

# 고장난 위성을 식별하는 TLS 에 기초한 새로운 시이퀀셜 알고리즘

## A New TLS-Based Sequential Algorithm to Identify an Errant Satellite

전 창 완\*  
(Chang-Wan Jeon)

**Abstract :** RAIM techniques based on TLS have rarely been addressed because TLS requires a great number of computations. In this paper, the particular form of the observation matrix  $\mathbf{H}$ , is exploited so as to develop a new TLS-based sequential algorithm to identify an errant satellite. The algorithm allows us to enjoy the advantages of TLS with less computational burden. The proposed algorithm is verified through a numerical simulation.

**Keywords :** GPS, RAIM, sequential algorithm, total least squares (TLS)

### I. 서론

어떤 응용분야에 있어서는 항법메시지에 의한 위성상태에 관한 정보가 제때에 충분히 사용되어질 수 없기 때문에 1980년대 이래로 사용자가 GPS 위성의 상태를 감시하는 RAIM (Receiver Autonomous Integrity Monitoring)이 많은 주목을 받아 왔다. 이러한 분야는 RAIM, FDI (Failure Detection and Isolation), 또는 FDE (Failure Detection and Exclusion) 라는 이름으로 많은 연구가 현재에도 진행되어지고 있다[1,2].

GPS를 이용한 위치측정과 위성의 건강상태를 측정하는 주된 방법 중의 하나는 선형측정모형을 수립하고 최소승법(Least Squares method)을 이용하여 해를 구하는 것이다[3,4]. 특별히 위성의 건강상태를 감시하는 방법에 있어서는 많은 알고리즘들이 순차적으로 하나의 위성을 제거하고 위치를 계산하여 그 결과에 따라 위성의 상태를 판별하는 방법을 취하고 있다[5-7].

최근에 Juang이 선형측정모형을 재구성하여 최소승법 대신 TLS (Total Least Squares)방법에 기초한 RAIM알고리즘을 제안하였다[8]. 그의 방법은 TLS의 장점을 살려 LS 방법에 기초한 것 보다 더 좋은 위성감시를 위한 지표(integrity monitoring metric)를 고안하였으나 TLS 문제를 푸는데 있어 많은 계산량을 요구하는 SVD (Singular Value Decomposition)를 사용하여 계산량이 많아지는 단점을 갖게되었다.

본 논문에서는 선형측정모형이 독특한 형태를 가지고 있다는 사실에 착안하여 TLS에 기초한 새로운 RAIM 알고리즘을 제안한다. 본 논문에서 제안하는 알고리즘은 이전의 결과를 이용하는 순차적인 형태를 갖기 때문에 Juang의 알고리즘보다 계산량이 훨씬 적은, 효율적인 알고리즘이다.

### II. 이론적 배경

본 장에서는 3장의 알고리즘 개발의 기초가 되는 선형측정모형과 혼합 LS-TLS 문제에 대한 간단한 설명을 한다.

\* 책임저자(Corresponding Author)

논문접수 : 2004. 12. 6., 채택확정 : 2005. 5. 4.

전창완 : 순천향대학교 정보기술공학부(jeoncw@sch.ac.kr)

※ 본 논문은 2002학년도 순천향대학교 산업기술연구소 학술연구조성비 일반 연구과제로 지원을 받아 수행하였음.

### 1. 선형측정모형

선형측정모형은 일반적으로 위성을 이용한 측위와 위성상태 감시에 이용되어 왔다. Juang의 논문에서는 관측행렬인  $\mathbf{H}$ 에 오차가 있다는 사실을 감안하여 선형측정 모형을 재구성하였다[8]. 기존의 LS에 기초한 방식에서는 관측행렬  $\mathbf{H}$ 에 오차가 없다고 가정하였으나 Juang의 모델에서는 고장난 위성에 의한 오차가 관측행렬인  $\mathbf{H}$ 에 포함되어있는 것으로 모델링하여 좀 더 실제에 가깝게 모델링하여 오차를 줄이는 방법을 사용하였다. 따라서 양변에 오차가 있는 문제를 풀기에 적당한 TLS방법이 자연스럽게 도입되었다. 본 논문에서는 Juang의 측정모형을 그대로 사용할 것이다. 따라서 아래에 Juang의 측정모형에 대하여 간단히 요약한다.  $n$ 개의 위성이 보인다고 가정하면 측정모형은 다음과 같다.

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

여기서  $\rho^i$ 는  $i$ 번째 GPS위성의 의사측정거리(pseudo-range measurement)이고,  $\mathbf{u}$ 는 사용자의 위치,  $c$ 는 클럭오프셋(clock offset),  $\mathbf{s}^i$ 는  $i$ 번째 GPS위성이 알려주는 자신의 위치,  $\Delta \mathbf{s}^i$ 는  $i$ 번째 GPS위성의 자신이 알려주는 위치와 실제위치와의 차이, 그리고  $e^i$ 는 그 외의 기타오차를 나타낸다.  $e^i$ 는 평균이 0인 잡음을 나타낸다. 의사측정거리  $\rho^i$ 와 위성이 알려주는 자신의 위치  $\mathbf{s}^i$ 는 모두 천체(ephemeris) 오차, SA(만약 존재한다면), 환경, 위성의 고장, 간섭, 잡음 등의 영향을 받는다. 다음과 같이  $\mathbf{u}$ ,  $\mathbf{s}^i$ ,  $\Delta \mathbf{s}^i$ 를 두자.

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

선형화 지점을 다음과 같이 표시한다.

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

그러면 의사측정거리의 추정치는 다음과 같다.

$$\begin{aligned} \rho_0^i &= \|\mathbf{u}_0 - \mathbf{s}^i - \Delta \mathbf{s}^i\| + c_0 \\ &= \sqrt{(x^i + \Delta x^i - x_0)^2 + (y^i + \Delta y^i - y_0)^2 + (z^i + \Delta z^i - z_0)^2} + c_0 \end{aligned} \quad (4)$$

다음과 같이  $\mathbf{r}$ 과  $\mathbf{p}$ 를 정의하자.

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

그러면 가시권에 있는  $n$ 개의 위성에 대한 (1)의 선형화된 행렬방정식은 다음과 같다.

$$\mathbf{H}\mathbf{p} = \mathbf{q} + \mathbf{e}, \quad (6)$$

여기서

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

그리고

$$\begin{aligned} h_{11} &= \frac{x^i + \Delta x^i - x_0}{\sqrt{(x^i + \Delta x^i - x_0)^2 + (y^i + \Delta y^i - y_0)^2 + (z^i + \Delta z^i - z_0)^2}} \\ h_{12} &= \frac{y^i + \Delta y^i - y_0}{\sqrt{(x^i + \Delta x^i - x_0)^2 + (y^i + \Delta y^i - y_0)^2 + (z^i + \Delta z^i - z_0)^2}} \\ h_{13} &= \frac{z^i + \Delta z^i - z_0}{\sqrt{(x^i + \Delta x^i - x_0)^2 + (y^i + \Delta y^i - y_0)^2 + (z^i + \Delta z^i - z_0)^2}} \end{aligned}$$

이다. 이 식에서 주목할 점은 행렬  $\mathbf{H}$ 의 마지막 열은 정확하게 알 수 있는 값들로 구성되어져 있다는 사실이다. 그래서 선형화된 행렬방정식은 다음절에서 설명할 LS-TLS 문제가 된다.

## 2. LS-TLS 혼합 문제

(6)에서  $\mathbf{b} = \mathbf{q} + \mathbf{e}$ 로 놓으면 잘 알려진 선형행렬방정식이 된다.

$$\mathbf{H}\mathbf{p} = \mathbf{b}. \quad (7)$$

고전적인 LS 방식에서는  $\mathbf{H}$ 의 모든 원소는 오차가 포함되어 있지 않다고 가정한다. 그러므로 모든 오차는 측정벡터인  $\mathbf{b}$ 에만 존재한다. 그러나 이러한 가정은 많은 적용분야에서 종종 비현실적이다. TLS 방법은 측정벡터  $\mathbf{b}$ 와 데이터 행렬  $\mathbf{H}$  모두에 오차가 있을 경우에 방정식을 풀 수 있는 적당한 방법 중의 하나이다. 본 논문에서 다루는 경우와 같이 데이터 행렬  $\mathbf{H}$ 의 일부 열에 오차가 없을 경우 우리는 그것을 LS-TLS 혼합문제라고 한다[9]. 이와 같은 LS-TLS 혼합문제를 풀 경우에 LS문제를 TLS문제를 분리하여 풀 수 있는 일괄처리 알고리즘(batch algorithm)을 사용할 수 있다[9]. 그러나 그 일괄처리방식은 많은 계산량을 요구하기 때문에 GPS 위성의 상태를 감시하는데 적용하기에는 부적절하다. 이와 같

은 문제에 있어서 계산량을 줄이는 하나의 방법은 알고리즘을 회귀적 또는 순차적인 형태로 바꾸는 것이다. 본 논문에서는 상기한 GPS 위성의 상태감시 문제에 나타난 딜레마를 해결할 수 있는 순차적인 알고리즘을 제안한다. 이 방식은 순차적인 형태를 가지므로 기존의 일괄처리방식 보다 계산량이 훨씬 준다는 잇점이 있다. 순차적인 알고리즘의 개발은 기존의 일괄처리 알고리즘으로부터 시작하므로 논문의 전개상 일괄처리 알고리즘을 간단히 설명하면 다음과 같다.

일반성을 상실하지 않으면서  $\mathbf{H}$ 행렬의 열들의 순서를 바꾸는 적당한 순서교환행렬(permutation matrix)을 통하여 (7)의 방정식을 다음과 같이 바꿀 수 있다.

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (8)$$

여기서

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

단순히 논리전개상의 편리를 위하여 이와 같은 기법을 사용하였다. 그러면 행렬  $\mathbf{A} = [\mathbf{A}_1; \mathbf{A}_2]$ 를 처음  $p$ 열( $\mathbf{A}_1$ )이 오차를 포함하지 않으면서 딱찬 행렬계수(full column rank)를 갖도록 할 수 있다. 본 연구에서는 오직 하나의 오차가 없는 열이 존재하지만 알고리즘이 일반성을 갖게하기 위하여  $p$ 개의 오차를 포함하지 않은 열이 존재한다고 가정하면 알고리즘은 다음과 같다. 행렬  $[\mathbf{A}; \mathbf{b}]$ 에  $p$ 번의 하우스홀더(Householder) 변환  $\mathbf{Q}$ 를 수행하면 다음과 같은 방정식이 얻어진다.

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

여기서  $\mathbf{R}_{11}$ 은  $p \times p$  상위삼각행렬(upper triangular matrix)이다(본 논문에서는 스칼라양이 된다.) 그 후에 (10)의 방정식 중에서  $\mathbf{R}_{22}\mathbf{x}_2 = \mathbf{y}_2$  방정식을 SVD(Singular Value Decomposition)를 이용해 풀어 TLS해  $\hat{\mathbf{x}}_2$ 를 구한다.  $\hat{\mathbf{x}}_2$ 를 이용해 전체해  $\hat{\mathbf{x}}$ 의 나머지  $n-p$ 개의 해  $\hat{\mathbf{x}}_1$ 를 구한다.  $\hat{\mathbf{x}}_1$ 를 구하기 위해서는 다음의 방정식을 풀면된다.

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

위 방정식의 해는 관측벡터  $\mathbf{b} - \mathbf{A}_2\hat{\mathbf{x}}_2$ 를 행렬  $\mathbf{A}$ 에 의해 생성되는 공간  $R(\mathbf{A}_1)$  투사시켜 구해지는 LS해이다.

여기서 주목할 점은 고장난 위성을 가려내기 위해 이 알고리즘은  $n-1$ 개의 위성으로 구성되어지는  $n$ 개의 부분집합군에 대한 위와 같은 방정식을 반복적으로 풀어야만 하는데 이 경우에 많은 계산량을 필요로하는 단점이 있다는 점이다.

## III. 설정 고장난 위성을 식별하기 위한 순차적인 알고리즘의 개발

$k$ 번째 위성이 제외되었다고 가정하자.

$$\mathbf{A}_k = [\mathbf{A}_{k,1} \quad \mathbf{A}_{k,2}] \quad \text{행렬을 } \mathbf{A} \text{ 행렬에서 } k \text{ 번째 행을 제}$$

거하여 이루어진 행렬로 놓는다.  $\mathbf{A}_{k,1}$  행렬은 정확하게 아는 것으로 간주한다.  $\mathbf{b}_k$  는 관측벡터  $\mathbf{b}$  에서  $k$  번째 행을 제거하여 이루어진 벡터로 놓는다. 다음과 같이  $[\mathbf{A}_k; \mathbf{b}_k]$  에 곱해지는 하우스홀더변환행렬을  $\mathbf{Q}_k$  라 한다.

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

여기서  $\mathbf{R}_k = \mathbf{Q}_k^T \mathbf{A}_{k,1} \in \mathbb{R}^{(n-1) \times p}$  는 상위삼각행렬이고  $\mathbf{B}_k = \mathbf{Q}_k^T [\mathbf{A}_{k,2}; \mathbf{b}_k]$  이다. 다음 단계에서  $k+1$  번째 위성이 제외되고  $k$  번째 위성은 다시 포함된다. 이 과정을 두 단계로 나누어 실행할 수 있다. 즉  $k+1$  번째 위성( $k+1$  번째 행)을 제외하는 단계와 이전에 제외되었던  $k$  번째 위성( $k$  번째 행)을 다시 포함시키는 단계로 나누어 수행할 수 있다.  $k+1$  번째 위성을 제외시킨다는 것은  $\mathbf{A}_k$  와  $\mathbf{b}_k$  에서 첫번째 행을 제외시키는 것과 같으므로  $\mathbf{A}_k$  와  $\mathbf{b}_k$  에서 첫행이 제외된 형태를 각각  $\bar{\mathbf{A}}_k = [\bar{\mathbf{A}}_{k,1}; \bar{\mathbf{A}}_{k,2}]$  와  $\bar{\mathbf{b}}_k$  로 나타낸다. 그리고 이전에 제외되었던  $k$  번째 행을 마지막 행으로 포함시킨 형태를 다음과 같이 표시한다.

$$\mathbf{A}_{k+1}^+ = \begin{bmatrix} \mathbf{A}_k \\ \alpha_k^T \end{bmatrix} \text{ and } \mathbf{b}_{k+1}^+ = \begin{bmatrix} \mathbf{b}_k \\ \beta_k \end{bmatrix}, \quad (13)$$

여기서  $\alpha_k$  와  $\beta_k$  는 이전에 제외되었던  $k$  번째 행을 나타낸다. 첫번째 단계로  $k+1$  번째 위성이 제외되는 경우를 생각하자. 순차적인 알고리즘의 형태를 유도하기 위해 다음과 같이  $\bar{\mathbf{A}}_{k,1}$  에 대하여 QR 분해(factorization)를 수행한다.

$$\bar{\mathbf{A}}_{k,1} = \bar{\mathbf{Q}}_k \bar{\mathbf{R}}_k \quad (14)$$

위에서 구한  $\bar{\mathbf{Q}}_k$  를 이용하여 다음의 행렬  $\bar{\mathbf{B}}_k$  를 계산한다.

$$\bar{\mathbf{B}}_k = \bar{\mathbf{Q}}_k^T [\bar{\mathbf{A}}_{k,2}; \bar{\mathbf{b}}_k]. \quad (15)$$

순차적인 형태를 구하기 위해서는 이전의  $\mathbf{Q}_k, \mathbf{R}_k, \mathbf{B}_k$  로부터  $\bar{\mathbf{Q}}_k, \bar{\mathbf{R}}_k, \bar{\mathbf{B}}_k$  를 구하는 것이 필요하다.

먼저  $\mathbf{Q}_k$  와  $\mathbf{R}_k$  로부터  $\bar{\mathbf{Q}}_k$  와  $\bar{\mathbf{R}}_k$  를 구하는 것을 살펴보자. 이 방법은 [10]에 잘 나타나 있다.  $\mathbf{q}_1^T$  를  $\mathbf{Q}_k$  의 첫행이라고 하면 다음과 같은 Givens 회전행렬  $\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_{n-2}$  을 구할 수 있다.

$$\mathbf{G}_1^T \mathbf{G}_2^T \dots \mathbf{G}_{n-2}^T \mathbf{q}_1 = \rho \mathbf{e}_1, \quad (16)$$

여기서  $\rho = \pm 1$  이다. 여기서

$$\mathbf{G}_1^T \mathbf{G}_2^T \dots \mathbf{G}_{n-2}^T \mathbf{R}_k = \begin{bmatrix} \mathbf{v}^T \\ \bar{\mathbf{R}}_k \end{bmatrix} \quad (17)$$

는 상위 Hessenberg (upper Hessenberg) 행렬이고

$$\mathbf{Q}_k \mathbf{G}_{n-2} \dots \mathbf{G}_1 = \begin{bmatrix} \rho & 0 \\ 0 & \bar{\mathbf{Q}}_k \end{bmatrix}, \quad (18)$$

식 안의  $\bar{\mathbf{Q}}_k \in \mathbb{R}^{(n-2) \times (n-2)}$  가 직각행렬이라는 점을 주목할 필요가 있다. 이렇게 갱신된 QR 분해를 구할 수 있다.

다음으로  $\mathbf{B}_k$  로부터  $\bar{\mathbf{B}}_k$  를 구하는 방법을 유도한다.  $[\mathbf{A}_{k,2}; \mathbf{b}_k]$  를 다음과 같이 놓는다.

$$[\mathbf{A}_{k,2}; \mathbf{b}_k] = \begin{bmatrix} \alpha_{k+1}^T & \beta_{k+1} \\ \bar{\mathbf{A}}_{k,2} & \bar{\mathbf{b}}_k \end{bmatrix}, \quad (19)$$

여기서  $\alpha_{k+1}$  와  $\beta_{k+1}$  는  $k+1$  번째 위성과 관련된 행이다. 그러면,

$$\begin{aligned} \mathbf{G}_1^T \mathbf{G}_2^T \dots \mathbf{G}_{n-2}^T \mathbf{B}_k &= \mathbf{G}_1^T \mathbf{G}_2^T \dots \mathbf{G}_{n-2}^T \mathbf{Q}_k^T [\mathbf{A}_{k,2}; \mathbf{b}_k] \\ &= \begin{bmatrix} \rho & 0 \\ 0 & \bar{\mathbf{Q}}_k^T \end{bmatrix} \begin{bmatrix} \alpha_{k+1}^T & \beta_{k+1} \\ \bar{\mathbf{A}}_{k,2} & \bar{\mathbf{b}}_k \end{bmatrix} \\ &= \begin{bmatrix} \rho \alpha_{k+1}^T & \rho \beta_{k+1} \\ \bar{\mathbf{Q}}_k^T \bar{\mathbf{A}}_{k,2} & \bar{\mathbf{Q}}_k^T \bar{\mathbf{b}}_k \end{bmatrix} \\ &= \begin{bmatrix} \rho \alpha_{k+1}^T; \rho \beta_{k+1} \\ \bar{\mathbf{B}}_k \end{bmatrix}. \end{aligned} \quad (20)$$

이와 같이  $\mathbf{B}_k$  로부터  $\bar{\mathbf{B}}_k$  를 구할 수 있다.

두번째 단계로 이전 단계에서 제외되었던 위성을 다시 추가하는 것을 생각하자. (14)와 (15)로부터 다음을 얻을 수 있다.

$$\mathbf{H}_{k+1} \equiv \text{diag}(1, \bar{\mathbf{Q}}_k^T) \mathbf{P} \mathbf{A}_{k+1} = \begin{bmatrix} \alpha_{k+1}^T; & \alpha_{k+1}^T; \beta_{k+1} \\ \bar{\mathbf{R}}_k; & \bar{\mathbf{B}}_k \end{bmatrix}, \quad (21)$$

여기서

$$\mathbf{A}_{k+1} = \begin{bmatrix} \bar{\mathbf{A}}_{k,1}; & \bar{\mathbf{A}}_{k,2}; & \bar{\mathbf{b}}_k \\ \alpha_{k+1}^T; & \alpha_{k+1}^T; & \beta_{k+1} \end{bmatrix} \quad (22)$$

이고 순서교환행렬은 다음과 같다.

$$\mathbf{P} = \begin{bmatrix} & 0 & 1 \\ \mathbf{I}_{(n-2) \times (n-2)} & & 0 \end{bmatrix}. \quad (23)$$

여기서  $\mathbf{H}_{k+1}$  의 처음  $p$  열은 상위 Hessenberg 행렬임을 주목하라. 다음과 같이 상위삼각행렬을 만드는 Givens 회전행렬  $\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_p$  는 어렵지 않게 구할 수 있다.

$$\mathbf{J}_p^T \mathbf{J}_{p-1}^T \dots \mathbf{J}_1^T \begin{bmatrix} \alpha_{k+1}^T \\ \bar{\mathbf{R}}_k \end{bmatrix} = \mathbf{R}_{k+1}, \quad \mathbf{R}_{k+1} \in \mathbb{R}^{(n-1) \times p} \quad (24)$$

이상의 결과들을 적용시켜 (21)을 다시 쓰면 다음과 같다.

$$\mathbf{J}_p \mathbf{J}_{p-1} \dots \mathbf{J}_1 \mathbf{H}_{k+1} = \begin{bmatrix} \mathbf{R}_{k+1,11} & \mathbf{R}_{k+1,12} & \mathbf{y}_{k+1,1} \\ 0 & \mathbf{R}_{k+1,22} & \mathbf{y}_{k+1,2} \end{bmatrix}, \quad (25)$$

여기서  $\mathbf{R}_{k+1} = \begin{bmatrix} \mathbf{R}_{k+1,11} \\ 0 \end{bmatrix}$ ,  $\mathbf{R}_{k+1,11} \in \mathbb{R}^{p \times p}$ ,  $\mathbf{R}_{k+1,12} \in \mathbb{R}^{p \times (n-p)}$ ,  $\mathbf{R}_{k+1,22} \in \mathbb{R}^{(n-p-1) \times (n-p)}$ ,  $\mathbf{y}_{k+1,1} \in \mathbb{R}^p$  and

$\mathbf{y}_{k+1,2} \in \mathbf{R}^{n-p-1}$ . 이렇게 원래의 문제

$$\begin{bmatrix} \bar{\mathbf{A}}_{k,1}; & \bar{\mathbf{A}}_{k,2} \\ \boldsymbol{\alpha}_{k,1}^T; & \boldsymbol{\alpha}_{k,2}^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k+1,1} \\ \mathbf{x}_{k+1,2} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{b}}_k \\ \boldsymbol{\beta}_k \end{bmatrix} \quad (26)$$

이 SVD를 사용하지 않고 다음과 같이 변환되어졌다.

$$\begin{bmatrix} \mathbf{R}_{k+1,11} & \mathbf{R}_{k+1,12} \\ 0 & \mathbf{R}_{k+1,22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k+1,1} \\ \mathbf{x}_{k+1,2} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{k+1,1} \\ \mathbf{y}_{k+1,2} \end{bmatrix} \quad (27)$$

그러면 이제 LS-TLS 혼합문제는 분리해서 풀 수 있다. TLS해  $\mathbf{x}_{k+1,2}$  는 다음의 방정식에 SVD를 이용하여 풀 수 있다.

$$\mathbf{R}_{k+1,22}\mathbf{x}_{k+1,2} = \mathbf{y}_{k+1,2} \quad (28)$$

비록 SVD를 적용하는 많은 수치적으로 안정된 알고리즘이 있지만 많은 계산량을 필요로 한다. 이 문제를 극복하기 위하여 다음의 행렬을 구성한다.

$$\mathbf{D}_{k+1} = \begin{bmatrix} \mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2} \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_{k+1,22}; \mathbf{y}_{k+1,2} \end{bmatrix} \quad (29)$$

그리고  $R_{k+1,22}$  의 최소특이치(minimum singular value)와 연관된 최소특이벡터(minimum singular vector)를 구하는 대신에 FALM (Fast Algorithm to Locate Minimum eigenpair) [11,12]을 이용하여 최소고유치(minimum eigenvalue)와 연관된 최소 고유벡터(minimum eigenvector)를 계산한다. 이렇게 구한 TLS해를 이용하여 다음 방정식을 풀어 LS해  $\mathbf{x}_{k+1,1}$  를 구한다.

$$\mathbf{R}_{k+1,11}\mathbf{x}_{k+1,1} = \mathbf{y}_{k+1,1} - \mathbf{R}_{k+1,12}\mathbf{x}_{k+1,2} \quad (30)$$

이렇게  $k+1$  단계에서 전체 해  $\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1,1}^T; \mathbf{x}_{k+1,2}^T \end{bmatrix}^T$  가 구해진다. 다음 단계에서 사용하기 위한  $\mathbf{Q}_{k+1}$  을 순차적인 형태로 구하면 다음과 같다.

$$\mathbf{Q}_{k+1} = \mathbf{P}^T \text{diag}(1, \bar{\mathbf{Q}}_k) \mathbf{J}_1^T \mathbf{J}_2^T \cdots \mathbf{J}_p^T \quad (31)$$

이상의 유도과정을 통하여 얻어진 알고리즘을 정리하면 다음과 같다. 알고리즘을 표현하는데 있어 잘 알려지고 간단한 MATLAB 문법을 사용하였다.

1. 알고리즘

Step 0: 의사측정거리로부터 얻어지는 (6)으로부터 선형화된 (12)를 구하고 첫행을 제외한 후 QR분해를 하여 LS와 TLS해를 각각 구한다. 그러면  $k=1$  일 때의  $\mathbf{Q}_k, \mathbf{R}_k, \mathbf{B}_k, \boldsymbol{\alpha}_k, \boldsymbol{\beta}_k$  이 주어진다. 이후에 step 1로부터 시작되는 순차적인 알고리즘으로 진행한다.

Step 1: 다음과 같은 Givens 회전행렬  $\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_{n-2}$  을 계산한다.

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

여기서  $\mathbf{q}_1^T$  는  $\mathbf{Q}_k$  의 첫번째 행이고  $\rho = \pm 1$  이다.

Step 2: 다음과 같은  $\bar{\mathbf{R}}_k, \bar{\mathbf{Q}}_k, \bar{\mathbf{B}}_k$  를 계산한다.

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

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

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

Step 3: 다음과 같은 Givens 회전행렬  $\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_p$  를 계산한다.

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

여기서  $\mathbf{R}_{k+1}$  은 위삼각행렬(upper triangular matrix)이다.

Step 4:  $\mathbf{R}_{k+1,12}, \mathbf{R}_{k+1,22}, \mathbf{y}_{k+1,1}, \mathbf{y}_{k+1,2}$  를 계산한다.

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

여기서  $\mathbf{R}_{k+1,12} \in \mathbf{R}^{p \times (n-p)}, \mathbf{R}_{k+1,22} \in \mathbf{R}^{(n-p-1) \times (n-p)}$ ,

$\mathbf{y}_{k+1,1} \in \mathbf{R}^p$  and  $\mathbf{y}_{k+1,2} \in \mathbf{R}^{n-p-1}$ .

Step 5: (29)의  $\mathbf{D}_{k+1}$  행렬을 구성하고 FALM을 사용하여  $\mathbf{D}_{k+1}$  의 최소고유벡터  $\mathbf{v}$  를 구한다. 그 후에  $\mathbf{v}$  를 이용하여 TLS 해  $\mathbf{x}_{k+1,2}$  를 계산한다.

$$\mathbf{x}_{k+1,2} = -\frac{1}{v_{k-p+1}} [v_1, v_2, \dots, v_{k-p}]^T$$

Step 6: (30)의 LS해  $\mathbf{x}_{k+1,1}$  를 계산한다. 그러면 전체해는

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1,1}^T; \mathbf{x}_{k+1,2}^T \end{bmatrix}^T \text{ 이 된다.}$$

Step 7: (31)의  $\mathbf{Q}_{k+1}$  을 계산한다. 만약 모든 위성이 차례로 제거되었으면 프로그램을 멈추고 그렇지 않으면 Step 1로 간다.

Givens 회전행렬을 구하는데 4플롭(flop)과 하나의 제곱근을 구하는 것이 필요하고 Givens 회전행렬을 벡터에 곱하는데  $4(m-1)$  플롭이 필요하므로[10], 위 알고리즘은 한 서브셋에 대하여 약  $44m+81$ 의 곱셈과 2m개의 제곱근을 구하는 것이 필요하다. 여기서  $m$  은  $n-1$ 이고  $n$ 은 보이는 위성의 개수다. 한편 SVD는 특이값과 특이벡터를 계산하는데  $50m + 256$ 의 곱셈이 필요하고 특이치과 좌우특이벡터를 계산하는데는  $10m^2 + 100m + \frac{14}{3}n^3$  플롭이 필요하다. 그러므로 본 논문에서 제시된 알고리즘이 계산량이 더 적은 것을 알 수 있다. 다음 장에서는 본 알고리즘이 기능적으로 잘 동작하는 것과 계산량이 SVD를 사용하는 알고리즘과 비교하여 상대적으로 적다는 것을 시뮬레이션 결과를 통하여 보이고자 한다.

IV. 시뮬레이션 결과

본 장에서는 시뮬레이션을 통하여 알고리즘의 성능을 보인다. 시뮬레이션의 초점은 위성이 고장난 상황에서 본 알고리즘이 얼마나 잘 동작하는지의 여부이다. 위성데이터는 MATLAB 툴박스를 통하여 생성되었다[13]. 위성데이터를 생

성할 때에 열에 의한 잡음(thermal noise), 대류권오차(tropospheric error), 다중경로오차(multipath error), 이온층오차(ionospheric error)를 고려하였다. 시간 t에서 1000m 의사측정거리 오차를 인가하여 결과를 관찰하였다. 그 외 다양한 크기의 의사측정거리 오차를 인가하여 보았으나 거의 비슷한 결과를 얻을 수 있었다. 알고리즘의 정상 동작 여부를 판단하기 위하여는 오차가 인가되기 전과 오차가 인가된 후의 결과를 비교하는 것이 중요하기 때문에 시간 t가 정확히 언제 인지는 중요성을 갖지 않는다. 시뮬레이션을 수행할 때에 PRN #7 위성이 t-1시간까지는 정상동작하다가 t-1과 t 사이에 고장이 일어났다고 가정하였다. 그러면 시간 t에서의 위성데이터는 부정확한 데이터가 될 것이다. 이 경우에 본 논문에서 제안한 알고리즘이 제대로 동작하는지 살펴보았다. 본 알고리즘이 시간 t-1과 t에서의 위치와 클럭바이어스를 계산한 결과를 아래 그림에 나타내었다. 그림 1은 계산된 위치를 나타내고 있다.

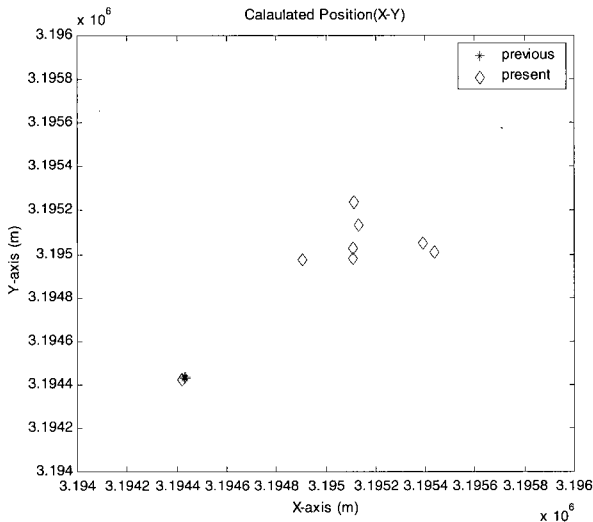


그림 1. 계산된 위성의 위치.  
Fig. 1. Calculated position.

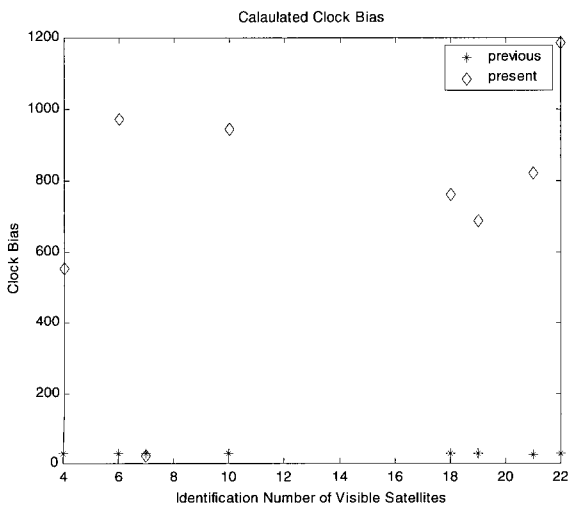


그림 2. 계산된 클럭바이어스(clock bias).  
Fig. 2. Calculated clock bias.

그림에서 다이아몬드는 시간 t에서의 위치를 나타내고 있다. 시간 t-1에서와 마찬가지로 8개의 위치가 계산되었는데 모두 서로 떨어져 있는 것을 볼 수 있다. 이 결과로 위성중 하나가 비정상적인 동작을 하고 있음을 알 수 있다. 단지 하나의 위치만이 이전 위치와 일치하는 것을 볼 수 있다. 이것을 통하여 어떤 위성이 비정상적으로 동작하는지를 알 수 있다.

그림 2는 계산된 클럭바이어스를 보여준다. 본 시뮬레이션에서는 클럭바이어스를 0으로 가정하였다. t-1시간에서 계산된 8개의 클럭바이어스(별표)는 모두 0 근처에 나타났다. 반면에 다이아몬드로 표시된 시간 t에서 계산된 클럭바이어스는 하나(PRN #7)만이 0 근처에 표시되고 있다. 이것을 통하여 PRN #7 위성이 비정상이라는 것을 알 수 있다. 이상과 같은 결과를 보인데 본 논문에서 제시된 알고리즘에 적당한 판별 기준(measure)과 문턱값(threshold)(본 논문의 주제는 아니지만)을 더한다면 TLS에 기초한 고장난 위성을 식별하는데 탁월한 성능을 보일 수 있음을 알 수 있다.

아래 표1은 24시간 동안 가시위성의 수가 변함에 따라 제안된 알고리즘과 기존의 SVD를 사용하는 알고리즘 간의 계산량을 비교한 것이다. 표에 나타난 계산량은 두 알고리즘의 같은 부분의 계산량은 제외하고 TLS 문제를 푸는 부분의 경우만 나타난 것이다. 그러므로 표에 나타난 계산량이 전체 알고리즘의 계산량은 아니다.

아래 그림 3은 위의 표를 그래프로 나타낸 것으로 본 논문에서 제안된 방법이 계산량이 더 적을 뿐만 아니라 가시위성의 수가 늘어날수록 계산량의 차이가 많이 벌어지는 것을 보여주고 있다. 이와 같은 현상은 SVD의 계산량은 가시위성의 수의 제곱에 비례하지만 본 논문에서 제안한 방법은 가시위성의 수의 제곱에 비례하기 때문이다. 그러므로 GPS 위성뿐만 아니라 장차 GLONASS와 유럽에서 추진하고 있는 갈릴레오 위성시스템이 동시에 수신되는 상황에서는 더 많은 장점이 있음을 알 수 있다.

표 1. 알고리즘의 계산량 비교(flops).

Table 1. Comparison of computational amount.

가시위성의 수	제안된 알고리즘	기존의 SVD를 사용하는 알고리즘
6	1463	1543
7	1496	1602
8	1622	1862
6	2080	2644

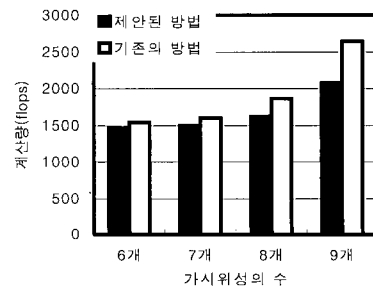


그림 3. 가시위성의 수에 따른 계산량 비교.

Fig. 3. Comparison of computational amount according to the number of visible satellites.

본 알고리즘은 TLS가 사용되어질 수 있는 많은 분야[14, 15]에 적용할 수 있다.

### V. 결론

본 논문에서는 고장난 위성을 판별하기 위한 새로운 방법인 TLS에 기초한 순차적인 알고리즘을 제안하였다. 본 논문의 중요한 공헌은 TLS에 기초한 알고리즘들이 SVD를 사용하기 때문에 계산량이 많은데 비해 알고리즘을 순차적인 형태로 유도하여 적은 계산량으로 TLS의 잇점을 살릴 수 있도록 한 점이다. 적당한 판별기준과 문턱값만 보강된다면 본 알고리즘은 고장난 위성을 판별하는데 유용하게 사용되어질 수 있을 것이다.

### 참고문헌

- [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] 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] B. W. Parkinson and J. J. Spilker, (Eds), *Global Position System: Theory and Applications*, vol. I, New York:AIAA, 1996.
- [4] D. Wells, et al, *Guide to GPS Positioning*, Canadian GPS Associates, 1987.
- [5] 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)*, June 24-26, pp. 1-4, 1986.
- [6] B. W. Parkinson and P. Axelard, "Autonomous GPS integrity monitoring using the pseudorange residual," *Navigation (Washington)*, vol. 35, no. 2, pp. 255-274, 1988.
- [7] B. S. Pervan, et al, "Parity space methods for autonomous fault detection and exclusion algorithms using carrier phase," *Proc. of PLANS'96*, 1996.
- [8] J.-C. Juang, "On GPS position and integrity monitoring", *IEEE Transactions on Aerospace and Electronics Systems*, vol. 36, no. 1, pp. 327-336, January 2000.
- [9] S. Van Huffel, and J. Vandewalle, *The Total Least Squares Problem Computational Aspects and Analysis*, SIAM, 1991.
- [10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, 1983.
- [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 Trans. on Information and Systems*, vol. E82-D, no. 6, pp. 1011-1019, June 1999.
- [13] GPSsoft, *Satellite Navigation TOOLBOX 2.0 User's Guide*, GPSsoft LLC, October 1999.
- [14] S.-H. and K.-J. Woo, "Serial communication-based fault diagnosis of a BLDC motor using bayes classifier," *International Journal of Control, Automation, and Systems*, vol. I, no. 3, pp. 308-314, Sept. 2003.
- [15] M. H. Choi and W. W. Lee, "Quantitative evaluation of an intuitive teaching method for industrial robot using a force/moment direction," *International Journal of Control, Automation, and Systems*, vol. I, no. 3, pp. 395-400, Sept. 2003.



### 전 창 완

1963년생. 1986년 서울대학교 제어계측 공학과 졸업. 1986년 동 대학원 석사. 1996년 동 대학원 박사. 1994년 3월~1998년 2월 삼성전자 정보통신본부. 2001년 2월~2002년 2월 캐나다 캘거리 대학교 방문교수. 1998년 3월~현재 순천향대학교 교수. 관심분야는 시스템 식별이론, GPS 응용, 칼만필터, 제어기 설계, S/W 개발.