A Large Doppler Weak Direct Spread Signal Acquisition Algorithm and Optimization Method

Zhicheng Liu*, Ju Wang*
School of Information and Electronics, Beijing Institute of Technology, Beijing, China

*Corresponding author e-mail: 584528581@qq.com, wangju@bit.edu.cn

Abstract. For satellite receivers, on the one hand, large Doppler frequency will cause accumulation loss for a long-term accumulation acquisition algorithm, which leads to the problem of correlation peak broadening; on the other hand, searching within a larger Doppler uncertainty range will consume more logic resources. Thus, this paper first analyses the impact of large Doppler frequency on coherent accumulation and non-coherent accumulation based on the traditional Double Block Zero Padding (DBZP) algorithm. Secondly, an improved DBZP algorithm is designed in combination with the Keystone transform of the transform domain, and simplify the variable-scale DFT to reduce the amount of calculation. Then, the key parameters of the algorithm are extracted to establish the optimization objective function, the algorithm parameters are optimized through genetic algorithm without reducing the performance of the algorithm. Finally, the simulation verifies that the optimized and improved DBZP algorithm can effectively solve the problem of code phase walking.

1. Introduction
More and more companies choose to develop low-altitude satellite networks because low-orbit satellites have the advantages of low transmission delay, high reliability, and abundant orbit resources. But low-orbit satellites also have the characteristics of fast flight speed and short residence time. The fast flight speed will cause the signal to have a larger Doppler frequency, which will lead to the pseudo code phase walk and accumulated peak broadening during the accumulation process; short residence time requires the receiver to be able to acquire the signal in a shorter time.

The DBZP algorithm [1] [2] [3], based on two-dimensional parallel search, is widely used in the acquisition of satellite signals because of its high computational efficiency and suitability for long-term accumulation. However, the traditional DBZP algorithm does not compensate for the code phase walk, so in the case of large Doppler frequency, the traditional DBZP algorithm will have serious accumulation losses.

An improved DBZP algorithm based on Keystone transformation of Knab interpolation method which can compensate for the code phase walk was proposed by Haikun Luo and other people [4]. However, the performance of Keystone transformation of interpolation method is affected by the number of interpolation points, and in engineering applications, the interpolation algorithm usually does not have a ready-made integrated operation module, so it is more complicated to implement the algorithm. This paper uses a simplified Keystone transformation of transform domain [5] [6] to improve the traditional DBZP algorithm. The simplified Keystone transformation of transform domain uses FFT fast
convolution operation instead of variable-scale DFT, which avoids a large number of operations that calculate DFT point by point. Moreover, many digital signal processing platforms provide ready-made FFT operation modules, so the simplified Keystone transformation of transform domain is simpler in engineering implementation.

Because the receiver of the low-orbit satellite needs to acquire the signal in a shorter time, it means that the algorithm needs more parallel processing in the engineering implementation, and this will make the logic resources more insufficient. Jun Wang and other people proposed an optimization method of acquisition algorithm based on genetic algorithm [7], this paper uses genetic algorithm to optimize the algorithm parameters to reduce resource consumption without reducing its performance.

2. Analysis of accumulated loss caused by large Doppler in DBZP algorithm

In the satellite receiver, the signal mathematical expression after orthogonal down convert frequency and sampling is:

\[
s(nT_s) = AD\left(nT_s - \tau + \frac{f_d}{f_c}nT_s\right)C\left(nT_s - \tau + \frac{f_d}{f_c}nT_s\right) \exp(j2\pi f_d nT_s + \varphi) + N(nT_s)
\] (1)

In formula (1), \(A\) is the signal amplitude, \(f_s\) is the sampling frequency, \(T_s\) is the sampling interval, \(D(t)\) is the data code, \(C(t)\) is the spreading code, \(f_c\) is the carrier frequency, and \(f_d\) is the Doppler frequency, \(\tau\) and \(\varphi\) is the code phase delay and carrier phase delay caused by signal transmission, \(f_d nT_s/f_c\) is the code phase delay caused by Doppler, \(N(nT_s)\) is complex Gaussian noise.

First, the DBZP algorithm divides the baseband signal within the coherent accumulation time \(T_{coh}\) into \(K\) blocks of size \(N\) sampling points. The signal of formula (1) can be written as:

\[
S(n,k) = AC\left(1 + \frac{f_d}{f_c}\right)(nT_s + kNT_s) - \tau \exp(j2\pi f_d(nT_s + kNT_s) + \varphi) + N(nT_s)
\] (2)

In formula (2), \(n = 0, 1, ..., N - 1\) means the number of points in the block; \(k = 0, 1, ..., K - 1\) means the number of blocks. The local pseudo code can also be written as:

\[
C(n, k) = C((nT_s + kNT_s) - \hat{\tau})
\] (3)

In formula (3), \(\hat{\tau}\) is the initial phase value of the locally regenerated pseudo code.

Secondly, according to the DBZP algorithm, the signal and the pseudo code are combined into a 2N-point double block, double block using 2N-point FFT to perform fast correlation operations on double block signals and pseudo code, the correlation results discard the \(N\) points at the end to obtain a partial correlation result matrix of \(K\) rows \(\times\) \(N\) columns. Then, the coherent accumulation result can be obtained through performing FFT operations on each column of partial correlation result matrix. Finally, the result of non-coherent accumulation can be obtained through sum up the squares of the results of several coherent accumulations.

The following will analyze the impact of Doppler on coherent accumulation and non-coherent accumulation in the traditional DBZP algorithm.

2.1. The effect of large Doppler on coherent accumulation

Without considering the accumulation of noise, the expression of some correlation results is

\[
R_{Part,corr}(n,k) = AR(\hat{\tau}) \exp(j2\pi f_d kNT_s + \varphi) \frac{\sin(\pi f_d T_s N)}{N \sin(\pi f_d T_s)}
\] (4)

Using FFT to operate the partial correlation result by column in formula (4) can obtain:
In formula (5), \( \exp(\varphi) \) is the residual phase, \( \hat{R}(\tilde{\tau}) \) is the normalized autocorrelation function of the pseudo code, \( \sin(\pi f_d T_s N) / N \sin(\pi f_d T_s) \) is the coherent accumulation loss caused by Doppler frequency. The pseudo code normalized autocorrelation function \( R(\tilde{\tau}) \) can expand to:

\[
R(\tilde{\tau}) = \frac{1}{N} \sum_{k=0}^{N-1} c \left( \left( 1 + \frac{f_d}{f_c} \right) (nT_s + kNT_s) - \tilde{\tau} \right) c' (nT_s - \tilde{\tau})
\]

Thus, the phase difference between the signal pseudo code phase and the locally regenerated pseudo code phase is:

\[
\Delta\tau = \frac{f_d}{f_c} (nT_s + kNT_s) - \tau + \tilde{\tau}
\]

In formula (7), \( \tau \) and \( \tilde{\tau} \) are constant when the receiver receives the signal. Because the time of a single block signal is short, the code phase walk mainly affects between blocks, so formula (7) can be simplified as:

\[
\Delta\tau \approx \frac{f_d}{f_c} kNT_s - \tau + \tilde{\tau}
\]

From formula (8), the peak value of each block in the partial correlation result matrix appears at the same code phase \( \Delta\tau \approx -\tau + \tilde{\tau} \) when the Doppler frequency is small, so the correct coherent accumulation result can be obtained through performing the FFT operation by columns. But in the case of larger Doppler, the position of the peak of partial correlation results change with the increase of \( k \), so it will inevitably cause accumulation losses if the FFT operation is performed directly by columns.

2.2. The effect of large Doppler on non-coherent accumulation

In the DBZP algorithm, the process of non-coherent accumulation is to accumulate the results of several coherent accumulations. It is known that the phase of the pseudo code will walk with time, and because the initial time of each coherent accumulation is different, so there will be a fixed code phase deviation between the two coherent accumulation results. The following analyzes the amount of pseudo code phase walk in the non-coherent accumulation process.

Code Doppler's calculation formula is:

\[
f_{\text{code, d}} = \frac{v}{c} f_{\text{code}} = \frac{f_d}{f_c} f_{\text{code}}
\]
It can be known from formula (10) that when the system sampling rate $f_s$ and carrier frequency $f_c$ are confirmed, the amount of code phase walk is only related to the Doppler frequency. Thus, the number of sampling points for each non-coherent cumulative initial code phase offset is:

$$N_{\text{Code, phase}} = \frac{mN_{\text{coh}}T_s}{T_{\text{code, f}}}, \quad m = 0, 1, ... , N_{\text{coh}} - 1$$ (11)

In formula (11), $N_{\text{coh}} = NK$ is the number of coherent accumulation points, $m = 0, 1, ... , N_{\text{coh}} - 1$ is the number of times of non-coherent accumulations. It is necessary to compensate code phase walk according to formula (11) before non-coherent accumulation.

3. An improved DBZP implementation based on Keystone transform

The traditional Keystone transform in the transform domain is implemented by variable-scale DFT + IFFT, and its mathematical expression is:

$$R_{\text{Part, Corr}}(f, l) = \sum_{k=0}^{K-1} \left( \sum_{k=0}^{K-1} R_{\text{Part, Corr}}(f, k) e^{-j \frac{2\pi k}{K} (f + f_{\text{st}})} \right) e^{j \frac{2\pi k}{K} l}$$ (12)

In formula (12), the formula in brackets is a variable-scale DFT transformation, and this operation must be calculated point by point. Operations outside the brackets can be directly calculated using IFFT. The following expands the operations in brackets of formula (12):

$$R_{\text{Part, Corr}}(f, k') = \sum_{k=0}^{K-1} R_{\text{Part, Corr}}(f, k) e^{-j \frac{\pi f + f_{\text{st}}}{f_c} (k^2 + k'^2 - k^2)}$$

$$= e^{-j \frac{\pi f + f_{\text{st}}}{f_c}} \sum_{k=0}^{K-1} R_{\text{Part, Corr}}(f, k) e^{-j \frac{\pi f + f_{\text{st}}}{f_c} k'^2} e^{j \frac{\pi f + f_{\text{st}}}{f_c} k^2}$$ (13)

In formula (13), the expression in $\sum(\cdot)$ can be equivalent to a convolution operation:

$$R_{\text{Part, Corr}}(f, k') = e^{-j \frac{\pi f + f_{\text{st}}}{f_c} k'^2} \left[ R_{\text{Part, Corr}}(f, k) e^{j \frac{\pi f + f_{\text{st}}}{f_c} k^2} \right]$$

$$= e^{-j \frac{\pi f + f_{\text{st}}}{f_c} k'^2} \times \text{IFFT} \left[ \text{FFT} \left( R_{\text{Part, Corr}}(f, k) e^{-j \frac{\pi f + f_{\text{st}}}{f_c} k^2} \right) \times \text{FFT} \left( e^{j \frac{\pi f + f_{\text{st}}}{f_c} k^2} \right) \right]$$ (14)

Formula (14) is simplified to avoid the point-by-point calculation of variable-scale DFT. The Keystone transformation is completed after performing an IFFT operation on formula (14).

The schematic diagram of the improved DBZP algorithm based on the transformation domain Keystone transformation is shown as below:
As shown in Figure 1, the improved DBZP algorithm adds the Keystone transform module after the complex multiplier and before the partial correlation result IFFT. The algorithm needs to perform operations on the matrix rows before and after the keystone transformation. During the Keystone transformation, the algorithm needs to perform FFT operations on the columns of matrix, so the partial correlation result matrix needs to be cached twice before and after the Keystone transformation.

The following analysis amount of the reduction in calculation after simplified Keystone transformation. All data to be processed in the Keystone transformation is a partial correlation result matrix of $K \times 2N$ columns, where $K$ and $N$ are both integer powers of 2. If using variable scale DFT for point-by-point calculation, the data at each point of the matrix needs to perform $K$ complex multiplications and $K-1$ complex additions, and each column of matrix needs to perform a $K$ point IFFT operation, to complete all the data operation requires a total of $K \times 2NK + NK \times \log_2(K)$ complex multiplications and $K-1 \times 2NK + 2NK \times \log_2(K)$ complex additions. In the simplified Keystone transformation, each point of data in the matrix needs to perform 3 multiplications, and each column of data in the matrix needs to perform a $2K$ point FFT operation and a $2K$ point IFFT operation, to complete all the data operation requires a total of $3 \times K \times 2N + 4NK \times \log_2(2K)$ complex multiplications and $8NK \times \log_2(2K)$ complex additions.

It can be seen in Figure 2 that the simplified Keystone transform calculation is significantly reduced. And the larger the $N$ and $K$ is, the lower the calculation amount is.

4. Algorithm parameter optimization
The optimization objective function is:
In Formula (15), \( N \) is the points in the block, \( K \) is the number of blocks, \( N_{\text{FFT}} \) is the number of FFT points for frequency discrimination, \( N_{\text{COH}} \) is the number of non-coherent accumulation, \( P_{\text{fa}} \) is the false alarm probability, \( A \) and \( B \) is the Tang detection parameters, \( M \) is the number of parallel processing links. \( \text{num}_{\text{mul}} \) is the number of complex multiplications, \( \text{num}_{\text{add}} \) is the number of complex additions, \( \lambda_1 \) and \( \lambda_2 \) are weighting coefficients. \( P_{\text{f_all}} \) is the algorithm's total false alarm probability of the algorithm, \( P_{\text{fgoal}} \) is the request false alarm probability, \( P_{\text{d_all}} \) is the algorithm's total detection probability, \( P_{\text{dgoal}} \) is the request detection probability, \( T_{\text{a_all}} \) is the algorithm's total acquisition time, and \( T_{\text{aggoal}} \) is the request acquisition time.

The problem become a single objective optimization problem by weighting coefficients. The constraints of detection probability, false alarm probability and acquisition time in the optimization function are implemented by the penalty function. Because the optimization function contains 8 variables and the relationship between the variables is complex, we turn to the genetic algorithm toolbox provided by MATLAB.

Assume that the acquisition algorithm requirements are: the code phase uncertainty range is 10ms, the Doppler frequency uncertainty range is 500kHz, the system sampling frequency is 81.92MHz, the input signal-to-noise ratio is -40dB, the required false alarm probability is less than \( 10^{-7} \), the required detection probability is greater than 90\%, the required capture time is less than 20 seconds.

Uncertainty range of pseudo code phase and Doppler frequency is too large to achieve a single accumulation to complete the search, thus, the entire uncertainty range needs to be divided into small one. We set \( \lambda_1 = 1, \lambda_2 = 1 \) to optimize for multiple objectives, the first optimization result is:

| \( N \) | \( K \) | \( N_{\text{FFT}} \) | \( N_{\text{COH}} \) | \( P_{\text{fa}} \) | \( B \) | \( A \) | \( M \) |
|-------|-------|----------------|----------------|------------|-----|-----|-----|
| 1024  | 256   | 256            | 5             | 0.01       | 1    | 4    | 10  |

In the case of the parameters shown in Table 1, the false alarm probability of the algorithm is about \( 10^{-8} \), and the detection probability of the algorithm is 90.54\%. 800 searches are required to complete the code phase uncertainty range, 13 searches are required to complete the Doppler frequency uncertainty range, and the estimated acquisition time is about 10.19s when algorithm 10-way parallel processing. In total, \( 40,964 \times 10 \) complex multiplications and \( 81,923 \times 10 \) complex additions are required.

It can be seen that the acquisition time only takes half of the requirement, so the algorithm parameters are not optimal. It is difficult to ensure that the optimization goal of the genetic algorithm can quickly reach the optimal because there are many variables and the relationship between variables is complex, thus, it is necessary to use the last optimized population as the next initial population for iterative optimization.

The optimization results after five iterations of optimization are:

| \( N \) | \( K \) | \( N_{\text{FFT}} \) | \( N_{\text{COH}} \) | \( P_{\text{fa}} \) | \( B \) | \( A \) | \( M \) |
|-------|-------|----------------|----------------|------------|-----|-----|-----|
| 512   | 512   | 512            | 4             | 0.01       | 1    | 5    | 8   |

In the case of the parameters shown in Table 2, the false alarm probability of the algorithm is about \( 10^{-10} \), and the detection probability of the algorithm is 96.8\%. 1600 searches are required to complete the code phase uncertainty range, 7 searches are required to complete the Doppler frequency uncertainty range.
range, and the estimated acquisition time is about 10.19s when algorithm 8-way parallel processing. In total, 31,492×8 complex multiplications and 62,979×8 complex additions are required.

It can be seen from the data in Table 1 and Table 2 that the optimization objective function can get the algorithm parameters that meet the design requirements, and it can effectively reduce the amount of algorithm calculation after iterations of optimization.

5. Improved DBZP simulation

The improved DBZP algorithm is simulated by MATLAB, and its parameters are show in Table 2. The initial code phase is set to the 256th sampling point and the Doppler frequency is set to 80 kHz in the simulation.

![Figure 3. Partial correlation results and non-coherent accumulation results before and after code walk compensation.](image)

As shown in Figure 3. (a) and 3. (b), the code walk within the coherent accumulation time has been compensated after Keystone transformation. The accumulated peak broadened in Figure 3. (c) is effectively concentrated on the peak shown in Figure 3. (d) after Keystone transformation and incoherent walk compensation.

6. Conclusion

The main conclusions of this paper are as follows:
(1) The improved DBZP algorithm through Keystone transform of transform domain and non-coherent walk compensation can adapt to the working environment of large Doppler and weak signals.

(2) The parameter optimization method based on genetic algorithm can effectively reduce the calculation amount of the algorithm under the premise of ensuring the performance of the algorithm.

References

[1] Qunfang G E, Wei X. A Scheme for Acquiring C/A Code in High Dynamic Environment [J]. modern electronics technique, 2007.

[2] Qin XX, Han CD, Xie YK. A High Sensitive Fast Acquisition Algorithm Suitable to Implement in Software GPS Receiver [J]. Acta Electronica Sinica, Jan. 2010, 38 (1).

[3] Lin D M, Tsui J B Y, Comparison of acquisition methods for software GPS receiver, Salt Lake City, USA: ION, 2000.

[4] Luo H K, Wang Y Q, Shen Y Y, et al. Long-term integration algorithm for weak DSSS signal in large dynamic Doppler frequency scenarios [J]. Tien Tzu Hsueh Pao/acta Electronica Sinica, 2013, 41 (12): 2397-2402.

[5] Xiaofei S, Yujun Z, Xu Z. Weak Target Accumulation Detection Based on Keystone Transform and Implementation Methods in Engineering [J]. fire control radar technology, 2018.

[6] Juan W, Yongbo Z. Research on Implementation of Keystone Transform [J]. fire control radar technology, 2011.

[7] Jun W, Xin S, Li W. Application of Integer Programming Genetic Algorithm in P Code Direct Acquisition Method [J]. transactions of beijing institute of technology, 2013.

[8] Kun L, Hu-Li S. Analysis and Simulation of High Sensitivity GPS Acquisition Techniques [J]. gnss world of china, 2007.

[9] C. O’Driscoll. Performance Analysis of the Parallel Acquisition of Weak GPS Signals [D]. Ireland :The National University of Ireland, 2007, 26-37

[10] D. Borio. A Statistical Theory for GNSS Signal Acquisition [D]. Italy: Tesi di Dottorato, 2008, 22-43