# Implementation of GPS Software Receiver Based on GNU Radio

Yuekun Pei, Haowei Chen, Bingnan Pei
College of Information Engineering
Dalian University
Dalian, China
peiyuekun@dlu.edu.cn

*Abstract*—**It's difficult for traditional hardware GPS receivers to expand from single-band to multi-band and from single-constellation to multi-constellation. In this work, the idea of software defined radio is applied and the technical solution of the GPS signal processing based on the open source software radio frame GNU Radio is proposed. GPS signal acquisition, tracking, navigation message demodulation and positioning are realized with C++, multi-channel parallel processing is achieved exploiting the scheduler module provided by GNU Radio, and it's shown that the design is able to run real time GPS signal processing. The software prototype meets the requirements of flexibility and reusability, and the positioning accuracy is proved to be comparable with the commercial hardware receiver. This work benefits the research and design of multi-band multi-constellation GNSS receivers.**

***Keywords—GNU radio; GPS; software receiver;***

## I. INTRODUCTION

GNU Radio is an open source frame brought up by Eric Blossom [1], which is used for software defined radio applications by providing signal processing modules. Recently, GNU Radio combing USRP has been widely applied [2-4]. With the modernization of GPS and GLONASS systems, and the completion of the Galileo system and China's BeiDou satellite navigation system, several Global Navigation Satellite System (GNSS) will be operational in the next decade, and multiple civil signals will be broadcasted. Traditional hardware receivers are difficult to upgrade and adapt to the developing needs of multi-system, multi-constellation and modern GNSS. Multi-mode, intelligent, and software-based highly-flexible receivers are drawing increasing research interests.

Compared with the traditional hardware receivers, the software receivers are more flexible as the acquisition and tracking algorithms can be programmed and implemented in a general processor. the first GPS and Galileo dual-mode software receiver was designed and presented in [5]. The work done in [6] realized a GPS software receiver based on USRP, However, since the software modules are programmed using MATLAB for post-processing, the collected satellite data occupies a large amount of disk space. GPU's high-speed computing features were also used to implement a software receiver based on CUDA-related algorithms [7]. Generally, GNU Radio has the following advantages: (1) Compared with application-specific integrated circuits (ASICs), pure software signal processing has higher flexibility; (2) GNU Radio itself contains a variety of signal processing modules, such as FFT, modulation/demodulation, filters, etc. (3) GNU Radio has a multi-threaded scheduler that satisfies the Khan network model and enables data to be maximized in processing speed to meet real-time performance. In this paper, GNU Radio thread scheduling management and the provided signal processing module are exploited, and a software receiver scheme based on GNU Radio framework is proposed and designed, and finally the functional verification and simulation are performed.

## II. GPS SIGNAL PROCESSING

The GPS transmits signals on the carrier L1 (1575.42 MHz) and L2 (1227.6 MHz), and the transmitted navigation sequence is a direct spread spectrum sequence (DSSS) that has unique pseudo-random noise (PRN) spread-spectrum modulation. GPS L1 C/A signal is represented as

$$s^k(t) = \sqrt{2P_C}(C^k(t) \otimes D^k(t))\sin(2\pi f_{L1}t) \tag{1}$$

where $P_C$ is the transmission power of C/A code, $C^k$ is the C/A code sequence with the satellite number k, $D^k$ is the navigation sequence, and the carrier frequency of L1 frequency band is $f_{L1}$=1575.42 MHz.

The GPS signal needs to be down converted to intermediate frequency or base band before being processed by software module. In this work, a USRP device was used as the frontend for signal down conversion and sampling. The sampled signal needs to undergo despreading after acquisition and tracking phase to obtain the navigation message $D^k$, and the navigation message contains the satellite's position information and signal transmission time. Combining the time of the local receiver and using the pseudorange positioning principle, the receiver's location information can be obtained.

## III. GNU RADIO BASED SOFTWARE SOLUTION

GNU Radio applies C++/Python programming. In GNU Radio, Python provides a top-level abstraction module that implements complex interactions between modules, similar to block diagram design; C++ is used to implement high-performance underlying signal processing modules.

Based on GNU Radio, we defined all the software models shown in Fig. 1, providing a common interface for different algorithms to facilitate the expansion. In the figure, the solid triangle represents inheritance, the subclass inherits all methods from the parent class, and the dotted arrow represents dependencies, the classes on which the dependent class calls are dependent. This level establishes the definition of different algorithms and implementations that will be instantiated according to the configuration file. The software model allows multiple modules to share a common interface that defines a common approach, interface-oriented programming, and the algorithm can change independently of the program that uses it. Finally, there is an adapter class Adapter between the interface and the module responsible for implementing the processing algorithm inherited from the class gr_block, which is responsible for controlling the input and output between the GNU Radio module and the module so that they can be connected to the software flow diagram. In this way, if you want to use a block of the GNU Radio resource library, you only need to write the appropriate adapter class to be called by the corresponding signal processing module.

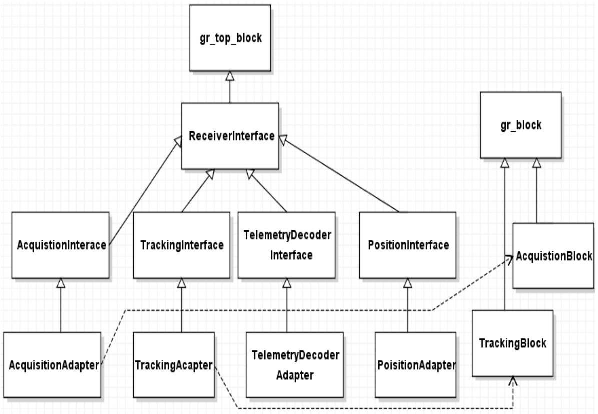


Fig. 1. GNU Radio-based receiver software model.

## IV. Experiment Results

### A. Experiment Setup

The experiment was conducted on the top of the building of Information Engineering College of Dalian University at 11:46 pm on October 5, 2017. The GNU Radio-based real-time GPS software receiver implemented in this paper consists of three parts: Comnav AT300 active antenna, USRP-2922, PC provided with Ubuntu 16.04 operating system and GNU Radio environment, with memory of 8G and Intel I5 processor.

### B. Results and Analysis

The time measured for the first-time positioning is shown in Table I, four fifth of the experiments required time within 50s and only one cost 73s.

The acquisition result of PRN 12 is shown in Fig. 2, in which the code delay is $\tau_{acq}$ =247.566chips, and Doppler frequency shift is $f_d$ =-1250Hz

TABLE I. First-time Positioning Time

| No. | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Time (s) | 34 | 73 | 44 | 43 | 35 |

Peaks shown in the figure imply that the correlation of the incoming signal and the local C/A code replica was aligned at the specific Doppler frequency and code delay. Once a correlation peak is found above the threshold, the corresponding PRN signal is captured and to be tracked afterwards.

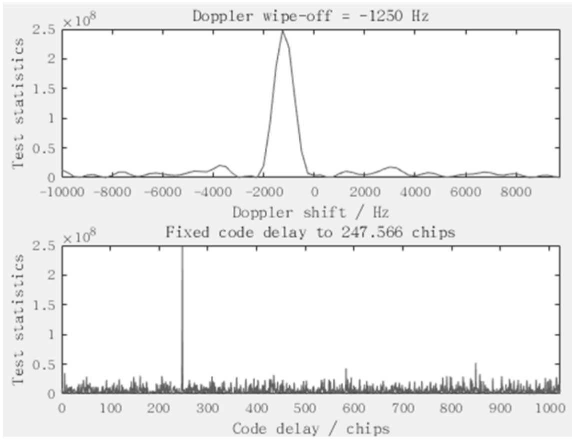


Fig. 2. Correlation peaks of Doppler shift (up) and code delay (down).

The tracking loop generates locally three code replicas: Early (E), Prompt (P) and Late (L), nominally with a spacing of ±0.5 chip. Through the DLL phase-detection output, the chip position is adjusted in real time to maximize the P-code correlation. Fig. 3 shows the three code correlations of the real-time processing tracking process. It can be seen that the correlation of the In-phase Early code is always at the maximum so that the navigation message can be subsequently demodulated.

After the demodulation of the navigation message, positioning results of 90s were plotted on Google Earth shown in Fig. 4. The polyline indicates the position variety every second.

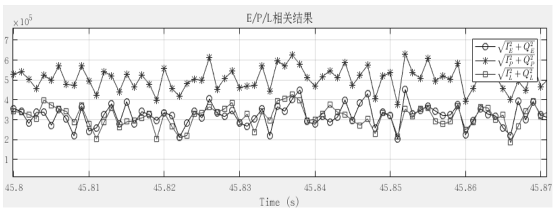


Fig. 3. E/P/L correlations in DLL-PLL tracking loop.

Fig. 4. Positioning solutions on Google Earth.

As shown in Fig. 5, 6 satellites were tracked, and the position information (39.10085853710839°, 121.81675011872109°, 123.959m) was solved. Comparing with the Comnav k528 hardware receiver, which tracked 7 satellites and gave the positioning results of (39.100893609°, 121.816776246°, 106.527m), the horizontal error is within 2m and the elevation error is about 20m. The experiment proves that the software receiver scheme is feasible and the performance is reliable.

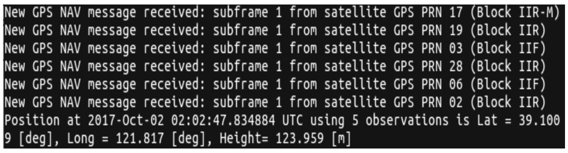


Fig. 5. Positioning outputs of the software receiver.

## V. Conclusions

This paper demonstrates a GPS software receiver scheme based on GNU Radio. The use of GNU Radio's scheduler multi-thread scheduling manager for multi-channel fast processing of baseband signals enables real-time positioning resolution. Meanwhile, C++ is used for object-oriented programming, which provides the software receiver with high extensibility and flexibility.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (61601076).

## References

[1] E. Blossom, “GNU Radio: tools for exploring the radio frequency spectrum. Linux Journal,” vol. 122, pp. 4, 2004.

[2] R. Gandhiraj, K. P. Soman, “Modern analog and digital communication systems development using GNU Radio with USRP,” Telecommunication Systems, 56(3), pp. 367-381, 2014.

[3] R. Zitouni, S. Ataman, M. Mathian and L. George, “Radio frequency measurements on a SBX daughter board using GNU radio and USRP N-210,” IEEE International Workshop on Measurements & Networking. IEEE, 2015, pp. 1-5.

[4] J. Demel, S. Koslowski, F. K. Jondral, “A LTE Receiver Framework Using GNU Radio,”. Journal of Signal Processing Systems, vol. 78(3), pp. 313-320, 2015,

[5] B. Kai, D. M. Akos, N. Bertelsen, P. Rinder and S. H. Jensen, “A Software-Defined GPS and Galileo Receiver,” Applied & Numerical Harmonic Analysis, pp. 1632-1637, 2005.

[6] E. A. Thompson, N. Clem, I. Renninger and T. Loos, “Software-defined GPS receiver on USRP-platform,” Journal of Network & Computer Applications, vol. 35(4), pp. 1352-1360, 2012.

[7] L. Xu, N. I. Ziedan, X. Niu and W. Guo, “Correlation acceleration in GNSS software receivers using a CUDA-enabled GPU,” GPS solutions, vol. 21(1), pp. 225-236, 2017.