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

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

[1]Research Institute of Tsinghua University in Shenzhen

[2]School of Aerospace Engineering, Tsinghua University

Conference on Communications and Networking in China, 2017

Outline

Outline

Background

- GNSS is widely used in various areas.
    - Precise and affordable
    - Orbit position and velocity sensor
- Four tracked navigational space vehicles in sight are needed to locate a GNSS receiver.

Basic Problems

- Lack of trackable GNSS space vehicles occasionally.
  - Swarm C, 530km, 2.6%
  - The number of trackable GNSS space vehicles will decrease as orbit altitude increases.
- Other undesirable factors.
  - Limited acquisition sensitivity
  - Erroneous bit synchronization
  - Space radiation effects

Previous Work

Short-arc Filtering Based on Runge-Kutta Algorithm

- Advantages:
    - High precision
    - Low short-term prediction error
- Disadvantages:
    - High computation complexity
    - Not robust enough

Outline

General Idea

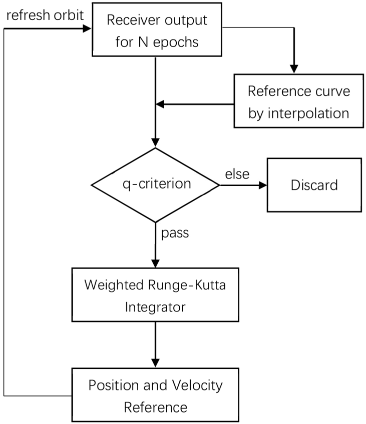

Figure: Flowchart of the proposed algorithm

Algorithm Details

- Exclude outliers

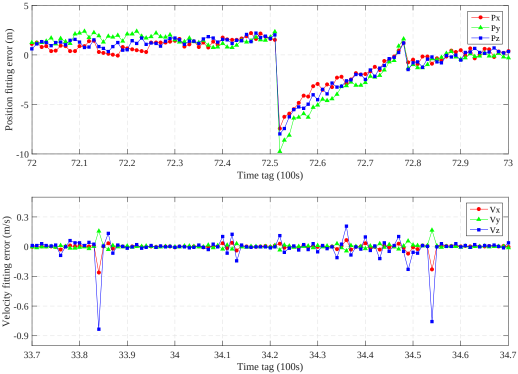

Figure: Third-order fitting error of position and velocity

Algorithm Details

- Gill's formula of the Runge-Kutta method

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

where

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

Algorithm Details

- Weighted sum

$$\mathbf{X}_{ref} = \sum_{n=1} \mathbf{X}_n / p \quad (3)$$

where

$$p = \sum_{n} \frac{1}{n}, \ n \in \{n \leq N \mid q - criterion\ pass\} \quad (4)$$

Parameter Choose

- Better choice for epochs number N

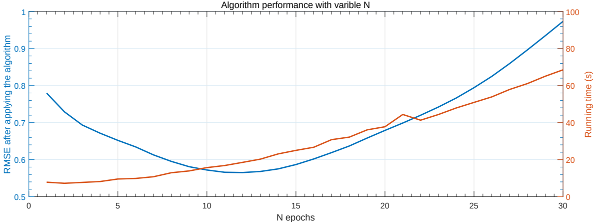

Figure: Relationship between the performance of the algorithm and N

Outline

Correctness Check

- Correctness check on PVT data collected from GRACE B receiver

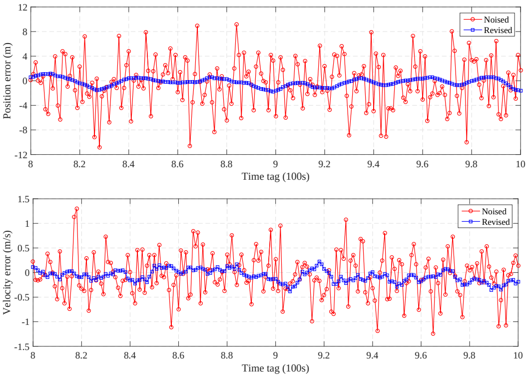

Figure: Comparison of position and velocity errors of x-axis

Simulation on LING QIAO

- Position errors of x, y and z axis before and after robustifying

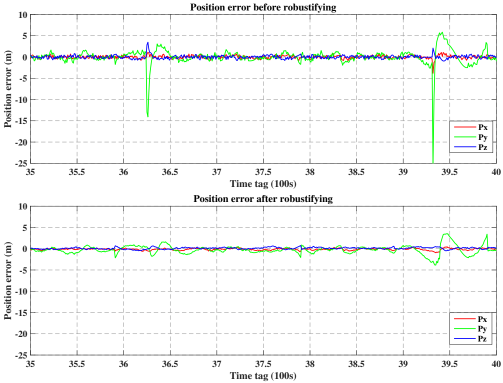

Figure: Comparison of position errors

Simulation on LING QIAO

- Velocity errors of x, y and z axis before and after robustifying

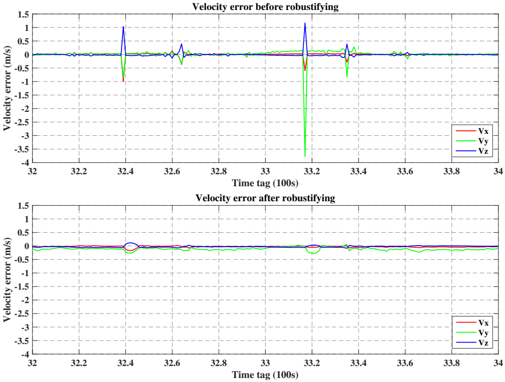

Figure: Comparison of velocity errors

Summary

- Weighted Runge-Kutta method guarantees accuracy.
- Fit or interpolate to obtain robustness.
- The algorithm effectively reduces RMSE of PVT data.

- Outlook
  - Different weighting parameters worth trying.
  - Can we train a group of weighting parameters?

Acknowledgement

Thank You For Your Listening!