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

Chang-Wan Jeon and Gérard Lachapelle

Abstract: With the development of RAIM techniques for single failure, increasing interest has been shown in the multiple failure problem. As a result, numerous approaches have been used in attempts to tackle this problem. This paper considers the two failure problem with total least squares (TLS) technique, a solution that has rarely been addressed because TLS requires an immense 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, thereby reducing computational load. The algorithm permits the advantages of TLS without the excessive computational burden. The proposed algorithm is verified through a numerical simulation.

Keywords: GPS, RAIM, sequential algorithm, total least squares.

1. INTRODUCTION

Since the mid 1980s, receiver autonomous integrity monitoring (RAIM) has been given a great deal of attention. This is because the integrity information provided via navigation messaging may not be timely enough in some applications. Up-to-date extensive research on this topic has 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 [2-4], there has been increasing interest in the multiple failure problem and consequently, several approaches have been developed to tackle the problem from various points of view. One of the major methods used to solve this problem is to form ${}_nC_2$ subsets of n-2 satellites by sequentially deleting two satellites which are not previously excluded combination and have calculated test statistic for each satellite subset using residual [5-6]. The general scheme of this kind of approach is as follows: 1) If there is a failure, perform test with complete set of measurements. 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 the two failure problem with total least squares (TLS) technique, with particular emphasis on the second step.

In comparison with the least squares method, the TLS method enjoys the properties of data consistency at the expense of increased numerical computation [7-8]. RAIM technique based on the TLS has rarely been addressed because TLS requires a significant number of computations. Recently, Juang [9] reformulated a linear measurement model and proposed a positioning and integrity monitoring scheme based on total least squares (TLS) instead of least squares.

He proposed a new test metric using TLS residual and proved the statistical distribution of the test metric is summation of an n-4 degrees of freedom chi-square distribution and four Gamma distributions with different parameters [9]. He only focused on failure detection using TLS corresponding to step 1 of the general scheme. If we extend his work to the isolation problem (step 2 of the general scheme), the increasing computational burden acts as a bottleneck. Therefore, an efficient algorithm to reduce the computational burden is required. This paper tackles the problem. Jeon et al. [10] suggested a sequential TLS-based RAIM algorithm for single failure problems.

In this paper, the result of Jeon et al. is extended for the two failure problem. The particular 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 entire 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.

Manuscript received November 2, 2004; revised April 4, 2005; accepted April 19, 2005. Recommended by Editorial Board member Jae Weon Choi under the direction of Editor Keum-Shik Hong.

Chang-Wan Jeon is with the Division of Information Technology Engineering, Soonchunhyang University, San 53-1 Eupnae-ri, Shinchang-myun, Asan-si, Choongnam 336-745, Korea (e-mail: jeoncw@sch.ac.kr).

Gérard Lachapelle is with the Department of Geomatics Engineering, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4 (e-mail: lachapel@geomatics.ucalgary.ca).

2. TECHNICAL BACKGROUND

This section describes the linear measurement model and mixed LS-TLS problem for 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 [9], a 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 no longer precisely known. Naturally, TLS is employed to solve this problem. In this paper, the linear model in [9] was used, however it was slightly modified to emphasize that the linear model has a mixed LS-TLS structure. 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 the pseudo-range measurement with respect to the i-th GPS satellite, $\mathbf{u}$ is the user's position, c is the 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, environmental effects, satellite failure, interferences, noises, and so on. Let

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

Suppose that the linearization point is at

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

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

$$\rho_0^i = \left\| \mathbf{u}_0 - \mathbf{s}^i - \Delta\mathbf{s}^i \right\| + c_0 \tag{4}$$

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

Define

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

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

$$\mathbf{H}\mathbf{p} = \mathbf{q} + \mathbf{e}, \tag{6}$$

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

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

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

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

The **H** matrix of equation (6) can be regarded as

$$\mathbf{H} = \left[\bar{\mathbf{H}} + \Delta\mathbf{H} \quad \mathbf{1}_{1\times 4} \right], \tag{7}$$

where

$$\bar{\mathbf{H}} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ \vdots & \vdots & \vdots \\ h_{n1} & h_{n2} & h_{n3} \end{bmatrix}, \quad \mathbf{1}_{1\times 4} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

and $\Delta\mathbf{H}$ is a perturbation matrix, which is a function of $\mathbf{s}^i$. The linear model was modified from Juang's to have a precisely recognized column.

Note that the last column of **H** matrix is exactly identified. 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,

$$\mathbf{H}\mathbf{p} = \mathbf{b}. \tag{8}$$

In the classical LS approach, all elements of **H** are assumed to be free of error; hence, all errors are confined to the observation vector **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 **b** and the data matrix **H**. Especially when only some of the columns of the data matrix **H** are free of error like the case considered in this paper, we refer to it as a mixed LS-TLS problem [8]. In this situation we can solve the mixed problem by solving the LS and TLS problem separately with a proper batch algorithm in [8]. We can permute the order of columns in **H** with a proper permutation matrix and obtain the following equation without loss of the generality,

$$\mathbf{Ax} = \mathbf{b}, \tag{9}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & h_{11} & h_{12} & h_{13} \\ 1 & h_{21} & h_{22} & h_{23} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & h_{n1} & h_{n2} & h_{n3} \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} \delta \\ \mathbf{r} \end{bmatrix}. \tag{10}$$

This trick is for convenience only. Let a matrix $\mathbf{A} = [\mathbf{A}_1; \mathbf{A}_2]$ be given whose first p columns $\mathbf{A}_1$ have no error and have full column rank. Suppose the matrix has precisely recognized p 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

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

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 locate the first p components $\hat{\mathbf{x}}_1$ of the solution vector, solve

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

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 **A**.

3. DERIVATION OF A SEQUENTIAL ALGORITHM FOR THE TWO FAILED PROBLEM

3.1. The Algorithm of a Sequential Algorithm for the Two Failed Problem

A brief description is needed concerning the algorithm for the single failure problem because the algorithm is extended to algorithms for the two failure problem. It is assumed that one satellite (that is, one row of (9)), is deleted one after another from the first satellite. The algorithm is composed of four parts, Initialize, Phase I, Phase II, and Phase III. The Initialize part initializes the primary values needed for sequential processing. Phase I and Phase II are core parts of the algorithm performing sequential processing. In order to form a new satellite subset the algorithm takes two steps: deletes a satellite (row) intended for deletion in the present step and inserts the previously deleted satellite (the previously deleted row) in the previous stage. For example, suppose the first row of (9) is deleted at the present stage and the second row of (9) 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 satellite subset. This sequential processing relieves computational burden greatly since it makes use of previous results without repeating the entire process. Phase III solves the linear matrix equation to determine the final solution of the subset.

Algorithm

Initialize

Step 1: Form the sub-matrix equation ($\bar{\mathbf{A}}\mathbf{x} = \bar{\mathbf{b}}$) by deleting the first column in (9) and compute $\bar{\mathbf{Q}}_1, \bar{\mathbf{R}}_1, \bar{\mathbf{B}}_1$ and the initial solution at t=1 from (11), (12), and

$$\bar{\mathbf{Q}}_1^T[\bar{\mathbf{A}}_{1,1}; \bar{\mathbf{A}}_{1,2}; \bar{\mathbf{b}}_1] = [\bar{\mathbf{R}}_1; \bar{\mathbf{B}}_1], \tag{13}$$

where $\bar{\mathbf{R}}_1 = \bar{\mathbf{Q}}_1^T \bar{\mathbf{A}}_{1,1} \in \Re^{(n-1)\times p}$ is an upper triangular matrix and $\bar{\mathbf{B}}_1 = \bar{\mathbf{Q}}_1^T[\bar{\mathbf{A}}_{1,2}; \bar{\mathbf{b}}_1]$. 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 (9) 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 $\mathbf{G}_1^T\mathbf{G}_2^T \cdots \mathbf{G}_{n-2}^T\mathbf{q}_1 = \mu\mathbf{e}_1$ where $\mathbf{q}_1^T$ be the first row of $\mathbf{Q}_k$ and $\mu = \pm 1$.

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

$$\bar{\mathbf{R}}_k = \mathbf{G}_1^T \cdots \mathbf{G}_{n-2}^T \mathbf{R}_k(2:(n-1),:)$$

$$\bar{\mathbf{Q}}_k = \mathbf{Q}_k \mathbf{G}_{n-2} \cdots \mathbf{G}_1(2:(n-1), 2:(n-1))$$

$$\bar{\mathbf{B}}_k = \mathbf{G}_1^T \cdots \mathbf{G}_{n-2}^T \mathbf{B}_k(2:(n-1),:)$$

Phase II: Inserting a row

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

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

is upper triangular.

Step 5: Compute $\mathbf{Q}_{k+1}$, $\mathbf{Q}_{k+1} = \mathbf{P}^T diag\left(1, \bar{\mathbf{Q}}_k\right) \mathbf{J}_1^T \mathbf{J}_2^T \cdots \mathbf{J}_p^T$.

Step 6: Compute $\mathbf{R}_{k+1,12}$, $\mathbf{R}_{k+1,22}$, $\mathbf{y}_{k+1,1}$, $\mathbf{y}_{k+1,2}$ $\begin{bmatrix} \mathbf{R}_{k+1,12} & \mathbf{y}_{k+1,1} \\ \mathbf{R}_{k+1,22} & \mathbf{y}_{k+1,2} \end{bmatrix} = \mathbf{J}_p^T \mathbf{J}_{p-1}^T \cdots \mathbf{J}_1^T \begin{bmatrix} \boldsymbol{\alpha}_{k,2}^T; \beta_k \\ \bar{\mathbf{B}}_k \end{bmatrix}$ 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 $\mathbf{D}_{k+1}$, $\mathbf{D}_{k+1} = \left[\mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2}\right]^T \left[\mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2}\right]$, and compute the minimum eigenvector $\boldsymbol{v}$ of the matrix $\mathbf{D}_{k+1}$ using the FALM [11-12] (or one can use SVD with the matrix $\mathbf{D}_{k+1} = \left[\mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2}\right]^T \left[\mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2}\right]$ to compute the TLS solution). Compute, then, the TLS solution $\mathbf{x}_{k+1,2}$ $\mathbf{x}_{k+1,2} = -\dfrac{1}{v_{k-p+1}}\left[v_1, v_2, \cdots, v_{k-p}\right]^T$.

Step 8: Compute the least squares solution $\mathbf{x}_{k+1,1}$ of the equation $\mathbf{R}_{k+1,11}\mathbf{x}_{k+1,1} = \mathbf{y}_{k+1,1} - \mathbf{R}_{k+1,12}\mathbf{x}_{k+1,2}$. Then the overall solution is $\mathbf{x}_{k+1} = \left[\mathbf{x}_{k+1,1}^T; \mathbf{x}_{k+1,2}^T\right]^T$. If every satellite is excluded one by one, stop. If not, proceed to Step 2.

In the following subsections, a couple of TLS-based sequential algorithms for the two failure problem will be derived. Basically, Phases I and II of the algorithm for the single failure problem must be repeated, 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 Phases I and II of the single failure algorithm are 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 the two failure case, a permutation technique is required. Therefore a permutation matrix and a permutation index vector are employed. The permutation index vector functions to keep track of the order of rows and supply information concerning which rows are deleted or inserted to the next step. 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

$$\mathbf{P} = \begin{bmatrix} 0_{1\times(i-1)} & 1 & 0_{1\times(n-i)} \\ \mathbf{I}_{(i-1)\times(i-1)} & & 0_{(i-1)\times(n-i+1)} \\ 0_{(n-i)\times i} & & \mathbf{I}_{(n-i)\times(n-i)} \end{bmatrix}, \tag{14}$$

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

Now, we consider how the permutation matrix affects $\mathbf{R}_k$, $\mathbf{Q}_k$ 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,

$$\mathbf{PA} = (\mathbf{PQ})\mathbf{R}. \tag{15}$$

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

To see the effect on $\mathbf{B}_k$, let $\mathbf{A} = \left[\mathbf{A}_1 \quad \mathbf{A}_2\right]$, then the matrix equation multiplied by **P** becomes

$$\left[\mathbf{PA}_1 \quad \mathbf{PA}_2\right]\mathbf{x} = \mathbf{Pb}. \tag{16}$$

If we let $\mathbf{A}_1 = \mathbf{QR}$, it becomes

$$\left[\mathbf{PQR} \quad \mathbf{PA}_2\right]\mathbf{x} = \mathbf{Pb}. \tag{17}$$

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

$$\begin{aligned} \mathbf{B}' &= (\mathbf{PQ})^{\mathbf{T}}\left[\mathbf{PA}_2 \quad \mathbf{Pb}\right] \\ &= \mathbf{Q}^{\mathbf{T}}\left[\mathbf{A}_2 \quad \mathbf{b}\right] \\ &= \mathbf{B}, \end{aligned} \tag{18}$$

because $\mathbf{P}^{\mathbf{T}}\mathbf{P}=\mathbf{I}$. As shown, **B** is not changed by permuting the rows.

In order to form the sub-matrix, every two satellites should be deleted. To delete systematically, we find a 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 determine the combination of satellite identification numbers. However it is more convenient to correspond the identification numbers to natural numbers from 1 to 8 and find combinations of 1 to 8, such as (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 combinations ([1:svn],2) where svn is the number of visible satellite numbers. When we follow the sequence, two cases are met. One case changes only one satellite, while the other changes two satellites. For example, (1,2) is deleted at the present step, then (1,3) must be deleted at the 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 must be changed. Therefore, some type of check routine is required.

Based on the above discussion, we summarize the algorithm shown 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 (11), (12), and

$$\overline{\mathbf{Q}}_1^T \left[\overline{\mathbf{A}}_{1,1}; \overline{\mathbf{A}}_{1,2}; \overline{\mathbf{b}}_1\right] = \left[\overline{\mathbf{R}}_1; \overline{\mathbf{B}}_1\right], \quad (19)$$

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 combinations ([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 cases.

While every subset is tested

{

Step 2: Check whether one or two rows should be changed.

Step 3: Determine which row(s) permuted and compute **P** as in (14)

Step 4: Compute

$$\overline{\mathbf{A}} = \mathbf{P}\overline{\mathbf{A}},$$
$$\overline{\mathbf{b}} = \mathbf{P}\overline{\mathbf{b}},$$
$$\mathbf{indvec} = \mathbf{P} * \mathbf{indvec}, \quad (20)$$
$$\mathbf{Q}_k = \mathbf{P}\mathbf{Q}_k,$$

where **indvec** denotes index vector.

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

Step 6: If two rows are changed, then repeat steps 3, 4 and 5 for the secondly changed row. Otherwise proceed 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 the computational burden of the algorithm. Comparing the algorithm for the two failure problem with the algorithm for the single failure algorithm, we can see computational amount increased when two rows require to be changed (Step 7). If we can avoid the repeat, computational burden of the algorithm for the two failure problem has only minor difference compared to that of the algorithm for the single failure problem. To avoid the repeat, the test order of subsets is important. A close examination leads to the following fact.

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

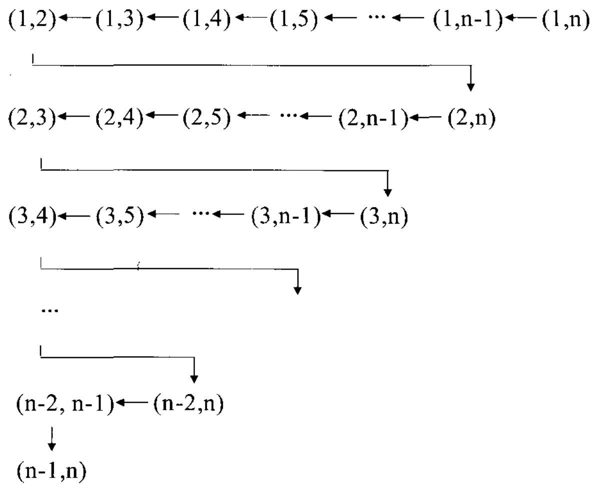


Fig 1. Subset test sequence for two failure problem when n satellites are visible.

Table 1. Comparison of computational amount.

Proposed Algorithm	SVD with singular values and right singular vectors	SVD with singular values and right and left singular vectors
44m+81	50m + 256	$10m^2 + 100m + \frac{14}{3}n^3$

Proof of the fact is not necessary because it is obvious from Fig. 1. Fig. 1 shows the subset test sequence for a 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, 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 a Givens rotation matrix to a vector requires 4(m-1) flops [13], 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 $10m^2 + 100m + 14n^3/3$ flops for computation of singular values and right and left singular vectors. As shown, the algorithm has more advantage as the number of subsets to be tested increases, as it should because it has sequential form. Table 1 presents the comparison of computational amount.

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 1 already shows how fast the algorithm is. The satellites data were generated using MATLAB toolbox [14]. No errors were considered in generating the satellite data to demonstrate 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 degrees 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). The value of 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. If the algorithm is working well, the algorithm must give the same position as the previous position only when the errant satellites are excluded (in this case #10 & #19). The following figures describe the results. Fig. 2 shows the calculated positions. The star and square denote previous positions at time t-1 and t-2 respectively. Since 8 satellites are invisible, 28 subset tests must be performed. The previous positions were exactly overlapped because no errors were assumed. On the contrary, only one subset position - the failed satellites, #10 & #19 were excluded - coincided with the previous position, which shows the algorithm worked well.

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.

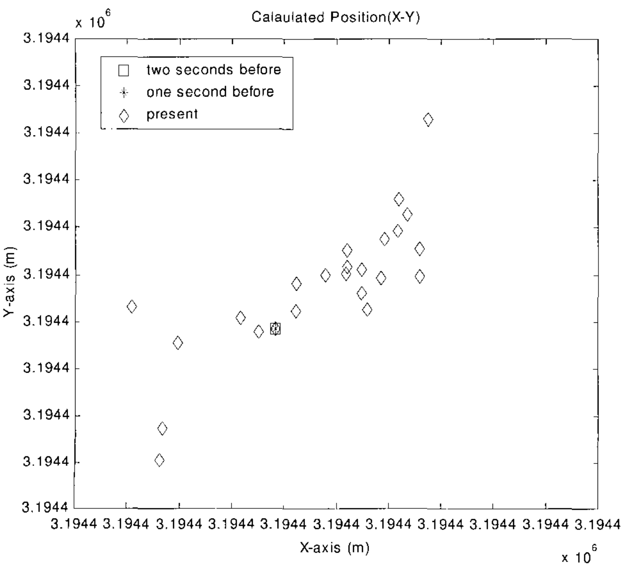


Fig. 2. Calculated position.

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 that the algorithm is new and it allows us to enjoy the advantages of TLS with less computational burden since it takes on a sequential form. With the proper measure and threshold that has been extensively studied to date, it can provide performance for failure detection and identification. The reliability analysis performed in the least squares problem [15-17] can be extended as a future work.

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*, AIAA, New York, 1996.

[2] R. J. Kelly, "The linear model, RNP, and the near-optimum fault detection and exclusion algorithm," *Global Positioning System*, vol. V, ION, pp.227-259, 1998.

[3] Y. C. Lee, "Analysis of range and position comparison methods as a means to provide GPS integrity in the user receiver," *Proc. of the Annual Meeting of the Institute of Navigation,* Seattle, WA, pp. 1-4, June 24-26, 1986.

[4] B. S. Pervan, D. G. Lawrence, C. E. Cohen, and B. W. Parkinson, "Parity space methods for autonomous fault detection and exclusion algorithms using carrier phase," *Prof. of IEEE Position, Location, and Navigation Symposium*, Atlanta, GA, pp. 649-656, April 1996.

[5] R. G. Brown, "Solution of the two-failure GPS RAIM problem under worst-case bias conditions: parity space approach," *Journal of The Institute of Navigation*, vol. 44, no. 4, pp. 425-431, 1998.

[6] G. Lachapelle and S. Ryan, "Statistical reliability measures for GPS," *IMA Workshop on Mathematical Challenges in GPS*, University of Minnesota, Aug. 16-18, 2000.

[7] S. Van Huffle and J. Vandewalle, "On the accuracy of total least squares techniques in the presence of errors on all data," *Automatica*, vol. 25, no. 5, pp. 765-769, 1989.

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

[9] J.-C. Juang, "On GPS position and integrity monitoring," *IEEE Trans. on Aerospace and Electronics Systems*, vol. 36, no. 1, pp. 327-336, January 2000.

[10] C. W. Jeon, G. I. Jee, and G. Lachapelle, "Development of a sequential algorithm for a

GNSS-based multi-sensor vehicle navigation system," *International Journal of Control, Automation, and Systems*, vol. 2, no. 2, pp. 165-170, June 2004.

[11] C. W. Jeon, H. J. Kim, and J. G. Lee, "A fast and accurate algorithm for computing desired eigenpairs of hermitian matrices," *IEICE Trans. on Information and Systems,* vol. E79-D, no. 3, pp. 182-188, March 1996.

[12] C. W. Jeon and J. G. Lee, "A new MRQI eigenpairs," *IEICE Transactions on Information and Systems*, vol. E82-D, no. 6, pp. 1011-1019, June 1999.

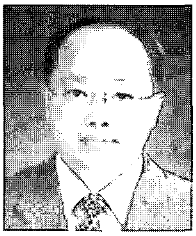
[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, 1983.

[14] GPSoft, *Satellite Navigation TOOLBOX 2.0 User's Guide,* GPSoft LLC, October 1999.

[15] S. Ryan and G. Lachapelle, "Augmentation of DGNSS with dynamic constraints for marine navigation," *Proc. of ION GPS 99*, Nashville, Tennessee, September 14-17, 1999.

[16] S. Ryan, M. Petovelle, and G. Lachapelle, "Augmentation of GPS for ship navigation in constricted water ways," *Proc. of ION National Technical Meeting*, Long Beach CA, January 21-23, 1998.

[17] S. Ryan and G. Lachapelle, "Augmentation of DGNSS with dynamic constraint for marine navigation," *Proc. of ION GPS 99*, Nashville, Tennessee, pp. 1303-1313, September 14-17, 1999.



Chang-Wan Jeon received his B.S., M.S., and Ph.D. degrees in Control and Instrumentation Engineering from Seoul National University, Seoul, Korea, in 1986, 1988 and 1996, respectively. In 1996, he was employed by the Samsung Electronic Company (SEC) to carry out development in the areas of vehicle navigation and integrated information and communication systems for the Incheon International Airport. In 1998, Dr. Jeon joined the faculty of the Division of Information Technology Engineering, Soonchunhyang University, where he has been an Associate Professor since 2004. His research interests include system identification, the Global Positioning System, Kalman filtering, and control system design.

Gérard Lachapelle received his B.Sc. in Surveying from Laval University, his M.Sc. in Geodesy from Oxford, his L.Ph. from the University of Helsinki, and his Dr.techn from the Technical University of Graz. He holds a CRC/iCORE Chair in Wireless Location in the Department of Geomatics Engineering at the University of Calgary where he has been a Professor since 1988. He has been involved in GNSS R&D since 1980 and has contributed to numerous aspects of ground and satellite-based navigation, including cellular telephone location. Professor Lachapelle has been active in numerous associations and was Western Vice President of the (U.S.) Institute of Navigation in 2002-03. He has received countless awards for his work, including the Institute of Navigation Johannes Kepler Award in 1997 and Fellowship of the Royal Society of Canada and the Institute of Navigation.