FFT-based Fast-Response Quasi-Open Tracking Loop for GPS Receiver

Yanpin He¹,a, Weimin Li²,b, and Bo Bi²,c

¹, ² Beijing Microelectronics Technology Institute, Beijing, 100076, China

¹ 13366120525@163.com, b liwm@mxtronics.com, c bibo_mxt@sina.com

Keywords: GPS Receiver, FFT, Matched Filter, CZT

Abstract. This article discusses a new GNSS signal tracking architecture with a quasi-open tracking loop. Matched filtering is adopted for PRN code phase discrimination and FFT is used for frequency discrimination, which greatly shortens the duration of the pull-in step after acquisition. There is no loop filter any more. The discrimination results are fed back to code NCO and carrier NCO directly to modify the local signal in the next round. A Chirp-Z Transform (CZT) algorithm is considered for further improving the frequency resolution of FFT, which can also be computed using FFT. All the modules in this architecture are realized on hardware platform, showing characteristic advantages compared to traditional software tracking loop.

Introduction

In traditional tracking loop, carrier frequency and code phase of local signal is slowly and finely tuned by the discriminators and the loop filters. But this on the other hand reflects the drawbacks of traditional tracking loop that it takes relatively long time for the loop to recover from interference as well as the acquisition-to-tracking transition, i.e. pull-in stage. So it limits the flexibility of the loop.

Matched filtering and FFT algorithm is first put forward for fast acquisition [1]. But the parallel processing it is based on can also be applied in tracking [2]-[4]. It discriminates carrier frequency and code phase directly so that the modified quantity is elastic. Even if the frequency or code phase shifting is large, it still guarantees high adjusting speed. The whole structure can be realized on hardware, which saves the resources used for integrating a CPU on chip and accelerates the computing speed. Moreover, the tracking FFT module can multiplex the FFT module used in acquisition stage. So it won’t take too many resources.

The main worry of the tracking structure in question is the frequency tracking precision because FFT is discrete. It is unpractical to do ultra long FFT, which takes too many resources and will cause the chip’s area too large. The fact is that only the acquisition needs to search full spectrum. Tracking needs only searching the spectrum around the estimated frequency from acquisition. So spectrum refining methods should be introduced. There are several kinds of approaches to refine spectrum, such as some linear or nonlinear interpolation algorithms, Zoom-FFT and CZT [3], [5]. Simple interpolations may be too stiff with characteristics of frequency domain not been considered, and the algorithms’ performances are difficult to assess. According to [5], CZT outperforms Zoom-FFT in not requiring filter and higher precision.

Tracking Structure

Acquisition. For better comprehension, it is necessary to introduce the acquisition method first, because tracking structure is built based on acquisition structure. In acquisition stage, matched filter and FFT is used for search in code phase dimension and frequency dimension respectively [1]. For GPS L1 signal, 10 ms 128 dots FFT is chosen to achieve a balance between performance and computing load. The acquisition process can be divided into following steps (Fig.1):

Suppose that the PRN number is already known (ignore the search in PRN dimension) and the sampling rate is $f_s$, local code NCO generates the 10 ms corresponding PRN code, the length of which is $0.01 \times f_s$ dots.
1) The phase shifter shifts the local code’s phase by $\frac{1}{2} chip$ at a time. At each phase the local code multiplies the received signal to get a result sequence $r(k, 1: 0.01 * fs), k = 1, 2 ..., 2046$.

2) Piecewise summation is processed on these result sequences for down sampling, which also helps to improve SNR. The sampling rate after down sampling is 12.8 KHz, which corresponds to the ±6.4 KHz frequency acquisition range. A 2046*128 array is built after down sampling:

$$s(k, n) = \sum_{i=0}^{0.01 * fs / 128} r(k, i)$$

3) Do 128-point FFT on every row of the array above to get a 2046*128 array in frequency domain. The size of the frequency bin is 12.8 KHz/128 = 100 Hz.

4) Find the location of the peak, the corresponding frequency and code phase of which is the acquisition results.

In practical application, the two adjacent 10 ms will both be used in detection. Because if one 10 ms is across a data bit, then the other will not be. The two correlation values are compared to select the larger one for detection.

**Tracking.** According to the acquisition (Fig. 1), the code phase error of the acq_result is $\pm\frac{1}{4} chip$ and the frequency error is $\pm 50 Hz$. The result is still too rough for ranging so that the tracking stage follows. The tracking process is like a fine acquisition (Fig. 2).

In pull-in stage, for wider pull-in range, faster update rate and bit synchronization not realized, 1 ms 128 dots FFT is chosen to find relatively rough frequency estimation.

Then chirp-z transform algorithm is used to refine the frequency domain near the rough frequency.

For code phase pull-in, different from acquisition, only correlations near the rough code phase from acquisition rather than all code phase is computed based on sampling points, i.e. code phase shifting by 1 sample point at a time. If the IF is 4.123968 MHz and the sampling rate
is 16.7667 MHz, then 1 code chip includes about 16 sampling points. The value of the IF and the
sampling rate will be used all over the article. If correlations are made for every sampling point,
then the code phase tracking resolution is ± \( \frac{1}{32} \) chip, corresponding to ± 9 m of pseudo-range. The
pseudo-range resolution can be more refined with higher sampling rate.

**Chirp-Z Transform Algorithm**

**Algorithm Description.** The discrete Fourier transform (DFT) is the sampling Z transformation
on the unit circle where the sampling points are uniformly-spaced, and the Chirp Z-transform (CZT),
more generally, samples along spiral arcs in the Z-plane with alterable sampling space by changing
the number of sampling points [5], defined as:

\[
z_k = A \cdot W^{-k}, k = 0, 1, ..., M - 1
\]  

(2)

Where \( A = A_0 e^{j\theta_0} \), is the start point of the sampling track with \( A_0 \) representing the distance
between start point and the origin and \( \theta_0 \) representing the angle of the start point; \( W = W_0 e^{-j\varphi_0} \),
is the ratio between two points with \( W_0 \) controlling the direction of the spiral arc and \( \varphi_0 \) being
the phase difference of two adjacent points; \( M \) is the number of CZT points. The track’s spiral arc
bends to inside circle when \( W_0 > 1 \), while bends to outside circle when \( W_0 < 1 \). When \( W_0 = 1 \), the
track is a circle arc.

With the property analyzed above, spectrum refining and peak detection is realized by taking out
a specific spectrum section and doing the CZT. The steps are as follows [5, 6]:

1) Set \( A_0 = W_0 = 1 \), given sampling rate as \( f_s \). Determine the spectrum section needed to be
refined as \( [f_1, f_2] \), and the value of \( M \).

2) Map \( [f_1, f_2] \) to unit circle as angle range \( \left[ \frac{2\pi f_1}{f_s}, \frac{2\pi f_2}{f_s} \right] \), and then start angle \( \theta_0 = \frac{2\pi f_1}{f_s} \).

3) In the angle range \( M \) points need to be calculated, so

\[
\varphi_0 = \frac{(2\pi f_2/f_s - 2\pi f_1/f_s)}{M}
\]

(3)

4) Calculate CZT according to (2).

**Computing Method in Application.** In practical engineering application, this calculation is
implemented using FFT [5]. For the receiver structure in question, the sampling frequency is
128 KHz after down sampling. The 128-point full spectrum FFT is calculated using the down
sampled signal in the pull-in stage, with the frequency resolution being 1 KHz. Suppose the
frequency corresponds to the peak is \( f_p \), it is reasonable to set the CZT spectrum section as \( [f_p - 500, f_p + 500] \) Hz. The value of \( M \) depends on the requirements of the refining resolution. But
remember, the bigger \( M \), the more points of FFT and thus the more computing load. For example, if
the resolution of 1.25 Hz is required, \( M \) should be 1000/1.25 = 800, which means 1024-point
FFT should be computed (1024 > 800 + 128 – 1).

**Parameter Selection.** The frequency resolution influences the accuracy of the code phase
estimation to eventually impact the pseudo-range accuracy. Low frequency resolution causes large
frequency error. Hence the code doppler frequency error is large accordingly, which means there
may be considerable difference between the length of 1 ms local PRN code and 1 ms received
signal. Suppose Doppler frequency of the received signal is \( f_d \), but local frequency estimation
is \( f_p + f_e \). Then code rate of received signal is:

\[
f_{cr} = \frac{RF + f_d}{1540} \text{ Hz}
\]

(4)

Whereas code rate of local PRN code is:

\[
f_{ct} = \frac{RF + f_d + f_e}{1540} \text{ Hz}
\]

(5)

When we calculate the number of sampling points corresponding to 1 ms (1 code cycle) using
local PRN code rate, we get:

\[dots_{1ms} = f_s \times \frac{1540 \times 1023}{RF + f_d + f_e} \text{ dots}\]  

(6)

But in the received signal such number of points corresponds to:

\[\frac{RF + f_d}{RF + f_d + f_e} \text{ code cycle} < 1 \text{ code cycle}\]  

(7)

So the correlation is done between 1 local code cycle and less than 1 received code cycle. Since \(RF\) is 1575.42 MHz, far larger than \(f_d\) and \(f_e\), so this consideration can be ignored.

Beside the respect analyzed above, large left-over frequency after mixing with the local carrier will also decrease the overall value of the correlation according to (8)[7]:

\[I(n) = aD(n)R(\tau) \text{sinc}(f_e T_{coh}) \cos \phi_e\]

\[Q(n) = aD(n)R(\tau) \text{sinc}(f_e T_{coh}) \sin \phi_e\]  

(8)

Where \(I, Q\) are coherent integration values from the in-phase and the quadra-phase paths; \(a\) is signal amplitude. \(D(n)\) is the modulated data bits. \(R(\tau)\) is the autocorrelation of C/A code. The \(\text{sinc}\) function corresponds to the influence of the residual frequency error, which is determined by the product of frequency error and coherent integration time. With larger value of \(f_e\) and \(T_{coh}\) fixed, the integration is smaller. Hence reasonable value of \(M\) should be chosen to make a compromise between computing load and tracking precision.

**Simulation Results**

**Diagrams for Detection.** The acquisition and tracking is simulated on MATLAB. Doppler frequency is set to 219.945 Hz ignoring dynamics. And code phase is set to 3.8 chip. \(CN_0 = 40 \text{ dB/Hz}\). The acquisition process can be presented in Fig. 3.

The x-value of the peak is 3, which corresponds to the third frequency bin. The representative frequency of the third frequency bin is 200 Hz (1\(^{st}\) bin is 0 Hz, 2\(^{nd}\) bin is 100 Hz). The y-value of the peak is 9, which corresponds to 4.5 chip of code phase. The reason why the setting code phase is 3.8 chip but the detection code phase is 4.5 chip (the error is over \(\pm \frac{1}{4}\) chip) is the same as is described in III.C. We use nominal value of IF (4.123968 MHz) and code frequency (1.023 MHz) to choose the two 10 ms sequences. The 1\(^{st}\) 10 ms sequence’s start code phase is 3.8 chip. But due to the doppler frequency, it actually lasts for

\[\frac{IF \times \text{code length}}{\text{code_freq} \times (IF + \text{doppler})} \times \text{Num.ms} = \frac{4.123968 \text{MHz}}{1.023 \text{MHz}} \times \frac{1}{4123968 + 219.945} \times 1023 \times 10\]  

(9)

Remaining

\[0.01s - 0.009999467 s = 5.33 \times 10^{-7}s\]  

(10)

This \(5.33 \times 10^{-7}\)s is made up by the head of the 11th ms. So the 2\(^{nd}\) 10 ms doesn’t start from the start of the 11th ms, but the moment that has passed \(5.33 \times 10^{-7}\)s from the strat of the 11th ms. \(5.33 \times 10^{-7}\)s corresponds to

\[\frac{1023 \times 5.33 \times 10^{-7}}{\text{code_freq} \times (IF + \text{doppler})} = 0.545 \text{ chip}\]  

(11)

So the 2\(^{nd}\) 10 ms start code phase is \(3.8 + 0.545 = 4.345\) chip. Also the 2\(^{nd}\) correlation value is larger than that of the 1\(^{st}\). So the final code phase estimation is 4.5 chip. From the analysis above, we can not just set the code phase pull-in range as \(\pm \frac{1}{4}\) chip. it should be larger.

After acquisition, the estimated doppler frequency is 200 Hz. For bigger code phase pull-in range, the code phase shift is from \(-2\) chip to 2 chip centering on the code phase computed from
the last 1 ms, so that is about 64 sampling points. Because the size of the frequency bin is 1 KHz, for high spectrum resolution, choose $M = 1000$ for CZT so that 1 CZT point corresponds to 1 Hz.

Fig. 4 shows the 64*128 array after FFT which represents the correlation value on every code phase and every frequency bin. The peak is obvious and the triangle area meets the autocorrelation of the gold PRN code. The peak indicates the first frequency bin and the modification of -1 in code phase. It means the frequency tracking error is less than 1 KHz and the code phase error is $-\frac{1}{16}$ chip. So do CZT in [0,1000 Hz] to get Fig. 5. It shows that No.21 point has the biggest CZT results. Given that 1 point corresponds to 1 Hz, so the frequency error is 20 Hz. This result is quite accurate, because with the 200 Hz obtained from the acquisition stage, the total local frequency estimation is $200 + 20 = 220$ Hz, leaving just 0.055 Hz of frequency error.

**Long Period Simulation in Tracking.** Fig. 6 and Fig. 7 present frequency tracking error and code phase error for 30 ms respectively. It is obvious to see, the two residual errors from acquisition are removed just in 1 ms. From the second ms, the errors start to be steady near zero.

The reason why the frequency error stays 0.055 Hz all the time is still the fence effect of FFT. Only that CZT samples the spectrum more densely to diminish the rest of error caused by the fence effect. Remember that the CZT frequency resolution is 1 Hz here.
Summary and Future Research

Primary analysis and simulation have shown that the tracking structure based on matched filter and FFT can achieve a faster response and will greatly shorten the pull-in time. The whole acquisition and tracking structure can be implemented in hardware with the FFT module multiplexed by acquisition, tracking and CZT. So there is potentiality of saving resources.

Besides the advantages mentioned above, there is research showing that this parallel correlate method improves tracking sensitivity [6]. In weak signal circumstances, SN0 is too small to keep the traditional tracking loop locked. The main approach to improve sensitivity is extending the coherent integration time. But in traditional tracking loop, it will cause the loop update rate being too low to keep tracking in dynamic circumstances. Since the extension of the coherent integration time is irreplaceable for improving the sensitivity, the introduction of parallel correlation can save a lot of time from another respect. Matched filter and FFT are such methods to compute all the correlations in frequency dimension and the code phase dimension at the same time. Further research will be put on improving the sensitivity performance of this structure.

References


