

적응형 알고리즘을 통한 재머 억제 빔형성

오관진*, 김준호**, 천창율*, 정용식**
 서울시립대*, 광운대**

Adaptive algorithm beamforming for jammer suppression

Kwan-Jin Oh*, Jun-ho Kim**, Chang-Yul Cheon*, Yong-Seek Jeong**
 University of Seoul*, KwangWoon University**

Abstract - 본 논문은 적응형 알고리즘(Adaptive Algorithm)을 이용하여 재머(Jammer)를 억제하는 것을 기존의 알고리즘을 시뮬레이션으로 확인한다. 본 논문에서는 원하는 방향으로 빔을 형성하는 동시에 특정방향 주로 부엽(sidelobe)으로 들어오는 재머의 신호를 억제시키는데 있어 여러 가지 적응형 알고리즘을 확인하고, 알고리즘을 이용하여 부엽으로 들어오는 재머신호를 어떻게 억제하는지 시뮬레이션을 통해 결과를 확인하도록 할 것이다.

1. 서 론

빔 형성(beamforming)이라는 말은 안테나에서 방사된 에너지가 공간에서 특정한 방향을 따라 집중되는 장치 또는 장비들에 의해 수행되는 기능과 관련이 있다. 원하는 사용자 방향으로의 빔형성을 위한 방법으로 스위칭형 빔형성(Switched Beamforming) 방식과 적응형 빔형성(Adaptive Beamforming) 방식이 있다. 스위칭형 빔형성 방식은 conventional beamforming 또는 fixed beamforming이라고도 불리며, 이는 미리 몇 개의 방향에 대한 가중치 벡터를 설정하여 빔형성을 하는 방법이다. 적응형 빔형성 방식은 원하는 사용자 신호 대 간섭 신호의 비를 최대화하도록 원하는 사용자의 위치에 따라서 가중치 벡터를 계속 갱신하는 방법이다. 빔 형성을 위해 Weight 벡터를 계산하기 위해 다양한 적응형 알고리즘(Adaptive Algorithm)이 사용된다.

2. 본 론

2.1 적응형 알고리즘(Adaptive Algorithm)

적응형 빔형성 방식에서는 Weight 벡터를 계산하기 위해 다양한 적응형 알고리즘이 사용된다. 본 연구내용에서는 원하는 방향으로 빔을 형성하는 동시에 특정방향에서 오는 방해전파(Jammer)를 Suppression하는데 사용되는 Algorithm은 LMS(Least mean square), SMI(Sample Matrix Inversion), LSMI(Loaded Sample Matrix Inversion), RLS(Recursive Least Square)가 있다. 이 중 주로 사용되는 Algorithm인 LMS, SMI, LSMI에 대해 다루고자 한다.

2.2 LMS Algorithm

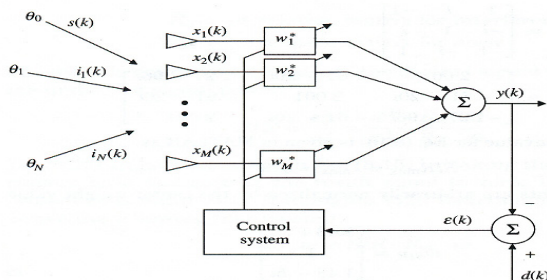
LMS 알고리즘은 각 배열소자에서의 출력과 원하는 패턴의 오차를 최소화 하는 방향의 가중치를 구해줌으로써 평균제곱오차(MSE : mean squared error)를 최소화시키는 방법이다.

오차에 관한 수식은 다음과 같다.

$$e(k) = d(k) - \mathbf{w}^H(k)\mathbf{x}(k)$$

제곱오차에 관한 식은

$$|e(k)|^2 = |d(k) - \mathbf{w}^H(k)\mathbf{x}(k)|^2$$



<그림 1> MSE adaptive system

위 식을 보다 간단히 하기 위한 목적으로 time dependence parameter 'k'를 없앨 것이다. 위의 식에서 양변에 기대값을 취함으로써 다음과 같은

식(cost function)을 얻을 수 있다.

$$E[|e|^2] = \mathcal{J}(\mathbf{w}) = D - 2\mathbf{w}^H \mathbf{r} + \mathbf{w}^H \mathbf{Q}_{xx} \mathbf{w}$$

이 식에서 D는 아래와 같다.

$$D = E[d^2]$$

위 식 $\mathcal{J}(\mathbf{w})$ 에 gradient method를 적용하여 최소가 되는 값을 찾아내어 \mathbf{w} 를 찾아낼 수 있다. gradient값이 0이 될 때 최소가 되므로 최적화된 \mathbf{w}_{opt} 은 다음과 같다.

$$\mathbf{w}_{opt} = \mathbf{Q}_{xx}^{-1} \mathbf{r}$$

이 식으로부터 만약 신호와 재머에 대한 정보를 안다면 공분산행렬(covariance matrix) \mathbf{Q}_{xx} 를 구하여 모든 신호에 대한 정보를 알 수 있게 되어 최적화된 가중치 값을 계산할 수 있게 된다. 하지만, 일반적으로 우리가 재머에 대한 정보를 알 수 없으므로 재머가 들어오는 각각의 시간(snapshots)내에서 즉시 공분산행렬과 상관행렬(correlation matrix)을 추정해야 한다. 이를 추정하는 식은 다음과 같다.

$$\hat{\mathbf{Q}}_{xx}(k) \approx \mathbf{x}(k)\mathbf{x}^H \text{ and } \hat{\mathbf{r}}(k) \approx \mathbf{d}^*(k)\mathbf{x}(k)$$

여기서 위에서 구한 cost function의 gradient값을 추정하기 위해 반복적으로 값을 추정하게 되는 steepest descent method를 적용할 것이다. steepest descent의 방향은 gradient vector로써 반대방향에 있다. steepest descent iterative approximation을 적용하게 되면 최종 식은 다음과 같이 유도된다.

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \frac{1}{2} \mu \nabla_{\mathbf{w}} (\mathcal{J}(\mathbf{w}(k))) \\ &= \mathbf{w}(k) - \mu [\hat{\mathbf{Q}}_{xx} \mathbf{w} - \hat{\mathbf{r}}] \\ &= \mathbf{w}(k) + \mu \mathbf{e}^*(k)\mathbf{x}(k) \end{aligned}$$

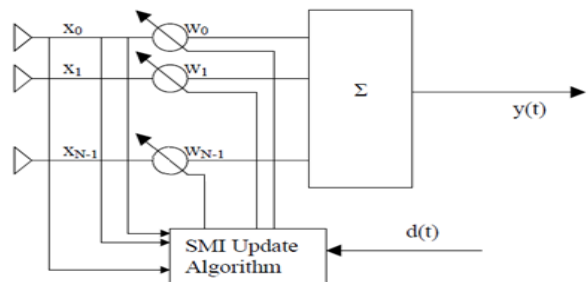
위의 식에서 $e(k)$ 는 다음과 같다.

$$e(k) = d(k) - \mathbf{w}^H(k)\mathbf{x}(k) = \text{error signal}$$

2.3 SMI Algorithm

LMS 알고리즘의 단점 중 하나는 만족하는 수렴을 얻기 전에 많은 Iteration을 거쳐야 된다는 점이다. 만약 들어오는 신호 특성이 급격하게 변하면, LMS 알고리즘은 원하는 신호의 패턴을 추적하는 것이 힘들게 된다. 이런 LMS 알고리즘의 문제점을 해결하기 위한 방법이 SMI 알고리즘이다. SMI 알고리즘은 공분산행렬로부터 직접 역행렬을 구하기 때문에 Direct matrix Inversion이라고 불리기도 한다. SMI 알고리즘은 앞 절의 gradient method를 이용해 가중치를 구하는 식에서부터 출발한다.

$$\mathbf{w}_{opt} = \mathbf{Q}_{xx}^{-1} \mathbf{r}$$



<그림 2> SMI adaptive system

위의 식에서 각각의 항은 아래와 같다.

$$\begin{aligned} \mathbf{r} &= E[\mathbf{d}^* \cdot \mathbf{x}] \\ \mathbf{Q}_{xx} &= E[\mathbf{x}\mathbf{x}^H] \end{aligned}$$

\mathbf{Q}_{xx} 는 공분산행렬로써 \mathbf{x} 에 신호와 재머와 잡음(noise)에 관한 정보가

있다면 그 식으로부터 아래와 같이 구할 수 있게 된다. 여기서 K(Snapshots)는 observation time interval이라고 보면 된다. \mathbf{r} 은 상관행렬로써 빔의 조향 각도와 관련이 있다.

$$\hat{Q}_{xx} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}(k) \mathbf{x}^H(k) \quad \hat{\mathbf{r}} = \frac{1}{K} \sum_{k=1}^K d^*(k) \mathbf{x}(k)$$

위 식에서 K 개의 스냅샷을 사용하기 때문에 가중치 구하는 식을 보다 간단하게 나타낼 수 있다.

$\mathbf{X}_K(K)$ 는 K개의 스냅샷에 관한 \mathbf{x} 벡터로 다음과 같이 나타낼 수 있다.

$$\mathbf{X}_K(K) = \begin{bmatrix} x_1(1+kK) & x_1(2+kK) & \dots & x_1(K+kK) \\ x_2(1+kK) & x_2(2+kK) & \dots & x_2(K+kK) \\ \vdots & \vdots & \ddots & \vdots \\ x_M(1+kK) & \dots & \dots & x_M(K+kK) \end{bmatrix}$$

따라서 공분산행렬의 추정값은 다음과 같이 쓸 수 있다.

$$\hat{Q}_{xx}(K) = \frac{1}{K} \mathbf{X}_K(K) \mathbf{X}_K^H(K)$$

수신하고자 하는 신호는 다음과 같이 정의 된다.

$$\mathbf{d}(k) = [d(1+kK) \ d(2+kK) \ \dots \ d(K+kK)]$$

따라서 상관행렬의 추정값도 다음과 같이 쓸 수 있다.

$$\hat{\mathbf{r}}(K) = \frac{1}{K} \mathbf{d}^*(K) \mathbf{X}_K(K)$$

SMI 알고리즘을 이용한 가중치 값은 아래의 식과 같이 계산된다.

$$\begin{aligned} \mathbf{w}_{SMI}(k) &= \hat{Q}_{xx}^{-1} \hat{\mathbf{r}}(k) \\ &= [\mathbf{X}_K(K) \mathbf{X}_K^H(K)]^{-1} \mathbf{d}^*(K) \mathbf{X}_K(K) \end{aligned}$$

2.4 LSMI Algorithm

SMI의 문제점중 하나는 공분산행렬을 추정함에 있어 \hat{Q}_{xx} 의 작은 고유값이 잡음 때문에 상대적으로 불확실하게 추정된다는 것이다. 따라서 작은 고유값은 최적의 가중치를 구하는데 있어 공분산행렬의 역행렬을 계산되게 되면 커지기 때문에 가중치 값을 불안정하게 만들게 된다. 또 이것은 적응 후의 빔 패턴에서 SLL(Side Lobe Level)을 증가시키고, SNIR을 불안정하게 하는 원인이 된다.

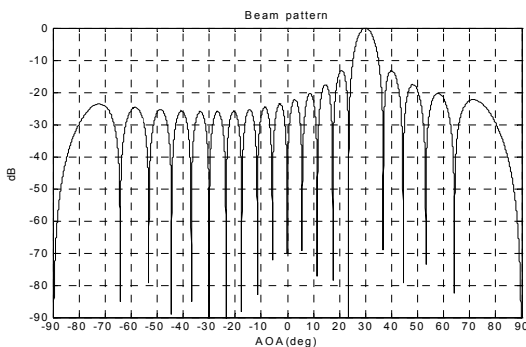
이러한 SMI 알고리즘의 불안정한 면의 단점을 보완하기 위한 방법이 LSMI 알고리즘이다. LSMI 알고리즘의 공분산 행렬을 구하는 식은 다음과 같다.

$$\hat{Q}_{xx}(K) = \frac{1}{K} \mathbf{X}_K(K) \mathbf{X}_K^H(K) + \alpha \mathbf{I}$$

이 식을 통해 공분산행렬을 구하게 되면 상대적으로 공분산행렬의 불안정한 작은 고유값의 수를 줄일 수 있게 된다. 또한, SMI 알고리즘은 취해야 하는 스냅샷의 개수가 배열소자의 개수에 의존하지만 LSMI 알고리즘은 제머의 개수에 의존한다. 따라서 스냅샷의 개수가 훨씬 줄어들 수 있어 보다 빠르게 공분산행렬을 구할 수 있게 된다. 위의 식에서 변수 α 값은 미지의 신호와 제머에 따라 달라진다. 따라서 실제 적용에 있어 적절한 α 값을 선택하는 것이 중요하다. 일반적으로 α 값은 rule of thumb에 의해 $\alpha = 2\sigma^2, \dots, 4\sigma^2$ 안에서 정하게 된다. (σ^2 는 수신 잡음의 분산)

2.5 적응형 알고리즘을 이용한 패턴 시뮬레이션

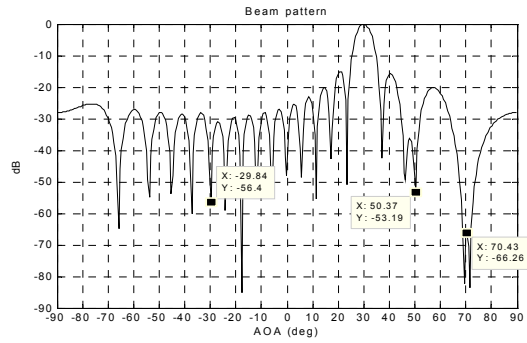
3가지 적응형 알고리즘 LMS, SMI, LSMI를 이용하여 제머의 방향에 따른 패턴을 비교해보았다. 일반적인 Linear array에서의 패턴과 적응형 알고리즘을 이용한 패턴과 비교해서 보도록 한다. 제머의 방향은 모두 동일하게 $-30, 50, 70$ 도로 놓고 시뮬레이션을 했다.



<그림 3> 선형배열(N=20, $d = \lambda/2$), 수신신호(30도)

2.5.1 LMS Algorithm을 이용한 시뮬레이션

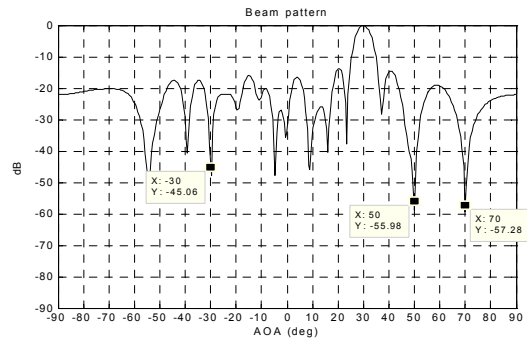
수신하고자 하는 신호의 입사각은 30° 로 설정하고 제머의 방향 따른 빔 패턴을 보도록 한다.



<그림 4> 선형배열(N=10, $d = \lambda/2$), 제머신호(3개, $-30, 50, 70$ 도)

2.5.2 SMI Algorithm을 이용한 시뮬레이션

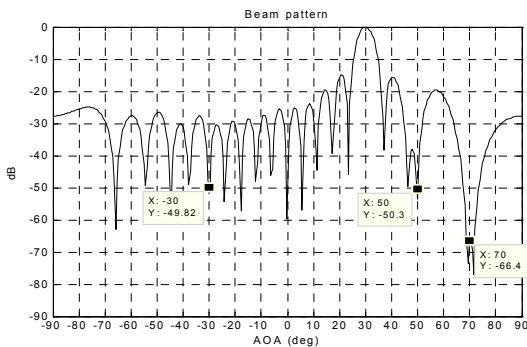
스냅샷의 개수는 100개로 설정하였다.



<그림 5> 선형배열(N=20, $d = \lambda/2$), 제머신호(3개, $-30, 50, 70$ 도)

2.5.3 LSMI Algorithm을 이용한 시뮬레이션

스냅샷의 개수는 10개, α 값은 0.04로 설정하였다.



<그림 6> 선형배열(N=20, $d = \lambda/2$), 제머신호(3개, $-30, 50, 70$ 도)

3. 결 론

LMS 알고리즘은 간단한 알고리즘으로 구현이 단순하다는 것이 장점이다. 또 어떤 신호조건에서도 안정적이고 좋은 성능을 나타낸다. 하지만 가장 큰 문제점은 수렴속도가 느리다는 점이다. 최적화된 가중치 값으로의 수렴속도가 느리다는 것은 실제 적용 시에 가장 큰 문제점이 될 수 있다. SMI 알고리즘의 장점은 LMS 알고리즘보다 빠른 수렴속도를 가진다는 점이다. 하지만 공분산행렬의 규모가 커지면 역행렬을 계산하는데 있어서 시간이 오래 걸리고 복잡해진다는 것이 단점이다. LSMI 알고리즘은 SMI 알고리즘보다 적은 스냅샷의 개수를 통해 공분산행렬을 추정하여 안정된 가중치 값과 패턴을 구할 수 있다. 하지만 SMI 알고리즘과 마찬가지로 SLL의 측면에서 봤을 때는 좋지 않다.

[참 고 문 헌]

- [1] 신성문, 방송찬, "스마트 안테나 기술 개발 동향", 주간기술동향 통권 1093호, pp16-18, 2003.
- [2] Frank Gross, "Smart Antenna for Wireless Communications", McGraw-Hill, pp227-234, 2005.
- [3] Wulf-Dieter Wirth, "Radar techniques using array antennas", The Institution of Electrical Engineers(IEE), 2001.
- [4] Monzingo, R. and T. Miller, "Introduction to adaptive arrays", Scitech, 2004.