GPS Solutions (2021) 25:106
https://doi.org/10.1007/s10291-021-01145-8

ORIGINAL ARTICLE

GPS satellite clock estimation using global atomic clock network

Jian Yao[1] · Sungpil Yoon[1] · Bryan Stressler[1] · Steve Hilla[1] · Mark Schenewerk[1]

Received: 3 August 2020 / Accepted: 18 May 2021 / Published online: 31 May 2021

Abstract
We report the GPS satellite clock estimation using 20 globally distributed receivers with an external hydrogen maser atomic clock. By applying corrections for the Sagnac effect, the relativistic effect due to orbit eccentricity, tropospheric and ionospheric delays, satellite and receiver antenna phase center offsets and variations, solid earth tides, ocean tide loading, phase wind-up effect, and P1-C1 bias, our satellite clock results matches the IGS final clock product within ± 1.4 ns with comparable frequency stability for an averaging time of less than 1000 sec and a 10–30% worse frequency stability for an averaging time of greater than 1000 sec, on MJD 58244. This small atomic clock network results in a fast computation that becomes increasingly appealing when the real-time satellite orbit and clock estimation is needed and as the GNSS constellations and the GNSS signals expand.

Keywords GPS satellite · Atomic clock · Satellite clock estimation

Introduction

The meter-level accuracy of orbit and clock broadcast by GNSS satellites fundamentally sets the positioning limit to ground users. To break this limit, precise GNSS satellite orbit and clock products need to be generated. Ten analysis centers in the world compute precise GNSS satellite orbit and clock independently. By giving each analysis center's results a proper weight, the International GNSS Service (IGS) (Johnston et al. 2017) provides a combined precise GNSS satellite orbit and clock product reaching an accuracy of 1–2 centimeters (Griffiths and Ray 2009). Using the precise GNSS satellite orbit and clock product together with sophisticated modeling of physical effects, modern GNSS software can achieve centimeter-level ground positioning (Kouba 2009).

The basic principle of computing precise GNSS satellite orbit and clock is to form a global network composed of all GNSS satellites and about 200 ground receivers, with code and phase measurements from receivers and accurate physical models/observations. In this network, the main unknown variables are satellite orbits and clock biases, as well as receiver's positions and clock biases. Many analysis centers, including the National Geodetic Survey (NGS) (Yoon et al. 2017), use the double-differencing technique to compute satellite orbits and receiver positions. Let us label the transmission time from Satellite A to Receiver j as t_j^A. The double-differencing technique is a mathematical operation of $\left(t_j^A - t_k^A\right) - \left(t_j^B - t_k^B\right)$, which pre-eliminates satellite clock biases and receiver clock biases. Therefore, after double-differencing, only satellite orbits and receiver positions are the main unknown variables, which can be solved by using the least squares method. However, double-differencing cannot compute the other indispensable part of GNSS satellites—the satellite clock biases. A satellite clock usually has white frequency noise at a very short time (e.g., 30 sec). This noise is at the level of a few centimeters for non-IIF Rb clocks and a few millimeters for IIF Rb clocks. Therefore, the epoch-to-epoch satellite clock estimation is critical to the general GNSS users who want to reach centimeter-level point positioning without resorting to double-differencing.

Although NGS has been computing the precise satellite orbits since 1989, it has not yet computed the satellite clocks. This paper proposes a novel method of computing the satellite clocks—forming a global atomic clock network to estimate the GPS satellite clocks. The next section provides details of this method. The Results section presents the satellite clock results using this method on MJD 58244 (i.e., May 06, 2018; Day 0 of GPS Week 2000). The Discussions section provides a comprehensive evaluation of this method

✉ Jian Yao
jian.yao@colorado.edu

[1] National Geodetic Survey, NOAA, Silver Spring, USA

from the aspects of accuracy, frequency stability, and computation burden. The last section concludes this paper and suggests future work.

Method of computing satellite clocks

After estimating the satellite/receiver positions using double-differencing, the remaining unknown variables in the global network are primarily satellite/receiver clock biases. The conventional least squares GNSS method, which solves these unknowns in a large matrix based on all ground observations over a whole day (Hugentobler et al. 2002), is only applicable in the post-processing mode and thus results in a latency that becomes increasingly problematic when real-time satellite orbits and clocks are needed. The correlation issue between satellite clocks and receiver clocks may also lead to a biased estimation of each clock in the conventional least squares method. With these concerns, we propose a new method—forming a network composed of 20 ground receivers referenced to hydrogen maser atomic clocks. To guarantee the full-time coverage of all GPS satellites, these receivers are selected to be distributed around the world as uniformly as possible (Fig. 1). Because hydrogen masers are orders of magnitude more stable than satellite clocks, the receiver clocks are available a priori with small uncertainties, and thus, this method could reduce the correlation issue between satellite clocks and receiver clocks, as well as the correlation issue between clocks and other noise sources. In addition, this small network significantly reduces the computation burden, which could be useful for real-time applications.

To implement this idea of using the hydrogen maser network to estimate GPS satellite clock biases, we have developed an algorithm with two steps. The first step is to compute the time difference between each satellite clock and each receiver clock, using code and phase measurements, respectively. The GPS observation equations for code and phase are (Yao 2014),

$$P_i^j = \left|\vec{x}^j - \vec{x}_i\right| - c\Delta t^j + \Delta_{\text{ion}} + \Delta_{\text{tropo}} + c\Delta t_i + \Delta_{\text{other}\_P} + \varepsilon_{\text{P}} \tag{1}$$

$$\Phi_i^j = \left|\vec{x}^j - \vec{x}_i\right| - c\Delta t^j - \Delta_{\text{ion}} + \Delta_{\text{tropo}} + c\Delta t_i + \Delta_{\text{other}\_\Phi} + \lambda N_i^j + \varepsilon_{\Phi} \tag{2}$$

where Δt^j is the clock bias of satellite j, Δt_i is the clock bias of station i, and Δ_{ion} and Δ_{tropo} are the ionospheric and tropospheric delay, respectively. Δ_{other} includes all other physical effects on code or phase measurements that are at the level of a few centimeters or larger, such as satellite/receiver antenna phase center offsets (PCO) and variations (PCV), solid earth tide, ocean tide loading, relativistic eccentricity effect, Sagnac effect, phase wind-up effects, and P1-C1 bias. ε is the noise term, and N_i^j is the phase ambiguity. Without loss of generality, we use the precise GPS satellite orbits and receiver positions provided by the IGS final sp3 file for $\vec{x}^j$ and $\vec{x}_i$, respectively. For the tropospheric delay, it can be divided into two parts—dry tropospheric delay (=$\text{zpd}_d \times M_{\text{d}}$, where zpd_d is the dry zenith path delay and M_d is the dry mapping function), and wet tropospheric delay (=$\text{zpd}_w \times M_{\text{w}}$). Following suggestions by Kouba (Kouba 2009), we get zpd_d, M_{d}, and M_{w} from models, while zpd_w is treated as an unknown variable that needs to be estimated in this algorithm.

A simple mathematical operation of (1) and (2) gives (3) and (4),

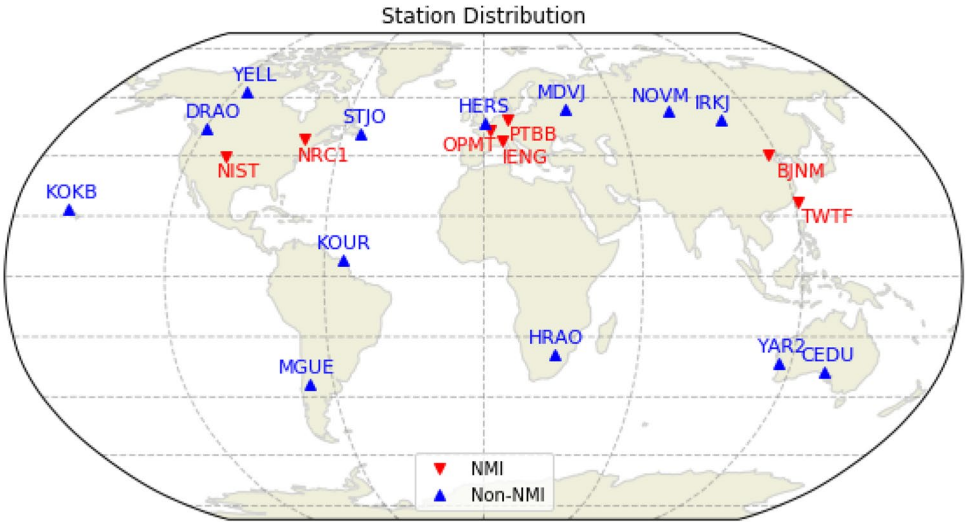

Fig. 1 Globally distributed atomic clock-referenced GNSS receiver network. Red labels are National Metrology Institutes (NMI). The UTC(k) generated by steering hydrogen maser atomic clocks gently at NMIs is a realization of Coordinated Universal Time and thus provides accurate and precise reference times to their GNSS receivers. Blue labels are non-NMIs that have free-running hydrogen maser atomic clocks. These clocks provide precise, though not accurate, reference times to the receivers

$$\Delta t^j - \Delta t_i - \frac{M_\text{w}}{c} \bullet \text{zpd}_w = \frac{\left|\vec{x}^j - \vec{x}_i\right| - P_i^j(\text{ion free}) + \text{zpd}_d \bullet M_\text{d} + \Delta_{\text{other}\_P} + \varepsilon_\text{P}}{c} \tag{3}$$

$$\Delta t^j - \Delta t_i - \frac{M_\text{w}}{c} \bullet \text{zpd}_w - n_i^j = \frac{\left|\vec{x}^j - \vec{x}_i\right| - \Phi_i^j(\text{ion free}) + \text{zpd}_d \bullet M_\text{d} + \Delta_{\text{other}\_\Phi} + \varepsilon_\Phi}{c} \tag{4}$$

The left side of (3) and (4) includes the unknown variables to be estimated in Step 2 -- Δt^j, Δt_i, zpd_w, and n_i^j. Here, n_i^j is the ionosphere-free version of the phase ambiguity and is a constant for a specific satellite-receiver pair after correcting for all cycle slips. Since the satellite orbits and receiver coordinates are already given, all terms on the right side of (3) and (4) are known. Plugging numbers on the right side of (3) and (4) gives a raw estimation of the time difference between satellite and receiver. Clearly, this raw estimation still includes the unknown wet tropospheric delay and the unknown phase ambiguity.

A single ground receiver can only observe a satellite for a few hours each day. Therefore, the result of Step 1 would not provide a continuous clock estimation for each satellite. In addition to know the actual time difference between satellite and receiver, we need to resolve the unknown wet tropospheric delay and the unknown phase ambiguity. Step 2 of the algorithm addresses these issues. A forward-only Kalman filter is designed to merge all results obtained in Step 1, to enable the clock observation of all GPS satellites every 30 sec, 24 hours a day. One receiver (e.g., the NIST receiver for our case) is selected to serve as the time reference for all GPS satellite clocks and other receiver clocks. The state vector of the Kalman filter includes the unknown variables – the time/frequency/frequency-drift offsets of all satellite clocks and all receiver clocks, the wet zenith path delays zpd_w, and the phase ambiguities of each satellite-to-receiver pair n_i^j. The noise level of the satellite/receiver clocks is determined by the corresponding Allan deviations (Senior et al. 2008; Strandjord and Axelrad 2018; Montenbruck et al. 2012; McGrew et al. 2019; Yao et al. 2019). The calculated noise of receiver clocks is negligible compared to that of satellite clocks. The initial uncertainties of the phase ambiguities are set to be at the level of 3 meters, which reflects the noise level of code measurement. The inherent variation in phase ambiguity is set to be tiny to reflect the fact that phase ambiguities are constant. The observations are weighted based on elevation angles (Dach et al. 2007), i.e., weight = $\sin^2$(elevation). With these settings, the Kalman filter runs epoch-by-epoch in a forward sense (note: the time interval is 30 sec in this paper unless specified). When the phase measurement is 4 standard deviations away from the prediction (this may come from the rise of a satellite, or cycle slips, or phase measurement outliers), we re-estimate the phase ambiguity, using (5) which can be derived from (4),

$$n_i^j = \Delta t^j - \Delta t_i - \frac{M_\text{w}}{c} \bullet \text{zpd}_w - \frac{\left|\vec{x}^j - \vec{x}_i\right| - \Phi_i^j(\text{ionfree}) + \text{zpd}_d \bullet M_\text{d} + \Delta_{\text{other}\_\Phi} + \varepsilon_\Phi}{c}. \tag{5}$$

The uncertainty of this re-estimation of the phase ambiguity is determined by the uncertainties of Δt^j, Δt_i, zpd_w, and the phase measurement noise in Φ_i^j(ion free). Depending on satellite blocks, the uncertainties of Δt^j are at the level of 100 ps or smaller (i.e., < = 3 cm), and the uncertainties of Δt_i are negligible thanks to hydrogen masers. zpd_w has little variation epoch-by-epoch; phase measurement noise is typically below 3 cm. Therefore, we can reach an accurate re-estimation of phase ambiguity at the level of a few centimeters, much smaller than its initial uncertainty of ~ 3 meters. This Kalman filter not only allows for real-time applications but also avoids the large matrix of combining all-epoch observations as in conventional least squares GNSS processing. In addition, the forward-only stream processing in the Kalman filter does not result in the day boundary that exists in conventional processing.

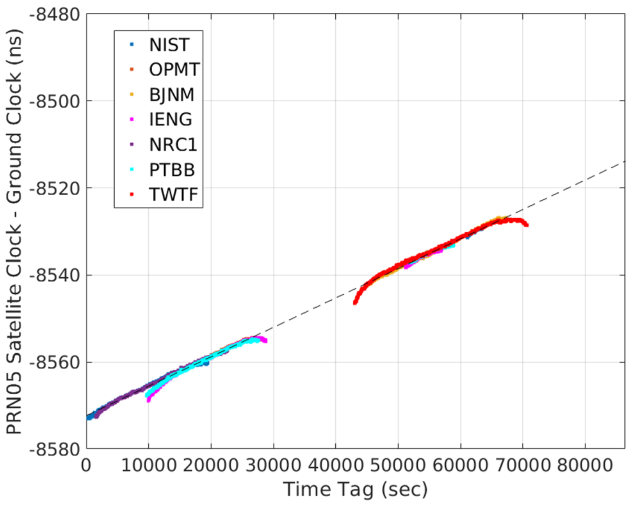

Fig. 2 Time difference between PRN05 and ground receivers at NMIs on MJD 58244, based on phase measurements. The black dashed line illustrates the slope of these curves. Note, phase ambiguities are not estimated, and curves are aligned with constant shifts

Results

According to the previous section, we can get a raw estimate of the time difference between each satellite and each receiver after Step 1. Here, as an example, we show the raw time difference between a GPS satellite clock (i.e., PRN05) and seven receivers located at National Metrology Institutes, calculated by (4). From Fig. 2, all receivers at NMIs observe the same slope of PRN05 indicated by the black dashed line. The curves deviate from the black dashed line at low elevation angles because the wet tropospheric delay has not yet been taken into account in Step 1. Indeed, as a sanity check, we applied the wet tropospheric delay correction and found that the deviations from the black dashed line disappeared. Figure 2 indicates that the seven receivers are not enough to provide coverage for 24 hours per day, nor do they offer sufficient redundancy in case data from one or more of these receivers are unavailable.

To achieve the 24-hour coverage for each day, we introduce thirteen ground receivers that refer to free-running hydrogen masers (i.e., the blue labels in Fig. 1). We eventually form a network of 20 stations providing full coverage. Figure 3 shows the time difference between PRN05 and these additional ground receivers. Although a hydrogen maser is comparable to UTC(k) in terms of precision, it is not as accurate as UTC(k) and usually has a nonzero frequency offset, which leads to the various slopes seen in Fig. 3. Like Fig. 2, the curves in Fig. 3 can also deviate from the black dashed lines at low elevation angles because the wet tropospheric delay was not taken into account in Step 1.

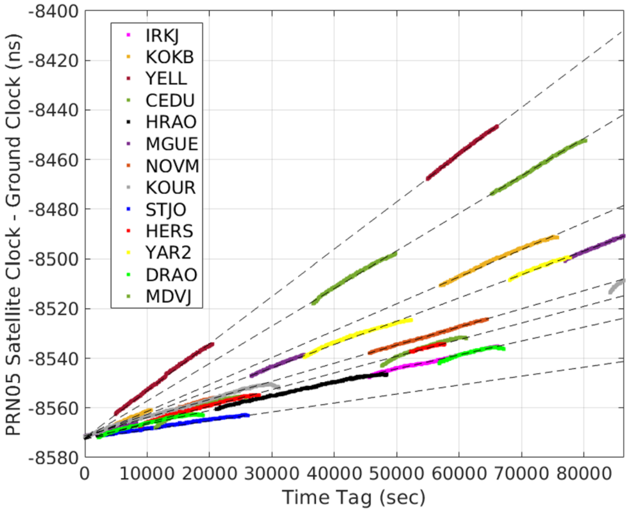

Fig. 3 Time difference between PRN05 and ground receivers at non-NMIs on MJD 58244, based on phase measurements. The black dashed lines illustrate the slopes of each curve. Note, phase ambiguities are not estimated, and curves are aligned with constant shifts; a large constant frequency offset is applied to YELL for a better presentation of this figure

Although the time difference based on code measurements is much noisier than that based on phase measurements, it helps the estimation of phase ambiguities inside the Kalman filter, and therefore, those code measurement data are included with appropriate weighting. For our case, the ratio of code weight to phase weight is 1:400. With the raw time differences between each GPS satellite and each ground station based on both phase and code as the inputs of the Kalman filter, we can get 24-hour clock estimation for all GPS satellites and all 20 ground stations. Here, we show our result of PRN05 as an example (blue curve in Fig. 4(top)). It is noted that to make a fair comparison, the reference time for both blue curve and red curve in Fig. 4(top) is the IGS final timescale. Remember that the Kalman filter gives our

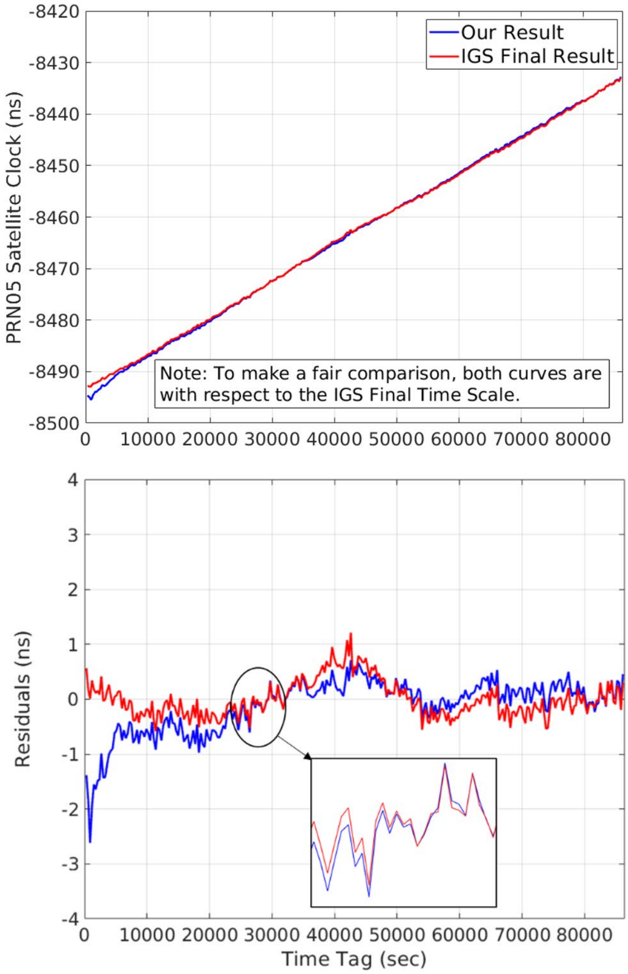

Fig. 4 Our result of PRN05 satellite clock on MJD 58244 (blue curve). The IGS final result of PRN05 (red curve) is shown for comparison. The IGS final result is a combination of all analysis centers' results. The inserted plot of the bottom panel enlarges the curves around 25,000 sec. Note, the reference time for both blue curves and red curves is the IGS final timescale

estimate of (PRN05 clock – NIST time), and the IGS final product gives the time difference of (NIST time – IGS final timescale). Therefore, we can get our estimate of (PRN05 clock – IGS final timescale) using the simple equation of (PRN05 clock – NIST time) + (NIST time – IGS final timescale) and thus generate the blue curve in Fig. 4 (top). The red curve is generated from the IGS final product directly. From Fig. 4 (top), our result experiences a convergence process for the first few hours, because the phase ambiguities are estimated by averaging down multi-epoch code measurements. We want to emphasize that since the filter runs forward and can process data across day boundaries, this initialization process only occurs once as long as the process runs continuously. After the initialization, our result of the PRN05 clock is mainly linear, which is what we expect for a GPS satellite up to a day. More importantly, our result has an excellent match with that computed by the IGS, indicating that we have observed the PRN05 clock behavior correctly.

To do a detailed comparison between our result and the IGS final result, we remove the slope of our result and get the residuals (blue curve in Fig. 4(bottom)). Similarly, we remove the same slope for the IGS result and plot the residuals (red curve in Fig 4(bottom)). Both curves have similar behaviors, such as a rise during 20,000–35,000 sec, a decline during 45,000–55,000 sec, a dent around 56,000 sec, and a small bump around 65,000 sec. The inserted plot of Fig. 4(bottom) shows the epoch-level comparison of our result and the IGS result. The same pattern of the two curves in the inserted plot demonstrates that we are able to observe the instantaneous satellite clock behavior. This result demonstrates that the atomic clock network enables us to monitor the PRN05 satellite clock behavior comparable to the IGS.

We extend the above analysis to other GPS satellites and confirm that the conclusion applies to all GPS satellites. As an example, Fig. 5 shows what we get for PRN08. Again, after the initial convergence, our result well matches the IGS result, and the curve pattern of our result is nearly the same as that of the IGS result, but a small divergence around 80,000 sec. Remember that the IGS result is based on more than 200 stations and is a combination of the contributions from over 10 analysis centers; in contrast, our result is based on only 20 stations and only involves a single analysis center. Considering this fact, the comparable performance of our result and IGS result demonstrates an advantage for our satellite clock estimation method.

Discussions

We further evaluated our method from three aspects—accuracy, frequency stability, and computation burden. The evaluation of the accuracy of our result is done by comparing it to the IGS result. Admittedly, the IGS result is still improving and can be affected by data processing and imperfect physical models (Coleman and Beard 2020); nevertheless, it provides the best available estimate of GPS satellite clocks and shall be considered as the best representation of the truth. Therefore, comparing our result to the IGS result could provide a good evaluation of the accuracy obtained. Indeed, we have started such a comparison in Figs 4 and 5. Differencing our result and the IGS result for all GPS satellites gives Fig. 6. Neglecting the first few hours, our result is within ±1.4 ns with respect to the IGS result, for all GPS satellites and for all epochs. We also observe that there is no systematic bias between our result and the IGS result. Most curves wander around 0 ns with a deviation of less than ±1 ns. Thus, our result exhibits an accuracy at the level of 1 ns. We want to mention that the clock results of other IGS analysis centers also deviate from the IGS result. For example, the CODE (Center for Orbit Determination in Europe)

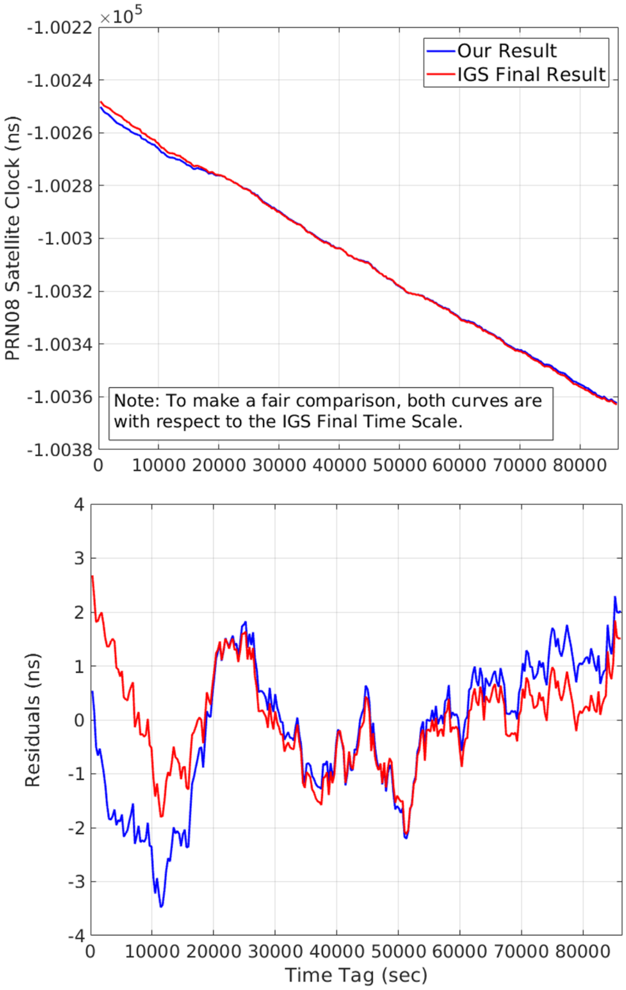

Fig. 5 Our result of PRN08 satellite clock on MJD 58244 (blue curve). The IGS result of PRN08 (red curve) is shown for comparison. Note, the reference time for both blue curves and red curves is the IGS final timescale

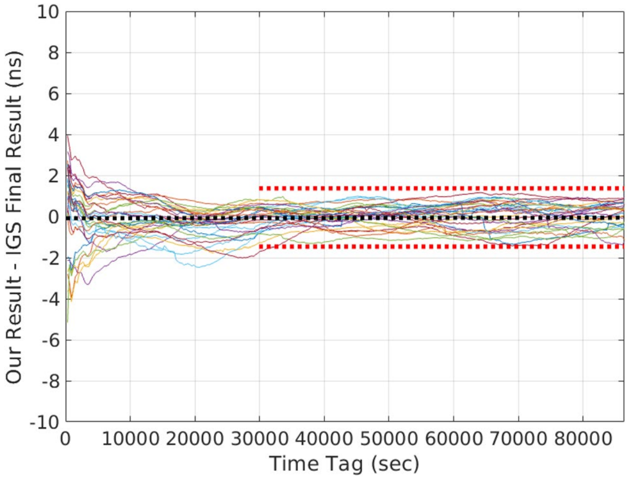

Fig. 6 Difference between our result and IGS final result, for all GPS satellites, on MJD 58244. Note, the red dotted lines illustrate ±1.4 ns and the black dotted line illustrates 0 ns, for a better presentation

result has a deviation of ±0.6 ns on the same date, the EMR (Natural Resources Canada) result has a deviation of ±0.9 ns, and the MIT (Massachusetts Institute of Technology) result has a deviation of ±0.9 ns. Admittedly, our current result is worse than the result of other IGS analysis centers because we believe it is limited by the code measurements rather than the phase measurements. If we would expand our small network of 20 stations to hundreds of stations like other analysis centers, we could have many stations monitoring one satellite at the same time and therefore could average down the code measurements noise and de-weight those contaminated code measurements, which can push our result closer to the IGS result and thus improve the accuracy of our result. In addition, the modeling of physical effects, such as the tropospheric model, as well as the forward-only Kalman filter processing, could introduce systematic biases/uncertainties contributing to the deviation of our result from the IGS result.

Next, we study the frequency stability of our GPS satellite clock results. Because a clock often exhibits random-walk noise or even higher-order noise, its standard deviation is divergent as time goes. The Allan deviation was introduced to solve the incompetence of the classical standard deviation in clocks. It is defined as $\sigma(\tau) = \sqrt{\frac{1}{2(M-1)}\sum_{i=1}^{M-1}\left(y_{i+1} - y_i\right)^2}$, where τ is the averaging time, y_i is the ith of M fractional frequency values averaged over τ. Neglecting the initial convergence time period, we calculate a series of Allan deviations by changing τ from 30 sec to ~ 1 day, which provides a full picture of the clock noise at a variety of time intervals. Instead of choosing the IGS time scale as the common reference that may have its own behavior, we directly compare two IIR block Rubidium clocks (e.g., PRN11 and PRN16)

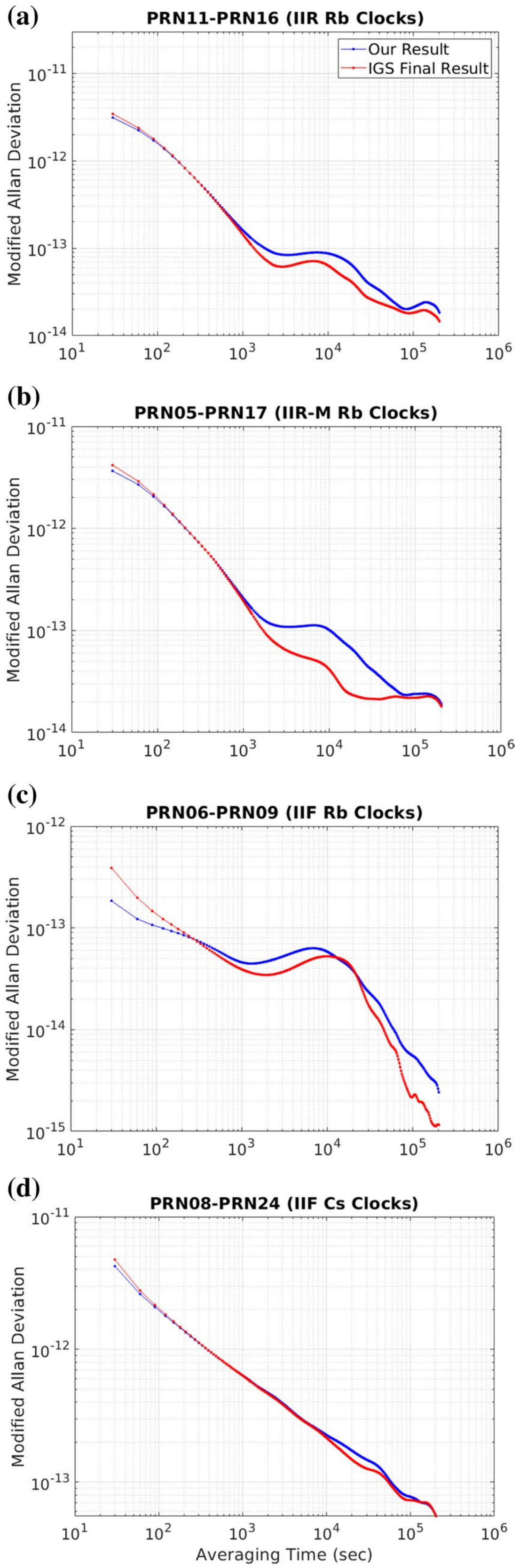

Fig. 7 Frequency stability of IIR Rubidium clocks (**a**), IIR-M Rubidium clocks (**b**), IIF Rubidium clocks (**c**), and IIF Cesium clocks (**d**)

in our result. The corresponding frequency stability is shown by the blue curve in Fig. 7(a). Similarly, we can calculate the frequency stability of these two clocks in the IGS result and get the red curve. We repeat the same process for IIR-M block Rubidium clocks, IIF block Rubidium clocks, and IIF block Cesium clocks (Fig. 7(b-d)). For all four clock types, the frequency stability of our result is comparable to that of the IGS result, up to 1000 sec. This matches the observation in Section Results that our result has the same instantaneous and short-term behavior as the IGS result. For an averaging time larger than 1000 sec, the frequency stability of our result is still comparable to that of the IGS result for the IIF block Cesium clocks. However, for the other three clock types, our result is roughly 1.3 times as noisy as the IGS result after 1000 sec. A few facts could lead to this worsened long-term performance. We only use 20 stations and involve a single analysis center, while IGS uses at least 200 stations and averages results from all analysis centers. Another fact is that although we have taken nearly all physical effects that are larger than a few centimeters into account, there are two remaining effects, i.e., pole tides and earth rotation parameters (ERP) variations (Kouba 2009), that need to be included in the future. Even more, the imperfect models for physical effects can deteriorate the performance of our result as well. Although our algorithm is designed with care, it can be further improved with additional features such as outlier detection/removal and parameter optimization in the Kalman filter. With these considerations, it is reasonable that our result is noisier than the IGS result after 1000 sec.

Furthermore, we explore the computation burden of our atomic clock network method. Step 1 computes all terms on the right side of (3) and (4). It takes a computer with a 3.10 GHz processor 106 sec to process one station for one day, and the processing of multiple stations can easily be parallelized. Remember that the double-differencing processing has calculated most terms on the right side of (3) and (4). With these terms readily available after the double-differencing processing, the computation burden of Step 1 becomes negligible. Thus, the computation burden of Step 2 is of more concern. Step 2 involves the estimation of unknown variables (i.e., Δt^j, Δt_i,n_i^j, and zpd$_w$) using measurements. Assuming that we have R stations, the number of unknown variables n equals 32 × 3 + R × 3 + 32 × R + R = 36 R + 96, where 32 × 3 and R × 3 are the number of parameters for satellite clocks and receiver clocks, respectively, since each clock has three parameters—time, frequency, and frequency drift, 32×R is the total number of phase ambiguities, and the last term R corresponds to the total number of wet zenith path delays for all receivers (each receiver has one wet zenith path delay to be estimated). The number of measurements m is proportional to R. According to (Zumberge et al. 1997; Malys and Jensen 1990), the computation burden of Step 2 is proportional to $n^2m = (36R + 96)^2R$. When R is significantly larger than 96/36 = 2.7 (for our case, R equals 20 and thus much larger than 2.7), the computation burden becomes proportional to R^3. Therefore, the computation of Step 2 is roughly one thousand times faster than if 200 stations were used. Indeed, it takes a computer with a 3.10 GHz processor only 590 sec to finish Step 2 and generate the 30-sec clock product for one day. If we were generating 5-min clock product, the computation time would further drop to 59 sec. This fast computation is particularly appealing when real-time satellite orbits and clocks become of great interest and as the GNSS constellations and the GNSS signals expand rapidly.

In addition to the above evaluation of our atomic clock network method from the aspects of accuracy, frequency stability, and computation burden, Appendix A addresses the performance of our method at day boundaries. Although the day boundary discontinuity (typically at the level of 100–200 ps) can be obscured by the noise of satellite clocks, this discontinuity becomes obvious when comparing ultra-low-noise ground atomic clocks. By comparing clocks at two NMIs, Appendix A demonstrates that our method does not have the day boundary discontinuity while the IGS result does. Appendix B evaluates the satellite clock estimation performance when using 20 receivers without external atomic clocks. A GPS receiver without an external atomic clock usually has its internal quartz clock aligned to the GPS system time with an accuracy of a few nanoseconds. If we use such 20 receivers to estimate satellite clocks with the same algorithm as described in Section “Method of Computing Satellite Clocks,” the frequency stability of IIR Rb clocks, IIR-M Rb clocks, and IIF Rb clocks becomes approximately 1.6 times as noisy as if the atomic clock network was used, for an averaging time of 1000 sec–12 hours (Appendix B). This indicates that the satellite clock estimation is improved by capitalizing on the ground atomic clock timing infrastructure.

Conclusions and future work

We have demonstrated an atomic clock network architecture for monitoring GPS satellite clocks. This network is composed of twenty globally distributed high-precision hydrogen maser atomic clocks. Thanks to these excellent ground clocks, we are able to monitor the instantaneous satellite clock behaviors. Our result for all GPS satellites is within ±1.4 ns with respect to the IGS result. Considering the fact that we use 20 stations while IGS uses hundreds of stations, it is satisfying that our result is just slightly noisier than the IGS result overall. An advantage of this architecture is that it is expected to be orders of magnitude faster than if hundreds of stations were used. In the era of a rapid expansion of GNSS constellations and a great emphasis on real-time satellite orbits and clocks, this feature is particularly favorable. In the future, we want to include more geophysical effects (e.g., pole tides) to further improve our result. Also, we plan to extend this work to multi-GNSS. In addition, it is worth exploring the feasibility of using this atomic clock network architecture to observe both clocks and orbits of GNSS satellites, which could potentially offer two benefits for the real-time high-precision satellite orbit and clock determination, i.e., significantly reducing the number of maintained stations and achieving much faster computation.

Appendix

Appendix A: performance of our method at day boundaries

When comparing ground high-precision atomic clocks using GPS carrier phase, such as hydrogen masers, the day boundary discontinuity at the level of 100 – 200 ps often occurs because the GPS data are processed on a daily basis which results in different estimations of phase ambiguities between two consecutive days (Yao and Levine 2012). This discontinuity problem becomes less serious for the satellite clock estimation since the noise of GPS satellite clocks obscures it.

Thanks to the forward-only processing in our method, we can avoid the discontinuity at day boundaries, as shown by Fig. 8. Remember that PTBB and OPMT (see the map in Fig. 1) are receivers referencing to external UTC(k). The bias of ~ 215.5 ns between the two receivers is caused by the time delays from cables connecting to receivers. We can see that there are six obvious day boundary discontinuities (i.e., at time tags of 1.0, 2.0, 3.0, 5.0, 6.0, and 9.0) in the IGS result. In contrast, our result exhibits no day boundary discontinuities. We also want to mention that the blue curve and the red curve have pattern similarities (e.g., both curves have a dent at 2.5 d and a spike at 4.1 d), which indicates that our method is able to observe the behavior of UTC(k). Admittedly, our result is noisier than the IGS result, consistent with what we have found in Fig. 7.

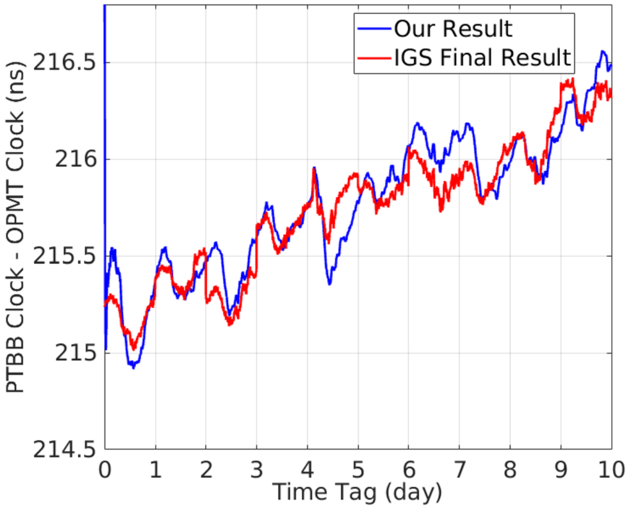

Fig. 8 Performance of our method at day boundaries. The time period ranges from MJD 58244.0 to MJD 58254.0. PTBB and OPMT (see Fig. 1) are receivers at NMIs

Appendix B: satellite clock estimation using 20 receivers without external atomic clocks

In this appendix, we investigate the satellite clock estimation performance using 20 receivers without external atomic clocks (for the sake of simplicity, we call these receivers "plain receivers"), instead of using 20 receivers with external hydrogen masers discussed in the body text. A plain receiver typically has its internal quartz clock disciplined to the GPS system time with an accuracy of a few nanoseconds (Misra 1996). To make a fair comparison between the disciplined quartz clock network and the atomic clock network, the distribution of the 20 plain receivers is nearly the same as the distribution shown in Fig. 1. To be specific, these plain receivers are VALD, BLYT, WILL, BAKE, SCH2, HNLC, BRAZ, FALK, ZAMB, RABT, VIS0, QAQ1, GRAZ, MOBN, POL2, YSSK, SHAO, PIMO, MRO1, and MOBS. We use the same algorithm described in Section "Method of Computing Satellite Clocks" to process the GPS data. The noise parameters for receiver clocks in the Kalman filter are adjusted accordingly.

An intuitive impression of the disciplined quartz clock network is that this network cannot accurately estimate satellite clocks since these quartz clocks are orders of magnitude noisier than satellite clocks. The quartz clocks are disciplined to the GPS system time formed by satellite clocks, and thus, lack of independence. As a commonsense, it is impossible to evaluate a standard using a device calibrated by this standard. However, for our case, the function of the disciplined quartz clock network is not to evaluate the satellite clocks directly but to establish links between satellite clocks. One satellite can be linked to other satellites via receivers. In other words, satellite clocks are monitoring each other with the help of this disciplined quartz clock network. Therefore, this network, though receiver clocks are noisy, still enables the estimation of satellite clocks. Nevertheless, the performance of this disciplined quart clock network is expected to be inferior to that of the atomic clock network because of the following reasons. First, considering the disciplined quartz clocks have little weights, the number

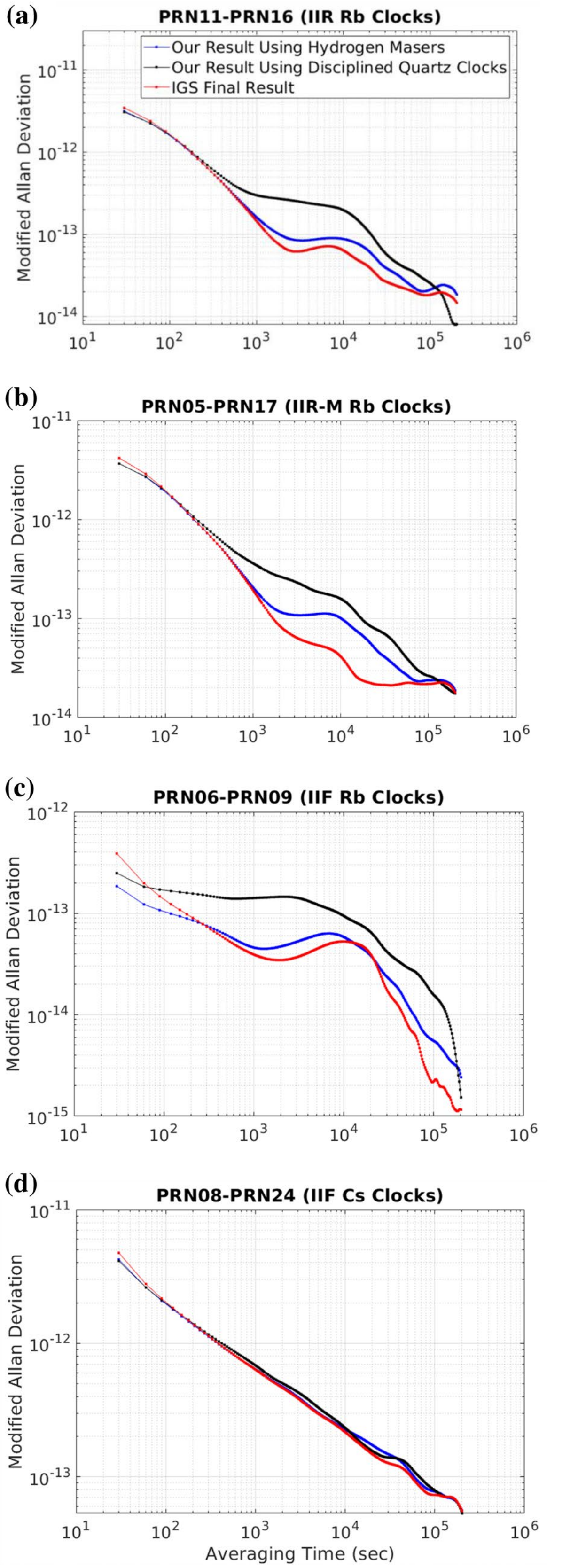

Fig. 9 Comparison between our result using disciplined quartz clocks (black curves) and our result using hydrogen masers (blue curves). The blue and red curves are the same as those in Fig. 7

of effective clocks becomes 32 (i.e., the number of satellite clocks). In contrast, the number of effective clocks in the atomic clock network method is $32+20=52$ (note: 20 is the number of receivers). Therefore, the disciplined quartz clock network method is not as robust as the atomic clock network method. More importantly, whenever we need to re-estimate the phase ambiguity, such as at the occurrence of cycle slips, the rise of a satellite, or measurement outliers, the uncertainty of the phase-ambiguity estimation in the atomic clock network method is only a few centimeters, as mentioned in the section on Method of Computing Satellite Clocks. However, this small uncertainty does not apply to the disciplined quartz clock network method, because the Δt_i uncertainty in (5) becomes as large as a few nanoseconds. Therefore, the uncertainty of the phase-ambiguity estimation shall be at the level of 1 meter, much larger than that in the atomic clock network method. Since phase ambiguity correlates with clocks, this large phase-ambiguity uncertainty makes satellite clock estimation susceptible. In addition, the atomic clock network method can easily identify the measurement anomalies and then remove/de-weight the measurements, while the disciplined quartz clock network method can hardly identify those anomalies that are below the noise of the disciplined quartz clock.

The black curves in Fig. 9 show our result using the disciplined quartz clock network. We can see that the frequency stability of IIR Rb clocks, IIR-M Rb clocks, and IIF Rb clocks is approximately 60% noisier than if the atomic clock network was used, for an averaging time of 1000 sec – 12 hours. This is consistent with the above analysis that we are able to observe the satellite clocks using disciplined quartz clocks, but not as good as using high-precision atomic clocks. As for the frequency stability of IIF Cs clocks (Fig. 9(d)), the black and blue curves are nearly the same. This can be understood by the fact that the IIF Cs clocks are among the noisiest clocks in the GPS constellation. The advantages brought by the atomic clock network, as discussed in the previous paragraph, are obscured by the noise of IIF Cs clocks.

Data availability The datasets generated during this study are available from the corresponding author upon request.

References

Coleman M, Beard R (2020) Autonomous clock ensemble algorithm for GNSS applications. Navigation 67:333–346

Dach R, Hugentobler U, Fridez P, Meindl M (2007) Bernese GPS software version 5.0. Astronomical Institute, University of Bern, Bern, page 144.

Griffiths J, Ray J (2009) On the precision and accuracy of IGS orbits. J Geodesy 83:277–287

Hugentobler U, et al (2002) CODE IGS analysis center technical report 2002. IGS Technical Reports, 43-52.

Johnston G, Riddell A, Hausler G (2017) The International GNSS service Springer handbook of global navigation satellite systems. springer handbooks. Springer

Kouba J (2009) A guide to using international GNSS service (IGS) products. Geodetic Survey Division, Natural Resources Canada. http://igscb.jpl.nasa.gov/igscb/resource/pubs/UsingIGSProductsVer21.pdf

Malys S, Jensen PA (1990) Geodetic point positioning with GPS carrier beat phase data from the CASA UNO experiment. Geophys Res Lett 17(5):651–654

McGrew W et al (2019) Towards the optical second: verifying optical clocks at the SI limit. Optica 6:448–454

Misra PN (1996) The role of the clock in a GPSreceiver. GPS World, 7(4):60–66

Montenbruck O, Hugentobler U, Dach R, Steigenberger P, Hauschild A (2012) Apparent clock variations of the Block IIF-1 (SVN62) GPS satellite. GPS Solut 16:303–313

Senior K, Rayk J, Beard R (2008) Characterization of periodic variations in the GPS satellite clocks. GPS Solut 12:211–225

Strandjord K, Axelrad P (2018) Improved prediction of GPS satellite clock sub-daily variations based on daily repeat. GPS Solut 22:58

Yao J (2014) Continuous GPS carrier-phase time transfer. University of Colorado at Boulder, Dissertation.

Yao J, Levine J (2012) GPS carrier-phase time transfer boundary discontinuity investigation. Proc. 44th Annual Precise Time and Time Interval Systems and Applications (PTTI) Meeting, Reston, Virginia, 317–326.

Yao J et al (2019) Optical-clock-based time scale. Phys Rev Appl 12:044069

Yoon S, Saleh J, Heck J, Choi K, Hilla S, Schenewerk M (2017) NGS analysis center technical report 2017. IGS Technical Reports, 90-92.

Zumberge J, Hefflin MB, Jefferson DC, Watkins MM, Webb FH (1997) Precise point positioning for the efficient and robust analysis of GPS data from large networks. J. Geophys. Res. 102:5005–5017

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jian Yao received his Ph.D. degree in physics from the University of Colorado at Boulder in 2014. He later joined U.S. National Institute of Standards and Technology (NIST) where he worked on GP time transfer and timescale architecture. During 2018–2020, He conducted research on the GPS/GNSS positioning, navigation and timing at the U.S. National Geodetic Survey (NGS). He is currently with the NIST Time Realization and Distribution group.

Sungpil Yoon received a Ph.D. degree in Aerospace Engineering from Auburn University in 1999. He worked on orbit determination of LEO satellites and deep space spacecraft before he joined NOAA′s National Geodetic Survey (NGS) in 2015. At NGS, he processed GPS data from the national and global network of continuously operating reference stations contributing to the International Terrestrial Reference System (ITRF) and National Spatial Reference System (NSRS. Since 2020, he is with the Amazon Project Kuiper.

Bryan Stressler earned a Master′s degree in Geophysics from the University of Iowa in 2017. He later joined NOAA′s National Geodetic Survey (NGS) where he has been working on research and development for the next-generation multi-GNSS processing software suite.

Steve Hilla joined NOAA′s National Geodetic Survey (NGS) in 1986 and has been involved with GPS processing and GPS/GNSS software development for over 30 years. He is the editor of The GPS Toolbox column in the journal GPS Solutions, which highlights algorithms and shareware developed for GNSS applications.

Mark Schenewerk joined the National Geodetic Survey in 1987 where he began developing GPS high-accuracy processing software. Current activities include development of the next generation GNSS processing software and supporting the OPUS suite.