

# DOA 추정을 위한 Subspace Tracking 기법들 간의 성능 비교

강경훈, 유경렬

한양대학교 전자전기제어계측공학과

## Study of Subspace Tracking Methods for Estimating DOA of Linearly Closely Spaced Time-Varying Signals

Kyoung Hoon Kang, Kyung Yul Yoo

Dept. of Electronic, Electrical, Control and Instrument Engineering, Hanyang Univ.

E-mail : ever0707@hanmail.net, yoo@emc.hanyang.ac.kr

### 요약

센서 array 신호처리에서 DOA(direction of arrival)의 추정에 사용되고 있는 root-MUSIC, TLS-ESPRIT 등과 같은 고해상도 스펙트럼 추정 기법들은 과도한 연산량으로 인하여 실시간 구현이 어렵고, 신호들의 DOA가 근접한 경우에는 추적 성능이 매우 불안정하게 된다. 이러한 문제점에 대한 대안으로 여러 형태의 부공간 추적 개념을 사용하는 수치기법이 제안되어 왔다 [2], [4], [6]. 본 논문에서는 이들 부공간 추정 기법들을 LS-ESPRIT 기법에 접목하여 그 성능을 비교하고, 개선 방안을 제시하였다.

### 1. 서론

Array 신호처리에서 DOA를 추정함에 있어 minimum-norm, ESPRIT, WSF, MUSIC 등의 기법들이 대표적으로 사용되어 왔다. 하지만 이런 고전적인 기법들은 많은 연산량 ( $O(n^3)$ )으로 인해 실시간 구현이 어려움은 물론, 신호원의 주파수들이 근접한 경우 불안정한 추적 성능을 보인다. 문제에 대한 해결책으로 제시되고 있는 것이 부공간 추적 개념을 사용하는 수치기법이다.

공간 입력 데이터의 신호원의 수(=rank)에 대한 사전정보를 갖고 있는 경우, 부공간 추적 알고리즘을 MUSIC이나 ESPRIT 등에 접목 시킴으로써 DOA 추정을 실시간 처리 가능케 할 수 있다. 또한, closed spaced signal들의 DOA의 해상도도 향상시킬 수 있다. Subspace tracking 알고리즘들은 연산량에 따라  $O(n^2r)$  기법,  $O(nr^2)$  기법,  $O(nr)$  기법들로 분류되어질 수 있는데, DOA의 추적 성능은 fast subspace 기법들의 성능을 보여주는 간접적인 잣대가 된다. Fast subspace 기법으로 분류되는 대표적인 알고리즘들은 다음과 같다 [1].

알고리즘	Complexity
KARA	$O(nr^2)$
ROSA	$O(nr)$
PAST	$O(nr)$

※  $n$  = window length,  $r$  = rank

### 2. Problem statement

선형 분포를 갖는 센서 array와 천천히 시변하는 채널을 가정하자. DOA 추정에 있어,  $t$  시각에서  $n$  개의 센서에 입력되는 데이터 벡터는 다음과 같다.

$$\mathbf{x}(t) = [\mathbf{x}_1(t) \ \mathbf{x}_2(t) \ \cdots \ \mathbf{x}_n(t)]^T$$

$$\mathbf{x}(t) = \sum_{i=1}^r s_i \mathbf{a}(w_i) + \mathbf{n}(t) = \mathbf{A}_{nr \times r} \mathbf{s}_{rx1}(t) + \mathbf{n}(t) \quad (1)$$

$\mathbf{A} = [\mathbf{a}(w_1) \ \mathbf{a}(w_2) \ \cdots \ \mathbf{a}(w_r)]$  는 deterministic  $n \times r$  행렬이고,  $\mathbf{a}(w_r) = [1 \ e^{jw_r} \ \cdots \ e^{j(n-1)w_r}]^T$  는 steering 벡터이고,  $\mathbf{s}(t) = [s_1 \ s_2 \ \cdots \ s_r]^T$  는 자기상관 행렬  $\mathbf{C}_s = E[\mathbf{s}(t)\mathbf{s}^H(t)]$  를 갖는 random source 벡터이다.  $w_i$  는  $i$  번째 정현파의 각 주파수이다.  $\mathbf{n}(t)$  는 동일한 분산  $\sigma^2$  를 갖는 공간 도메인에서의 백색 노이즈 벡터이다. 입력 데이터 벡터들의 자기상관 행렬은 다음과 같이 정의된다.

$$\mathbf{C} = E[\mathbf{x}(t)\mathbf{x}^H(t)] = \mathbf{A}\mathbf{C}_s\mathbf{A}^H + \sigma^2\mathbf{I} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^H \quad (2)$$

여기서,  $\mathbf{\Sigma} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  는  $\lambda_1 > \dots > \lambda_{r+1} = \dots = \lambda_n = \sigma^2$  의 관계를 갖는 고유값 행렬이고,  $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_n]$  는 고유 벡터를 갖는 행렬이다.

EVD(eigenvalue decomposition)를 기반으로 하는 root-MUSIC이나 SVD를 기반으로 하는 TLS-ESPRIT와 같은 기법들이 시변 신호들의 DOA를 고해상도로 추적하기 위해서는 많은 수의 sensor를 가정하는데 이는 많은 연산량을 의미한다. 특히, closed spaced signal들의 경우 sensor의 수를 많이 늘리더라도 주파수들이 가까워지는 부근에 가서는 추정능력이 불안정하게 된다. 그림 1, 2는 선형적으로 변화하는 주파수를 갖는 두 신호로 구성된 입력신호에 대하여 TLS-ESPRIT를 사용한 DOA 추정 결과를 보여주고 있다. 두 신호의 주파수가 500 sample 부근에서 겹치고 있으며, TLS-ESPRIT가 이러한 근접 주파수 성분을 추적하지 못하고 있음을 볼 수 있다. 이러한 불안정한 추정은 신호의 고유값 중 하나가 너무 작아져서 노이즈의 고유값에 근접해 가기 때문에 발생한다 [2]. Root-MUSIC를 사용하더라도 TLS-ESPRIT와 결과가 비슷하다.

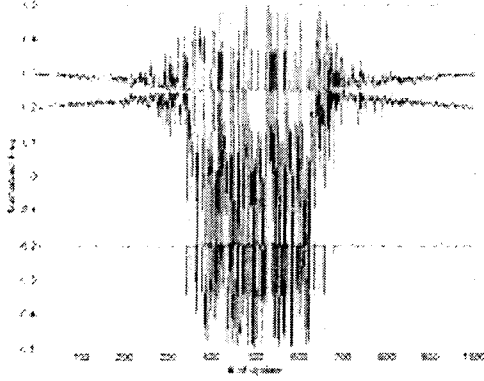


그림1. TLS-ESPRIT을 사용한 DOA 추정 결과,  
sensor=10, 10dB, update=1000

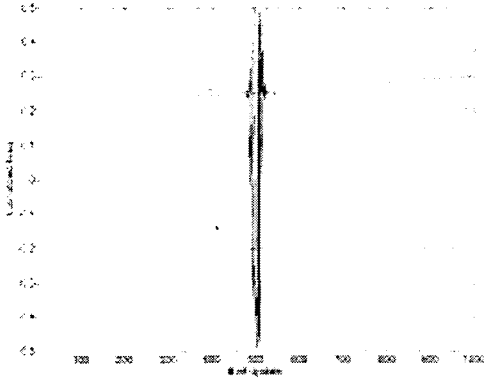


그림2. TLS-ESPRIT, sensor=128, 10dB, update=1000

### 3. ROSANA, ROSA

이 장에서 소개할 ROSANA, ROSA 알고리즘들은 Bunch의 rank-one EVD update 알고리즘을 근간으로 하고 있다 [2], [3]. Bunch 알고리즘은 시변 subspace를 추적하는데 적합하지만, 많은 업데이트를 필요로 하는 환경에서는 round-off error의 누적으로 부적합하다. 이런 단점을 극복하기 위해 DeGroat의 ROSANA와 ROSA 알고리즘이 제안 되었는데, (1) 자기상관 행렬을 forgetting factor를 사용하여 업데이트 함으로써 고유값의 안정화를 이루고, (2) pairwise Gram-Schmidt라는 직교화 과정으로 고유 벡터의 안정화를 갖고 온다. 그 결과 ROSANA와 ROSA 알고리즘들은 수치적 안정성을 갖게 되고, 많은 업데이트 환경에서도 실시간 구현을 가능케 해준다.

DOA를 추정하는 다른 subspace 기법들과는 달리 DeGroat의 알고리즘들은 block processing이 아니다. 어느 시점의 업데이트에서 그 바로 전 시점과의 subspace들 사이에는 연속적인 연관성을 갖는 특성이 있다.

시간  $k$ 에서 시변 자기상관 행렬의 추정자는 다음과 같다.

$$R_k = \alpha R_{k-1} + (1-\alpha)x_k x_k^H \quad (3)$$

$\alpha$ 는 forgetting factor이고,  $R_{k-1} = U_{k-1} D_{k-1} U_{k-1}^H$ 는 시각  $k-1$

에서의 EVD이고,  $R_k = U_k D_k U_k^H$  시각  $k$ 에서 EVD이다.

#### 3.1 ROSANA

$$\begin{aligned} \tilde{R} &= \alpha R_{k-1} + (1-\alpha)x_k x_k^H \\ &= \alpha U_{k-1} D_{k-1} U_{k-1}^H + x_k x_k^H \\ &= U(\alpha D + \bar{\beta} \bar{\beta}^H) U^H, \quad \bar{\beta} = \sqrt{1-\alpha} U^H x \\ &= UG(\alpha D + \bar{\gamma} \bar{\gamma}^H) G^H U^H, \quad \bar{\gamma} = G^H \bar{\beta} \end{aligned} \quad (4)$$

$G = \text{diag}(\beta_1/\beta_1, \dots, \beta_m/\beta_m)$ 는 복소수 행렬  $\alpha D + \bar{\gamma} \bar{\gamma}^H$ 을 실수 행렬로 바꾸는 변환 행렬인데, 연산량을 줄여주는 역할을 한다.

$$\begin{aligned} S &= (\alpha D + \bar{\gamma} \bar{\gamma}^H) \\ &= \left( \alpha \begin{bmatrix} d^{(s)} I_r & 0 \\ 0 & d^{(n)} I_{m-r} \end{bmatrix} + \bar{\gamma} \bar{\gamma}^T \right) \\ &= \begin{bmatrix} H_r^{(s)} & 0 \\ 0 & H_{m-r}^{(n)} \end{bmatrix} \left( \alpha \begin{bmatrix} d^{(s)} I_r & 0 \\ 0 & d^{(n)} I_{m-r} \end{bmatrix} + \bar{\zeta} \bar{\zeta}^T \right) \\ &\quad \begin{bmatrix} H_r^{(s)} & 0 \\ 0 & H_{m-r}^{(n)} \end{bmatrix}^T, \quad \bar{\zeta} = H^T \bar{\gamma} \\ &= \begin{bmatrix} H_r^{(s)} & 0 \\ 0 & H_{m-r}^{(n)} \end{bmatrix} \begin{bmatrix} I_{r-1} & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & I_{m-r-1} \end{bmatrix} \\ &\quad \begin{bmatrix} \alpha d^{(s)} I_{r-1} & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 \\ 0 & 0 & 0 & \alpha d^{(n)} I_{m-r-1} \end{bmatrix} \\ &\quad \begin{bmatrix} I_{r-1} & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & I_{m-r-1} \end{bmatrix}^T \begin{bmatrix} H_r^{(s)} & 0 \\ 0 & H_{m-r}^{(n)} \end{bmatrix}^T \end{aligned} \quad (5)$$

$d^{(s)}$ 와  $d^{(n)}$ 은 신호의 고유값과 노이즈의 고유값을 각각 평균을 취한 것이다. Householder 행렬  $H$ 는 새로운 데이터 벡터  $\bar{\gamma}$ 에 포함된 신호와 노이즈의 모든 전력을 각 subspace의 한 성분으로 집중시켜 주는 역할을 한다. 신호와 노이즈 subspace 업데이트 문제는 다음과 같이  $2 \times 2$  EVD 문제로 축소된다.

$$\left( \begin{bmatrix} d_s & 0 \\ 0 & d_n \end{bmatrix} + \begin{bmatrix} |\gamma^{(s)}| \\ |\gamma^{(n)}| \end{bmatrix} \begin{bmatrix} |\gamma^{(s)}| & |\gamma^{(n)}| \end{bmatrix} \right) = Q \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} Q^T \quad (6)$$

신호와 노이즈의 새로운 subspace는 다음과 같이 재구성되어진다.

$$\begin{aligned} \tilde{U} &= [\tilde{U}^{(s)}, \tilde{U}^{(n)}] \\ &= [U^{(s)} G^{(s)} H^{(s)}, U^{(n)} G^{(n)} H^{(n)}] \begin{bmatrix} I_{r-1} & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & I_{m-r-1} \end{bmatrix} \end{aligned} \quad (7)$$

ROSANA의 경우는 신호의 subspace와 노이즈의 subspace를 동시에 추적한다. 업데이트할 때 마다 각각의 subspace에 해당하는 고유값을 평균을 취함으로써 신호의 고유값의 평균

레벨과 잡음의 고유값의 평균 레벨은 어느 정도 차이를 유지하게 된다. 따라서, 2장에서 서로 다른 주파수가 가까워질 때 DOA의 추적이 실패했던 문제는 더 이상 발생하지 않게 된다. 하지만, ROSANA의 경우 연산량이  $O(n^2 + nr)$ 으로 다소 많다.

### 3.2 ROSA

ROSANA는 신호와 노이즈의 subspace 각각을 모두 추적한다. 그런데, 두 개의 subspace 중 dimension이 작은 subspace만을 추적하고, dimension이 큰 subspace는 필요로 할 때만 작은 subspace를 이용하여 구할 수 있다. 대부분의 경우, 신호의 dimension이 노이즈의 dimension보다 훨씬 작아 많은 연산량의 이득을 보게 된다. ROSA는 신호의 고유값만 평균을 취함으로써  $O(nr)$ 의 연산량을 갖게 된다.

신호의 subspace를 업데이트 하기 위해서는, 신호의 subspace와 하나의 노이즈 벡터만 있으면 된다. 이 노이즈 벡터는 다음과 같이 정의 되어진다.

$$u^{(n)} = \frac{1}{|\gamma^{(n)}|} (x - U^{(s)} \gamma^{(s)}) \quad (8)$$

$\gamma^{(n)}$ 의 구하는데 연산량을 줄이기 위해 아래와 같은 식을 사용한다.

$$|\gamma^{(n)}|^2 = x^T x - |\gamma^{(s)}|^2 \quad (9)$$

ROSA의 전체적인 업데이트 식은 다음과 같은 식으로 재구성된다.

$$\begin{aligned} & [\tilde{U}^{(s)} \quad u^{(n)}]_{m \times (r+1)} \\ & = [U^{(s)} G^{(s)} H^{(s)} \quad u^{(n)}]_{m \times (r+1)} \begin{bmatrix} I_{r-1} & 0 \\ 0 & Q \end{bmatrix}_{(r+1) \times (r+1)} \end{aligned} \quad (10)$$

### 4. PAST

Yang이 제안한 PAST 알고리즘은 data space를 signal subspace로 투영함으로써 'unconstrained minimization problem'을 만들고, 이 문제의 해답을 RLS(recursive least square) 방법으로 구하고 있다 [4].

$x(t)$ 가 자기상관 행렬  $C = E[x(t)x^H(t)]$ 를 갖는 랜덤 벡터일 때, 비용 함수는 다음과 같다.

$$\begin{aligned} J(W(t)) &= \sum_{i=1}^r (\beta)^{t-i} \|x(i) - W(t)W^H(t)x(i)\|^2 \quad (11) \\ &= tr[C(t)] - 2tr[W^H(t)C(t)W(t)] \\ &\quad + tr[W^H(t)C(t)W(t)W^H(t)W(t)] \quad (W \in C^{n \times r} (r < n)) \end{aligned}$$

$\beta (0 < \beta \leq 1)$ 는 시변 환경에서 subspace를 보다 잘 추적하기 위한 'forgetting factor'이다. 시변 환경에서 자기상관 행렬의 업데이트 식은 다음과 같다.

$$C(t) = \sum_{i=1}^r \beta^{t-i} x(i)x^H(i) = \beta C(t-1) + x(t)x^H(t) \quad (12)$$

위의 비용 함수  $J(W(t))$ 를 최소화 하는 행렬  $W(t)$ 는  $C(t)$ 의 signal subspace를 span하는 orthonormal basis가 된다. PAST 알고리즘의 핵심 개념은  $W^H(t)x(i)$ 를  $W^H(t-1)x(i)$ 로

의 근사화이다. 즉,  $x(i)$ 의  $W(t)$ 로의 투영을  $W(t-1)$ 로의 투영으로 근사화하는 것이다. 이렇게 하면, 위에서  $W(t)$ 의 4차 식인 비용함수  $J(W(t))$ 가 2차로 수정되어 질 수 있다.

$$J(W(t)) = \sum_{i=1}^r \beta^{t-i} \|x(i) - W(t)W^H(t-1)x(i)\|^2 \quad (13)$$

이 수정되어진 비용함수는  $J(W(t))$ 의 좋은 근사화가 된다. 수정된 비용함수  $J'(W(t))$ 는 다음과 같은 조건을 갖을 때 최소화 된다.

$$W(t) = C_{xy}(t)C_{yy}^{-1}(t) \quad (14)$$

$$C_{xy}(t) = \sum_{i=1}^r \beta^{t-i} x(i)y^H(i) = \beta C_{xy}(t-1) + x(t)y^H(t)$$

$$C_{yy}(t) = \sum_{i=1}^r \beta^{t-i} y(i)y^H(i) = \beta C_{yy}(t-1) + y(t)y^H(t)$$

좀 더 효율적이고 강한 해법으로는 (1)  $C_{yy}^{-1}(t)$ 를 계산하기 위해 matrix inversion lemma를 적용하거나, (2)  $C_{yy}(t)$ 의 Cholesky factor를 구하기 위해 QR 업데이트 방법을 쓰는 것이다. 그 결과 나타나는 RLS 알고리즘 중 하나를 이용한다.

### 표2. PAST 알고리즘

$$\begin{aligned} & C_{yy}^{-1}(0) = I_r, \\ & \text{columns of } W(0) = r \text{ leading unit vectors of } I_{m \times n} \\ & \text{For } t=1,2,\dots \text{ Do} \\ & \quad y(t) = W^H(t-1)x(t), \quad h(t) = C_{yy}^{-1}(t-1)y(t) \\ & \quad g(t) = \frac{h(t)}{\beta + y^H(t)h(t)}, \\ & \quad C_{yy}^{-1}(t) = \frac{1}{\beta} \text{Tri} \{ C_{yy}^{-1}(t-1) - g(t)h^H(t) \} \\ & \quad e(t) = x(t) - W^H(t-1)y(t), \quad W(t) = W(t-1) + e(t)g^H(t) \end{aligned}$$

### 5 실험 및 고찰

동일한 실험환경으로 선형적으로 분포된 10개(=window의 크기)의 센서를 가정하고, 신호원의 갯수(= rank)는 3으로 하였다. SNR은 10dB로 하였고, 주파수의 가정은 표3과 같다. DOA를 signal subspace에서 추출해내기 위해 사용한 스펙트럼 추정 기법은 LS-ESPRIT이다.

그림3은 다른 fast subspace 알고리즘들의 기준이 되는 Karasalo의 알고리즘을 구현한 것이다 [6]. 이 방법 역시 window의 수를 64로 늘렸음에도 불구하고 업데이트 5000부근에서 DOA를 추적하지 못하고 있다.

ROSANA의 경우, 시변 신호의 DOA의 추적하는데 해상도가 뛰어나고, 알고리즘의 안정적(stable)이다. ROSA의 경우, ROSANA의 경우보다 연산량이 훨씬 적을 뿐더러 해상도와 안정성 역시 뛰어난 것을 모의실험 결과에서 볼 수 있다.

PAST의 경우, 1000번의 업데이트에서  $\beta = 0.97$ 일 때, DOA의 해상도와 안정성이 최적의 결과를 얻을 수 있었다. 업데이트를 10000번으로 늘렸을 때  $\beta = 0.999$ 가 실험적인 최적 조건이 되는데, 그림6에서 볼 수 있듯이 알고리즘의 안정성은 가져올 수 있으나, 업데이트 5000정도에서 만나야 할 주파수가 5300-5400 정도에서 만나고 있다.

표3 각 신호원의 주파수 가정

# of update	frequencies
1000	$f_1 = 0.2 + 0.0001 \times t$ $f_2 = 0.3 - 0.0001 \times t$ $f_3 = -0.2$
10000	$f_1 = 0.2 + 0.00001 \times t$ $f_2 = 0.3 - 0.00001 \times t$ $f_3 = -0.2$

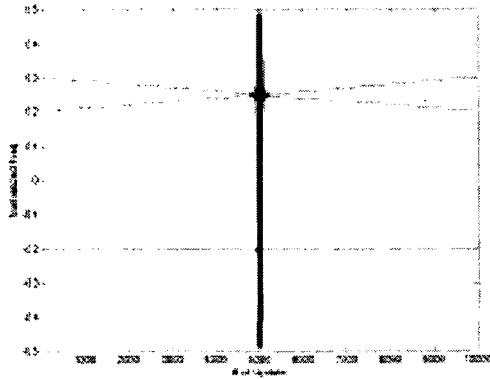


그림3. KARA-ESPRIT, sensor=64, 10dB, update=10000

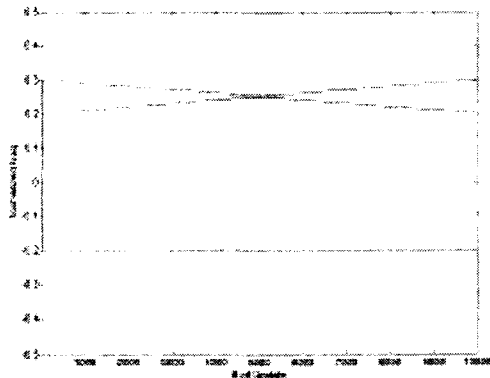


그림4. ROSANA-ESPRIT, sensor=10, 10dB, update=10000,  $\alpha = 0.95$

## 6. 결론

Fast subspace 기법들의 응용분야가 무수히 많고, 그 응용분야에 따라 성능의 우위가 다르게 나타나서 하나의 잣대로 비교하기가 대단히 어렵다. 본 논문에서는 동일한 모의실험 환경을 제공함으로써 fast subspace 기법들의 성능을 간접 비교 하였다. 여러 subspace 기법들 중 연산량에 있어서는 ROSA와 PAST가 가장 좋고, 많은 업데이트시에 안정성과 해상도를 고려할 때 PAST보다는 ROSA가 우위에 있다고 말할 수 있다.

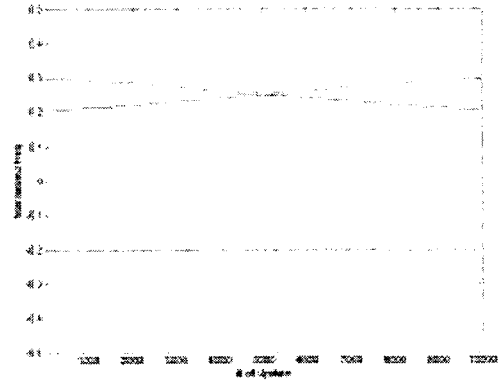


그림5. ROSA-ESPRIT, sensor=10, 10dB, update=10000,  $\alpha = 0.85$

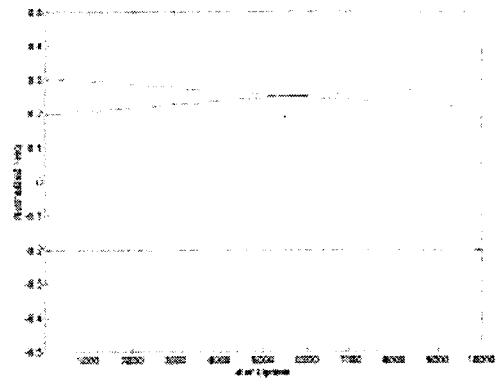


그림6. PAST-ESPRIT, sensor=10, 10dB, update=10000,  $\beta = 0.999$

## [참고문헌]

- [1] C. E. Davila, "Efficient, high performance, subspace tracking for time-domain data," IEEE Trans. Signal Processing, vol. 48, pp. 3307-3315, Dec. 2000.
- [2] R. D. DeGroat, "Noniterative subspace tracking," IEEE Trans. Signal processing, vol. 40, no. 3, pp. 571-577, Mar. 1992.
- [3] R. D. DeGroat and R. A. Roberts, "Efficient, numerically stabilized rank-one eigenstructure updating," IEEE Trans. Acoust., Speech, Signal Processing, vol. 38, no. 2, pp. 301-316, Feb. 1990.
- [4] B. Yang, "Projection approximation subspace tracking," IEEE Trans Signal Processing, vol. 43, pp.95-107, Jan. 1995.
- [5] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," IEEE Signal Processing Letters, vol. 7, no. 3, Mar. 2000.
- [6] P. Strobach, "Bi-iteration SVD subspace tracking algorithms," IEEE Trans. Signal Processing, vol. 45, pp.1222-1240, May. 1997.
- [7] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, "Statistical and adaptive signal processing," McGraw-Hill, 2000