



An efficient strategy for multi-GNSS real-time clock estimation based on the undifferenced method

Xingxing Li¹ · Yuhao Li¹ · Yun Xiong^{2,3} · Jiaqi Wu¹ · Hongjie Zheng¹ · Linyang Li¹

Received: 17 June 2022 / Accepted: 28 October 2022 / Published online: 18 November 2022
© The Author(s) 2022

Abstract

Precise satellite clock product is an indispensable prerequisite for the real-time precise positioning service. To meet the requirement of numerous time-critical applications, real-time satellite clock corrections need to be broadcast to users with an update rate of 5 s or higher. With the rapid development of global navigation satellite systems (GNSS) over the past decades, abundant GNSS tracking stations and modern constellations have emerged, and the computation for multi-GNSS real-time clock estimation has become rather time-consuming. In this contribution, an efficient strategy is proposed to achieve high processing efficiency for multi-GNSS real-time clock estimation, wherein undifferenced method based on sequential least square is adopted. In the proposed strategy, parallel data processing and high-performance matrix operations are introduced to accelerate the processing of multi-GNSS clock estimation. The former is based on OpenMP (Open Multi-Processing), while the latter is achieved by the implementation of the Schur complement and the open-source library OpenBLAS. Multi-GNSS observations from 85 globally distributed tracking stations are employed for the generation of real-time precise clock products. The average elapsed time per epoch with the proposed strategy is 0.35, 0.68, and 2.30 s for GPS-only, dual-system, and quad-system solutions, respectively. Compared to the traditional serial strategy, the computation efficiency is significantly improved by 76.0%, 77.3%, and 77.7%, respectively. The accuracy of the estimated clocks is evaluated with respect to IGS final GPS clock products and GFZ final multi-GNSS clock products (GBM0MGXRAP), and multi-GNSS real-time precise point positioning (PPP) experiments are further carried out. All the results indicate that the proposed strategy is efficient, accurate, and can promise high-rate multi-GNSS real-time clock estimation.

Keywords Multi-GNSS · Real-time clock estimation · Undifferenced method · Parallel computation · Sequential least square

Introduction

With the rapid development of global navigation satellite systems (GNSS) over the past decades, real-time precise point positioning (PPP) has extended to multiple fields, such as real-time atmospheric sounding (Li et al. 2014; Liu et al. 2018), tsunami prediction (Samper and Merino. 2013; Saito et al. 2016), earthquake monitoring and early warning

(Allen and Ziv. 2011; Li et al. 2013). Real-time precise orbit and clock products are indispensable prerequisites to support the real-time precise positioning service. Aiming at this, the International GNSS Service (IGS) initiated the Real-Time Working Group (RTWG) in 2001 and formally launched the Real-time Service (RTS) in 2013 (<http://www.igs.org/rts>). Since precise orbit can be predicted based on observations in the recent past with high precision, most of the efforts have been carried out in the aspect of real-time clock estimation (Ge et al. 2012; Dai et al. 2019). The target accuracy of real-time clock products is 0.3 ns (Weber et al. 2007; Hadas and Bosy. 2015). However, the predicted clocks of IGS ultra-rapid products can only reach a precision of 1.5 ns for GPS (<http://www.igs.org/products>). Therefore, quantities of globally distributed real-time stations are usually used in the real-time precise clock estimation (Dixon 2006; Hauschild and Montenbruck 2009), and the precision

✉ Yun Xiong
yxiong@gfz-potsdam.de

¹ School of Geodesy and Geomatics, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

² German Research Centre for Geosciences GFZ, 14473 Telegrafenberg Potsdam, Germany

³ Technische Universität Berlin (TUB), Straße Des 17. Juni, 10623 Berlin, Germany

of multi-GNSS real-time clock products is generally better than 0.2 ns compared to IGS final products (Liu et al. 2019; Cao et al. 2022).

Three primary methods are commonly used in real-time clock estimation, including the undifferenced (UD) method, epoch-differenced (ED) method (Mervart et al. 2008; Zhang et al. 2011), and mix-differenced (MD) method (Zhang et al. 2016; Chen et al. 2018). Both ED and MD methods use epoch-differenced carrier phase observations to remove large quantities of ambiguity parameters and speed up the computation. Although ED and MD methods are efficient to realize and can reach comparable accuracy with that of UD method, high computation performance is achieved at the loss of some useful information. The float ambiguity parameters, for example, can be further fixed to get more reliable clock estimation and improve the performance of real-time PPP (Li et al. 2019b; Dai et al. 2019). As to UD method, satellite clocks are estimated based on undifferenced phase and range observations, and all the useful information is kept. However, a large number of ambiguity parameters are estimated together with clock parameters, which leads to the heavy computation burden. In addition, to meet the needs of time-critical applications, real-time satellite clock corrections need to be broadcast to the users at an update rate of 5 s or higher. Nowadays, abundant GNSS tracking stations and modern constellations contribute to better observation geometry which ensures a higher clock precision and stability (Zuo et al. 2021), whereas the huge number of observations burden the computation and hamper its real-time applications. Improving the computation efficiency of UD method becomes a challenging issue to be urgently solved.

Benefitting from the development of computing technologies, parallel data processing and high performance of software libraries provide a new brainstorming for the GNSS network solutions (Cui et al. 2015; Li et al. 2019a). Gao et al. (2017) introduced the OpenMP (Open Multi-Processing) to accelerate the estimation of real-time GPS satellite clocks. Gong et al. (2018) improved the efficiency of square root information filter (SRIF) by taking advantage of the block QR factorization and decreasing the algorithm complexity of GNSS data processing. In the study of Kuang (2019), the sequential extended Kalman filter (EKF) with OpenMP was introduced to improve the efficiency of real-time GNSS orbit and clock estimation. Liu et al. (2019) developed an efficient UD method for precise clock estimation (PCE), wherein two-thread processing is adopted. The full-parameter thread is time-consuming and estimates all the parameters, including the large number of ambiguity parameters. The high-rate thread estimates the clock corrections only with phase observations, in which the latest ambiguity estimates and corresponding variance information are introduced as phase corrections. Fu et al. (2019) implemented an adapted online quality control strategy based on

sequential least square (LSQ) adjustment to meet the 5 s update rate of the IGS real-time clock product. In fact, with the formal commission of the global BeiDou navigation satellite system (BDS-3) in 2020, more satellites have become available for PCE and further burden the computation. Since the computation time should be shorter than the interval of the real-time clock products, it is a challenging work to finish the multi-GNSS clock estimation within a few seconds. Hence, it is essential to carry out more studies on improving the computation efficiency of multi-GNSS real-time clock estimation based on undifferenced observations.

In this contribution, an efficient strategy is proposed to achieve high processing efficiency for multi-GNSS real-time clock estimation, wherein the UD method based on sequential LSQ is adopted. In the proposed strategy, parallel data processing and high-performance matrix operations are introduced to accelerate the processing of multi-GNSS clock estimation. The former is based on OpenMP, since its API (Application Program Interface) gives programmers a simple and flexible interface for developing parallel applications on different platforms (<http://www.openmp.org>). The latter is achieved by the implementation of the Schur complement and the open-source library OpenBLAS (<http://www.openblas.net>).

After introducing the undifferenced real-time clock estimation observation model, the parallel processing strategy of PCE using sequential LSQ is presented and discussed. Subsequently, multiple experiments are conducted to evaluate the computation efficiency of the proposed strategy. The performance of precise clock products is then analyzed by clock comparison and real-time PPP validation. Finally, the summary and conclusions are drawn.

Methodology

This section introduces the undifferenced mathematical model for multi-GNSS clock estimation. Then, the principle of parameter elimination in sequential LSQ is presented. Finally, a parallel processing strategy based on OpenMP and a parallel processing procedure for multi-GNSS real-time clock estimation is summarized in detail.

Undifferenced model for multi-GNSS clock estimation

The original undifferenced GNSS observation equations for pseudorange $P_{r,j}^s$ and carrier phase $L_{r,j}^s$ can be expressed as follows,

$$P_{r,j}^s = \rho_r^s + c(t_r - t^s) + c(d_{r,j} - d^s) + I_{r,j}^s + T_r^s + \varepsilon_{P,r,j}^s \quad (1)$$

$$L_{r,j}^s = \rho_r^s + c(t_r - t^s) + \lambda_j(b_{r,j} - b_j^s) + \lambda_j N_{r,j}^s - I_{r,j}^s + T_r^s + \epsilon_{L,r,j}^s \tag{2}$$

where $s, r,$ and j refer to satellite, receiver, and frequency band, respectively; ρ_r^s is the geometric distance from the satellite to the receiver; c is the speed of light in vacuum; t^s and t_r are the clock offsets for the satellite and receiver in seconds; $d_{r,j}$ and d_j^s are the code bias of the receiver and satellite in seconds; $b_{r,j}$ and b_j^s are the phase delay of the receiver and satellite in cycles; λ_j denotes the wavelength of the carrier for frequency j in meters; $N_{r,j}^s$ denotes the integer ambiguity in cycles; $I_{r,j}^s$ is the ionospheric delay on the frequency band j in meters along the signal path; T_r^s is the slant tropospheric delay in meters; $\epsilon_{P,r,j}^s$ and $\epsilon_{L,r,j}^s$ denote the sum of observation noise, multi-path error, and other unmodeled errors for the pseudorange and carrier phase observations, respectively. The systematic error corrections such as the earth tides, ocean tide loading, relativistic delays, phase wind-up as well as phase center offsets (PCOs) and variations (PCVs) can be corrected according to the existing models (Schaer et al. 1999; Dach et al. 2006), and they are already applied in the equations.

In order to eliminate the first-order ionospheric delay, the ionosphere-free (IF) combination is widely used in real-time clock estimation. As for the tropospheric delay, its dry component is usually corrected with an a priori model (Li et al. 2019b). In this way, the IF observation model can be expressed as follows,

$$P_{r,IF}^s = \rho_r^s + c(t_{r,IF} - t_{IF}^s) + c(d_{r,IF} - d_{IF}^s) + m_r^s Z_r + \epsilon_{P,r,IF}^s \tag{3}$$

$$L_{r,IF}^s = \rho_r^s + c(t_{r,IF} - t_{IF}^s) + \lambda_{IF}^s (b_{r,IF} - b_{IF}^s) + m_r^s Z_r + \lambda_{IF}^s N_{r,IF}^s + \epsilon_{L,r,IF}^s \tag{4}$$

where $d_{r,IF}$ and d_{IF}^s refer to the IF code bias of receiver and satellite; $b_{r,IF}$ and b_{IF}^s are the IF phase delay of the receiver and satellite signal; $N_{r,IF}^s$ refer to the carrier phase IF ambiguity; Z_r is the zenith wet delay and m_r^s is the corresponding mapping function; $\epsilon_{P,r,IF}^s$ and $\epsilon_{L,r,IF}^s$ are the sum of IF observation noise, multi-path error, and other unmodeled errors for the pseudorange and carrier phase observations.

In clock estimation, the satellite orbits and station coordinates are held fixed. It should be noted that the satellite clocks are not estimable directly. Two types of the linear dependencies cause rank defects, the linear dependency between satellite clocks and receiver clocks (De Jonge 1998; Zhang et al. 2014) and the linear dependency between clocks and code hardware delays (Odijk et al 2016, 2017). To eliminate rank defects caused by the linear dependency between the satellite and receiver clocks, one clock datum should be chosen and the other clocks are aligned to the datum. In this study, one specific satellite clock is set as the datum and constrained according to the

broadcast ephemeris. To eliminate rank defects caused by the linear dependency between clocks and code hardware delays, satellite and receiver code biases are absorbed by the corresponding clock offset parameters, which is also an IGS recognized convention (Kouba and Héroux 2001; Dach et al. 2015). The phase delays are merged into the phase ambiguity. Hence, the observation model can be expressed as:

$$v_{Pc}^s = \delta \bar{t}_r - \delta t^s + m_r^s Z_r + \epsilon_{Pc} \tag{5}$$

$$v_{Lc}^s = \delta \bar{t}_r - \delta t^s + \lambda_{IF}^s \bar{N}_{r,IF}^s + m_r^s Z_r + \epsilon_{Lc} \tag{6}$$

where $\delta \bar{t} = c(t_{r,IF} + d_{r,IF})$ and $\delta t^s = c(t_{IF}^s + d_{IF}^s)$ refer to the receiver clock and satellite clock in meters, respectively; $\bar{N}_{IF}^s = N_{IF}^s + b_{r,IF} - b_{IF}^s - c(d_{r,IF} - d_{IF}^s) / \lambda_{IF}^s$ is the re-parameterized IF phase ambiguity; v_{Pc}^s and v_{Lc}^s denote "observed minus computed" (OMC) residuals of phase and pseudo-range observation, respectively.

In the multi-GNSS scenario, the measurements could be systematically biased from satellite to satellite due to the signal delay caused by hardware. For the GNSS using code division multiple access (CDMA) signals, including GPS, BDS, and Galileo, the inter-system biases (ISB) are added to handle the biases between different systems. As for the GLONASS satellites using frequency division multiple access (FDMA) signals, the biases between different satellites are parameterized as the inter-frequency bias (IFB). Using GPS time as time reference, the combined multi-GNSS clock estimation models can be expressed as:

$$\begin{cases} v_{Pc}^G = \delta \bar{t}_r - \delta t^{s,G} + m_r^s Z_r + \epsilon_{Pc}^G \\ v_{Pc}^E = \delta \bar{t}_r - \delta t^{s,E} + ISB^{E-G} + m_r^s Z_r + \epsilon_{Pc}^E \\ v_{Pc}^C = \delta \bar{t}_r - \delta t^{s,C} + ISB^{C-G} + m_r^s Z_r + \epsilon_{Pc}^C \\ v_{Pc}^{R_k} = \delta \bar{t}_r - \delta t^{s,R_k} + IFB^{R_k-G} + m_r^s Z_r + \epsilon_{Pc}^{R_k} \end{cases} \tag{7}$$

$$\begin{cases} v_{Lc}^G = \delta \bar{t}_r - \delta t^{s,G} + \lambda_{IF}^s \bar{N}_{r,IF}^{s,G} + m_r^s Z_r + \epsilon_{Lc}^G \\ v_{Lc}^E = \delta \bar{t}_r - \delta t^{s,E} + ISB^{E-G} + \lambda_{IF}^s \bar{N}_{r,IF}^{s,E} + m_r^s Z_r + \epsilon_{Lc}^E \\ v_{Lc}^C = \delta \bar{t}_r - \delta t^{s,C} + ISB^{C-G} + \lambda_{IF}^s \bar{N}_{r,IF}^{s,C} + m_r^s Z_r + \epsilon_{Lc}^C \\ v_{Lc}^{R_k} = \delta \bar{t}_r - \delta t^{s,R_k} + IFB^{R_k-G} + \lambda_{IF}^s \bar{N}_{r,IF}^{s,R_k} + m_r^s Z_r + \epsilon_{Lc}^{R_k} \end{cases} \tag{8}$$

where $G, E,$ and C refer to GPS, Galileo, and BDS systems, respectively; R_k is the GLONASS satellite with frequency factor k ; $ISB^{E-G} = d_{r,IF}^E - d_{r,IF}^G$; $ISB^{C-G} = d_{r,IF}^C - d_{r,IF}^G$; $IFB^{R_k-G} = d_{r,IF}^{R_k} - d_{r,IF}^G$. It should be noted that GLONASS code IFBs are usually neglected to simplify data processing, especially in real time (Cai and Gao 2013; Chen et al. 2017, 2018).

Sequential least square with parameter elimination

In GNSS data processing, the unknown parameter vector x can be divided into the active parameter vector x_1 and the inactive parameter vector x_2 . The Gauss-Markov model of the aforementioned observation equations can be expressed in the following as:

$$x = [x_1 \ x_2]^T \tag{9}$$

$$v = Ax_1 + Bx_2 + \epsilon, P \tag{10}$$

where A and B are the corresponding design matrices with full column rank of x_1 and x_2 ; v denotes OMC residuals of observation; ϵ is normally distributed observation error vector with zero mean; P denotes the weight matrix of observables.

In the sequential LSQ adjustment, inactive parameters, i.e., satellite and receiver clocks, and tropospheric delays, are usually estimated as white noise or random-walk process, and they will be eliminated before new parameters are added in the normal equation (NEQ). The NEQ of the least squares can be expressed as:

$$\begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix} \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} A^T P v \\ B^T P v \end{pmatrix} \tag{11}$$

where

$$N = \begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix} = \begin{pmatrix} A^T P A & A^T P B \\ B^T P A & B^T P B \end{pmatrix} \tag{12}$$

and \hat{x}_2 refers to the inactive parameters which will never be used for the observations afterward.

In fact, not all the inactive parameters are eliminated at the same epoch and parameters to be eliminated at current epoch are usually nonadjacent. Hence, eliminating parameters one by one is the traditional way in the sequential LSQ implementation. However, to fully exploit the power of modern CPUs, the granularity of the operations should be larger (Schreiber and Loan 1989; Demmel 1997). Figure 1 shows a typical multi-level memory of the cache-based processors. Data must be moved from memory layers lower in the pyramid into the registers for CPU to perform floating point operations. Since such movement represents unproductive overhead, reusing data that is up the memory pyramid many times before allowing it to slide back down becomes important (Low and van de Geijn 2004). Consequently, casting algorithms in terms of larger granularity operations, i.e., matrix–matrix multiplication, will achieve better performance for data processing (Gong et al. 2018). Concerning the parameter elimination in sequential LSQ, inactive parameters

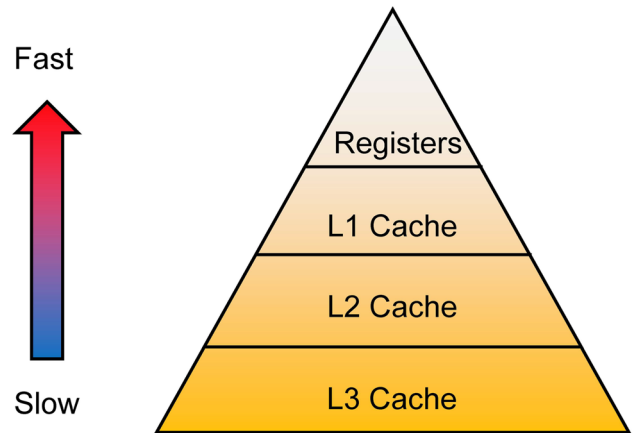


Fig. 1 Hierarchical memories of modern CPU viewed as a pyramid

at the current epoch are rearranged in a continuous row and eliminated thoroughly by the implementation of the Schur complement.

Note that N_{11} and N_{22} are square and invertible, the Schur complement of N_{22} in N is $N_{11} - N_{12}N_{22}^{-1}N_{21}$ (Gabe et al. 2010). Reducing the parameters \hat{x}_2 onto the parameters \hat{x}_1 gives,

$$\begin{pmatrix} N_{11} - N_{12}N_{22}^{-1}N_{21} & 0 \\ N_{21} & N_{22} \end{pmatrix} \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} A^T P v - N_{12}N_{22}^{-1}B^T P v \\ B^T P v \end{pmatrix} \tag{13}$$

Let $J = BN_{22}^{-1}B^T P$ and $\tilde{A} = (I - J)A$, then,

$$\begin{aligned} \tilde{N}_{11} &= N_{11} - N_{12}N_{22}^{-1}N_{21} \\ &= A^T P A - A^T P B N_{22}^{-1} B^T P A \\ &= A^T (I - J)^T P (I - J) A \end{aligned} \tag{14}$$

$$\tilde{N}_{11} \hat{x}_1 = \tilde{A}^T P v \tag{15}$$

$$\tilde{A} \hat{x}_1 = v \tag{16}$$

Inactive parameters are marginalized out, and active parameters are stored for the next epoch processing.

In addition, to further improve the performance of matrix operations, the open-source library OpenBLAS is implemented. Since the Basic Linear Algebra Subprograms (BLAS) is one of the most basic linear algebra kernels and provides numerous matrix–matrix operations with a portable interface (Low and van de Geijn 2004), OpenBLAS optimized level 3 BLAS (matrix–matrix operations; Wang et al. 2013). Compared to other open-source libraries commonly used in the traditional serial strategy, such as Eigen (<http://www.eigen.tuxfamily.org>),

OpenBLAS shows significant improvements in the following experiments.

Parallel processing strategy based on OpenMP

To fully exploit the power of modern CPUs with multi-core and multi-thread, parallel processing strategy based on OpenMP, multi-GNSS real-time clock estimation is implemented. Figure 2 shows the OpenMP parallel working

principle, namely a fork-join design pattern. OpenMP is an implementation of multithreading, a method of parallelizing whereby a primary thread (a series of instructions executed consecutively) forks a specified number of sub-threads and the system divides a task among them.

The parallel processing flowchart of multi-GNSS real-time clock estimation is summarized in Fig. 3. At the beginning, real-time observation data and broadcast ephemeris are derived from BKG Ntrip Client (BNC, <https://igs.bkg>.

Fig. 2 Illustration of the working principle of OpenMP

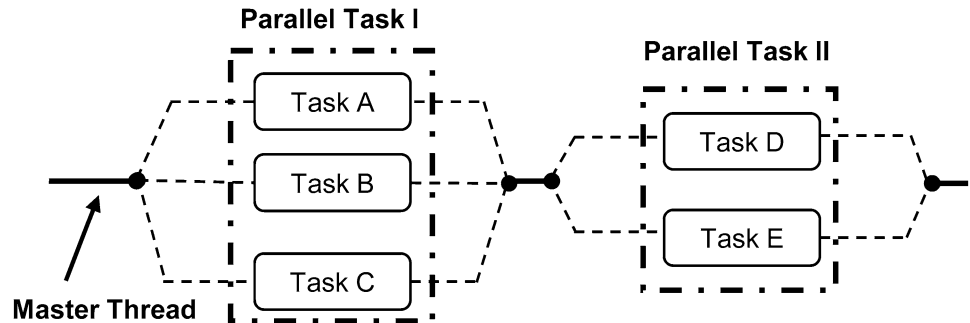
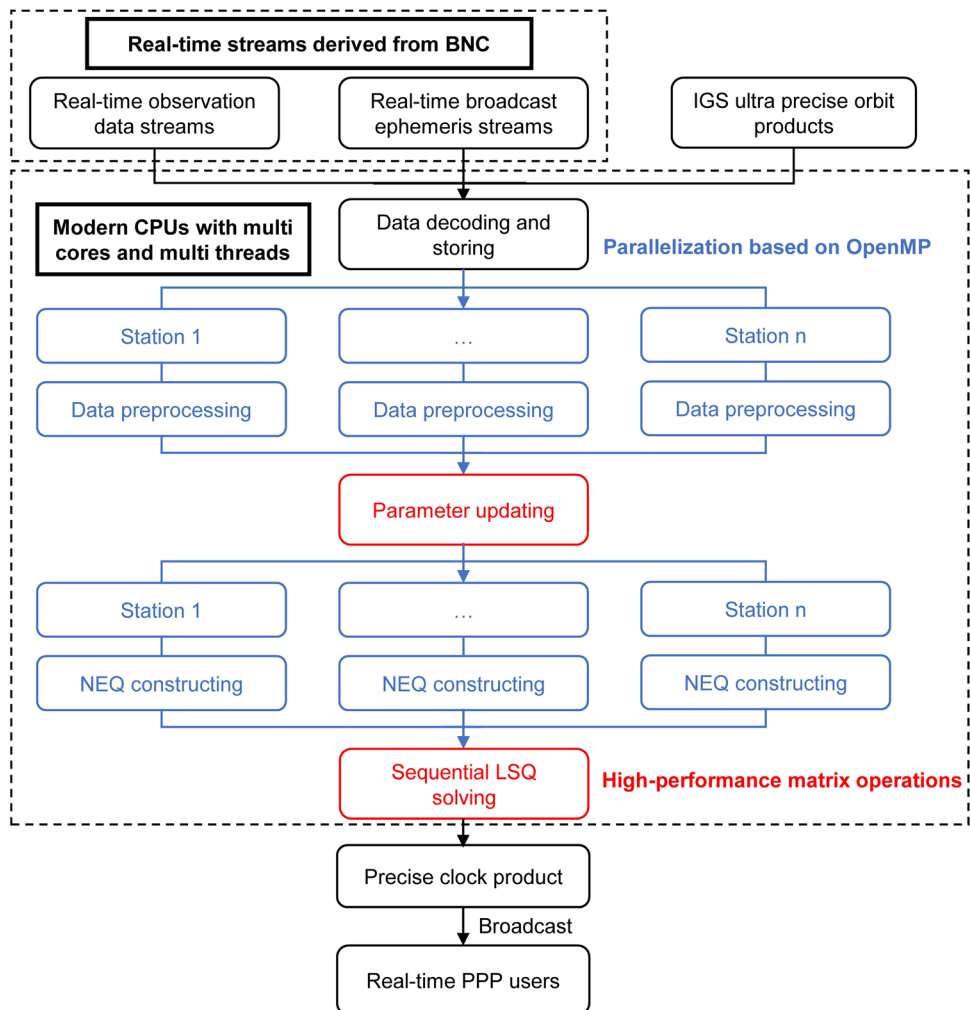


Fig. 3 Parallel processing procedure for multi-GNSS real-time clock estimation



bund.de/). The satellite orbits are retrieved and updated from IGS ultra-rapid (predicted half) products. Since epoch observations of different stations are independent, a parallel region based on OpenMP is created when performing data preprocessing and NEQ construction. In the step of data preprocessing, quality control is carried out to detect the gross errors and cycle slips. Next, in the parameter updating step, inactive parameters are marginalized out by the implementation of the Schur complement and new parameters are initialized and added in. Subsequently, parallelization based on OpenMP is introduced to accelerate the step of NEQ construction. As shown in Fig. 4, parallelization is implemented before the outer station loop in consideration of the overheads due to the synchronization. Thereafter, OpenBLAS matrix library contributes to the high performance of matrix operations and speeds up the step of sequential LSQ solving. It should be noted that OpenBLAS also supports multi-threading based on OpenMP to accelerate matrix operations.

```

#pragma omp parallel for
for all stations do
    for all satellites do
        Get epoch data from the shared memory;
        Calculate OMC residuals according to Eq.
(5) and Eq. (6);
        Construct the design and weight matrix;

```

Fig. 4 Pseudo-code of OpenMP to accelerate the step of NEQ construction

Finally, multi-GNSS real-time clock products are generated and broadcast to support real-time precise positioning service at the user side.

Software and hardware

In order to satisfy various demands of scientific and engineering applications in geodesy and navigation fields, the GREAT (GNSS + REsearch, Application and Teaching) software was designed and developed at the school of Geodesy and Geomatics of Wuhan University (Li et al. 2022). Written in standard C++ language and compiled with Object-Oriented principles, GREAT software can achieve multiple applications such as GNSS real-time orbit and clock determination, PPP-RTK (real-time kinematic), LEO augmentation, multi-sensor integration navigation, etc. The proposed strategy is implemented in real-time clock estimation with the GREAT software package in this contribution. Specifications of the processing strategy for multi-GNSS real-time clock estimation are summarized in Table 1. To verify the performance of the proposed strategy, we perform all computations on a cloud server. Detailed information about the hardware resources is listed in Table 2.

Experiments on computation efficiency

Series of experiments were designed to validate the effectiveness and reliability of the proposed computation strategy. First of all, the overall elapsed time per epoch of different parts in the traditional serial strategy of PCE is analyzed. Afterward, high-performance matrix operations based on the Schur complement and OpenBLAS library are added to accelerate the processing of parameter elimination and NEQ

Table 1 Detailed processing strategy for multi-GNSS real-time clock estimation

Items	Models
Basic observations	IF combinations
Frequency	GPS: L1&L2; GLONASS: L1&L2; BDS: B1&B3; Galileo: E1&E5a
Elevation angle	10°
Tropospheric delay	Corrected by Saastamoinen model (Saastamoinen 1972) with GMF mapping function (Boehm et al. 2006) The residual ZTD is estimated as piecewise constant with 2 h
Satellite orbits	Fixed to the ultra-rapid products provided by WUM
Station coordinate	Fixed to IGS weekly solutions
Phase wind-up effect	Corrected (Wu et al. 1993)
Relativistic and earth rotation effects	Corrected (Dach et al. 2015)
Satellite antenna PCO&PCV	Corrected with igs14.atx (Rebischung and Schmid 2016)
Receiver antenna PCO & PCV	Corrected with igs14.atx
Receiver clock	Epoch-wise estimated as white noise; estimate BDS, Galileo, and GLONASS ISBs relative to GPS

Table 2 CPU architecture information of the workstation

Items	Information
Architecture	X86-64
Platform	CentOS Linux release 7.4.1708 (Core)
Number of CPU	32
Thread(s) per core	1
Core(s) per socket	16
Socket number	2
NUMA node number	2
Model name	Inter(R) Xeon(R) Gold 6146 CPU @ 3.20 GHz
L1d cache size	32 K
L1i cache size	32 K
L2 cache size	4096 K
L3 cache size	28,160 K

solving, respectively. Then, parallel processing based on OpenMP is introduced to speed up data preprocessing and NEQ construction and further accelerate matrix operations. Finally, elapsed time with different computation strategies against the growth in the number of stations is compared and analyzed.

This contribution uses 85 globally distributed multi-GNSS stations with real-time streams. The distribution and supported constellations of selected tracking stations are

shown in Fig. 5. All stations provide GPS and GLONASS observations, while 83 and 78 stations can receive Galileo and BDS signals, respectively. The real-time streams are obtained by the BNC software in real time. This section uses observation data from DOY 50 to 52, 2022.

Elapsed time of the traditional serial strategy

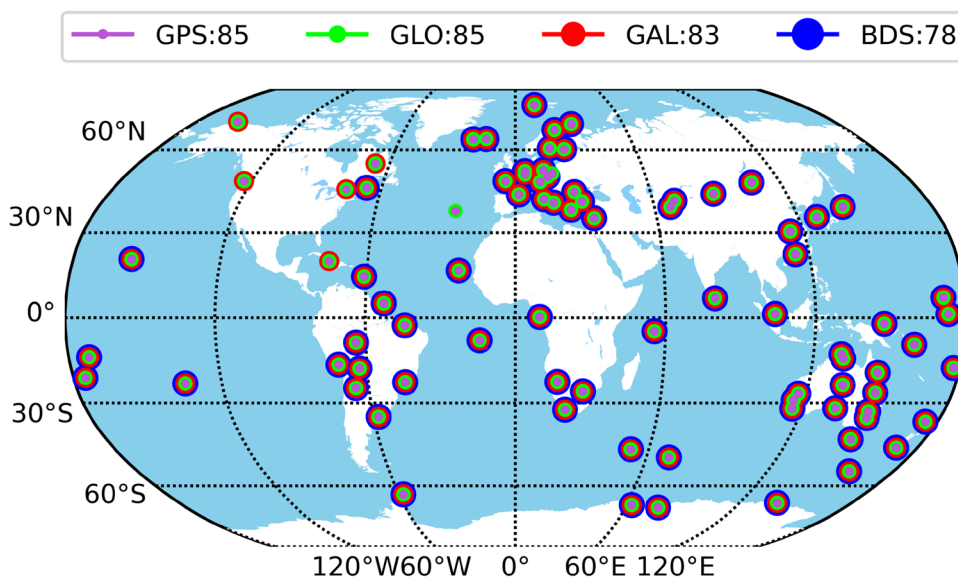
Figure 6 shows the overall elapsed time per epoch and average number of parameters for PCE using the traditional serial strategy with different GNSS constellations. With serial algorithm and open-source matrix library Eigen, it takes 1.46, 2.99, and 10.30 s to process the GPS-only, GE, and GREC systems for each epoch, respectively. It can be seen that, with the increase of GNSS satellites, the number of parameters and elapsed time increase rapidly.

Specifically, the overall computation time is subdivided into four parts: preprocessing, parameter elimination, constructing NEQ, and solving NEQ. Figure 7 shows the elapsed time and percentages of the four parts in GREC quad-system scene. Solving NEQ consumes most of the computation time and costs 6.06 s, while constructing NEQ, preprocessing, and parameter elimination costs 3.45, 0.33, and 0.46 s, respectively. Therefore, it is impossible for the traditional strategy to meet the 5 s update rate of the IGS real-time clock product.

Optimization of matrix operations

Considering the analysis mentioned above, solving NEQ takes most of the computation time, especially the inversion of the epoch-wise high-order NEQ. With the implementation of optimized Cholesky factorization, OpenBLAS can significantly accelerate the step of solving NEQ. Figure 8 shows

Fig. 5 Distribution and supported constellations of the selected IGS tracking stations (purple for GPS-capable stations, green for GLONASS, red for Galileo, and blue for BDS)



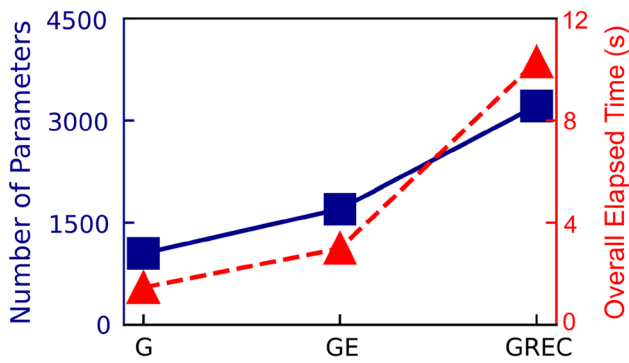


Fig. 6 Overall elapsed time per epoch and average number of parameters for PCE with different GNSS constellations using the traditional serial strategy

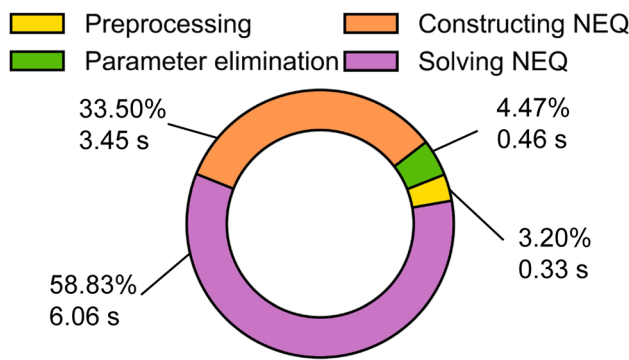


Fig. 7 Elapsed time and percentages of the four parts in GREC quad-system scene

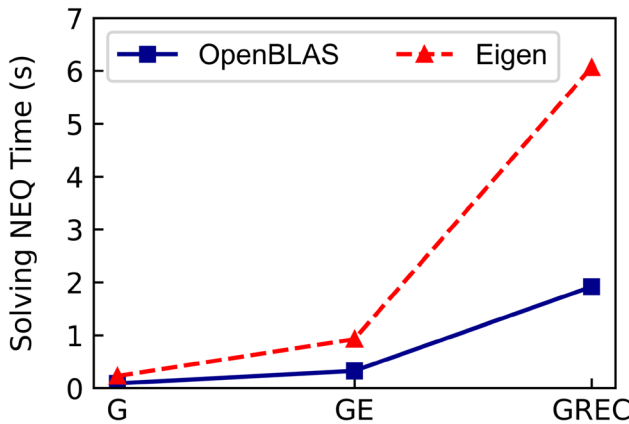


Fig. 8 Elapsed time of solving NEQ with different GNSS constellations using OpenBLAS and Eigen

the elapsed time of solving NEQ with different GNSS constellations using OpenBLAS and Eigen, respectively. With the increase of GNSS satellites, the elapsed time for solving NEQ using OpenBLAS grows much slower than that of using Eigen, which makes OpenBLAS more competitive in

Table 3 Elapsed time of parameter elimination with different GNSS constellations using the cycle scheme and matrix-remove scheme, respectively

GNSS constellation	Cycle scheme (s)	Matrix-remove scheme (s)	Number of parameters
GPS-only	0.02	0.02	1049
GE	0.07	0.06	1700
GREC	0.46	0.25	3212

the multi-GNSS scene. Compared with Eigen, the average time for the GPS-only, GE, and GREC using OpenBLAS is 0.09, 0.32, and 1.91 s, with an improvement of 63.6%, 65.2%, and 68.3%, respectively.

As for the step of parameter elimination, the inactive parameters from the last epoch are eliminated one by one in the traditional serial strategy, namely the cycle scheme. With the implementation of the Schur complement, the proposed computing strategy eliminates inactive parameters one time and shortens the processing time. Matrix-remove scheme is used to describe the proposed strategy for parameter elimination. Table 3 shows the elapsed time of parameter elimination with different GNSS constellations using the cycle scheme and matrix-remove scheme. It can be seen that the two schemes show comparable performance for GPS-only and GE. With the increase of GNSS satellites, inactive parameters increase and matrix-remove scheme shows much better performance than the cycle scheme, with a remarkable improvement of 45.7% in the quad-system scene.

OpenMP-based parallel optimization

In order to speed up the step of preprocessing and constructing NEQ, parallel processing based on OpenMP is utilized. Since the epoch observations of different stations are independent, parallel regions are created and OpenMP is implemented to fully exploit the power of modern CPUs with multiple cores and threads. Meanwhile, OpenBLAS also supports multithreading based on OpenMP to accelerate matrix operations.

Figure 9 shows the overall elapsed time for different GNSS constellations with various numbers of threads. It can be seen that the increase in the number of satellites burdens the computation, while the increase in threads can accelerate the processing. Besides, the improvement of parallelization is rather significant when the number of threads increases from 1 to 4, compared to that from 6 to 12. Taking the quad-system scene as an example, the former improvement reaches 51.2%, while the latter is only 10.8%. In the serial single thread scheme, the overall elapsed time is over 5 s and cannot meet the 5 s update rate of the IGS real-time clock product. When the number of threads reaches 12, the

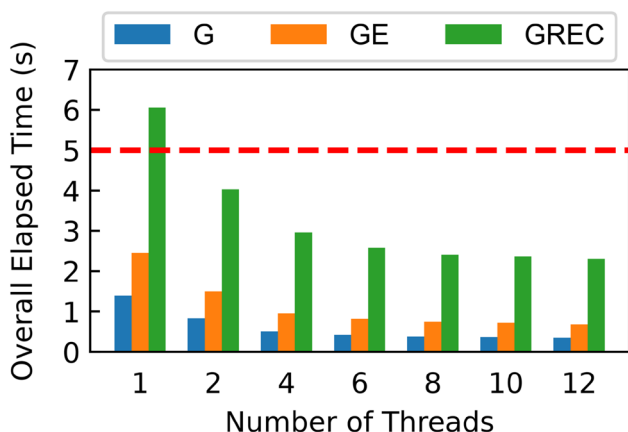


Fig. 9 Overall elapsed time for different GNSS constellations with various numbers of threads

Table 4 Elapsed time with the traditional strategy and the proposed strategy using 12 threads with different GNSS constellations

GNSS constellation	Traditional strategy (s)	Proposed strategy (s)	Improvement
GPS-only	1.46	0.35	76.0%
GE	2.99	0.68	77.3%
GREC	10.30	2.30	77.7%

Table 5 Components of elapsed time for GREC system with the traditional strategy and the proposed strategy using 12 threads

Components of elapsed time	Traditional strategy (s)	Proposed strategy (s)	Improvement
Preprocessing	0.33	0.24	27.3%
Parameter Elimination	0.46	0.16	65.2%
Constructing NEQ	3.45	0.52	84.9%
Solving NEQ	6.06	1.38	77.2%
Total	10.30	2.30	77.7%

average time per epoch is less than 2.5 s, with a remarkable improvement of 62.0%.

To be more intuitionistic, Table 4 summarizes the performance of the traditional serial strategy and the proposed strategy with 12 threads in single-, dual-, and quad-system scenarios. Compared to the traditional serial strategy, the computation efficiency is significantly improved by 76.0%, 77.3%, and 77.7%, respectively.

Table 5 shows the components of elapsed time for GREC systems with the traditional serial strategy and the

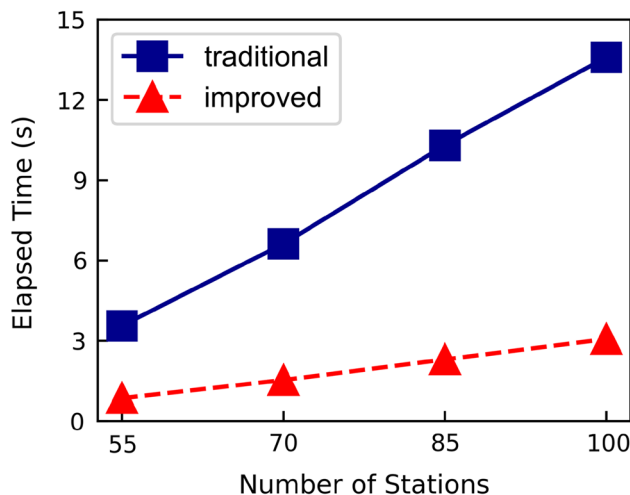


Fig. 10 Elapsed time of quad-system clock estimation with 55-, 70-, 85-, and 100-station networks based on two computing strategies

proposed strategy using 12 threads. With the proposed strategy, elapsed time for preprocessing, parameter elimination, constructing NEQ and solving NEQ is improved by 27.3%, 65.2%, 84.9%, and 77.2%, respectively. Multi-GNSS real-time clock estimation can meet the 5 s update rate of the IGS real-time clock product.

Computation efficiency with different number of stations

Abundant GNSS networks contribute to better observation geometry and ensure a higher clock precision and stability. However, such a huge number of observations greatly burdens the processing. To further assess the computation efficiency of multi-GNSS clock estimation with the proposed strategy, the average processing time for different numbers of stations is compared and analyzed. Additional 15 multi-GNSS stations are utilized to enrich the experiment results. Figure 10 shows the elapsed time of quad-system clock estimation with 55-, 70-, 85-, and 100- station networks based on the traditional serial and proposed strategies. When the number of stations is more than 70, the elapsed time with the traditional serial strategy is over 6 s. By contrast, the elapsed time using the proposed strategy grows quite slowly with the increase of stations. The computation efficiency is significantly improved by 76.1%, 76.9%, 77.7%, and 77.5%, respectively. When the network size reaches 100 stations, the elapsed time with the proposed strategy is 3.06 s and can still meet the 5 s update rate of the IGS real-time clock product.

Evaluation of real-time clock products

In this section, the accuracy of real-time clock products is evaluated, and real-time PPP validation experiments are performed to further verify the reliability of the proposed method. All data processing for precise clock estimation and PPP validation are fully based on real-time data streams.

Clock comparison

To evaluate the accuracy of multi-GNSS real-time clock products, the derived clock estimates are compared with the IGS final GPS products and the GFZ final multi-GNSS products (GBM0MGXRAP). The comparison lasts from DOYs 50–52, DOYs 164–166, and DOYs 250–252, 2022. The clock time series are aligned to a reference satellite in order to eliminate the systematic errors caused by the clock datum. G01, R01, E02, and C21 are selected as the reference

Fig. 11 STDs of the clock difference with respect to IGS final GPS products

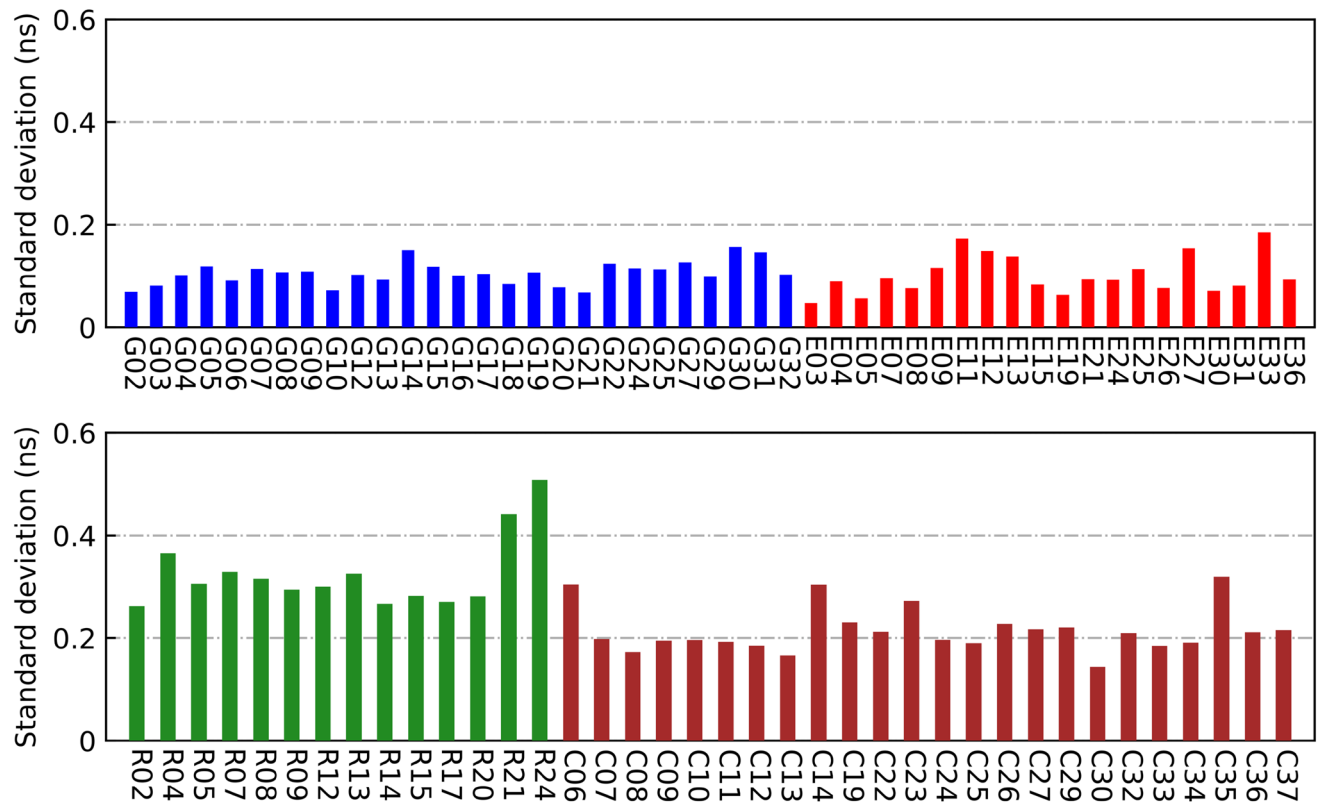
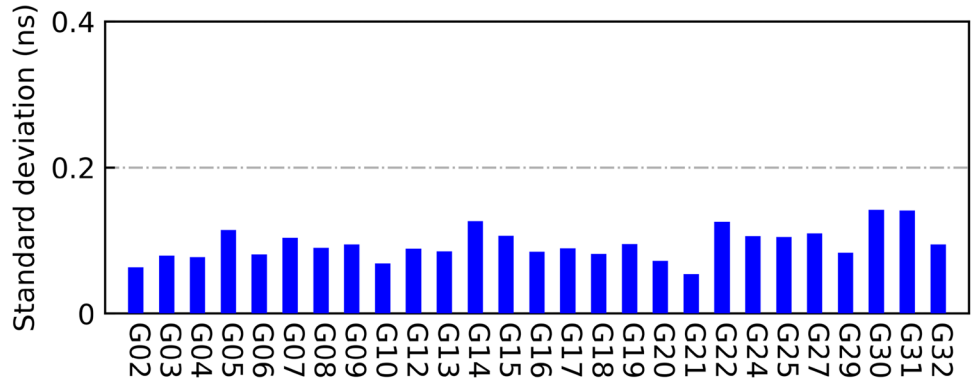


Fig. 12 STDs of the clock difference with respect to GFZ final multi-GNSS products (GBM0MGXRAP) for different constellations (blue for GPS, red for Galileo, green for GLONASS, and brown for BDS)

clock for GPS, GLONASS, Galileo, and BDS, respectively. Note that the performance of orbit products, especially the radial component, greatly influences on the performance of real-time clock estimation. The radial orbit differences between the different orbit products used in the proposed solution and IGS post-processing products, are removed. Figures 11 and 12 present the average STDs of the clock differences with respect to IGS final GPS products and GFZ final multi-GNSS products, respectively.

For GPS satellites, the STDs are all smaller than 0.16 ns and the average value is 0.095 ns and 0.106 ns compared to IGS final GPS products and GFZ final multi-GNSS products, respectively. Comparable to that of GPS, the average STD of Galileo satellites is 0.102 ns. Due to the poor quality of GLONASS orbits, the average STD is only 0.325 ns and shows the worst performance. In terms of BDS satellites,

the STDs are generally smaller than 0.25 ns and the average STD is 0.215 ns. The results indicate that multi-GNSS real-time products generated by the proposed strategy are of high quality and have a good agreement with the IGS post-processing products.

Real-time PPP validation

In this contribution, real-time clock and orbit corrections are encoded to state-space representative (SSR) format and broadcast through the Ntrip protocol to support the real-time PPP service. In order to further validate the positioning performance of real-time clock products, 30 multi-GNSS stations are selected to perform the real-time PPP experiments and the distribution of the stations is shown in Fig. 13.

Fig. 13 Distribution of 30 multi-GNSS stations selected for real-time PPP experiments

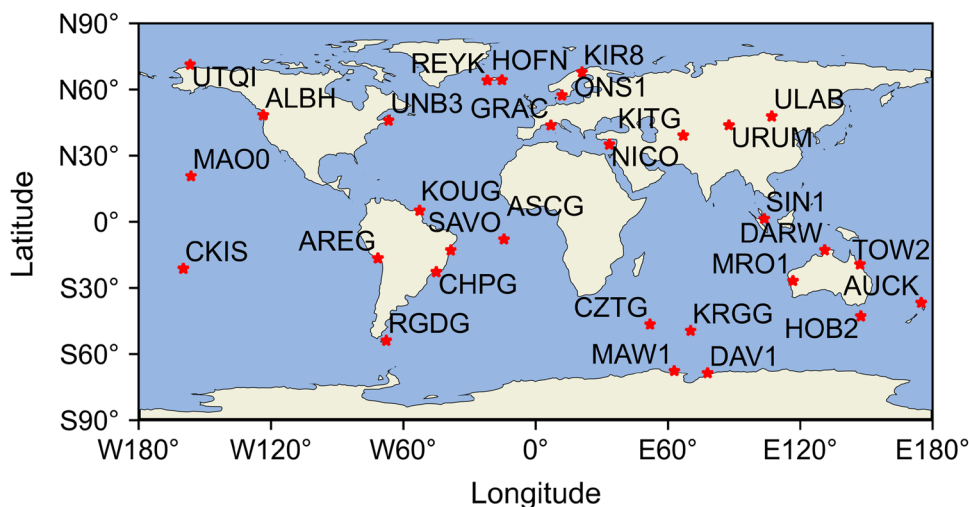


Fig. 14 Time series of the real-time PPP positioning errors for station TOW2 (top for GREAT clock, and bottom for CNES clock)

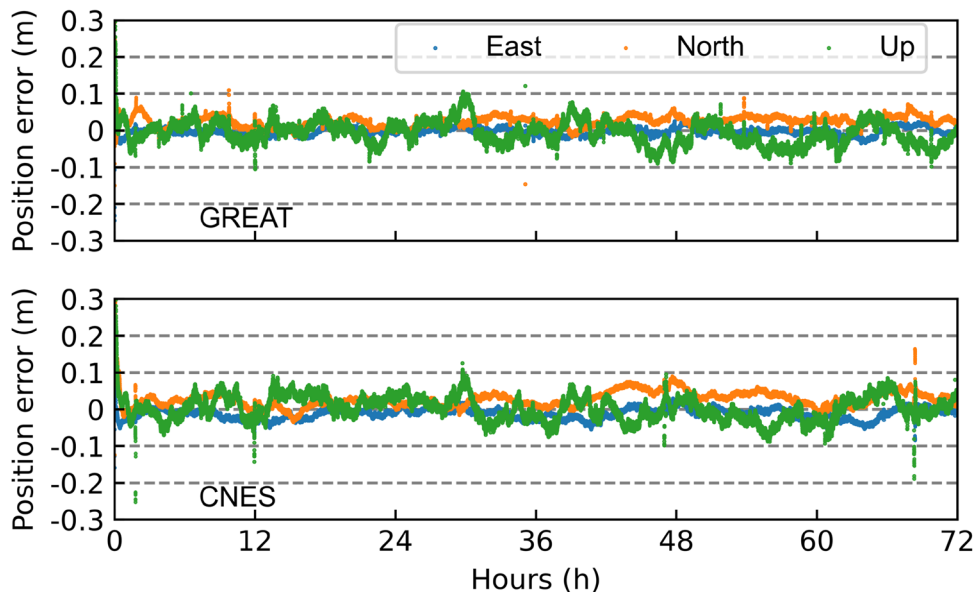


Fig. 15 Time series of the real-time PPP positioning errors for station KIR8 (top for GREAT clock, and bottom for CNES clock)

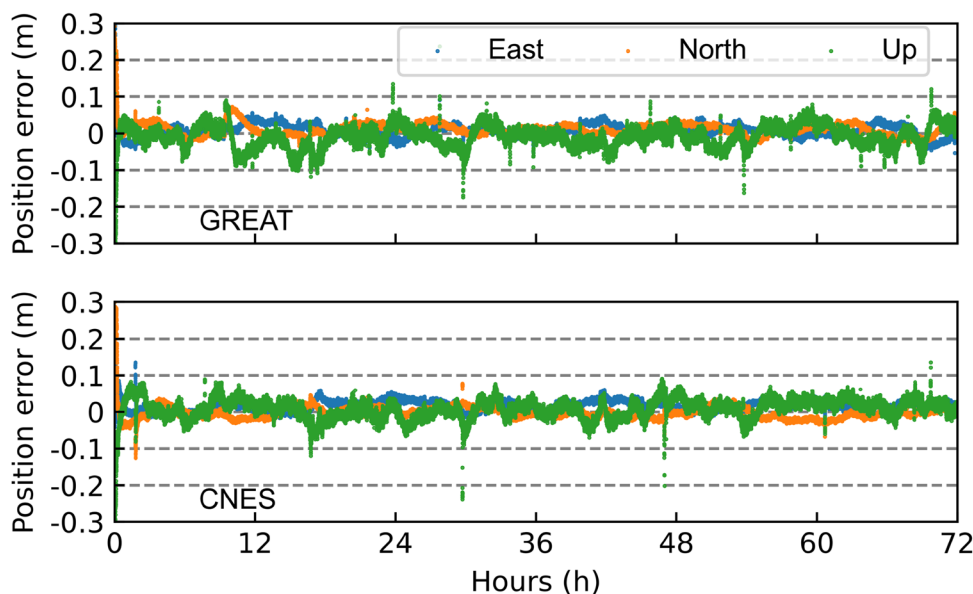


Table 6 Average RMS of positioning errors and convergence time for 30 stations using CNES and GREAT real-time clock

Multi-GNSS real-time clock products	RMS of positioning errors (cm)			Convergence time (min)
	East	North	Up	
CNES	2.34	3.49	6.47	9.50
GREAT	1.83	3.00	4.85	8.79

The experiments last from 2022-09-06 04:17:20-2022-09-09 04:17:15 (GPST), and the positioning accuracy is determined by the difference between PPP results and IGS weekly combined solutions. To evaluate the performance of real-time PPP solutions using the estimated clock (GREAT clock), real-time PPP solutions using the CNES clock (SSRA00CNE0) are also listed for comparisons. The time series of the real-time PPP results for stations TOW2 and KIR8 are shown as examples in Figs. 14 and 15, respectively.

It can be seen that the positioning errors of the stations in the east, north, and up components are within ± 10 cm for most epochs. The average RMS of positioning errors and convergence time for 30 stations are summarized in Table 6. For CNES clock, the average RMS values are 2.34, 3.49, and 6.47 cm in the east, north, and up components, respectively, and the average convergence time is 9.50 min. For GREAT clock, the average RMS values are 1.83, 3.00, and 4.85 cm in the east, north, and up components, respectively, and the

average convergence time is 8.79 min. All results indicate that high positioning accuracy can be achieved using the estimated multi-GNSS real-time clock products.

Conclusion

Real-time precise clock products are essential prerequisites to support real-time precise positioning service. In this contribution, an efficient strategy based on the UD method is proposed to achieve high processing efficiency of multi-GNSS clock estimation, wherein parallel data processing and high performance of matrix operations are introduced to accelerate the processing of multi-GNSS clock estimation. Benefitting from parallelization and effective use of modern CPUs, the computation time can be significantly reduced compared to the traditional serial strategy. Multi-GNSS observations from 85 globally distributed tracking stations are utilized for the generation of real-time clock products with the proposed strategy.

The computation efficiency of the proposed strategy is analyzed. Specifically, the overall computation time is subdivided into four parts: preprocessing, parameter elimination, constructing NEQ, and solving NEQ. With the implementation of optimized Cholesky factorization, OpenBLAS can accelerate the step of solving NEQ significantly. Compared with the open-source library Eigen, the average time for solving NEQ in GPS-only, GE, and GREC scene is 0.09, 0.32, and 1.91 s, with an improvement of 63.6%, 65.2%,

and 68.3%, respectively. With the implementation of the Schur complement, the elapsed time of parameter elimination is shortened by 45.7% in the quad-system scene. To fully exploit the power of modern CPUs with multi-core and multi-thread, parallel regions applying OpenMP are created in the step of preprocessing and constructing NEQ. With the proposed strategy with 12 threads, the average elapsed time per epoch is 0.35, 0.68, and 2.30 s for GPS-only, dual-system, and quad-system solutions, respectively. Compared to the traditional serial strategy, the computation efficiency is significantly improved by 76.0%, 77.3%, and 77.7%, respectively. Moreover, the computation efficiency with different numbers of stations is compared and analyzed. When the network size reaches 100 stations, the elapsed time of quad-system clock estimation with the proposed strategy is 3.06 s and can still meet the 5 s update rate of the IGS real-time clock product.

Furthermore, the performance of the real-time clock products is evaluated to verify the reliability of the proposed strategy. Compared to IGS final GPS products, the average STD value is 0.095 ns, and compared to GFZ final multi-GNSS satellite clock products, the average STD values are 0.106, 0.102, 0.325, and 0.215 ns for GPS, Galileo, GLONASS, and BDS, respectively. Based on the real-time clock corrections, 30 multi-GNSS stations are employed for real-time PPP experiments. The average RMS values of the stations are 1.83, 3.00, and 4.85 cm in the east, north, and up components, respectively. All results demonstrate that the proposed strategy is effective and accurate and can guarantee high-rate multi-GNSS real-time clock estimation.

Acknowledgements We would like to express our gratitude to IGS MGEX for providing multi-GNSS data and products. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University. This study is financially supported by the National Natural Science Foundation of China (No. 41974027), the National Key Research and Development Program of China (2021YFB2501102), and the Sino-German mobility program (Grant No. M-0054).

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The real-time Multi-GNSS data streams are provided by IGS. The precise orbits are provided by Wuhan University (WUM), <ftp://igs.gnsswhu.cn/pub/gnss/products/mgex>. The IGS final GPS clock and orbit products are available in the IGS repository, <ftp://igs.gnsswhu.cn/pub/gps/products>. The GFZ final clock and orbit products are available in the IGS repository, <ftp://ftp.gfz-potsdam.de/pub/GNSS/products/mgex>. The IGS weekly combined solutions are available in the IGS repository, <ftp://igs.gnsswhu.cn/pub/gps/products>.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Allen RM, Ziv A (2011) Application of real-time GPS to earthquake early warning. *Geophys Res Lett* 38(16) <https://doi.org/10.1029/2011GL047947>
- Boehm J, Niell A, Tregoning P, Schuh H (2006) Global Mapping Function (GMF): a new empirical mapping function based on numerical weather model data. *Geophys Res Lett* 33(7):L07304. <https://doi.org/10.1029/2005gl025546>
- Cai C, Gao Y (2013) Modeling and assessment of combined GPS/GLONASS precise point positioning. *GPS Solut* 17:223–236. <https://doi.org/10.1007/s10291-012-0273-9>
- Cao X, Kuang K, Ge Y, Shen F, Zhang S, Li J (2022) An efficient method for undifferenced BDS-2/BDS-3 high-rate clock estimation. *GPS Solut* 26:66. <https://doi.org/10.1007/s10291-022-01252-0>
- Chen Y, Yuan Y, Ding W, Zhang B, Liu T (2017) GLONASS pseudorange inter-channel biases considerations when jointly estimating GPS and GLONASS clock offset. *GPS Solut* 21:1525–1533. <https://doi.org/10.1007/s10291-017-0630-9>
- Chen Y, Yuan Y, Zhang B, Liu T, Ding W, Ai Q (2018) A modified mix-differenced approach for estimating multi-GNSS real-time satellite clock offsets. *GPS Solut* 22(3):72. <https://doi.org/10.1007/s10291-018-0739-5>
- Cui Y, Lu Z, Lu H, Li J, Wang Y, Huang L (2015) A Parallel Processing Strategy of Large GNSS Data Based on Precise Point Positioning Model. CSNC. https://doi.org/10.1007/978-3-662-46632-2_12
- Dach R, Schaer S, Hugentobler U (2006) Combined multi-system GNSS analysis for time and frequency transfer. In: *Proceedings of the European Frequency and Time Forum*, pp 530–537
- Dach R, Lutz S, Walser P, Fridez P (2015) *Bernese GNSS Software Version 5.2*. Astronomical Institute. University of Bern, Switzerland
- Dai Z, Dai X, Zhao Q, Liu J (2019) Improving real-time clock estimation with undifferenced ambiguity fixing. *GPS Solut* 23:44. <https://doi.org/10.1007/s10291-019-0837-z>
- De Jonge P (1998) A processing strategy for the application of the GPS in networks. Ph.D. thesis, Delft University of Technology

- Demmel J (1997) Applied numerical linear algebra. SIAM, Philadelphia
- Dixon K (2006) StarFire: A global SBAS for sub-decimeter precise point positioning. In: Proceedings of ION GNSS 2006, 26–29 Sept. 2006, Fort Worth, TX, USA, pp 2286–2296
- Fu W, Huang G, Zhang Q, Gu S, Ge M, Schuh H (2019) Multi-GNSS real-time clock estimation using sequential least square adjustment with online quality control. *J Geodesy* 93(7):963–976. <https://doi.org/10.1007/s00190-018-1218-z>
- Gabe S, Larry M, Gaurav S (2010) Sliding window filter with application to planetary landing. *J Field Robotics* 27(5):587–608. <https://doi.org/10.1002/rob.20360>
- Gao K, Zhang S, Li J, Cao X, Kuang K (2017) Real-Time GPS Satellite Clock Estimation Based on OpenMP. CSNC. https://doi.org/10.1007/978-981-10-4594-3_19
- Ge M, Chen J, Douša J, Gendt G, Wickert J (2012) A computationally efficient approach for estimating high-rate satellite clock corrections in realtime. *GPS Solut* 16(1):9–17. <https://doi.org/10.1007/s10291-011-0206-z>
- Gong X, Gu S, Lou Y, Zheng F, Ge M, Liu J (2018) An efficient solution of real-time data processing for multi-GNSS network. *J Geod* 92(7):797–809. <https://doi.org/10.1007/s00190-017-1095-x>
- Hadas T, Bosy J (2015) IGS RTS precise orbits and clocks verification and quality degradation over time. *GPS Solut* 19(1):93–105. <https://doi.org/10.1007/s10291-014-0369-5>
- Hauschild A, Montenbruck O (2009) Kalman-filter-based GPS clock estimation for near real-time positioning. *GPS Solut* 13(3):173–182. <https://doi.org/10.1007/s10291-008-0110-3>
- Kouba J, Héroux P (2001) Precise Point Positioning Using IGS Orbit and Clock Products. *GPS Solut* 5:12–28. <https://doi.org/10.1007/PL00012883>
- Kuang K, Zhang S, Li J (2019) Real-time GPS satellite orbit and clock estimation based on OpenMP. *Adv Space Res* 63(8):2378–2386. <https://doi.org/10.1016/j.asr.2019.01.009>
- Li X, Ge M, Zhang X, Zhang Y, Guo B, Wang R, Klotz J, Wichert J (2013) Real-time high-rate co-seismic displacement from ambiguity-fixed precise point positioning: application to earthquake early warning. *Geophys Res Lett* 40(2):295–300. <https://doi.org/10.1002/GRL.50138>
- Li X, Dick G, Ge M, Heise S, Wickert J, Bender M (2014) Real-time GPS sensing of atmospheric water vapor: precise point positioning with orbit, clock, and phase delay corrections. *Geophys Res Lett* 41(10):3615–3621. <https://doi.org/10.1002/2013gl058721>
- Li L, Lu Z, Chen Z, Cui Y, Kuang Y, Wang F (2019a) Parallel computation of regional CORS network corrections based on ionospheric-free PPP. *GPS Solut* 23(3):1–12. <https://doi.org/10.1007/s10291-019-0864-9>
- Li X, Xiong Y, Yuan Y, Wu J, Li X, Zhang K, Huang J (2019b) Real-time estimation of multi-GNSS integer recovery clock with undifferenced ambiguity resolution. *J Geod* 93(12):2515–2528. <https://doi.org/10.1007/s00190-019-01312-3>
- Li X, Qin Y, Zhang K, Wu J, Zhang W, Zhang Q, Zhang H (2022) Precise orbit determination for LEO satellites: single-receiver ambiguity resolution using GREAT products. *Geo Spat Inform Sci* 25(1):63–73. <https://doi.org/10.1080/10095020.2021.2022966>
- Liu T, Zhang B, Yuan Y, Li M (2018) Real-Time Precise Point Positioning (RTPPP) with raw observations and its application in real-time regional ionospheric VTEC modeling. *J Geod* 92(11):1267–1283. <https://doi.org/10.1007/s00190-018-1118-2>
- Liu T, Zhang B, Yuan Y, Zha J, Zhao C (2019) An efficient undifferenced method for estimating multi-GNSS high-rate clock corrections with data streams in real time. *J Geod*. <https://doi.org/10.1007/s00190-019-01255-9>
- Low T, van de Geijn R (2004) An API for manipulating matrices stored by blocks. FLAME Working Note, Computer Science Department, University of Texas at Austin, May 11, 2004, <https://www.cs.utexas.edu/users/flame/pubs/flash.pdf>
- Mervart L, Lukes Z, Rocken C, Iwabuchi T (2008) Precise point positioning with ambiguity resolution in realtime. In: Proceedings of ION GNSS 2008, 16–19 Sept 2008, Savannah, GA, USA, pp 397–405
- Odiijk D, Zhang B, Khodabandeh A, Odolinski R, Teunissen P (2016) On the estimability of parameters in undifferenced, uncombined GNSS network and PPP-RTK user models by means of S-system theory. *J Geod* 90(1):15–44. <https://doi.org/10.1007/s00190-015-0854-9>
- Odiijk D, Khodabandeh A, Nadarajah N, Choudhury M, Zhang B, Li W, Teunissen P (2017) PPP-RTK by means of S-system theory: Australian network and user demonstration. *J Spat Sci* 62(1):3–27. <https://doi.org/10.1080/14498596.2016.1261373>
- Rebischung P, Schmid R (2016) IGS14/igs14.atx: a new framework for the IGS products. In: AGU fall meeting, San Francisco, CA
- Saastamoinen J (1972) Contributions to the theory of atmospheric refraction. *Bull Geod* 105(1):279–298. <https://doi.org/10.1007/BF02521844>
- Saito E, Kubo N, Shimoda K (2016) Performance evaluation and a new disaster prevention system of precise point positioning at sea. In: Proceedings of the 29th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2016), pp 3412–3432. <https://doi.org/10.33012/2016.14809>
- Samper L, Merino M (2013) Advantages and drawbacks of the Precise Point Positioning (PPP) technique for earthquake, tsunami prediction and monitoring. *Proc Ion Pacific Pnt Meet* 8900(6):9–26
- Schaer S, Beutler G, Rothacher M, Brockmann E, Wiget A, Wild U (1999) The impact of the atmosphere and other systematic errors on permanent GPS networks. Presented at IAG Symp on Positioning, Birmingham, UK, 19–24 July, p 406
- Schreiber R, Van Loan C (1989) A storage-efficient WY representation for products of householder transformations. *SIAM J Sci Stat Comput* 10(1):53–57. <https://doi.org/10.1137/0910005>
- Wang Q, Zhang X, Zhang Y, Yi Q (2013) AUGEM: automatically generate high performance dense linear algebra kernels on x86 CPUs. In SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp 1–12. IEEE. <https://doi.org/10.1145/2503210.2503219>
- Weber G, Mervart L, Lukes Z, Rocken C, Dousa J (2007) Real-time clock and orbit corrections for improved point positioning via NTRIP. In: Proceedings of ION-GNSS-2007, Institute of Navigation, 25–28 Sept, Fort Worth, TX, USA, pp 1992–1998
- Wu J, Wu S, Hajj G, Bertiger W, Lichten S (1993) Effects of antenna orientation on GPS carrier phase. *Manuscr Geod* 18(2):91–98
- Zhang X, Li X, Guo F (2011) Satellite clock estimation at 1 Hz for realtime kinematic PPP applications. *GPS Solut* 15(4):315–324. <https://doi.org/10.1007/s10291-010-0191-7>
- Zhang B, Ou J, Yuan Y (2014) Method of processing GNSS reference network data with refined datum definition for rank-deficiency elimination. *Acta Geodaetica Et Cartographica Sinica* (in Chinese) 43(9):895–901
- Zhang W, Lou Y, Gu S, Shi C, Haase J, Liu J (2016) Joint estimation of GPS/BDS realtime clocks and initial results. *GPS Solut* 20(4):665–676. <https://doi.org/10.1007/s10291-015-0476-y>
- Zuo X, Jiang X, Li P, Wang J, Ge M, Schuh H (2021) A square root information filter for multi-GNSS real-time precise clock estimation. *Satellite Navigation* 2(1):1–14. <https://doi.org/10.1186/s43020-021-00060-0>



Xingxing Li completed his B.Sc. degree at the School of Geodesy and Geomatics at Wuhan University and obtained his Ph.D. at the Department of Geodesy and Remote Sensing of the German Research Centre for Geosciences (GFZ). He is currently a professor at Wuhan University. His current research mainly involves high-precision GNSS and multi-sensor integration.



Jiaqi Wu is currently a Ph.D. candidate at Wuhan University. He has completed his Master's at the School of Geodesy and Geomatics at Wuhan University in 2020. His area of research currently focuses on multi-GNSS precise orbit determination.



Yuhao Li is currently a Master's candidate at Wuhan University. He has completed his B.Sc. at the School of Geodesy and Geomatics at Wuhan University in 2020. At present, his area of research mainly focuses on precise clock estimation for GNSS and its real-time applications.



Hongjie Zheng is currently a Master's candidate at Wuhan University. He has completed his B.Sc. at the School of Geodesy and Geomatics in Wuhan University in 2019. His area of research currently focuses on multi-GNSS precise orbit determination.



Yun Xiong is currently a Ph.D. candidate at the Technische Universität Berlin (TUB). He received his Master's degree at the School of Geodesy and Geomatics at Wuhan University in 2019. His current research focuses on multi-GNSS precise satellite clock estimation and precise point positioning.



Linyang Li obtained his Ph.D. from the Institute of Surveying and Mapping, Information Engineering University, China, in 2019, where he is now an associate professor. His research focuses on the rapid processing of GNSS data.