A New TLS-Based Sequential Algorithm to Identify Two Failed Satellites

Chang-wan Jeon*, and Gérard Lachapelle**

* Division of Information Technology Engineering, Soonchunhyang University, Asan, Korea
(Tel : +82-41-530-1372; E-mail: jeoncw@sch.ac.kr)
**Department of Geomatics Engineering, Calgary University, Calgary, Canada
(Tel : +1-403-220-7104; E-mail: lachapel@geomatics.ucalgary.ca)

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. In this paper, the special form of the observation matrix $\mathbf{H}$, that is, one column is exactly known, is exploited so as to develop an algorithm in a sequential form, which reduces computational burden. The algorithm makes us enjoy the advantages of TLS without much computational burden. The proposed algorithm is verified through a numerical simulation.

Keywords: TLS, Sequential Algorithm, RAIM, Two Failed Satellites

1. INTRODUCTION

Since middle 1980s, receiver autonomous integrity monitoring (RAIM) has been widely focused on. This is because the integrity information provided via navigation message may not be timely enough in some applications. In these days, extensive researches on this topic have been performed under the name of RAIM, FDI (failure detection and isolation), or FDE (failure detection and exclusion) [1-2].

With the development of RAIM techniques for single failure, there has been increasing interest in the multiple failure problem. There have been several approaches to tackle the problem from various points of view. One of major streams on solving this problem is to form ${}_nC_2$ subsets of n-2 satellites by sequentially deleting two satellites not previously excluded combination and have calculated test statistic for each satellite subset using residual. The general scheme of this kind of approach is like this: 1) Perform test with whole measurements set if there exist a failure. 2) If there is a failure, form ${}_nC_2$ subsets of n-2 satellites by sequentially deleting two satellites and perform subset test to identify which satellites fail.

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[8] reformulated a linear measurement model and proposed a positioning and integrity monitoring scheme based on total least squares (TLS) instead of least squares. Jeon et al. proposed a sequential TLS-based RAIM algorithm for single failure problem.

In this paper, the result of Jeon et al. is extended for two failure problem. The special form of the observation matrix $\mathbf{H}$, that is, one column is exactly known, is exploited so as to develop an algorithm in a sequential form, which reduces computational burden. It makes use of previous results without repeating the whole process. Therefore one can enjoy, with less computational burden, the advantages for integrity monitoring provided by Juang who employed TLS as a tool for positioning and integrity monitoring.

2. TECHNICAL BACKGROUND

This section describes the linear measurement model and mixed LS-TLS problem for the discussion in the remainder of the paper.

2.1 Linear Measurement Model

A linear model is generally employed for proper positioning and integrity monitoring. In [8], linear measurement model was reinvestigated considering errors in observation matrix H. In this model, error due to a failed satellite is included in observation matrix H. Therefore the observation matrix H is not exactly known any more. Naturally, TLS is employed to solve this problem. In this paper, the linear model in [8] will be used. Therefore a brief description of the model is given.

Suppose n satellites are visible. The measurement model is

$$\rho^i = \left\| \mathbf{u} - \mathbf{s}^i - \Delta\mathbf{s}^i \right\| + c + e^i \tag{1}$$

where ρ^i is pseudo-range measurement with respect to the i-th GPS satellite, $\mathbf{u}$ is the user's position, c is clock offset, $\mathbf{s}^i$ is the broadcast position of the i-th GPS satellite, $\Delta\mathbf{s}^i$ is the difference between the broadcast position and true position of the i-th GPS satellite, and e^i accounts for the other errors. e^i is treated as zero mean noise. Both the pseudo-range ρ^i and the broadcast position $\mathbf{s}^i$ are subject to errors due to ephemeris errors, SA effects if it exists, environment effects, satellite failure, interferences, noises, and so on. Let

$$\mathbf{u} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{s}^i = \begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix}, \text{ and } \quad \Delta\mathbf{s}^i = \begin{bmatrix} \Delta x^i \\ \Delta y^i \\ \Delta z^i \end{bmatrix}. \tag{2}$$

Suppose that the linearization point is at

$$\mathbf{u} = \mathbf{u}_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \text{ and } \quad c = c_0 \tag{3}$$

then the estimated of the psudo-range measurement is given by

$$\rho_0^i = \left\| \mathbf{u}_0 - \mathbf{s}^i - \Delta\mathbf{s}^i \right\| + c_0$$

$$= \sqrt{\left(x^i + \Delta x^i - x_0\right)^2 + \left(y^i + \Delta y^i - y_0\right)^2 + \left(z^i + \Delta z^i - z_0\right)^2} + c_0 \tag{4}$$

Define

$$\mathbf{r}=\begin{bmatrix} x-x_0 \\ y-y_0 \\ z-z_0 \end{bmatrix},\ \delta=-c+c_0\text{, and }\ \mathbf{p}=\begin{bmatrix}\mathbf{r}\\ \delta\end{bmatrix}. \tag{5}$$

Then, the linearized matrix equation of (1) with respect to n observable satellites becomes

$$\mathbf{Hp}=\mathbf{q}+\mathbf{e} \tag{6}$$

where

$$\mathbf{H}=\begin{bmatrix} h_{11} & h_{12} & h_{13} & 1 \\ h_{21} & h_{22} & h_{23} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ h_{n1} & h_{n2} & h_{n3} & 1 \end{bmatrix},\ \mathbf{q}=\begin{bmatrix} \rho_0^1-\rho^1 \\ \rho_0^2-\rho^2 \\ \vdots \\ \rho_0^n-\rho^n \end{bmatrix},\ \mathbf{e}=\begin{bmatrix} e^1 \\ e^2 \\ \vdots \\ e^n \end{bmatrix},$$

and

$$h_{i1}=\frac{x^i+\Delta x^i-x_0}{\sqrt{\left(x^i+\Delta x^i-x_0\right)^2+\left(y^i+\Delta y^i-y_0\right)^2+\left(z^i+\Delta z^i-z_0\right)^2}}$$

$$h_{i2}=\frac{y^i+\Delta y^i-y_0}{\sqrt{\left(x^i+\Delta x^i-x_0\right)^2+\left(y^i+\Delta y^i-y_0\right)^2+\left(z^i+\Delta z^i-z_0\right)^2}}$$

$$h_{i3}=\frac{z^i+\Delta z^i-z_0}{\sqrt{\left(x^i+\Delta x^i-x_0\right)^2+\left(y^i+\Delta y^i-y_0\right)^2+\left(z^i+\Delta z^i-z_0\right)^2}}.$$

Note that the last column of H matrix is exactly known. Therefore solving the linearized matrix equation is a mixed LS-TLS problem described in the next section.

2.2 Mixed LS-TLS Problem

Let $\mathbf{b}=\mathbf{q}+\mathbf{e}$ for simplicity. Then (6) becomes a well-known linear matrix equation,

$$\mathbf{Hp}=\mathbf{b} \tag{7}$$

In the classical LS approach, all elements of $\mathbf{H}$ are assumed to be free of error; hence, all errors are confined to the observation vector $\mathbf{b}$. This assumption, however, is frequently unrealistic in some applications. The TLS is one method of fitting that is appropriate when there are errors in both the observation vector $\mathbf{b}$ and the data matrix $\mathbf{H}$. Especially when some of the columns, not all, of the data matrix $\mathbf{H}$ are free of error like the case considered in this paper, we call it a mixed LS-TLS problem [9]. In this case, we can solve the mixed problem by solving the LS and TLS problem separately with a proper batch algorithm in [9].

We can permute the order of columns in $\mathbf{H}$ with a proper permutation matrix and obtain the following equation without loss of the generality,

$$\mathbf{Ax}=\mathbf{b} \tag{8}$$

where

$$\mathbf{A}=\begin{bmatrix} 1 & h_{11} & h_{12} & h_{13} \\ 1 & h_{21} & h_{22} & h_{23} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & h_{n1} & h_{n2} & h_{n3} \end{bmatrix}\ \text{and}\ \mathbf{b}=\begin{bmatrix}\delta\\ \mathbf{r}\end{bmatrix} \tag{9}$$

This trick is for convenience only. Let a matrix $\mathbf{A}=[\mathbf{A}_1;\mathbf{A}_2]$ be given whose the first p columns $\mathbf{A}_1$ have no error and have full column rank. Suppose the matrix have p exactly known columns to generalize the discussion, although it has only one exactly known column in this case. Then, the algorithm is as follows. Perform p Householder transformations $\mathbf{Q}$ on the matrix $[\mathbf{A};\mathbf{b}]$ so that

$$\mathbf{Q}^T[\mathbf{A}_1;\mathbf{A}_2;\mathbf{b}]=\begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{y}_1 \\ 0 & \mathbf{R}_{22} & \mathbf{y}_2 \end{bmatrix} \tag{10}$$

where $\mathbf{R}_{11}$ is a $p\times p$ upper triangular matrix. (scalar in this case). Then, compute the TLS solution $\hat{\mathbf{x}}_2$ of $\mathbf{R}_{22}\mathbf{x}_2=\mathbf{y}_2$ by the SVD. $\hat{\mathbf{x}}_2$ yields the last *n-p* components of the solution vector $\hat{\mathbf{x}}$. To find the first p components $\hat{\mathbf{x}}_1$ of the solution vector, solve

$$\mathbf{R}_{11}\mathbf{x}_1=\mathbf{y}_1-\mathbf{R}_{12}\mathbf{x}_2 \tag{11}$$

This is simply the LS solution obtained by projecting the reduced observation vector $\mathbf{b}-\mathbf{A}_2\hat{\mathbf{x}}_2$ into the space $R(\mathbf{A}_1)$ generated by the known columns of $\mathbf{A}$.

3. DERIVATION OF A SEQUENTIAL ALGORITHM FOR TWO FAILED PROBLEM

3.1 The Algorithm For Single Failure Problem

It is needed a brief description on the algorithm for single failure problem because the algorithm is extended to algorithms for two failure problem. It is assumed that one satellite (that is, one row of (8)), is deleted one after another from the first. The algorithm is composed of four parts, Initialize, Phase I, Phase II, and Phase III. Initialize part initializes initial values needed for sequential processing. Phase I and Phase II are core parts of the algorithm performing sequential processing. In order to form new subset of satellite the algorithm takes two steps : deleting a satellite (row) supposed to be deleted at present step and inserting the previously deleted satellite (the previously deleted row) in the previous stage. For example, suppose the first row of (8) is deleted at present stage and the second row of (8) will be deleted at the next stage. At the next stage, the second row is deleted firstly from the sub-matrix of the previous stage (Phase I) and then the first row deleted at the previous stage is inserted (Phase II). This forms the new subset of satellite. This sequential processing relieves computational burden greatly since it makes use of previous results without repeating the whole process. The Phase III solves the linear matrix equation to find final solution of the subset.

ALGORITHM

Initialize

Step 1: Form the sub-matrix equation ($\overline{\mathbf{A}}\mathbf{x}=\overline{\mathbf{b}}$) by deleting the first column in (9) and compute $\overline{\mathbf{Q}}_1,\overline{\mathbf{R}}_1,\overline{\mathbf{B}}_1$ and the initial solution at t=1 from (10), (11), and

$$\overline{\mathbf{Q}}_1^T[\overline{\mathbf{A}}_{1,1};\overline{\mathbf{A}}_{1,2};\overline{\mathbf{b}}_1]=[\overline{\mathbf{R}}_1;\overline{\mathbf{B}}_1] \tag{12}$$

where $\overline{\mathbf{R}}_1=\overline{\mathbf{Q}}_1^T\overline{\mathbf{A}}_{1,1}\in\Re^{(n-1)\times p}$ is an upper triangular matrix and $\overline{\mathbf{B}}_1=\overline{\mathbf{Q}}_1^T[\overline{\mathbf{A}}_{1,2};\overline{\mathbf{b}}_1]$. 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 of (8) 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

$$\mathbf{G}_1^T\mathbf{G}_2^T\cdots\mathbf{G}_{n-2}^T\mathbf{q}_1=\rho\mathbf{e}_1 \tag{13}$$

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

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

$$\overline{\mathbf{R}}_k = \mathbf{G}_1^T \cdots \mathbf{G}_{n-2}^T \mathbf{R}_k (2:(n-1),:) \quad (14\text{-a})$$

$$\overline{\mathbf{Q}}_k = \mathbf{Q}_k \mathbf{G}_{n-2} \cdots \mathbf{G}_1 (2:(n-1),2:(n-1)) \quad (14\text{-b})$$

$$\overline{\mathbf{B}}_k = \mathbf{G}_1^T \cdots \mathbf{G}_{n-2}^T \mathbf{B}_k (2:(n-1),:) \quad (14\text{-c})$$

Phase II : Inserting a row

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

$$\mathbf{J}_p^T \mathbf{J}_{p-1}^T \cdots \mathbf{J}_1^T \begin{bmatrix} \boldsymbol{\alpha}_{k,1}^T \\ \overline{\mathbf{R}}_k \end{bmatrix} = \mathbf{R}_{k+1}, \qquad \mathbf{R}_{k+1} \in \mathbf{R}^{(n-1)\times p} \quad (15)$$

is upper triangular.

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

$$\mathbf{Q}_{k+1} = \mathbf{P}^T diag(1, \overline{\mathbf{Q}}_k) \mathbf{J}_1^T \mathbf{J}_2^T \cdots \mathbf{J}_p^T . \quad (16)$$

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

$$\begin{bmatrix} \mathbf{R}_{k+1,12} & \mathbf{y}_{k+1,1} \\ \mathbf{R}_{k+1,22} & \mathbf{y}_{k+1,2} \end{bmatrix} = \mathbf{J}_p^T \mathbf{J}_{p-1}^T \cdots \mathbf{J}_1^T \begin{bmatrix} \boldsymbol{\alpha}_{k,2}^T ; \beta_k \\ \overline{\mathbf{B}}_k \end{bmatrix} \quad (17)$$

where $\mathbf{R}_{k+1,12} \in \mathbf{R}^{p\times(n-p)}$, $\mathbf{R}_{k+1,22} \in \mathbf{R}^{(n-p-1)\times(n-p)}$, $\mathbf{y}_{k+1,1} \in R^p$ and $\mathbf{y}_{k+1,2} \in R^{n-p-1}$.

Phase III : Compute Solution.

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

$$\mathbf{D}_{k+1} = [\mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2}]^T [\mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2}] \quad (18)$$

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

$$\mathbf{x}_{k+1,2} = -\frac{1}{v_{k-p+1}} [v_1, v_2, \cdots, v_{k-p}]^T \quad (19)$$

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

$$\mathbf{R}_{k+1,11} \mathbf{x}_{k+1,1} = \mathbf{y}_{k+1,1} - \mathbf{R}_{k+1,12} \mathbf{x}_{k+1,2} . \quad (20)$$

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.

In the following subsections, a couple of TLS-based sequential algorithms for two failure problem will be derived. Basically, it is needed to repeat the Phase I and II of the algorithm for single failure problem, since two satellites are changed for one subset test. However, there are important facts to consider..

3.2 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

$$\mathbf{P} = \begin{bmatrix} 0_{1\times(i-1)} & 1 & 0_{1\times(n-i)} \\ \mathbf{I}_{(i-1)\times(i-1)} & & 0_{(i-1)\times(n-i+1)} \\ 0_{(n-i)\times i} & & \mathbf{I}_{(n-i)\times(n-i)} \end{bmatrix} \quad (21)$$

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 $\mathbf{B}_k$. 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,

$$PA = (PQ)R . \quad (22)$$

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

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

$$[PA_1 \quad PA_2] x = Pb . \quad (23)$$

If we let $A_1 = QR$, it becomes

$$[PQR \quad PA_2] x = Pb \quad (24)$$

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

$$\begin{aligned} B' &= (PQ)^T [PA_2 \quad Pb] \\ &= Q^T [A_2 \quad b] \\ &= \mathbf{B} \end{aligned} \quad (25)$$

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 combntns([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.

Based on the above discussion, we summarize the algorithm below. The algorithm is expressed via MATLAB grammar, since it is simple and well known.

ALGORITHM

Initialize

Step 1: Form the sub-matrix equation ($\overline{\mathbf{A}}\mathbf{x} = \overline{\mathbf{b}}$) by deleting the first two columns in (9) and compute $\overline{\mathbf{Q}}_1, \overline{\mathbf{R}}_1, \overline{\mathbf{B}}_1$ and the initial solution at *t*=1 from (10), (11), and

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

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 (21)

Step 4: Compute

$\overline{\mathbf{A}} = \mathbf{P}\overline{\mathbf{A}}$

$\overline{\mathbf{b}} = \mathbf{P}\overline{\mathbf{b}}$

$\mathbf{indvec} = \mathbf{P} * \mathbf{indvec}$

$\mathbf{Q}_k = \mathbf{P}\mathbf{Q}_k$

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.

}

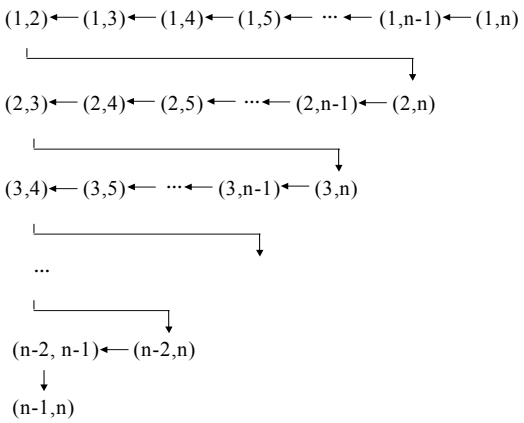
3.3 Consideration of Reducing computational Burden

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. Figure 1 shows the subset test sequence for two failure problem when n satellites are visible. 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.

Since to compute a Givens rotation matrix requires 4 flops and one square root and to multiply the Givens rotation matrix to a vector requires 4(m-1) flops[10], the algorithm requires about 44m+81 multiplications and 2m square roots for each subset test. The m is n-2, where n is the number of visible satellites. On the other hand, the SVD requires 50m + 256 for computation of singular values and right singular vectors and requires $10m^2 + 100m + \frac{14}{3}n^3$ flops for computation of singular values and right and left singular vectors. As shown, the algorithm has more advantage as number of subset to be tested increase, as it should be because it has sequential form.



4. SIMULATION RESULTS

In this section, some simulation results are discussed. The simulation is focused on how the proposed algorithm working well under two satellites failed circumstance, because Table I already shows how fast the algorithm is. The satellites data were generated using MATLAB toolbox[13]. 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). What is the time t is not important because the proposed algorithm runs between adjacent epochs (1 epoch = 1sec. 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.

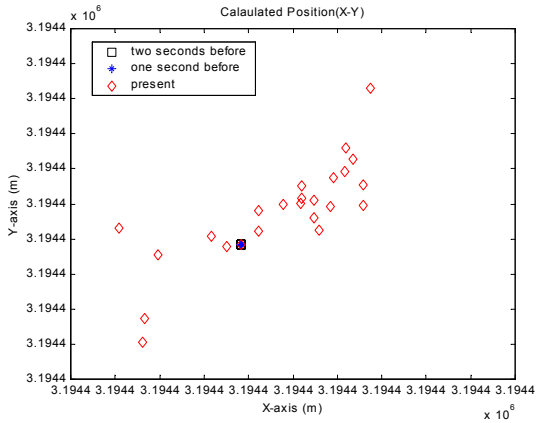


Fig 1. Calculated position

With a proper measure and threshold even though it is not the focus of this paper, the proposed algorithm can provide good performance for failure detection and isolation based on TLS technique.

5. 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. With a proper measure and threshold that is extensively studied until now, it can provide performance for failure detection and identification.

REFERENCES

[1] R. G. Brown, Receiver Autonomous Integrity Monitoring, In Bradford W. Parkinson and James J. Spilker, (Eds), *Global Position System: Theory and Applications*, Vol. II, New York:AIAA, 1996.

[2] Robert J. Kelly, "The Linear Model, RNP, and the Near-Optimum Fault Detection and Exclusion Algorithm", Global Positioning System, Vol. V, ION, 1998, pp.227-259.

[3] Bradford W. Parkinson and James J. Spilker, (Eds), *Global Position System: Theory and Applications*, Vol. I, New York:AIAA, 1996.

[4] David Wells, et al, Guide to GPS Positioning, Canadian GPS Associates, 1987.

[5] Lee, Y.C., "Analysis of Range and Position Comparison Methods as a Means to Provide GPS Integrity in the User Receiver," Proceedings of the Annual Meeting of the Institute of Navigation (Seattle, WA), June 24-26, 1986, pp. 1-4

[6] Parkinson, B.W., and Axelard, P., "Autonomous GPS Integrity Monitoring Using the Pseudorange Residual," Navigation (Washington), Vol. 35, No. 2, 1988, pp. 255-274.

[7] Pervan, B. S., et al, Parity Space Methods for Autonomous Fault Detection and Exclusion Algorithms Using Carrier Phase, Proceedings of PLANS'96.

[8] 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.

[9] Van Huffel, S. and Vandewalle, J., *The Total Least Squares Problem Computational Aspects and Analysis*, SIAM, 1991.

[10] Golub, G. H. and Van Loan, C. F., *Matrix Computations*, John Hopkins University Press, 1983.

[11] Chang Wan Jeon, Hyoung Joong Kim, and Jang Gyu Lee, "A Fast and Accurate Algorithm for Computing Desired Eigenpairs of Hermitian Matrices", *IEICE Transactions on Information and Systems*, Vol.E79-D, No.3, March 1996, pp. 182-188.

[12] 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.

[13] GPSoft, Satellite Navigation TOOLBOX 2.0 User's Guide, GPSoft LLC, Oct. 1999.