An Efficient Algorithm for Refining Position and Velocity Outputs of Space borne GNSS Receivers

Shuhao Chang[1], Xi Chen[1], and Menglu Wang[2]

[1] Research Institute of Tsinghua University in Shenzhen, Shenzhen 518057, China,
chang-sh15@mails.tsinghua.edu.cn,
chenxiee@tsinghua.edu.cn,
[2] School of Aerospace Engineering, Tsinghua University, Beijing 100084, China,
wangml14@mails.tsinghua.edu.cn

Abstract. Space borne GNSS receivers may not have enough tracked GNSS space vehicles due to the geometry caused by high altitude and limited receiver sensitivity. In many cases, such as real-time orbit determination for communication satellites, it is necessary to refine the position and velocity outputs of a space borne GNSS receiver for improved accuracy and robustness before actually using them. Toward this problem, an efficient algorithm jointly using weighted Runge-Kutta integration and cubic Hermite polynomial interpolation is proposed in this work. Simulations are conducted based on the GNSS data of GRACE-B satellite and LING QIAO satellite, the result of which show the proposed algorithm can effectively eliminate outliers and significantly reduce the root mean square error of GNSS position and velocity outputs.

Key words: GNSS, space borne, Runge-Kutta, refinement

1 Introduction

As one of the most important technologies of human history, global navigation satellite systems (GNSS) have been used for various purposes in both civil and military areas. It can provide precise, continuous, world-wide, three dimensional position and velocity information to users with appropriate positioning algorithms. Space borne GNSS receivers are those GNSS receivers used by space crafts and satellites. The use of a GNSS receiver as the orbit position, velocity and even attitude sensor, has become a standard component of satellites nowadays, especially LEO satellites. Precision and cost-effectiveness, among others, are main contributors to the popularity of GNSS in space missions.

Due to higher altitude, space borne GNSS receivers generally have less average trackable GNSS space vehicles than their ground counterparts. In order to obtain a position and velocity reference of LEO satellites, at least four tracked navigational space vehicles should be in sight. However, space borne GNSS receivers may have not enough tracked GNSS satellites in view. For LEO satellites, a lack of trackable GNSS space vehicles can only occur occasionally. Swarm C

satellite of European Space Agency (ESA) flies in an orbit 530km above the earth approximately and the possibility that it tracks less than four GPS satellites is 2.6% [1]. The number of trackable GNSS space vehicle will decrease as orbit altitude increases. Actually, Space-borne GNSS receivers can be used not only for orbit determination of LEO satellites, but also for orbit determination of satellites with higher altitude. NASA Goddard Space Flight Center (GSFC) had developed a new space-borne GPS receiver called Navigator that can operate effectively in the full range of Earth orbiting missions from LEO to GEO and beyond. For such GNSS receivers, the opportunity of tracking four or more satellites drops more significantly than LEO GNSS receivers. Besides orbit height, there are other reasons causing a space borne GNSS receiver to have inferior observations of tracked GNSS space vehicle signals. For example, erroneous bit synchronization due to limited acquisition sensitivity, scintillations on GNSS systems are much more severe and frequent for LEO satellites, particularly during posts unset hours [2], because of which observations may contain big error. Space radiation effects such as single element upset, may also cause large observation error.

A space borne GNSS receiver typically uses iterative least square or a Kalman filter to figure out a positioning solution. These L2 algorithms are not robust enough when there are large observation deviation. Theoretically, large observation deviation can be checked out and eliminated in pretreatment. However, preprocessing only works well when there are enough observations.

In existing work, we proposed a position and velocity refining algorithm based on short-arc filtering [3] at the expense of computation complexity. As a continuation of existing work, a more robust and efficient algorithm for refining position and velocity outputs of space borne GNSS receivers is proposed in this work. It is based on weighted Ronge-Kutta method and cubic Hermite polynomial interpolation. To overcome speed deficiency of Runge-Kutta method, enough but less data with outliers being screened out are used. Simulations with orbit data from GRACE-B satellite and GNSS data of LING QIAO(NORAD ID: 40136, International code: 2014-051A) were undertaken to show the efficiency and robustness of the proposed algorithm.

The paper is organized as follows: Section 2 gives the details and discussion of the proposed algorithm. Section 3 presents simulation performance and numerical results. The paper concludes in Section 4.

2 The Proposed Algorithm

2.1 General Idea

The proposed algorithm for refining position and velocity outputs of space borne GNSS receivers works as follows:

1. Save the output of a GNSS receiver for the last N epochs.

2. Approximately obtain the curve of position and velocity to get knowledge of a gentle and non-anomalous trajectory through cubic Hermite polynomial interpolation or polynomial fitting.
3. Calculate the Euclidean distance from each point to the three-order curve and omit anomalous points once they are too far away from the curve. A confirmed threshold q is required to determine whether the point we are aiming at is largely affected by non-ideal factors.
4. Gill's formula of weighted forth-order Runge-Kutta method is involved to attain better reference of position and velocity of the current epoch. We only use the "fine points" we have already screened out in the former step and reciprocals are used as weights.
5. The robustified position and velocity reference of the current epoch we have already got can be used in the following epochs to revised the orbit.

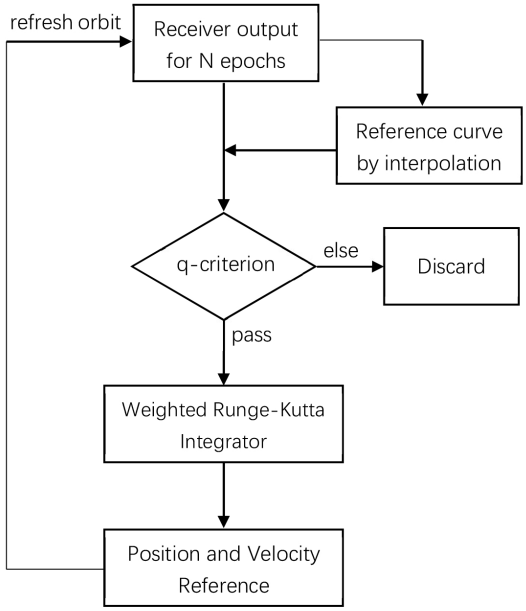

Fig. 1. Flowchart of the proposed algorithm

2.2 Algorithm Details

For the sake of convenience, we first declare a mathematical abstraction as a unified language in this paper. We map each output of GNSS receiver into the seven-dimension space as a point, epoch by epoch. Consider the six-tuple containing positions and velocities of x,y and z axis as a description of a point's

state and, naturally, we obtain a series of six-tuples as

$$\mathbf{X}(t) = \begin{bmatrix} P_x(t) \\ P_y(t) \\ P_z(t) \\ V_x(t) \\ V_y(t) \\ V_z(t) \end{bmatrix} = \begin{bmatrix} P(t) \\ V(t) \end{bmatrix} \tag{1}$$

Now we can easily write down a relational expression:

$$V(t) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \begin{bmatrix} P(t) \\ V(t) \end{bmatrix} = \mathbf{A}\mathbf{X}(t) \tag{2}$$

Low earth orbit communication satellites' movements are mainly affected by J2 perturbation in earth's non-spherical perturbation among all kinds of perturbation forces [4]. Taking J2 perturbation into consideration, the acceleration of x, y and z axis can be denoted as follows:

$$\begin{cases} a_x = \left[-\frac{GM}{r^3}(1+\frac{3}{2}J_2(\frac{r_e}{r})^2(1-5(\frac{r_e}{r})^2)) + \omega_e^2\right] P_x + 2\omega_e V_y \\ \quad = \xi_1 P_x(t) + 2\omega_e V_y \\ a_y = \left[-\frac{GM}{r^3}(1+\frac{3}{2}J_2(\frac{r_e}{r})^2(1-5(\frac{r_e}{r})^2)) + \omega_e^2\right] P_y - 2\omega_e V_x \\ \quad = \xi_1 P_y(t) - 2\omega_e V_x \\ a_z = \left[-\frac{GM}{r^3}(1+\frac{3}{2}J_2(\frac{r_e}{r})^2(3-5(\frac{r_e}{r})^2))\right] P_z(t) \\ \quad = \xi_2 P_z \end{cases} \tag{3}$$

where GM is the geocentric gravitational constant, $J_2 = 1.082628 \times 10^{-3}$ is the Earth's second dynamic form factor in the Nodal precession, ω_e is the angular velocity of the earth, r_e is the equatorial radius of the earth and $r = \sqrt{P_x^2 + P_y^2 + P_z^2}$ is the orbit radius. In a unified format, we arrange the formulas above to get the derivative of state variable X:

$$a(t) = \begin{bmatrix} \xi_1 & 0 & 0 & 0 & 2\omega_e & 0 \\ 0 & \xi_1 & 0 & -2\omega_e & 0 & 0 \\ 0 & 0 & \xi_2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P(t) \\ V(t) \end{bmatrix} = \mathbf{B}\mathbf{X}(t) \tag{4}$$

Thus

$$\frac{d}{dt}\mathbf{X}(t) = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{X}(t) = \mathbf{\Phi}_{6\times6}\mathbf{X}(t) \tag{5}$$

Now we involve the forth-order Runge-Kutta method. We adopt Gill's formula of the Runge-Kutta method, for rounding errors are largely reduced in it [6]. In case we are given the initial value $\mathbf{X}_0 = \mathbf{X}(t_0)$ as well as the derivative $\frac{d}{dt}\mathbf{X}(t) = f(t, \mathbf{X}_0)$, then the value $\mathbf{X}(t_0 + h)$ could be estimated with confidence as follows:

$$\mathbf{X}(t_0 + h) = \mathbf{X}_0 + \frac{1}{6}(k_1 + (2-\sqrt{2})k_2 + (2+\sqrt{2})k_3 + k_4) \tag{6}$$

where

$$\begin{cases} k_1 = hf(t_0, \mathbf{X}_0) \\ k_2 = hf(t_0 + \frac{h}{2}, \mathbf{X}_0 + \frac{k_1}{2}) \\ k_3 = hf(t_0 + \frac{h}{2}, \mathbf{X}_0 + \frac{\sqrt{2}-1}{2}k_1 + \frac{2-\sqrt{2}}{2}k_2) \\ k_4 = hf(t_0 + h, \mathbf{X}_0 - \frac{\sqrt{2}}{2}k_2 + \frac{2+\sqrt{2}}{2}k_3) \end{cases} \tag{7}$$

Take each one of the N points as a initial point and calculate a reference result for the state of current epoch and give it a weight inversely proportional to time interval. For instance, the result we calculate from the point n epochs before should be conferred with weight $1/n$. Note that the initial points we use ought to be screened out by the judgement whether they are too *far* from the three-order fitting curve. Each point with a six-dimension Euclidean distance to the fitting curve larger than a delicately designed threshold q should be omitted. Add all reference results to a temporary register X_{temp} and calculate the sum of wights p. Thus, a better state reference for current epoch is derived as follows:

$$\mathbf{X}_{ref} = \mathbf{X}_{temp}/p \tag{8}$$

where

$$p = \sum_n \frac{1}{n} \ , \ n \in \{n \leq N \mid q - criterion\ pass\} \tag{9}$$

2.3 Rationale Analysis

In this part, several topics worthy for discussion are expounded. Feasibility and designation matters are included here.

During the motion of satellites, measurement errors including offset in bit synchronization, tracking loop instability as well as hatching effect can cause inaccuracy of orbit parameters. Analysis based on the real in-orbit data show that an abrupt decrease of actual tracked satellite number may be a major reason for outliers. Thus, in order to get accurate orbit parameters, we need an approximate and available reference to pick out those outliers which should be excluded from the following integration. It has been testified that piecewise polynomial interpolation, such as splines or piecewise Hermite interpolation, adopted on short intervals is such a desirable choice. Furthermore, piecewise cubic interpolation with sampling intervals of about 100-200 seconds is considered to have a perfect behaviour with slight error [5]. Negligible computational complexity is required during this step and continuity at the junction point is ensured. Meanwhile, cubic polynomial fitting is also feasible to generate the reference curve. More time cost, however, is inevitable to solve the linear equation group with Vandermonde matrix of time.

Here gives a third-order fitting curve to illustrate this procedure. Fig. 2 and 3 shows a practical local fitting error and we can determine the threshold q from

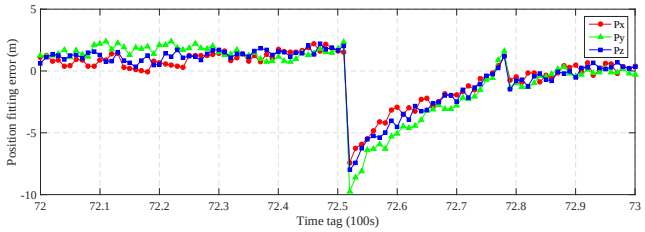

Fig. 2. Third-order fitting error of position

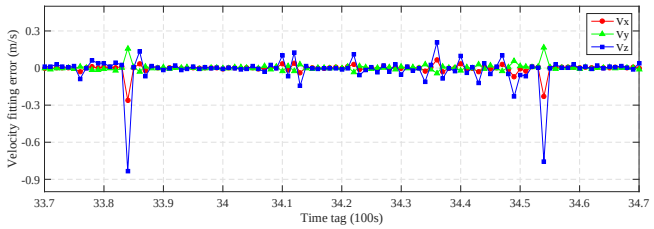

Fig. 3. Third-order fitting error of velocity

these graphs as priori knowledge. For deep fading situations, it both works to exclude outliers by thresholds for separate parameters or six-dimension Euclidean distance. Outliers can be ignored by q-criterion in this way and will not have impact on following integration procedures. It's worth noting that according to different circumstances, the parameters are supposed to be given different values. As for the epochs involved in the algorithm, computing time will increase and the interpolated or fitting curve will not work well with current fixed order if N is too large. While if N is chosen improperly small, the long-time error of the satellite will be neglected and acts out as wave property in the final estimate of error. It demonstrates the relationship between the performance of the algorithm and N in Fig. 4, taking LING QIAO satellite for example. And we can conclude that with the value of N between 10 and 15 a relatively reasonable performance of the algorithm on LING QIAO satellite can be achieved.

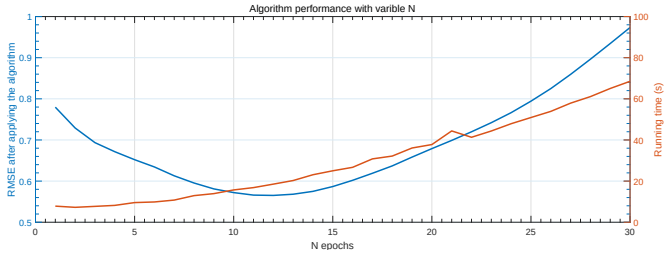

Fig. 4. Relationship between the performance of the algorithm and N

3 Simulation Results

3.1 Correctness Check

Before applying the algorithm to LING QIAO satellite, we first take correctness check on PVT data collected from GRACE B receiver to prove that the algorithm is Gaussian noise-resisting. Since GRACE-B flies in a polar orbit $500km$ over earth, the precision of its position and velocity data given by the receiver is relative high. To imitate a bad situation, additive white Gaussian noise(AWGN) is applied on original position and velocity data from the receiver. The position and velocity errors before going through the process system is depicted in Fig. 5 and Fig. 6 with red circles, while blue cubes shows the effect after going through the system. Here we involve last $N = 15$ epochs in integration and threshold is set as $q = 10m$. Results shows the RMSE of position reference decreased from 3.911m to 0.863m and for velocity reference it decreased from 0.434m/s to 0.131m/s, which is the normal range of GRACE's position error. This check shows the robustness of the proposed algorithm.

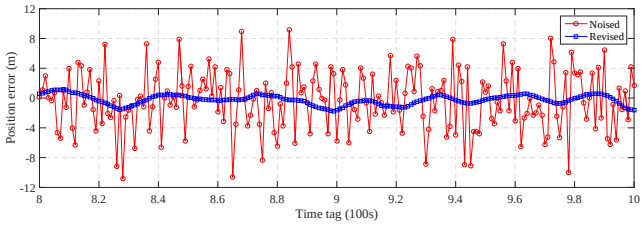

Fig. 5. Comparison of position errors of x-axis

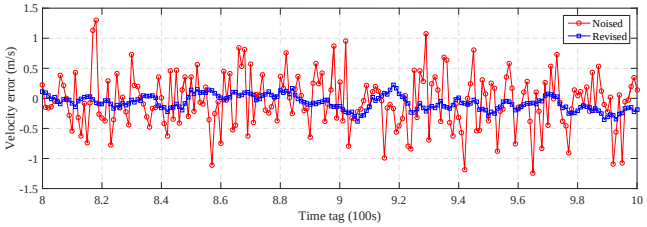

Fig. 6. Comparison of velocity errors of x-axis

3.2 Simulation on LING QIAO

In this part, simulations are conducted based on the original data provided by LING QIAO GPS receiver on Mar. 11th, 2015. LING QIAO satellite flies in an

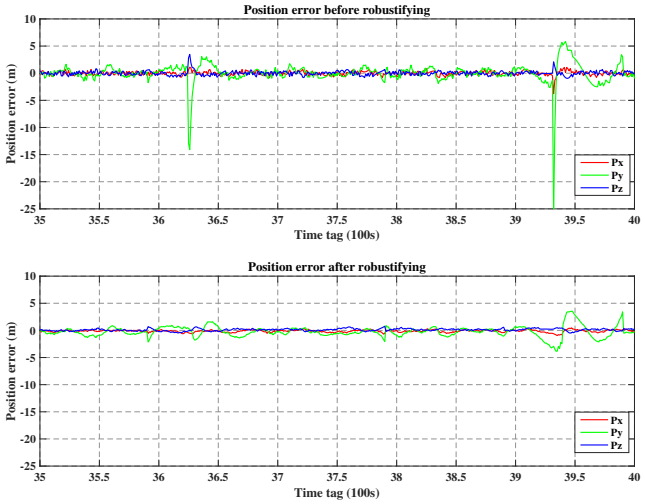

Fig. 7. Comparison of position errors

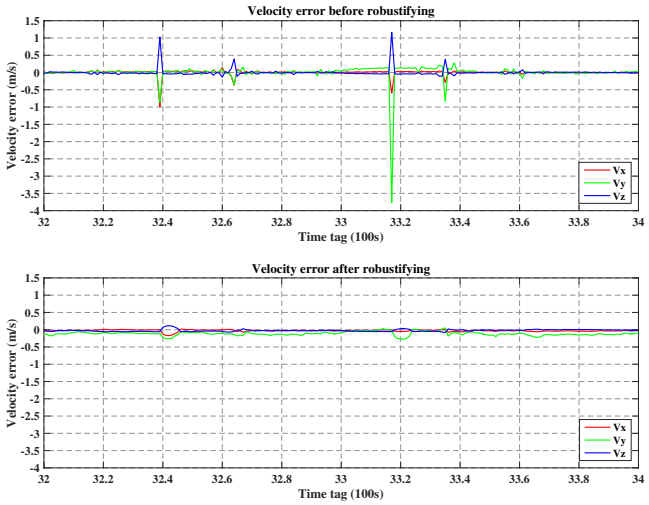

Fig. 8. Comparison of velocity errors

orbit $780km$ above the earth approximately and the GPS receiver output suffers stronger interference than GRACE satellites. In the simulations, the algorithm factors are adopted as follows: $N = 10$, $q = 10m$. In Fig. 7, position errors of x, y and z axis before and after robustifying are depicted and the results show a perfect property against signal deep-fading of the algorithm. The thing is, outliers of the original orbit data, such as the point at $t = 36.25, 31.30$, are revised by the algorithm. These outliers are very likely caused by abrupt short-time decrease of actual tracked satellite number. Through the process of

screening outliers and weighted integration, better reference for actual position is obtained. Analysis of velocity simulation is the same in Fig. 8. The RMSE of position reference decreased from 1.672m to 0.577m and for velocity reference it decreased from 0.086m/s to 0.029m/s. Thus the algorithm can approximately reduce the root mean square error of receiver output by 60% to 70%.

4 Conclusions

In this paper, we proposed an algorithm to acquire refined position and velocity as orbit state reference from space borne GNSS receiver outputs. Piecewise Hermite interpolation or similar fitting method are used to screen out nonanomalous points and reciprocals of time epochs are used to weight the Runge-Kutta solutions. It is also shown by simulation results based on real in-orbit data of GRACE-B and LING QIAO that the proposed algorithm can effectively eliminates outliers and significantly reduce the root mean square error of position and velocity data.

Acknowledgement

The authors would like to thank the Science, Technology and Innovation Commission of Shenzhen Municipality (No.JSGG20160429165838848).

References

1. Xiong, C., Stolle, C., Lxefhr, H.: The Swarm satellite loss of GPS signal and its relation to ionospheric plasma irregularities. SPACE. WEATHER. 14, 563-577 (2016)
2. Basu, S., Groves, K.M., Basu, S., Sultan, P.J.: Specification and forecasting of scintillations in communication/navigation links: current status and future plans. J. ATMOS. SOL-TERR. PHY. 64, 1745-1754 (2002)
3. Xi, C., Menglu, W., Wenyun, G.: Robustifying position and velocity references for communication satellites by short-arc filtering. International Conference on Wireless Communications & Signal Processing (WCSP), Nanjing (2015)
4. Montenbruck, O., Gill, E., Lutze, F.: Satellite Orbits: Models, Methods and Applications. APPL. MECH. REV. 55, 2504-2510 (2002)
5. Korvenoja, P., Piche, R.: Efficient satellite orbit approximation. Proceedings of International Technical Meeting of the Satellite Division of the Institute of Navigation, pp. 1930-1937 (2000)
6. Fuyang, K., Qing, W., Shuguo, P.: GLONASS orbit Runge-Kutta integral algorithm by automatic integral step length. J. SOUTHEAST. UNIV. 40, 755-759 (2010)