acquisition is to perform a correlation with the incoming signal and a PRN code. It is more convenient to make a circular correlation between the input and the PRN code without shifted code phase.

The cross-correlation between two finite length sequences x(n) and y(n) both with length N is calculated as:

\[ z(n) = \sum_{m=0}^{N-1} x(n) y(n+m) \]  \hspace{1cm} (2)

The convolution between the two finite length sequence x(n) and y(n) is calculated as:

\[ z(n) = x(n) \otimes y(n) = \sum_{m=0}^{N-1} x(n) y(n-m) \]  \hspace{1cm} (3)

Equation (4) and (5) show only difference between cross-correlation and convolution between finite length sequence is the sign y(n+m). The convolution can be transferred to frequency domain through Fourier transform.

\[ Z(k) = \sum_{m=0}^{N-1} x(m) y(n-m) e^{-j2\pi km/N} = X(k)Y(k) \]  \hspace{1cm} (4)

The combination of equation (5) and (6) give the DFT of cross-correlation between x and y:

\[ Z(k) = \sum_{m=0}^{N-1} x(m) y(n+m) e^{-j2\pi km/N} = X^*(k)Y(k) \]  \hspace{1cm} (5)

Therefore, the (6) and (7) are almost similar to each other. The only difference is that the Fourier transform of one of the two input sequences should be complex conjugated before multiplication. Fig. 2 is a block diagram of the parallel code phase search algorithm.

The incoming signal is multiplied by locally generated carrier signal. Multiplication with the carrier signal generates the I-arm signal, and multiplication with a 90° phase shifted version of the carrier signal generates the Q-arm signal. The two I and Q signals and combined to form a complex input signal \( x(n) = I(n) + jQ(n) \) to DFT function. The generated PRN code is transformed into the frequency domain and result is complex conjugated. The Fourier transform of the input is multiplied with the Fourier transform of the PRN code. The result of multiplication is transform into the time domain by an inverse transform. The absolute value of the output of the inverse Fourier transform represents the correlation between the input and the PRN code. If a peak is present in the correlation, the index of this peak marks the PRN code phase of the incoming signal.
3. Split-radix 2/8 Fast Fourier Transform

Discrete Fourier transform (DFT) is an important tool in digital signal processing. Many fast algorithms have been proposed to compute the discrete Fourier transform of signals. The split-radix-2/4 fast Fourier transform algorithm and the split-radix-2/8 fast Fourier transform are known as the FFT algorithms that use the lowest total number of operations. But the split-radix-2/8 FFT saves 25% of data loads and stores compared with the split-radix -2/4 fast Fourier transform algorithm.

A. Fast Fourier Transform

The radix-2 fast Fourier transform algorithm decomposes the N-point FFT into two (N/2)-point FFT. The decimation-in-frequency version is shown as follows:

\[ X_{2k} = \sum_{n=0}^{(N/2)-1} x_n + x_{n+(N/2)} W_N^{nk} \]

\[ X_{2k+1} = \sum_{n=0}^{(N/2)-1} x_n - x_{n+(N/2)} W_N^{nk} \]

where \( k = 0, \ldots, (N/2) - 1 \)

Even Part:

The split-radix-2/4 fast Fourier transform algorithm and the split-radix-2/8 FFT decomposes the even part using radix-2 method while using radix-8 for the odd part. This developed by Duhamel and Hollmann uses radix-2 and two (N/4) –point FFT. Using the split-radix -2/4 FFT method, the formula for the even part of FFT is as follows:

\[ X(2k) = \sum_{n=0}^{N/2-1} x(n) + x(n + N/2) W_N^{nk} \]

Odd Part:

\[ X(2k+1) = \sum_{n=0}^{N/2-1} x(n) - x(n + N/2) W_N^{nk} \]

B. Split-radix-2/4 Fast Fourier Transform

The split-radix-2/4 fast Fourier transform algorithm developed by Duhamel and Hollmann uses radix-2 decomposition for even part and radix-4 for odd part. This method decomposes the N-point FFT into one (N/2)-point FFT and two (N/4) –point FFT. Using the split-radix -2/4 FFT method, the formula for the even part of FFT is as follows:

Even Part:

\[ X(2k) = \sum_{n=0}^{N/4-1} x(n) + x(n + N/2) W_N^{nk} \]

Odd Part:

\[ X(2k+1) = \sum_{n=0}^{N/4-1} x(n) - x(n + N/2) W_N^{nk} \]

The structure of the split-radix-2/4 FFT algorithm is shown in Fig.5.

4 Peak- Finding method and Search Detector

There is an interesting issue will be discussed. When the autocorrelation function is calculated, how the program detects where is the peak value? Especially when the signal power is weak or it has a Doppler frequency shift, the problem is hard to get the optional solution. In Fig.6 the peak value is
very evident. The difference between the maximum and the second maximum value is not very large. So there exits probabilities that the receiver might not find the peak value correctly. So there are two problems here: one is how to find the peak value correctly? And another is how to find the most efficient integration? It is reasonable to assume the noise is Gaussian distribution, Monte-Carlo method and statistical properties are combined to decide suitable integration time.

There are many methods used to decide the peak value. Such as histogram, estimate the distribution of the autocorrelation function, detect the same peak location more than two times, and so on. In order to reduce the computation load on receiver, a simpler algorithm is used:

1. Find maximum value, second maximum value and mean.
2. Normalize the function (maximum value = 1 )
3. If (maximum value – mean) > VTH1, AND (maximum value –second maximum value ) > VTH2, then the time has peak value is code delay
4. maximum value > VTH3 × second maximum value

The algorithm is very immediate. The meaning of “peak value” is the maximum value of all. To avoid miscarriage, detects the difference between maximum value and second maximum value is needed. If the difference between maximum value and second maximum value is big enough, the probability of false alarm will be decreased.

\[
\text{SNR (dB)} = 10 \log \left( \frac{\text{max} \left[ d(n) \right] - \text{mean} \left[ d(n) \right]}{\text{std} \left[ d(n) \right]} \right)
\]

(9)

And combine the third step of the algorithm, it can be rewritten as:

(maximum value -mean)

\[
= \text{SNR (dB)} \times (\text{standard deviation}) > V_{\text{TH1}}
\]

(10)

A good performance signal has SNR more than 10dB and standard deviation and standard deviation is about 0.03 (after normalization):

So SNR (dB) × (standard deviation) > 0.3, and V_{TH1} = 0.3.

Then, if maximum value is two times bigger than second maximum value, it means the probability that maximum value is real peak is very high. The math represents:

Two times is decibel representation is 3 dB

\[
\text{SNR} \text{max_value(dB)} - \text{SNR}_{2\text{nd max value(dB)}} > 3dB
\]

(11.1)

\[
\log \left( \frac{\text{max} \left[ d(n) \right] - \text{mean} \left[ d(n) \right]}{\text{std} \left[ d(n) \right]} \right) > 0.3
\]

(11.2)

\[
V_{\text{TH2}} = 0.15
\]

(11.3)

Now, we will introduce false alarm, missing and probability of detection. If the PDF of noise exceeds threshold voltage we assume, the result could be declared as false alarm. If the PDF of signal with noise is below threshold voltage we assume, the result could be declared as missing. If the PDF of signal exceeds threshold voltage we assume, the result could be declared as probability of detection. Fig.8 shows an example of probability density function, PDF, of the correlation power distribution, and a detection threshold (V_t). As shown in the Fig.7, both Pd and Pfa are determined by the distribution of the correlation power and the threshold value.

To decide V_{TH3} is a experience rule. But, how do we decide the V_{TH3}? We assume maximum value is bigger than V_{TH3} multiplying second maximum value.

After all the factors above have been colligated, we will find the peak value. Hence, we will try to find V_{TH3}, we can discover the probability of false alarm, the probability of...
missing and the probability of detection. Table 1 will represent this.

Table 1 Detection Probability, Missing Probability and False Probability

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6%</td>
<td>1.3%</td>
<td>-110dBm</td>
<td>-120dBm</td>
</tr>
<tr>
<td>1.9</td>
<td>5.3%</td>
<td>1.3%</td>
<td>-110dBm</td>
<td>-120dBm</td>
</tr>
<tr>
<td>1.8</td>
<td>5.1%</td>
<td>1.7%</td>
<td>-110dBm</td>
<td>-120dBm</td>
</tr>
<tr>
<td>1.7</td>
<td>4.7%</td>
<td>2.1%</td>
<td>-110dBm</td>
<td>-120dBm</td>
</tr>
<tr>
<td>1.6</td>
<td>4.7%</td>
<td>2.2%</td>
<td>-110dBm</td>
<td>-120dBm</td>
</tr>
<tr>
<td>1.5</td>
<td>4.1%</td>
<td>2.3%</td>
<td>-110dBm</td>
<td>-120dBm</td>
</tr>
<tr>
<td>1.4</td>
<td>3.7%</td>
<td>2.3%</td>
<td>-110dBm</td>
<td>-120dBm</td>
</tr>
<tr>
<td>1.3</td>
<td>3.6%</td>
<td>2.6%</td>
<td>-110dBm</td>
<td>-120dBm</td>
</tr>
</tbody>
</table>

According to sum of detection probability, missing probability and false probability, we will choose 1.5. The error criteria sum up the probability of missing and the probability of false alarm. Because the error criteria is minimum, we will choose $V_{THD} = 1.4$.

We implemented a modified version of Tong detector described in Kaplan. Fig. 10 presents the Tong Search Detector algorithm. To utilize the Tong algorithm a few variable must be initialized. These variables are $K$ and $A$. The counter variable $K$ and the confirmation threshold $A$ must be initialized based on the operational environment. The operational environment will determine the relative importance of acquisition speed versus probability of detection and false alarm. To achieve a high probability of detection and a lower probability of false alarm, at the expense of speed, $K$ may be set to 2 or higher. When $K$ reaches $A$ is declared present. Assuming the search space has been defined as described in the previous section execution of the Tong Algorithm is relatively simple. Autocorrelation envelope $\sqrt{I^2 + Q^2}$ is formed for I channel and Q channel. If the correlation envelope exceeds the threshold $V$, then the counter $K$ is incremented by one, else $K$ is decremented by one. If $K$ equals $A$ then the signal is declared present and the search over. If $K$ equals zero then the signal is determined not to be present and we will search once time.

But how do we decide $A$? We assume each trial is Bernoulli trial, we can calculate the error probability. If we assume $\alpha = 0.06$ ($\alpha$ is error probability), we will choose $A = 5$ and the error probability is $1.97 \times 10^{-5}$. According to the method 1 used previously, the result has about 7% error probability. In other words, we only use the threshold voltage 1 and the threshold voltage 2 but no threshold voltage 3. The result has about 7% error probability. I submitted a brand new method to establish another threshold, which will decrease the error probability. That is to say we will assume a new threshold voltage 3, which will decrease the error probability. So it will reduce 1%, it will enhance the probability of detection.

Fig. 10 Tong Search Detector

6. Simulation results

Now, an important problem is discussed before acquisition process: sampling rate problem. An important factor in selecting the sampling frequency is related to the C/A code chip rate. The C/A code chip is 1.023MHz, and the sampling would be a multiple number of the chip rate. For our example, a sampling frequency of 16.384 MHz is used.

Now, see our data specification. The chip rate is of 1.023MHz, so 16$x$ chip rate is equal to 16.384MHz. But the sampling rate = 16.384 MHz, it means that in one millisecond, 16384 points will be chosen. But 16$x$chip rate = 16.368MHz means one millisecond, the picked up sample is 16368. Thus, zero must be inserted. Inserting zero is only sure process, because in tracking mode, synchronized circuit such as Costas PLL will be used.
If signal strength is more than threshold, we will declare the signal is found. We will send some information to tracking mode. When the autocorrelation function is calculated, how does the program detects where is the peak value? Especially when the signal power has a Doppler frequency shift, the problem is hard to get an optimal solution.

Simulation results show the chip delay and Doppler frequency for different GPS satellites. The input SNRs are set at -110 dBm. In Fig. 6, the autocorrelation functions are conducted using the split-radix-2, split-radix-2/4, and split-radix-2/8 methods to demonstrate the acquisition process. The split-radix 2/8 scheme can have the better computational efficiency and preserve the information of code offset effectively.

Table 2: Number of nontrival real multiplication and two real additions per complex multiplication for N complex number

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Loads</th>
<th>Stores</th>
<th>Mults</th>
<th>Adds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radix-2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Split-radix-2/4</td>
<td>2</td>
<td>4/3</td>
<td>4/3</td>
<td>8/3</td>
</tr>
<tr>
<td>Split-radix-2/8</td>
<td>3/2</td>
<td>1</td>
<td>5/4</td>
<td>11/4</td>
</tr>
</tbody>
</table>

From Table II, and III the Split-radix-2/4 FFT and the Split-radix 2/8 FFT are compared. Obviously, the conventional split-radix -2/8 saves 25% of data loads and stores compared with the split-radix-2/4 FFT algorithm.

6. Conclusions

In this paper, three very useful algorithms have been presented in software GPS receiver design. The software modules, taking such IF samples as its input, perform the baseband signal processing and navigation solution computation. The proposed schemes for acquisition in frequency domain are the Fast Fourier Transform: radix-2, split-radix-2/4 and split-radix-2/8. It is more efficient and speedy to use the split-radix-2/8 algorithm than the split-radix-2/4 algorithm because of the less computation burdens and memory requirements. The acquisition based on modified Tone detector method is implemented and evaluated on PC to demonstrate the civil applications.

ACKNOWLEDGMENT

This research was supported by the Evermore Inc., Hsinchu City, Taiwan, R. O. C.

REFERENCES