

Total Least Squares

선형계수추정문제는 자동제어, 신호처리, 시스템이론, 통계학, 경제학, 그리고 식물학 등 많은 분야에서 발생하는 중요한 문제 중의 하나이다. 이 문제는 선형회귀방정식(linear regression equation)에서 출발하여 overdetermined 된 선형방정식 $Ax \approx b$ 를 푸는 문제로 귀착된다. 데이터 행렬 A 와 측정 벡터 b 모두에 오차가 있는 경우 이 문제를 푸는 효과적인 방법 중의 하나가 total least squares (TLS)이다. 그러나 TLS가 사용되어지면 좋은 문제에도 효과적인 알고리즘을 모른다든지 계산량이 많다든지 하는 등의 이유로 least squares (LS) 방법이 많이 사용되어져 왔다. 본 논문에서는 TLS의 기초 이론과 응용 예를 소개하여, TLS가 사용되어져야 할 곳에 정확하게 사용되어져 더 좋은 연구 결과가 도출되도록 한다.

■ 전창완
(순천향대학교 전기통신공학과)

1. 서론

본 논문에서는 선형계수추정에 사용되어질 수 있는 기법 중의 하나인 Total Least Squares(TLS)의 원리와 응용에 관하여 소개한다. "Total Least Squares"란 용어가 문헌에 나타나기 시작한 것은 1980년대이지만 통계학 분야에서는 "orthogonal regression"이나 "errors-in-variables regression"란 이름으로 100년 이상의 긴 역사를 가지고 있는 개념이다[1-2]. 본 논문에서는 공학의 입장에서 계수 추정의 문제를 해결할 수 있는 하나의 도구로써 접근할 것이며 응용에 초점을 맞추어 실제적인 도움이 되도록 할 것이다.

본 논문의 구성은 다음과 같다. 제2장에서는 기초적인 TLS문제의 이론과 해를 구하는 간단한 알고리즘을 소개하고 제3장에서는 Least Squares(LS)와 TLS가 혼합된 문제의 해결방법을 소개하고 제4장에서는 실제 응용 예를 들어 실제적인 문제에 어떻게 적용할 수 있는지를 설명하고 제5장에서 결론을 맺는다.

2. TLS의 개념

본 장에서는 TLS의 기초 이론 및 TLS문제를 푸는 기본적인 알고리즘을 소개한다.

TLS의 기초 이론을 설명하기 위하여 다음과 같은 선형방정식을 생각한다.

$$\alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n = \beta \tag{1}$$

위와 같은 선형방정식은 자동제어, 신호처리, 시스템 모델링 등 공학 분야뿐만 아니라, 통계학, 물리학, 경제학, 생물학 등의 분야에서도 자주 등장하는 익숙한 방정식이다. 예를 들어 다음과 같은 간단한 시스템을 생각할 때에 (1)의 형태로 다음 그림 1의 시스템을 (2)와 같이 모델링 할 수가 있다.

$$u(t-1)x_1 + u(t-2)x_2 + \dots + u(t-n)x_n = y(t) \tag{2}$$

여기서 $u(t)$ 는 시스템의 입력, $y(t)$ 는 시스템의 출력이고 $x_i, i=1,2,\dots,n$ 는 시스템의 특성을 결정짓는 모델계수이다. 본 논문에서 우리의 목적은 TLS를 이용하여 시스템 모델의 계수를 추정하는 것이므로 (2)에서 입력($u(t)$)과 출력($y(t)$) 정보를 이용하여 미지의 모델계수 $x_i, i=1,2,\dots,n$ 를 구한다. 미지의 계수를 추정하기 위하여 $t=1,2,\dots,m(m > n)$ 에 대하여 입출

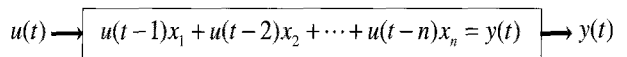


그림 1. 간단한 입출력 시스템의 모델링

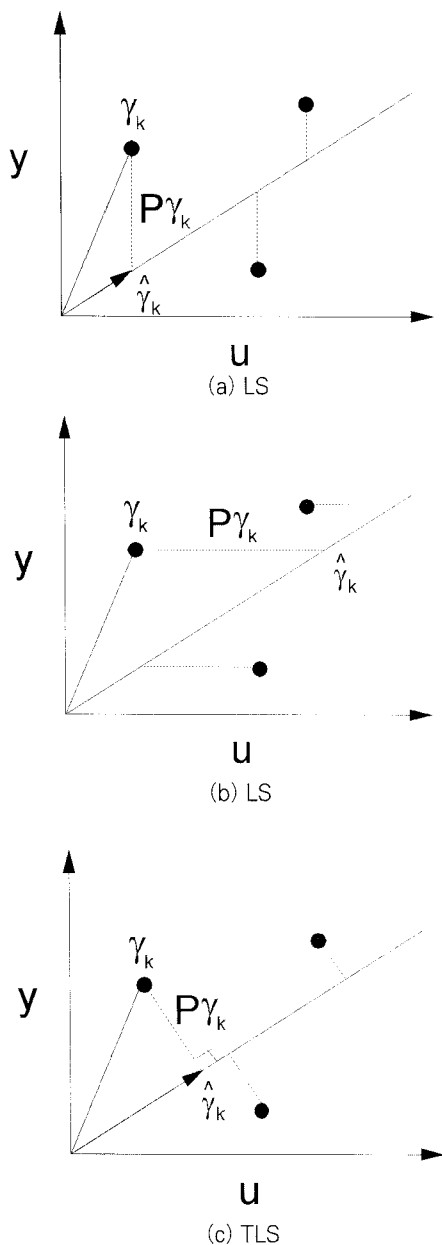


그림 2. 3가지 linear regression에 대한 기하학적 해석

력 관계식을 행렬로 표시하면 다음과 같다.

$$A\underline{x} \approx \underline{b} \tag{3}$$

여기서

$$A = \begin{bmatrix} u(0) & u(-1) & \cdots & u(1-n) \\ u(1) & u(0) & \cdots & u(2-n) \\ \vdots & \vdots & \ddots & \vdots \\ u(m-1) & u(m-2) & \cdots & u(m-n) \end{bmatrix}, \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \underline{b} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

(3)에서 '=' 대신 ' \approx ' 기호를 사용한 것은 실제 시스템에서는 측정 잡음 등으로 인한 오차가 있기 때문에 이를 나타내기 위한 것이다. 위와 같이 overdetermined된 선형방정식의 경우 보통 잘 알려진 LS해를 구하여 사용하여 왔다. 그러나 이러한 경우에는 시스템의 오차가 출력 벡터 \underline{b} 에만 존재한다는 가정이 기저에 깔려있다. 만약 오차가 데이터 행렬 A 에도 존재한다면 LS해는 데이터 행렬에 존재하는 오차를 무시한 만큼의 부정확한 값을 주게 될 것이다. TLS 방법은 데이터 행렬 A 에도 존재하는 오차를 고려하여 overdetermined된 선형방정식의 해를 좀 더 정확히 구하는 방법이다.

일차원 시스템인 경우 위에서 설명한 LS와 TLS의 개념을 그림으로 설명하면 그림 2와 같다[3]. 그림에서 $t = k$ 일 때 $\underline{\gamma}_k$ 를 $\underline{\gamma}_k = [y(k) \ u_k^T]^T$ 로 정의했고 투영 \mathbf{P} 는 각각의 경우에 적당한 형태를 취한다고 가정한다. 이때 $y(k)$ 는 오차가 포함된 출력벡터이고 $\underline{u}_k^T = [u(k) + n_i(k) \ u(k-1) + n_i(k-1) \ \cdots \ u(k-n+1) + n_i(k-n+1)]$ 는 오차가 포함된 입력벡터이다.

그림 2에서 LS해는 각각의 데이터에서 종속변수(여기서는 출력 $y(t)$)의 방향을 따라 hyperplane까지의 거리의 제곱의 합을 최소화시킨 것이고(a), 일명 "data least squares"로 불리는 LS해는 각각의 데이터에서 독립변수(여기서는 출력 $u(t)$)의 방향을 따라 hyperplane까지의 거리의 제곱의 합을 최소화시킨 것이다(b). 한편 그림 (c)에서 보듯이 TLS해는 각각의 데이터에서 hyperplane까지의 수직 거리의 제곱의 합을 최소화시킨 것이다.

3. TLS의 해

본 장에서는 SVD(Singular Value Decomposition)에 기초한 TLS를 푸는 기본적인 알고리즘에 대하여 설명한다[4].

(3)을 다음과 같이 바꿔 표현한다.

$$[A \ \underline{b}] \begin{bmatrix} \underline{x}^T & -1 \end{bmatrix}^T \approx 0. \tag{4}$$

$[A \ \underline{b}]$ 의 SVD를 다음과 같이 구한다.

$$[A \ \underline{b}] = U \Sigma V^T \tag{5}$$

여기서

$$\begin{aligned} U &= [\underline{u}_1, \cdots, \underline{u}_m], \underline{u}_i \in R^m, U^T U = I_m, \\ U &= [\underline{u}_1, \cdots, \underline{u}_m], \underline{u}_i \in R^m, U^T U = I_m, \\ \Sigma &= \text{diag}(\sigma_1, \cdots, \sigma_{n+1}) \in R^{m \times (n+1)}, \sigma_1 \geq \cdots \geq \sigma_{n+1} \geq 0. \end{aligned} \tag{6}$$

만약 $\sigma_{n+1} \neq 0$ 라면 $[A \ b]$ 는 rank가 $n+1$ 이 되고 의 행들에 $[A \ b]$ 의해서 생성되는 공간 S 는 R^{n+1} 이 되게 된다. 그러나 이 경우에 $[A \ b]$ 의 null space가 존재하지 않게 되므로 (4)의 해를 구할 수 없게 된다. 따라서 $[A \ b]$ 의 rank가 n 이 되어야만 한다. $[A \ b]$ 를 rank n 으로 가장 잘 근사화시키는 것을 $[\hat{A} \ \hat{b}]$ 로 나타내면 $[\hat{A} \ \hat{b}]$ 를 다음과 같이 표현할 수 있다.

$$[\hat{A} \ \hat{b}] = U \hat{\Sigma} V^T \quad (7)$$

여기서 $\hat{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_n, 0)$ 이다. 이 때 minimal TLS correction은 다음과 같다.

$$\hat{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_n, 0) \quad (8)$$

그리고 다음과 같은 관계가 성립한다.

$$[A \ b] - [\hat{A} \ \hat{b}] = [\Delta \hat{A} \ \Delta \hat{b}] = \sigma_{n+1} \mathbf{u}_{n+1} \mathbf{v}_{n+1}^T \quad (9)$$

(9)에서 TLS correction 행렬은 rank 1을 갖게 되고 다음의 관계가 성립한다.

$$[\hat{A} \ \hat{b}] [\mathbf{x}^T \ -1]^T = 0 \quad (10)$$

(10)이 만족하려면 벡터 $[\mathbf{x}^T \ -1]^T$ 이 $[\hat{A} \ \hat{b}]$ 의 null space에 속해야 하므로 TLS해 $[\mathbf{x}^T \ -1]^T$ 는 다음과 같이 \mathbf{v}_{n+1} 을 스케일링함으로써 구할 수 있다.

$$[\hat{\mathbf{x}}^T \ -1]^T = \frac{-1}{v_{n+1, n+1}} \mathbf{v}_{n+1} \quad (11)$$

만약 $v_{n+1, n+1} \neq 0$ 이면

$\hat{\mathbf{b}} = \hat{A} \hat{\mathbf{x}} = -1 / (v_{n+1, n+1}) \hat{A} [v_{1, n+1}, \dots, v_{n, n+2}]^T \in R(\hat{A})$ 가 되어 $\hat{\mathbf{x}}$ 가 TLS의 해가 됨을 알 수 있다.

한편, 만약 σ_{n+1} 가 0이면 $[A \ b]$ 는 rank가 n 이 되어 자연스럽게 $v_{n+1} \in N([A \ b])$ 이 되므로 근사화 과정이 필요 없게 된다.

정리하면 σ'_n 을 A 의 가장 작은 singular value라 할 때 $\sigma'_n > \sigma_{n+1}$ 에 라면 TLS해는 존재하며 유일하고 다음과 같이 주어진다.

$$\hat{\mathbf{x}} = -\frac{1}{v_{n+1, n+1}} [v_{1, n+1}, \dots, v_{n, n+1}]^T \quad (12)$$

이 TLS 해는 minimum norm 해와 같다[5].

이상과 같은 SVD를 이용한 방법은 SVD를 구하는데 A 의 차

원이 n 이라면 $O(n^3)$ 의 계산량이 필요하므로 이를 최소고유치 (minimum eigenpair) 문제로 변형시켜 계산량을 줄이는 방법도 있다[6-8].

4. LS-TLS 혼합 문제

(3)의 A 에서 일부 열에 오차가 없는 경우에 이를 LS-TLS 혼합 문제라고 하며 본 장에서는 이 문제를 푸는 방법을 다루도록 한다.

(3)의 A 를 $A = [A_1; A_2]$ 로 할 때 A_1 를 오차가 없는 p 개의 열로 이루어진 행렬이라고 가정한다. 만약 오차가 없는 열이 한 곳으로 몰려 있지 않고 퍼져 있을 경우에는 적당한 permutation 행렬을 이용하여 위와 같이 맨 앞의 한 곳으로 모을 수 있다. $[A; b]$ 행렬에 Householder transformations Q 를 A_1 행렬의 열의 수인 p 만큼 행하면 다음과 같이 된다.

$$Q^T [A_1; A_2; b] = \begin{bmatrix} R_{11} & R_{12} & r_1 \\ 0 & R_{22} & r_2 \end{bmatrix} \quad (13)$$

여기서 R_{11} 은 $p \times p$ upper triangular 행렬이다. 그러면 SVD를 이용하여 $R_{22} \mathbf{x} \approx r_2$ 의 TLS해 $\hat{\mathbf{x}}_2$ 를 구할 수 있다. 그 후에 이를 이용하여 해의 처음 p 개의 원소인 $\hat{\mathbf{x}}_1$ 을 다음의 식에서 구할 수 있다.

$$R_{11} \mathbf{x} = r_1 - R_{12} \hat{\mathbf{x}}_2 \quad (14)$$

위 식은 R_{11} 에는 오차가 없기 때문에 LS해를 구하면 된다.

위와 같은 알고리즘은 계산량이 많기 때문에 실시간에 적용하기 위하여 순차적인 알고리즘도 발표되었다.

5. TLS의 응용

본 장에서는 TLS기법이 응용되는 분야를 예를 들어 설명한다.

5.1 GPS 분야에의 응용

일반적으로 GPS를 이용한 측위에 사용되는 측정 모델은 다음과 같다.

$$\rho^i = \|\mathbf{u} - \mathbf{s}^i - \Delta \mathbf{s}^i\| + c + e^i \quad (15)$$

여기서 n 은 보이는 위성의 수, ρ^i 는 i 번째 위성의 pseudo-range measurement, \mathbf{u} 는 사용자의 위치, c 는 clock offset, \mathbf{s}^i 는 i 번째 위성의 broadcast position, $\Delta \mathbf{s}^i$ 는 i 번째 위성의 broadcast

position과 true position의 차이, e^i 는 평균 0으로서 그 외의 오차들을 나타낸다. pseudo-range ρ^i 와 broadcast position \mathbf{s}^i 는 ephemeris 오차, 환경에 의한 오차, 위성의 고장, 간섭, 잡음 등을 포함한다. 선형화되는 점을 (16) 같이 하고

$$\mathbf{u} = \mathbf{u}_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, c = c_0 \quad (16)$$

다음 (17)과 같이 정의하면

$$\mathbf{u} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \mathbf{s}^i = \begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix}, \Delta \mathbf{s}^i = \begin{bmatrix} \Delta x^i \\ \Delta y^i \\ \Delta z^i \end{bmatrix} \quad (17)$$

추정된 pseudo-range measurement는 다음과 같이 된다.

$$\begin{aligned} \rho_0^i &= \|\mathbf{u}_0 - \mathbf{s}^i - \Delta \mathbf{s}^i\| + c_0 \\ &= \sqrt{(x^i + \Delta x^i - x_0)^2 + (y^i + \Delta y^i - y_0)^2 + (z^i + \Delta z^i - z_0)^2} + c_0 \end{aligned} \quad (18)$$

n 개의 위성이 보인다고 가정할 때 선형행렬방정식 (1)과 같은 형태로 표현하면 다음과 같다.

$$\mathbf{H}_p = \mathbf{q} + \mathbf{e} \quad (19)$$

여기서

$$\begin{aligned} \mathbf{r} &= \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix}, \delta = -c + c_0, \mathbf{p} = \begin{bmatrix} \mathbf{r} \\ \delta \end{bmatrix}, \\ \mathbf{H} &= \begin{bmatrix} h_{11} & h_{12} & h_{13} & 1 \\ h_{21} & h_{22} & h_{23} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ h_{n1} & h_{n2} & h_{n3} & 1 \end{bmatrix}, \mathbf{q} = \begin{bmatrix} \rho_0^1 - \rho^1 \\ \rho_0^2 - \rho^2 \\ \vdots \\ \rho_0^n - \rho^n \end{bmatrix}, \mathbf{e} = \begin{bmatrix} e^1 \\ e^2 \\ \vdots \\ e^n \end{bmatrix}, \end{aligned}$$

그리고

$$\begin{aligned} h_{i1} &= \frac{x^i + \Delta x^i - x_0}{\sqrt{(x^i + \Delta x^i - x_0)^2 + (y^i + \Delta y^i - y_0)^2 + (z^i + \Delta z^i - z_0)^2}} \\ h_{i2} &= \frac{y^i + \Delta y^i - y_0}{\sqrt{(x^i + \Delta x^i - x_0)^2 + (y^i + \Delta y^i - y_0)^2 + (z^i + \Delta z^i - z_0)^2}} \\ h_{i3} &= \frac{z^i + \Delta z^i - z_0}{\sqrt{(x^i + \Delta x^i - x_0)^2 + (y^i + \Delta y^i - y_0)^2 + (z^i + \Delta z^i - z_0)^2}} \end{aligned}$$

이다. H행렬의 마지막 열이 정확히 아는 값이므로 mixed LS-TLS 문제가 된다.

5.2 가속도계와의 응용

가속도계는 자이로스코프와 함께 관성항법장치 Inertial Navigation System (INS)의 기본적인 센서이다. 자이로스코프의 오차보상 방법에는 deterministic 오차보상법과 random 오차보상법이 있다. deterministic 오차보상의 원리는 시불변 계수들을 추정하여 보상해주는 것으로서 이를 위하여 다음과 같은 입력력 모델을 고려한다. 이 모델에서 세계의 가속도계를 가정했고 이 가속도계는 서로 직각으로 배치되었다고 가정한다.

$$\begin{bmatrix} K_x V_x \\ K_y V_y \\ K_z V_z \end{bmatrix} = \begin{bmatrix} d_{xx} & d_{xy} & d_{xz} \\ d_{yx} & d_{yy} & d_{yz} \\ d_{zx} & d_{zy} & d_{zz} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} n_x a_x^2 \\ n_y a_y^2 \\ n_z a_z^2 \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (20)$$

여기서 K_i 는 i 축의 scale factor, V_i 는 i 축의 출력, $\{d_{ij}\}$ is the 방향 코사인행렬, a_i 는 IMU (Inertial Measurement Unit) 좌표계의 i 축 입력가속도, n_i 는 i 축 가속도계의 quadratic error coefficient 그리고 b_i 는 i 축가속도계의 바이어스를 나타낸다.

오차계수들을 추정하기 위하여 (20)을 (21)과 같이 변형한다.

$$K_x V_x = [1 \ a_x \ a_y \ a_z \ a_x^2] \begin{bmatrix} b_x \\ d_{xx} \\ d_{xy} \\ d_{xz} \\ n_x \end{bmatrix} \quad (21-a)$$

$$K_y V_y = [1 \ a_x \ a_y \ a_z \ a_y^2] \begin{bmatrix} b_y \\ d_{yx} \\ d_{yy} \\ d_{yz} \\ n_y \end{bmatrix} \quad (21-b)$$

$$K_z V_z = [1 \ a_x \ a_y \ a_z \ a_z^2] \begin{bmatrix} b_z \\ d_{zx} \\ d_{zy} \\ d_{zz} \\ n_z \end{bmatrix} \quad (21-c)$$

위 식들은 서로 독립이고 유사한 형태를 가지므로 동일한 방법으로 계수를 추정할 수 있으므로 (21)의 측정방정식을 $i (=x,y,z)$ 축에 대하여 다음과 같이 표현한다.

$$V_i = [1 \quad a_x \quad a_y \quad a_z \quad a_i^2] \begin{bmatrix} b_i / K_i \\ d_{ix} / K_i \\ d_{iy} / K_i \\ d_{iz} / K_i \\ n_i / K_i \end{bmatrix} \quad (22)$$

만약 일반적으로 n개의 위치를 고려하면 (22)의 측정방정식은 다음과 같이 표현된다.

$$\begin{bmatrix} V_i(1) \\ V_i(2) \\ \vdots \\ V_i(n) \end{bmatrix} = \begin{bmatrix} 1 & a_x(1) & a_y(1) & a_z(1) & a_i^2(1) \\ 1 & a_x(2) & a_y(2) & a_z(2) & a_i^2(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_x(n) & a_y(n) & a_z(n) & a_i^2(n) \end{bmatrix} \begin{bmatrix} b_i / K_i \\ d_{ix} / K_i \\ d_{iy} / K_i \\ d_{iz} / K_i \\ n_i / K_i \end{bmatrix} \quad (23)$$

$a_i(k)(k=1,2,\dots,n)$ 는 실제로 불확실성이 있는 값이고 행렬의 첫번째 열이 정확이 아는 값이므로 이 또한 mixed LS-TLS 문제가 되므로 이를 풀 후에 방향코사인 행렬의 원소들이 갖는 $d_{ix}^2 + d_{iy}^2 + d_{iz}^2 = 1$ 관계를 이용하면 원하는 오차계수를 구할 수 있다.

6. 결론

본 논문은 TLS를 소개하는 논문으로써 추정론의 관점에서 TLS에 관한 일반적인 이론과 해를 구하는 방법 그리고 응용의 실제를 예들 들어 기술하였다. 논문은 응용에 초점을 맞추어 응용하는데 필요한 내용들 위주로 기술하였다. TLS에 관한 연구는 TLS해를 구하는 데 계산량이 많으므로 속도 향상과 메모리 사용량을 줄이기 위한 연구가 많이 진행되어져 왔다. 그러나 요즘은 컴퓨터 기술의 발달로 이러한 연구의 필요성이 많은 부분 해소되었으나 여전히 임베디드 시스템 같이 한정된 자원을 사용하는 시스템의 경우에는 이와 같은 연구가 필요하다. 본 논문에서 다루지 않은 내용들과 최근에 TLS를 적용한 많은 예들을 참고문헌에 수록하였다.[9-20]

참고문헌

[1] G. H. Golub and C.F. Van Loan, An Analysis of the total least squares problem, SIAM J. Numer. Anal., 17, 1980, pp. 883-893.
 [2] R. J. Adcock, A problem in least squares, The analyst, 5, 1878, pp. 53-54.

[3] C. E., Davila, "An Efficient Recursive Total Least Squares Algorithm for FIR Adaptive Filtering," *IEEE Trans. Signal Processing*, Vol. ASSP-42, pp. 268-280, February 1994.
 [4] S. Van Huffel, and J. Vandewalle, *The Total Least Squares Problem Computational Aspects and Analysis*, SIAM, 1991.
 [5] E. M. Dowling and R. D. DeGroat, "The Equivalence of the Total Least Squares and Minimum Norm Method", *IEEE Transaction on Signal Processing*, Vol. 39, No. 8, August 1991, pp. 1891-1892.
 [6] C. W. Jeon, H. J. Kim, "An LS-TLS Based Sequential Identification Algorithm: Application to an accelerometer", *IEEE Transactions on Aerospace and Electronics*, Vol. 37, No. 2, pp. 607-618, 2001.
 [7] 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, 2004.
 [8] C. W. Jeon and G. Lachapelle, "A New TLS-Based Sequential Algorithm to Identify Two Failed Satellites", *International Journal of Control, Automation, and Systems*, Vol. 3, No. 2, pp. 166-172, 2005.
 [9] P. Stoica, and T. Soderstrom, "Bias Correction in Least-Squares Identification," *International Journal of Control*, Vol. 35, No. 3, pp. 449-457, 1982.
 [10] C. Jie. et. al., "Hybrid Filter Optimization Design Using Total Least Square Solving," *Proc. ICMMT 2008*.
 [11] I. Markovsky, et. al., "Application of Structured Total Least Squares for System Identification and Model Reduction," *IEEE Trans. Autom. Contr.*, Vol.50, No.10, 2005.
 [12] K. Hirakawa, T. W. Parks, "Image Denoising Using Total Least Squares," *IEEE Trans. Image Processing*, Vol.15, No.9, 2006
 [13] W. Zhu, et. al., "Regularized Total Least Squares Approach for Nonconvolutional Linear Inverse Problems," *IEEE Trans. Image Processing*, Vol.8, No.11, 1999.
 [14] D.-X. Chang, et. al., "A Fast Recursive Total Least Squares Algorithm for Adaptive IIR Filtering," *IEEE Trans. Signal Processing*, Vol. 53, No.3, 2005.
 [15] L. T. Yang, "Total Least Squares Filter," *IEEE Proc. DSP*,

pp. 741-744, 2002.

- [16] J.-s. Kim, *et. al.*, "A Frequency Domain Identification Method using Total Least Squares," *Proc. ISIE 2001*, Pusan, Korea, 2001.
- [17] V. Z. Mesarovic, *et. al.*, "Regularized Constrained Total Least Squares Image Restoration," *IEEE Trans. Image Processing*, Vol. 4, No.8, 1995.
- [18] K. Gao, *et. al.*, "A Constrained Anti-Hebbian Learning Algorithm for Total Least-Squares Estimation with Applications to Adaptive FIR and IIR Filtering," *IEEE Tran. Circuits and Systems-II; Analog and Digital Signal Processing*, Vol.41, No.11, 1994.
- [19] P. Lemmerling, *et. al.*, "On the Equivalence of Constrained Total Least Squares and Structured Total Least Squares," *IEEE Trans. Signal Processing*, Vol.44, No.11, 1996.
- [20] G. Vandersteen, "On the Use of Compensated Total Least Squares in System Identification," *IEEE Trans. Automat. Contr.*, Vol.43, No.10, 1998.

저자 약력



전창완

- 1986년 서울대학교 제어계측공학과 (공학사).
- 1988년 서울대학교 제어계측공학과(공학석사).
- 1996년 서울대학교 제어계측공학과(공학박사).
- 삼성전자 근무.
- 현재 순천향대학교 전기통신공학과 교수.
- 관심분야는 System Identification, GPS 응용,

BMS.