GPS 응용을 위한 새로운 RAIM 알고리즘 개발 II

전창완°
순천향대학교 정보기술공학부

Development of A New RAIM Algorithm for GPS Applications II

ChangWan Jeon
Division of Information Technology Engineering, Soonchunhyang Univ.

Abstract

With the development of RAIM techniques for single failure, there has been increasing interest in the multiple failure problem. There have been many approaches to tackle the problem from various points of view. This paper approaches to two failure problem with total least squares (TLS) technique, which has rarely been addressed because TLS requires a great number of computations

1. Introduction

This paper approaches to two failure problem with total least squares (TLS) technique, especially focusing on the second step. RAIM technique based on the TLS has rarely been addressed because TLS requires a great number of computations. Recently, Juang[1] reformulated a linear measurement model and proposed a positioning and integrity monitoring scheme based on total least squares (TLS) instead of least squares. Jeon[2] proposed a sequential TLS-based RAIM algorithm for single failure problem. In this paper, the result of Jeon is extended for two failure problem.

2. The Algorithm for Single Failure Problem

Initialize

Step 1: Form the sub-matrix equation ($\bar{\mathbf{A}}\mathbf{x} = \bar{\mathbf{b}}$) by deleting the first column and compute $\bar{\mathbf{Q}}_1, \bar{\mathbf{R}}_1, \bar{\mathbf{B}}_1$ and the initial solution at t=1 and

$$\bar{\mathbf{Q}}_1^T\left[\bar{\mathbf{A}}_{1,1}; \bar{\mathbf{A}}_{1,2}; \bar{\mathbf{b}}_1\right] = \left[\bar{\mathbf{R}}_1; \bar{\mathbf{B}}_1\right] \quad (1)$$

Let $\boldsymbol{\alpha}_1$ and β_1 be satellite measurements to be inserted at the next step. In this case, the previously deleted first row becomes $\boldsymbol{\alpha}_1$(the first row of A) and β_1(the first row of b), since at the next step the second row will be deleted.

Phase I : Deleting a row

Step 2: Compute Givens rotations $\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_{n-2}$ such that

$$\mathbf{G}_1^T\mathbf{G}_2^T \cdots \mathbf{G}_{n-2}^T\mathbf{q}_1 = \rho\mathbf{e}_1 \quad (2)$$

where $\mathbf{q}_1^T$ be the first row of $\mathbf{Q}_k$ and $\rho = \pm 1$.

Step 3: Compute $\bar{\mathbf{R}}_k$, $\bar{\mathbf{Q}}_k$ and $\bar{\mathbf{B}}_k$.

$$\overline{\mathbf{R}}_k = \mathbf{G}_1^T \cdots \mathbf{G}_{n-2}^T \mathbf{R}_k(2:(n-1),:) \quad \text{(3-a)}$$

$$\overline{\mathbf{Q}}_k = \mathbf{Q}_k \mathbf{G}_{n-2} \cdots \mathbf{G}_1(2:(n-1),2:(n-1)) \quad \text{(3-b)}$$

$$\overline{\mathbf{B}}_k = \mathbf{G}_1^T \cdots \mathbf{G}_{n-2}^T \mathbf{B}_k(2:(n-1),:) \quad \text{(3-c)}$$

Phase II : Inserting a row

Step 4: Compute Givens rotations $\mathbf{J}_1, \mathbf{J}_2, \cdots, \mathbf{J}_p$ such that

$$\mathbf{J}_p^T \mathbf{J}_{p-1}^T \cdots \mathbf{J}_1^T \begin{bmatrix} \boldsymbol{\alpha}_{k,1}^T \\ \overline{\mathbf{R}}_k \end{bmatrix} = \mathbf{R}_{k+1}, \qquad \mathbf{R}_{k+1} \in \mathbf{R}^{(n-1)\times p} \quad (4)$$

is upper triangular.

Step 5: Compute $\mathbf{Q}_{k+1}$,

$$\mathbf{Q}_{k+1} = \mathbf{P}^T diag(1, \overline{\mathbf{Q}}_k) \mathbf{J}_1^T \mathbf{J}_2^T \cdots \mathbf{J}_p^T. \quad (5)$$

Step 6: Compute $\mathbf{R}_{k+1,12}$, $\mathbf{R}_{k+1,22}$, $\mathbf{y}_{k+1,1}$, $\mathbf{y}_{k+1,2}$

$$\begin{bmatrix} \mathbf{R}_{k+1,12} & \mathbf{y}_{k+1,1} \\ \mathbf{R}_{k+1,22} & \mathbf{y}_{k+1,2} \end{bmatrix} = \mathbf{J}_p^T \mathbf{J}_{p-1}^T \cdots \mathbf{J}_1^T \begin{bmatrix} \boldsymbol{\alpha}_{k,2}^T ; \beta_k \\ \overline{\mathbf{B}}_k \end{bmatrix}. \quad (6)$$

Phase III : Compute Solution.

Step 7: Construct the matrix $\mathbf{D}_{k+1}$,

$$\mathbf{D}_{k+1} = [\mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2}]^T [\mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2}], \quad (7)$$

and compute the minimum eigenvector $\mathbf{v}$ of the matrix $\mathbf{D}_{k+1}$ using the FALM[3]. Compute, then, the TLS solution $\mathbf{x}_{k+1,2}$

$$\mathbf{x}_{k+1,2} = -\frac{1}{v_{k-p+1}} [v_1, v_2, \cdots, v_{k-p}]^T \quad (8)$$

Step 8: Compute the least squares solution $\mathbf{x}_{k+1,1}$ of the equation

$$\mathbf{R}_{k+1,11} \mathbf{x}_{k+1,1} = \mathbf{y}_{k+1,1} - \mathbf{R}_{k+1,12} \mathbf{x}_{k+1,2}. \quad (9)$$

Then the overall solution is $\mathbf{x}_{k+1} = [\mathbf{x}_{k+1,1}^T ; \mathbf{x}_{k+1,2}^T]^T$. If every satellite is excluded one by one, stop. If not, go to *Step* 2.

3. Algorithm for Two Failure Problem

One of the facts to consider is that Phase I and II of the single failure algorithm is assumed to delete or add the first row of the matrix equation sequentially. However, since two rows to be deleted are apart from each other for two failure case, a permutation technique is required. Therefore a permutation matrix and a permutation index vector are employed. The permutation index vector has a role to keep track the order of the rows and provide to next step with information which rows are deleted or inserted. One more point considered is the permutation matrix affect to the $\mathbf{R}_k$, $\mathbf{Q}_k$ and $\mathbf{B}_k$.

The permutation matrix is

$$\mathbb{P} = \begin{bmatrix} 0_{1\times(i-1)} & 1 & 0_{1\times(n-i)} \\ \mathbb{I}_{(i-1)\times(i-1)} & & 0_{(i-1)\times(n-i+1)} \\ 0_{(n-i)\times i} & & \mathbb{I}_{(n-i)\times(n-i)} \end{bmatrix} \quad (10)$$

where n is the number of visible satellites and i is the row to be placed as first row.

Now, we consider how the permutation matrix affect to the $\mathbf{R}_k$, $\mathbf{Q}_k$ and $\mathbb{B}_k$. The subscript k denotes that the k-th subset is considered. Therefore, it will be omitted in the following equation for convenience. Suppose the permutation matrix $A = QR$ is multiplied by P. Then, $PA = (PQ)R$. (11)

Only Q is changed, multiplied by P, when a row of a matrix is permuted.

To see the effect on $\mathbb{B}_k$, let $A = [A_1 \quad A_2]$, then the matrix equation multiplied by P becomes

$$[PA_1 \quad PA_2]x = Pb. \quad (12)$$

If we let $A_1 = QR$, it becomes

$$[PQR \quad PA_2]x = Pb \quad (13)$$

If we let B' be the $\mathbf{B}_k$ of the above equation, then by definition of $\mathbf{B}_k$

$$B' = (PQ)^T [PA_2 \quad Pb] = Q^T [A_2 \quad b] = \mathbf{B} \qquad (14)$$

because $P^T P = I$. As shown, B is not changed by permuting the rows.

In order to form sub-matrix, every two satellites should be deleted. To delete systematically, we find combination of visible satellites. For example, suppose 8 satellites are visible and their identification numbers are 4, 6, 7, 10, 18, 19, 21, and 22. We can find combination of satellite identification numbers. However it is more convenient to correspond the identification numbers to natural number from 1 to 8 and find combination of 1 to 8, like (1,2), (1,3), ⋯ (1,8), (2,3), (2,4), ⋯ (2,8), (3,4), (3,5), ⋯, (7,8). This can be easily implemented by MATLAB command com bntns([1:svn],2) where svn is number of visible satellite numbers. When we follow the sequence, two cases is met. One is changing only one satellite, the other is changing two satellites. For example, (1,2) is deleted at present step, then (1,3) has to be deleted next step. In this case, only one satellite, 2, comes to be changed by 3. On the other hand, when we proceed from (1,8) to (2,3), two satellites have to be changed. Therefore, we need a routine to check it.

ALGORITHM

Initialize

Step 1: Form the sub-matrix equation ($\overline{\mathbf{A}}\mathbf{x} = \overline{\mathbf{b}}$) by deleting the first two columns and compute $\overline{\mathbf{Q}}_1, \overline{\mathbf{R}}_1, \overline{\mathbf{B}}_1$ and the initial solution at t=1 and

$$\overline{\mathbf{Q}}_1^T \left[\overline{\mathbf{A}}_{1,1}; \overline{\mathbf{A}}_{1,2}; \overline{\mathbf{b}}_1\right] = \left[\overline{\mathbf{R}}_1; \overline{\mathbf{B}}_1\right] \qquad (12)$$

where $\overline{\mathbf{R}}_1 = \overline{\mathbf{Q}}_1^T \overline{\mathbf{A}}_{1,1} \in \Re^{(n-2)\times p}$ is an upper triangular matrix and $\overline{\mathbf{B}}_1 = \overline{\mathbf{Q}}_1^T \left[\overline{\mathbf{A}}_{1,2}; \overline{\mathbf{b}}_1\right]$. Determine the sequence using MATLAB command combntns ([1:svn],2). The rows to be deleted or inserted are automatically determined by the sequence.

Initialize the index vector, like [3 4 5 6 7 8] for 8 visible satellite case.

While every subset is tested

{

Step 2: Check how many rows should be changed, one or two.

Step 3: Determine which row(s) permuted and compute P as in (10)

Step 4: Compute

$$\overline{\mathbf{A}} = \mathbf{P}\overline{\mathbf{A}}, \quad \overline{\mathbf{b}} = \mathbf{P}\overline{\mathbf{b}},$$

$$\mathbf{indvec} = \mathbf{P} * \mathbf{indvec}, \quad \mathbf{Q}_k = \mathbf{P}\mathbf{Q}_k$$

Step 5: Perform Phase I and Phase II of the algorithm for single failure problem.

Step 6: If two rows are changed, then repeat step 3,4 and 5 for secondly changed row. Otherwise go to Step 7.

Step 7: Perform Phase III of the algorithm for single failure problem.

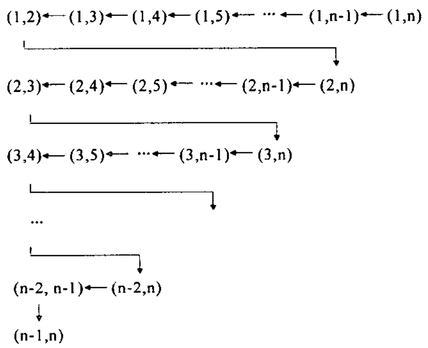
}

4. Consideration of Reducing computational Burden

One of the factors considered is computational burden of the algorithm. Comparing the algorithm for two failure problem with the algorithm for single failure algorithm, we can see computational amount increased when two rows have to be changed (Step 7). If we can avoid the repeat, computational burden of the algorithm for two failure problem has only minor difference compared to that of the algorithm for single failure problem. To avoid the repeat, test order of subsets is important. A close examination leads to the following fact.

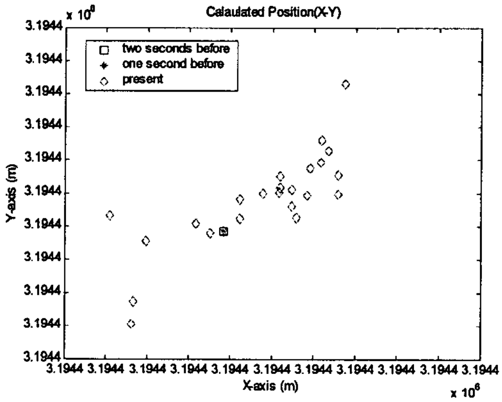
Fact : When performing a subset test for two failure problem, a test sequence changing only one satellite always exists.

Proof of the fact is not necessary because it is obvious from figure 1. In the figure only one satellite is inserted and deleted respectively from one subset to the other subset. Therefore the Step 7 can be omitted if we use this sequence.



5. Simulation Results

The satellites data were generated using MATLAB toolbox. No errors were considered in generating the satellite data to show how the algorithm works clearly. For this arbitrary simulation, midnight at the beginning of the GPS week has been chosen. The specified user location has been chosen at 0 degree latitude, 0 degrees longitude and 0 metes above geoid. A simulated pseudo-range error was injected to a satellite at time t (t=4 in this simulation. In this simulation, we assume that there is no failure until time t-1 and a satellite (PRN #10 & #19 in this case) fails between t-1 and t. Then, the satellite pseudorange measurements at time t have blunders. We examine how the algorithm is working in this case. The following figures describe the results. Figure 1 shows the calculated positions. The star and square denote previous positions at time t-1 and t-2 respectively. Since 8 satellites are in visible, it needs to perform 28 subset test. The previous positions were exactly overlapped because no errors were assumed. On the contrary, only one subset position which the failed satellites, #10 & #19, were excluded coincided with the previous position, which shows the algorithm worked well.



[그림 1] Calculated position

6. Conclusions

In this paper, a new TLS-based sequential algorithm to identify an errant satellite is proposed. A major contribution of this paper might be the fact the algorithm is new and it makes us enjoy the advantages of TLS with less computational burden since it takes sequential form.

[References]

[1] Jyh-Ching Juang, “On GPS Position and Integrity Monitoring”, IEEE Transactions on Aerospace and Electronics Systems, Vol. 36, No. 1, Jan. 2000, pp. 327-336.

[2] Chang Wan Jeon, “Development of A New RAIM Algorithm for GPS Applications”, Proceedings of 13th KMMS, May 2004.

[3] Chang Wan Jeon and Jang Gyu Lee, "A New MRQI Eigenpairs", *IEICE Transactions on Information and Systems*, Vol.E82-D, No.6, June 1999, pp. 1011-1019.