An Improvement of RINEX-Shift Algorithm for Continuous GPS Carrier-Phase Time Transfer

Jian Yao and Judah Levine
Time and Frequency Division and JILA, National Institute of Standards and Technology and University of Colorado, Boulder, Colorado 80305, USA
E-mail*: jian.yao@colorado.edu*

BIOGRAPHY

Jian Yao was born in Nancheng, Jiangxi Province, China, in 1988. He received a bachelor of science degree in physics from Nanjing University, China, in 2009, and a master of science degree in physics from the University of Colorado at Boulder in 2012. He is currently a Ph.D. candidate at the University of Colorado at Boulder and the National Institute of Standards and Technology (NIST), where he performs studies in GPS carrier-phase time and frequency transfer under the supervision of Dr. Judah Levine.

Judah Levine is a Fellow of the National Institute of Standards and Technology and is the leader of the Network Synchronization Project in the Time and Frequency Division, which is located in the NIST laboratories in Boulder, Colorado. Dr. Levine is responsible for the design and implementation of the time scales AT1 and UTC(NIST), which provide the reference signals for all of the NIST time and frequency services. In addition, he designed and built the servers that support the Automated Computer Time Service (ACTS) and the Internet Time Service, which provide time and frequency information to users in a number of different digital formats. The ACTS service is realized using a number of parallel computers that control a 12-line telephone rotary. The ACTS service receives about 4,000 requests per day. The Internet Time Service uses 24 computers which are located at several sites in the US. These computers receive about 6.5 billion requests per day for time stamps in 3 different standard formats. He received his Ph.D. in Physics from New York University in 1966. Dr. Levine is a member of the IEEE and a Fellow of the American Physical Society.

ABSTRACT

The wide application of GPS carrier-phase (CP) time transfer is limited by the problem of boundary discontinuity. The RINEX-Shift (RS) algorithm was designed to solve this problem [1]. However, if there are GPS data anomalies, the time transfer result computed by this algorithm oscillates around the true value. The deviation from the true value can be as large as a few nanoseconds. The origin of this oscillation behavior lies in the fact that the precise point positioning (PPP) technique needs some time (e.g., 2 hours) to converge. If there are only a short segment (e.g., < 2 hours) of valid data before or after the missing data, PPP does not have enough time to converge so that the solution for this short segment deviates from the true value. We propose the “revised RINEX-Shift” (RRS) algorithm to solve the oscillation problem in the RS algorithm. RRS extracts the result at the middle epoch, rather than the first epoch as in the RS algorithm. In this way, PPP has enough time to converge. Tests of the RRS algorithm show that the oscillation problem is solved successfully and there is a 10–55% improvement of fractional frequency stability over the RS algorithm. Thus, the RRS algorithm provides the best GPS time transfer result.

KEY WORDS

GPS, carrier phase, time transfer, precise point positioning (PPP), boundary discontinuity, RINEX-Shift algorithm, revised RINEX-Shift algorithm, isolated island effect.

I. INTRODUCTION

Global Positioning System (GPS) carrier-phase (CP) time transfer, a widely accepted method for high precision time transfer, provides much lower short-term noise than other time transfer methods, such as Two Way Satellite Time and Frequency Transfer (TWSTFT) and GPS Common View (CV) Time Transfer [2]. However, independent daily CP time transfer solutions frequently show boundary discontinuities of up to 1 ns, because of the inconsistency of the phase ambiguities between two independent days [3–5]. Many researchers have studied the behavior and origins of the boundary discontinuity in recent years [3, 6–9]. A few algorithms were proposed to eliminate the boundary discontinuity to achieve continuous GPS CP time transfer [1, 5, 10–11]. One promising algorithm among them is the RINEX-Shift (RS) algorithm [1].

The RS algorithm basically executes the precise point positioning (PPP) program for a certain period (e.g., 10 days) and extracts the PPP result at the first epoch. Then it shifts the Receiver Independent Exchange Format (RINEX) data by a time step (e.g., 10 min) and executes PPP again, and extracts the result at the current first epoch again. So on and so forth. The results at all first epochs form the final result. We can see that the RS algorithm actually converts the day boundary discontinuity into a short-term (i.e., 10 min) noise. The RS algorithm typically results in a fractional frequency stability of 2×10^{-15}, for an averaging time of 1 day [1].

In this paper, we focus on the improvement of the RS algorithm. First, we show that the time transfer result computed by the RS algorithm deviates from the true value if there are bad data or missing data. Then, we explore the origin of this problem in Sections III-IV. In order to solve the problem, the revised RINEX-Shift (RRS) algorithm is proposed in Section V. Section VI tests the performance of the RRS algorithm. It shows that it solves the oscillation problem in the RS algorithm. Also, it decreases the time transfer noise by 10–55%.

II. PROBLEM WITH RINEX-SHIFT ALGORITHM

The RS algorithm (or the NEW PPP program based on the RS algorithm) showed excellent performance in terms of eliminating the boundary discontinuity [1].

However, this is only the case when all of the GPS data are good. Although most timing laboratories have installed state-of-the-art GPS receivers and antennas, it is still inevitable that a GPS receiver malfunctions (e.g., losing track), or the satellite-receiver line is blocked by an object, or the reference time for the receiver is adjusted, or even a man-made error occurs. All these problems lead to GPS data anomalies. We find that if there are a few epochs of data anomalies, the result of the RS algorithm deviates from the true value (Figures 1–2).

In Figure 1, we compare the time difference between *PTBB* (a GPS receiver in Physikalisch-Technische Bundesanstalt (PTB), Germany) and International GNSS Service (IGS) final time. The black curve is the result of NRCan PPP software package [12]. We can see that there is an obvious day boundary discontinuity between Modified Julian Date (MJD) 56610 and MJD 56611. Since the standard deviation of the boundary discontinuity of *PTBB* is only 138.5 ps [3], we can hardly distinguish other small day boundary discontinuities at the scale of Figure 1. The red curve is the result computed by the RS algorithm. It eliminates the boundary discontinuity very well (e.g., there is no jump between MJD 56610 and MJD 56611).

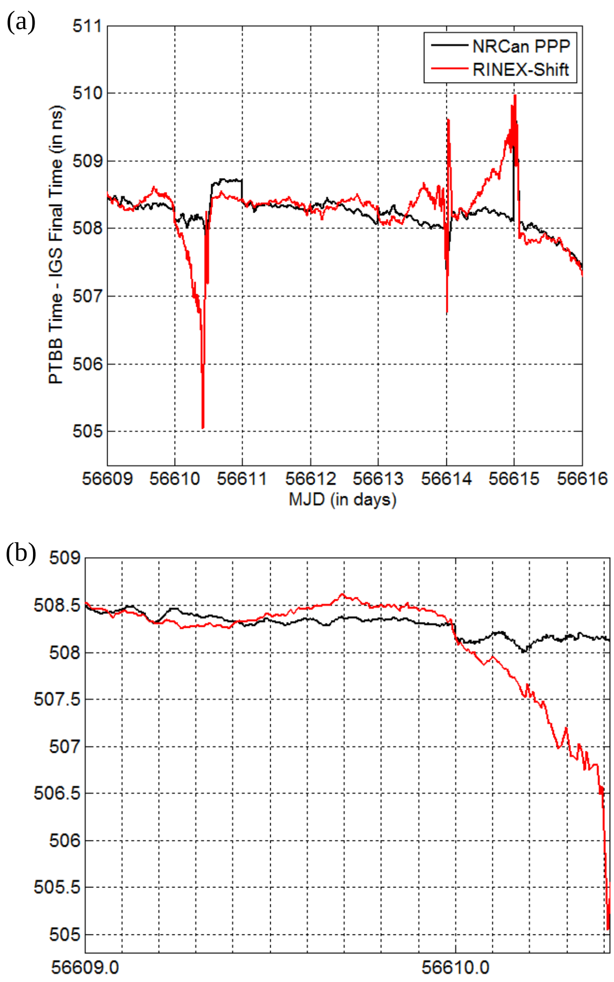

Figure 1. Comparison of NRCan PPP (black curve) and the RINEX-Shift algorithm (red curve) at anomalies at PTB. (b) is enlarged from (a).

However, the RINEX data of *PTBB* show that there are three time periods of missing GPS data during MJD 56609.0–56616.0. The first one occurs between 10:00:00 and 10:49:30 on MJD 56610, the second one 01:00:00–01:59:30 on MJD 56614, and the third one 01:00:00–01:59:30 on MJD 56615. The red curve in Figure 1(a) shows bad time transfer results of the RS algorithm whenever the missing data occur. The deviation from the NRCan PPP result can be as large as 3 ns. As an example, at 9:50:00 on MJD 56610 (i.e., MJD 56610.410), the red curve reads 505.049 ns, while the black curve reads 508.140 ns (see Figure 1(b)). This indicates an error of 3.091 ns in the RS algorithm. Besides, the large deviation occurs only on the left side of the bad epochs. As the time gets away from the bad epochs, the difference between the two curves shows a damped oscillation. For example, from MJD 56610.4 to 56610.0, the difference between the two curves decreases quickly. At MJD 56610.0, the difference is almost 0. Then from MJD 56610.0 to 56609.7, it increases to about 0.25 ns, and again decreases to 0 at MJD 56609.4. For the epochs before MJD 56609.4, the red curve becomes very close to the black curve. This indicates that after about 1 day, the damped oscillation due to a data anomaly in the RS algorithm becomes negligible. Analyzing the two curves from MJD 56612.0–56614.1 and from MJD 56614.1–56615.0 also shows that the damped oscillation almost disappears after approximately 1-day decay.

Figure 2 provides another example. The reference time for the *NIST* receiver was adjusted by approximately +23 ns around MJD 56646 and thus it leads to a GPS-data anomaly. In addition, no data were recorded during 18:22:00–18:28:30 on MJD 56647 and thus another anomaly. These two anomalies both result in damped oscillations in the RS algorithm (red curve). Again, the oscillation is negligible after about 1-day decay.

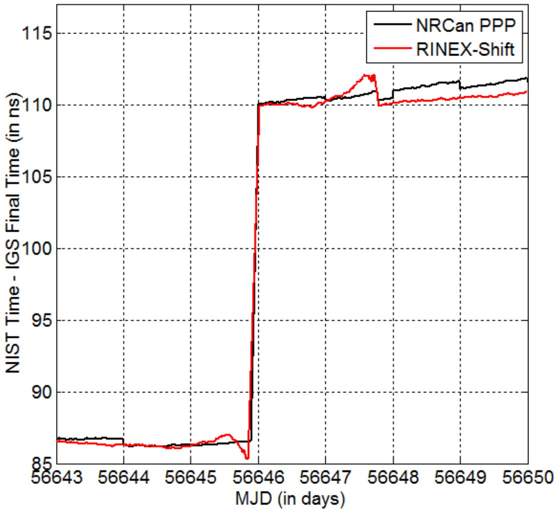

Figure 2. Comparison of NRCan PPP (black curve) and the RINEX-Shift algorithm (red curve) at anomalies at NIST.

From the above analysis, the RS algorithm suffers from a damped oscillation with a maximum amplitude of a few nanoseconds, for the epochs before an anomaly. It takes 1 day for the oscillation to decay to a negligible value. In the next section, we will introduce the “isolated island effect” and then explore the mechanism of the damped oscillation in the result of the RS algorithm in Section IV.

III. ISOLATED ISLAND EFFECT

The PPP method must estimate phase ambiguities for each GPS data-arc. If there is a few minutes of data anomaly, the data-arc is split into two pieces: the first sub-data-arc and the second sub-data-arc. PPP typically treats the second sub-data-arc (i.e., the data after the anomaly) as a new data-arc and thus re-estimates phase ambiguities which are usually different from the estimations for the first sub-data-arc. Because of different phase-ambiguity estimations for the first and second sub-data-arcs, we have a boundary discontinuity at the anomaly (Figure 3).

The blue curve in Figure 3 shows the PPP result for the original RINEX data of *PTBB* on MJD 56489. It is continuous because of no anomaly. However, if we delete 10 minutes of RINEX data, e.g., 13:00:30–13:10:00, then the PPP result has a boundary discontinuity of 1.07 ns at 13:00:00 (red curve). If the 10-min missing-data window occurs at other epochs (grey, orange, black, green, and magenta curves), we still have a boundary discontinuity, although the jump value varies. This illustrates that an anomaly leads to a boundary discontinuity. Notice that the slopes of all curves are more or less the same. Thus, an anomaly is not too bad, if only frequency transfer, instead of time transfer, is our main concern.

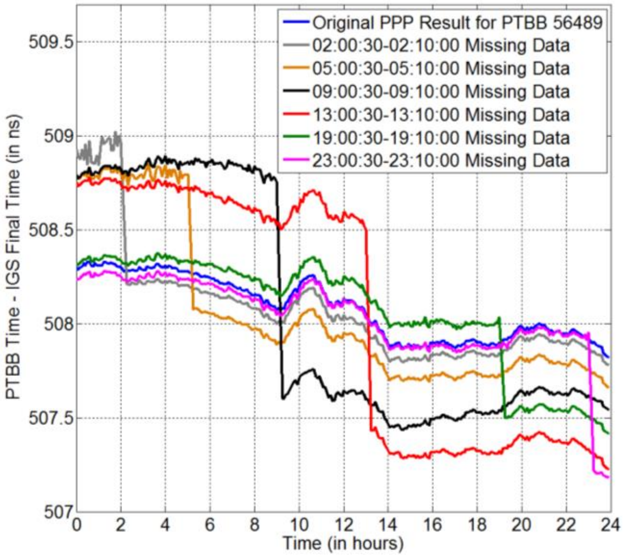

Figure 3. Anomaly in the middle range of the data-arc and boundary discontinuity. The blue curve is the original PPP result for *PTBB* on MJD 56489. All other curves are the PPP results with 10-min missing data.

However, the above analysis is based on the premise that an anomaly occurs in the middle range of a data-arc. If an anomaly occurs at the beginning or at the end of a data-arc, the behavior of the PPP result becomes quite different.

Figure 4 shows the PPP results when the 10-min missing-data window is close to the beginning of a data-arc. The blue curve is the PPP result of the original RINEX data, for reference. If the missing-data window occurs at 00:00:30–00:10:00 (that means that 00:00:00 is the only epoch of valid data before the anomaly since the GPS receivers record RINEX data every 30 s.), the time transfer error can be as large as 1.7 ns (solid red curve at 00:00:00). This is because the phase ambiguity cannot be solved for if only one epoch of measurements are provided [13]. This makes the phase measurements useless.

Thus, the time solution at 00:00:00 is only determined by the pseudorange measurements, which are much noisier than the phase measurements. As the missing-data window shifts away from the data-arc edge (e.g., the dashed red curve), we have a longer period of valid data (in this case, it is 30 min) between the edge and the anomaly. Therefore, we are getting a better estimation of the phase ambiguities. However, this estimation is still not good enough. Reference [12] tells us that the PPP program requires at least 1 hour of data to converge to an accurate estimation of phase ambiguities. 30 min is still not sufficient for PPP to converge to a reasonable result, which makes the dashed red curve still quite noisy during the first sub-data-arc. When the missing-data window occurs more than 1 hour away from the edge (e.g., the dashed green curve in Figure 4), the PPP result for the first sub-data-arc becomes smooth.

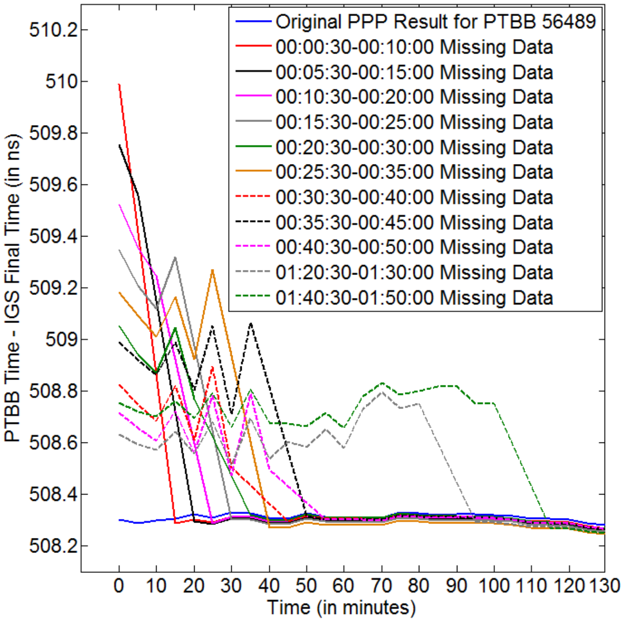

Figure 4. Anomaly at the beginning of the data-arc and boundary discontinuity. The blue curve is the original PPP result for *PTBB* on MJD 56489. All other curves are the PPP results with 10-min missing data.

From the above analysis, we know that an anomaly in the middle range of data-arc results in a boundary discontinuity, but the time-transfer slope is almost unchanged; an anomaly at the edge of data-arc (less than 1 hour from the edge) results in a completely-damaged PPP result between the anomaly and the edge. We call this phenomenon the “isolated island effect”.

IV. MECHANISM OF DAMPED OSCILLATION IN RINEX-SHIFT ALGORITHM

In the RS algorithm, we run PPP for a data-arc of 10 days and extract the PPP result at the first epoch (here, we define the PPP result at the first epoch as “*PPP_FE”*). If the anomaly is more than 10 days away from the first epoch, it does not affect the result of the RS algorithm. However, as we shift the RINEX data step by step, the new data-arc starts to cover the anomaly. The anomaly first shows at the end of the data-arc, and thus the length of the first sub-data-arc is 10 days (here, we define the length of the first sub-data-arc, i.e., the time difference between the anomaly and the first epoch, as *deltaT*). Then the RINEX data continue shifting and *deltaT* changes from 10 days to 0. At the same time, all *PPP_FE*s are extracted to form the RS solution. In the end, the data-arc goes over the anomaly (i.e., *deltaT* becomes negative) and thus the anomaly no longer affects the result of the RS algorithm. From the above, we can see that the RS solution after the anomaly is not affected by the anomaly; but the RS solution before the anomaly is determined by the relationship between *PPP_FE* and *deltaT*. Thus, we next study this relationship.

Actually, we have had the “*PPP_FE*–*deltaT*” relationship if we interpret Figure 3 and Figure 4 in a different perspective. For the grey curve in Figure 3, the length of the first sub-data-arc “*deltaT*” is 2 hours, and the PPP result at the first epoch (i.e., 00:00:00) is 508.926 ns, which is +0.624 ns away from the original PPP result (blue curve in Figure 3) at 00:00:00. Thus, we have (*deltaT*, *PPP_FE*) = (2.008 hours, 508.926 ns). Similarly, for the orange curve in Figure 3, we have (*deltaT*, *PPP_FE*) = (5.008 hours, 508.794 ns). So on and so forth. This gives a series of (*deltaT*, *PPP_FE*) pairs. The red curve in Figure 5 shows the result (note, it uses some curves not shown in Figures 3–4). The original PPP result at 00:00:00 is given for reference (blue curve).

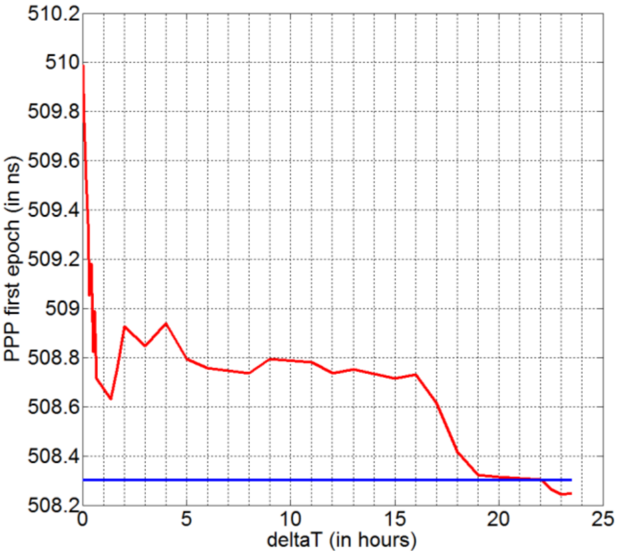

Figure 5. *PPP_FE*–*deltaT* graph. The *PPP_FE*–*deltaT* relationship determines the RS solution before the anomaly.

We can see that Figure 5 and Figure 1(b) have many common features, which demonstrates that the "*PPP_FE–deltaT*" relationship does determine the result of the RS algorithm. First, in Figure 5, we can clearly see that *PPP_FE* (red curve) is approximately 1.7 ns away from the original PPP result (blue curve) when *deltaT* is close to 0. This indicates that the error in the RS algorithm can be as large as several nanoseconds when the anomaly is very close to the first epoch. This matches our results in Figure 1(b). Second, when *deltaT* increases from 0 to 40 min, *PPP_FE* decays toward the original PPP result very quickly (Figure 5). This is very similar to the behavior of the RS solution during MJD 56610.375–56610.403 (red curve in Figure 1(b)). Third, for the range of 2 hours to a few hours, Figure 5 shows that *PPP_FE* decreases slowly and steadily toward the original PPP result, which is again quite similar to the RS solution for the time range of 56610.05–56610.33 in Figure 1(b). Last, when *deltaT* is more than 20 hours, *PPP_FE* becomes smaller than the original PPP result (Figure 5). Besides, the absolute maximum difference between *PPP_FE* and the original PPP result is only about 0.059 ns, which is much smaller than if *deltaT* is in the range of 2 hours to a few hours. This indicates a damped oscillation in *PPP_FE*. Again, this matches the damped oscillation behavior in the result of the RINEX-Shift algorithm (Figure 1(b)).

The above analysis shows that the *PPP_FE–deltaT* relationship does determine the RS result. Because the *PPP_FE–deltaT* relationship is derived from the isolated island effect, we conclude that it is the isolated island effect that leads to the damped oscillation behavior in the RS algorithm. Besides, the scope of the isolated island effect is approximately 1 day. Once a sub-data-arc is longer than 1 day, the impact of isolated island effect on the RS algorithm is negligible.

V. PRINCIPLE OF REVISED RINEX-SHIFT ALGORITHM

Based on the discussion in Sections II–IV, we know that if an anomaly is less than 1 day away from the edge of a data-arc (i.e., *deltaT* < 1 day), the time-transfer result of the RS algorithm in between is incorrect, because of the isolated island effect. If *deltaT* > 1 day, the result in between becomes reasonable. Thus we must avoid the situation when the anomaly is less than 1 day away from the data-arc edge. If we can achieve this, we can solve the damped oscillation problem in the RS algorithm. One good way is to extract the PPP result at the middle epoch of the data-arc, instead of the first epoch of the data-arc as in the RS algorithm. We call this "revised RINEX-Shift (RRS) algorithm". The following example explains how it works.

In Figure 1, there is an anomaly occurring at 01:00:00 on MJD 56614 (i.e., 56614.042). When we run PPP for the data-arc of MJD 56604.044–56614.044, the anomaly starts to affect the time-transfer result. We know that the result of the second sub-data-arc (i.e., 56614.043–56614.044) is completely damaged because the length of this sub-data-arc is too short for PPP to converge. However, the result of the first sub-data-arc (i.e., 56604.044–56614.041) is still good because the length of this sub-data-arc is greater than 1 day and the isolated island effect is negligible. If the RRS algorithm is applied, the PPP result at the middle epoch (i.e., 56609.044) is extracted. Since the middle epoch is within the first sub-data-arc, the PPP result at this epoch is good. As we continue shifting the data-arc until 56609.040–56619.040, the first sub-data-arc changes from 56604.044–56614.041 to 56609.040–56614.041. When the data-arc is 56609.040–56619.040, the length of the first sub-data-arc is 5.001 days, which is still long enough to make the isolated island effect negligible. Thus, the result at the middle epoch (i.e., 56614.040) extracted by the RRS algorithm is still good. Next, the data-arc shifts to 56609.043–56619.043. Now the PPP result at 56614.043 is extracted. This epoch is within the second sub-data-arc (i.e., 56614.042–56619.043). Since the second sub-data-arc is again longer than 1 day, the PPP result at 56614.043 is still good.

From the above example, we can see that once we extract the middle epoch of the data-arc, the sub-data-arc where the middle epoch lies in is always longer than 5 days, which is long enough to eliminate the impact of the isolated island effect. Thus, the RRS algorithm should provide reasonable results at all epochs. We will test its performance in the next section.

VI. PERFORMANCE OF REVISED RINEX-SHIFT ALGORITHM

In this section, we first test the performance of the RRS algorithm when there is an anomaly in the RINEX data. Figures 6–7 show the results of the RRS algorithm for the same periods as Figures 1–2. Similar to NRCan PPP (black curve) and the RS algorithm (red curve), the revised RINEX-Shift algorithm (blue curve) has a boundary discontinuity at the anomaly (e.g., MJD 56610.417). This is because the epochs before and after the anomaly belong to two different sub-data-arcs and thus different phase ambiguities are estimated. Like the RS algorithm, the RRS algorithm eliminates the day boundary discontinuities successfully. Most importantly, as compared to the RS algorithm, there is no damped oscillation problem in the RRS algorithm.

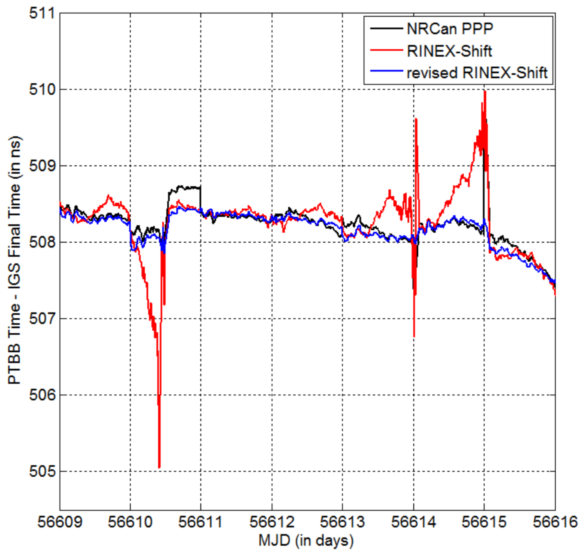

Figure 6. Comparison of NRCan PPP (black curve), RS algorithm (red curve) and RRS algorithm (blue curve) at anomalies at PTB.

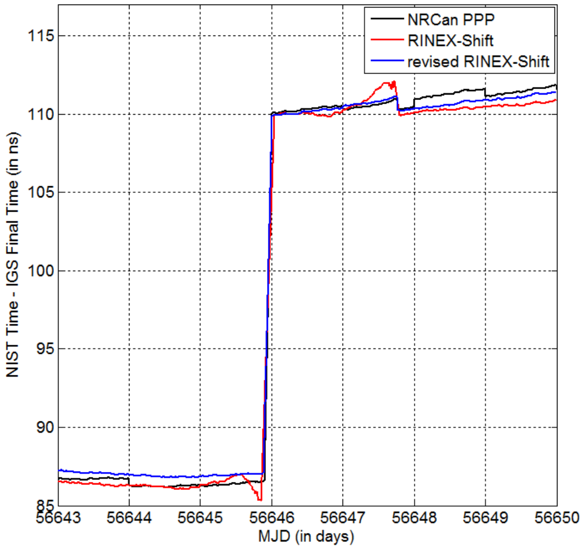

Figure 7. Comparison of NRCan PPP (black curve), RS algorithm (red curve) and RRS algorithm (blue curve) at anomalies at NIST.

Next, we test the performance of the RRS algorithm when the RINEX data are all good. To be consistent with the figures in [1], we run the RRS algorithm for the same MJDs as [1]. Besides, we use IGS 30-sec clock products, instead of IGS 5-min clock products, as the input of the RRS algorithm, because [1] shows that the time-transfer results using IGS 30-sec clock products as the input are closer to the true values than if using IGS 5-min clock products. In Figure 8, we do a time comparison between the *NIST* receiver and the *PTBB* receiver for MJD 56389–56409 by NRCan PPP (black curve), RS algorithm (red curve), and RRS algorithm (blue curve). We can see that both RS and RRS provide continuous time transfer results. The modified total deviation (MTD) is used to characterize the frequency stability of the three time-transfer methods (Figure 9). Obviously, the RRS algorithm provides the best frequency stability. It reduces the time-transfer noise of the RS algorithm by 10–55%. The most significant improvement of the RRS algorithm over the RS algorithm occurs at the averaging time of 1–4 days. For an averaging time of 1.75 days, the frequency stability of the RRS algorithm is only 8.6 × 10^{-16}, which makes it the best GPS time transfer result. Besides, the MTD of the RRS algorithm increases for an averaging time of greater than 1.75 days, which indicates that we have already seen the clock noise after 1.75 days.

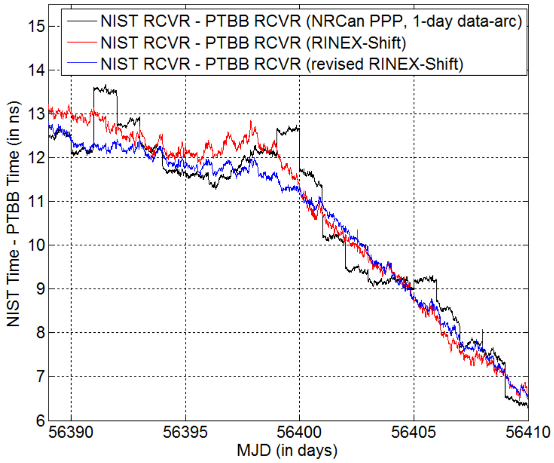

Figure 8. Time comparison between *NIST* and *PTBB* during MJD 56389–56409 by NRCan PPP (black curve), RS algorithm (red curve), and RRS algorithm (blue curve). The curves are shifted to overlay each other for better comparison.

As stated in [10], the smoothest time-transfer solution is not necessarily the most accurate solution. The above analysis only shows that the RRS algorithm does provide the smoothest solution. Next, we need to test the accuracy of the RRS algorithm. This can be done by comparing two receivers at the same station, because the reference clock noise is cancelled out and only the time transfer noise remains. Figure 10 shows the time difference between the *NIST* receiver and the *NIS2* receiver at NIST by using different time transfer methods [1]. We can clearly see that NRCan PPP solutions (blue, black, and orange curves) are quite artificial. The time difference between the two common-reference-clock receivers is not continuous. The RS algorithm (red curve) provides a continuous time transfer. However, compared to the RRS result, the RS result seems more artificial. For example, at MJD 56400.0, the red curve decreases very quickly. In contrast, there is almost no decrease in the NRCan PPP solutions (black and orange curves) and the RRS solution at this epoch. For another example, there is a peak with an amplitude of about 0.6 ns during MJD 56411–56416 in the red curve. In contrast, the peak is tiny in the RRS solution (green curve), which matches the NRCan PPP 30-day data-arc solution (orange curve). Both the NRCan PPP result and the RRS result for MJD 56440–56460 are smooth. On contrary, the RS solution for this period is quite noisy. All these make us conclude that the RRS algorithm provides time-transfer results closer to the true values than the RS algorithm.

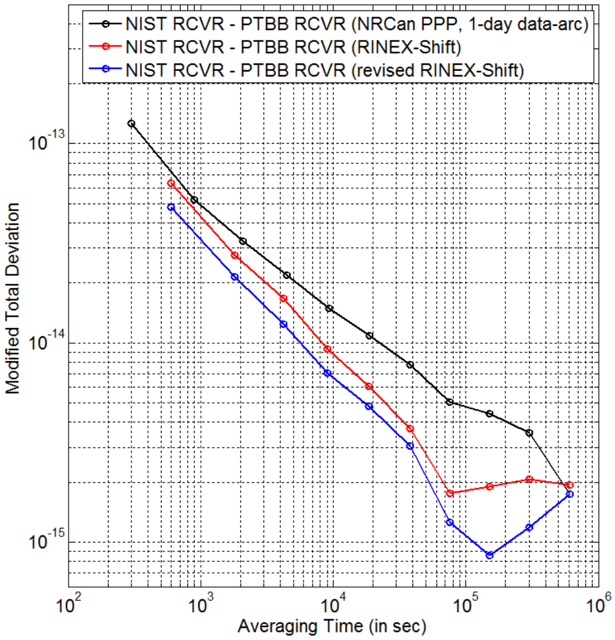

Figure 9. MTD of time difference between *NIST* and *PTBB* for MJD 56389–56409, by using NRCan PPP (black curve), RS algorithm (red curve), and RRS algorithm (blue curve).

To further verify the above conclusion, we do another test. We run the RRS algorithm for NIST and PTB for MJD 56375–56476 and compute the MTD of the difference between TWSTFT and RRS. According to [1], this characterizes how well RRS matches TWSTFT. The result is shown by the green curve in Figure 11. Note, to be consistent with the settings of the NEW PPP result in Figure 19 of [1], we first implement the RRS algorithm in the NEW PPP program (we call the updated version of NEW PPP "NEW PPP Version 2"), and then run NEW PPP Version 2 for two receivers at NIST (*NIST* and *NIS2*) and two receivers at PTB (*PTBB* and *PTBG*). Figure 11 clearly shows that the RRS result (or NEW PPP V2) matches TWSTFT best. An obvious improvement of the RRS algorithm over the RS algorithm occurs at the averaging time of 2–15 days. In addition, the slope of the green curve during 1–15 days is approximately −1. This indicates that the flicker phase modulation noise [14] dominates in the RRS time transfer during 1–15 days. Last, the green curve in Figure 11 also sets the upper limit of the RRS (or NEW PPP V2) time transfer noise (e.g., the RRS time-transfer noise is less than 4×10^{-16} for an averaging time of 4.25 days).

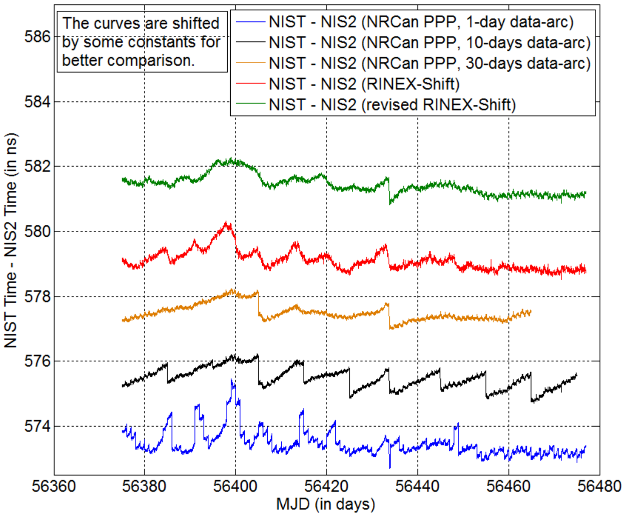

Figure 10. Time difference between *NIST* and *NIS2* for MJD 56375–56476, by using different PPP time transfer methods (NRCan PPP with 1-day data-arc (blue curve), NRCan PPP with 10-day data-arc (black curve), NRCan PPP with 30-day data-arc (orange curve), RS algorithm (red curve), and RRS algorithm (green curve)). Note, the blue, black, orange, and red curves are copied from [1].

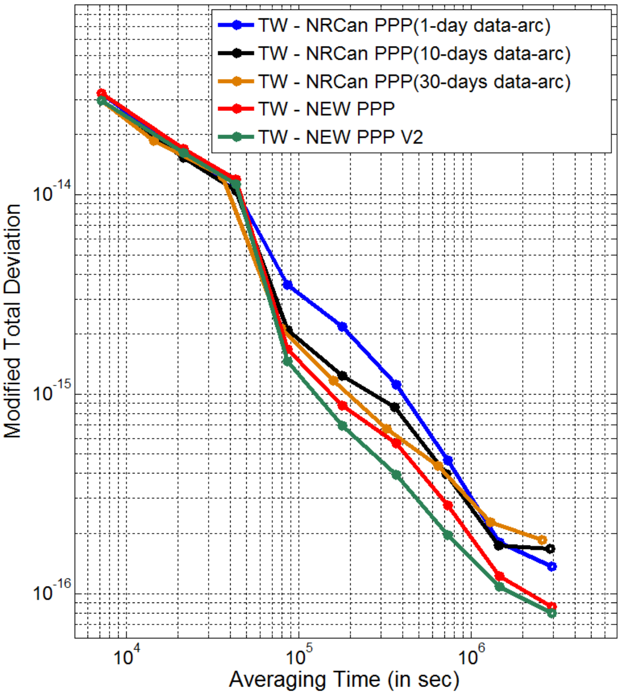

Figure 11. MTD of the double-difference between TWSTFT and different PPP time transfer methods (NRCan PPP with 1-day data-arc (blue curve), NRCan PPP with 10-day data-arc (black curve), NRCan PPP with 30-day data-arc (orange curve), RS algorithm or NEW PPP (red curve), and RRS algorithm or NEW PPP Version 2 (green curve)), for MJD 56375–56476. Note, blue, black, orange, and red curves are copied from [1].

As a whole, the RRS algorithm successfully removes the RS algorithm's damped oscillation problem occurring before the anomaly. More importantly, it matches TWSTFT best and reduces the time transfer noise of the RS algorithm by 10–55%. This makes the RRS algorithm the best GPS time transfer method.

VII. CONCLUSIONS

In this paper, we find that the isolated island effect leads to the damped oscillation behavior in the RS algorithm. In order to eliminate the impact of the isolated island effect on the RS algorithm, we propose the RRS algorithm, which extracts the PPP result at the middle epoch of a data-arc, instead of the first epoch as in the RS algorithm. It shows that the RRS algorithm solves the damped oscillation problem in the RS algorithm successfully. In addition, the RRS algorithm reduces the time transfer noise of the RS algorithm by 10–55%. The biggest improvement occurs at an averaging time of 1–4 days. Last, the RRS algorithm matches the TWSTFT result better than the RS algorithm. This indicates that the RRS algorithm approaches the true value more closely than the RS algorithm. These results indicate that the RRS algorithm is the best GPS time transfer method.

ACKNOWLEDGMENTS

The authors thank Francois Lahaye for sharing the NRCan PPP software. Victor Zhang and Marc Weiss are thanked for some helpful discussions about the NIST receiver anomaly. We also thank Trudi Peppler for sharing PCWork, a software package of frequency stability analysis. IGS is gratefully acknowledged for providing GPS tracking data, station coordinates, and satellite ephemerides. Finally, we thank those people who maintain the GPS receivers at NIST and PTB.

REFERENCES

[1] J. Yao and J. Levine, “A new algorithm to eliminate GPS carrier-phase time transfer boundary discontinuity,” Proc. 45th PTTI Meeting, pp. 292-303, 2013.

[2] J. Levine, “A review of time and frequency transfer methods,” Metrologia, 45, S162-S174, 2008.

[3] J. Yao and J. Levine, “GPS carrier-phase time transfer boundary discontinuity investigation,” Proc. 44th PTTI Meeting, pp. 317-326, 2012.

[4] J. Ray and K. Senior, “IGS/BIPM pilot project: GPS carrier phase for time/frequency transfer and time scale formation,” Metrologia, vol. 40, pp. 270-288, 2003.

[5] R. Dach, T. Schildknecht, T. Springer, G. Dudle, and L. Prost, “Continuous time transfer using GPS carrier phase,” IEEE Trans. Ultrason., Ferroelect., Freq. Contr., vol. 49, no. 11, pp. 1480-1490, 2002.

[6] P. Defraigne and C. Bruyninx, “On the link between GPS pseudorange noise and day-boundary discontinuities in geodetic time transfer solutions,” GPS Solutions, vol. 11, pp. 239-249, 2007.

[7] K. Senior, E. Powers, and D. Matsakis, “Attenuating day-boundary discontinuities in GPS carrier-phase time transfer,” Proc. 31st PTTI Meeting, pp. 481-490, 1999.

[8] N. Guyennon, G. Cerretto, P. Tavella, and F. Lahaye, “Further characterization of the time transfer capabilities of precise point positioning (PPP): the sliding batch procedure,” IEEE Trans. Ultrason., Ferroelect., Freq. Contr., vol. 56, no. 8, pp. 1634-1641, 2009.

[9] M. Weiss, J. Yao, and J. Li, “In search of a new primary GPS receiver for NIST,” Proc. 44th PTTI Meeting, pp. 179-186, 2012.

[10] R. Dach, T. Schildknecht, U. Hugentobler, L.-G. Bernier, and G. Dudle, “Continuous geodetic time transfer analysis method,” IEEE Trans. Ultrason., Ferroelect., Freq. Contr., vol. 53, no. 7, pp. 1250-1259, 2006.

[11] C. Bruyninx and P. Defraigne, “Frequency transfer using GPS codes and phases: short- and long-term stability,” Proc. 31st PTTI Meeting, pp. 471-479, 1999.

[12] J. Kouba and P. Heroux, “Precise point positioning using IGS orbit and clock products,” GPS Solutions, vol. 5, pp. 12-28, 2001.

[13] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, “GNSS global navigation satellite systems: GPS, GLONASS, Galileo and more,” Chapter 6, Springer, Berlin, 2008.

[14] W. J. Riley, “Handbook of frequency stability analysis,” NIST Special Publication 1065, 2008.