

# 효율적 계산량을 가지는 QRD-LS 적응 알고리즘

조해성\*, 조주필\*\* 정회원

## QRD-LS Adaptive Algorithm with Efficient Computational Complexity

Hae-Seong Cho\*, Ju-Phil Cho\*\* *Regular Members*

### 요약

본 논문은 계산량이  $O(N)$ 인 새로운 형태의 QRD-LS 적응 알고리즘을 제안한다. 제안한 알고리즘의 주요 사항은 입력벡터의 단위 벡터 계산이 Given Rotation 과정에서 이루어짐에 근거하고 있다. 알고리즘의 성능 평가는 FIR 시스템 식별 문제를 컴퓨터 시뮬레이션을 통하여 수행하였다. 이 알고리즘은 시뮬레이션의 결과 좋은 성능을 나타내었다. 그리고 이론적으로 평균 측면에서 알고리즘이 최적 계수 벡터에 수렴함을 보였다

### ABSTRACT

This paper proposes a new QRD-LS adaptive algorithm with computational complexity of  $O(N)$ . The main idea of proposed algorithm(D-QR-RLS) is based on the fact that the computation for the unit vector of is made from the process during Givens Rotation. The performance of the algorithm is evaluated through computer simulation of FIR system identification problem. As verified by simulation results, this algorithm exhibits a good performance. And, we can see the proposed algorithm converges to optimal coefficient vector theoretically.

Key Words : Adaptive filtering; Givens Rotation; QR decomposition.

### I. 서론

적응필터의 파라미터 추출방법은 지속적으로 연구되어왔으며, 필터링, 제어, 스펙트럼 분석등에 다양하게 응용되어 왔다. 파라미터 추출에는 계산량 때문에 주로 LMS 계열의 알고리즘이 많이 사용되었지만, LMS 계열의 알고리즘은 계산량이  $O(N^2)$ 인 LS 계열 알고리즘에 비하면 그 계산 결과 및 추적 능력 등에서 성능이 좋지 못하다. 반면에 LMS 계열의 알고리즘의 적응상수를 적절히 조절하여 수렴특성과 추적능력을 극대화 시키려는 연구들도 있었다. 많은 연구자들이 계산량  $O(N)$ 인 LS 계열 알고리즘을 연구하여 고속 LS 계열의 알고리즘들을 발표하였다[1,2]. 그러나 고속 알고리즘들은 알려진 바와 같이 수치적 안정도 때문에 효율적이지 못하였다[3,4]. 수치적 안정도 문제로 안정도가 좋은 QR 분해를 LS 계열 알고리즘에 적용한 연구자들의 결과가 발표되었다. 그러나 QR분해에 근거한 LS 알고리즘 역시 계

산량은  $O(N^2)$ 이어서 이것 역시 계산량을 줄이려는 연구들이 있었다[5,6].

QR을 적용한 논문들은 입력 벡터의 요소들에 대하여 Givens 회전 또는 Householder 변환을 수행하는 하는데 계산량은  $O(N^2)$ 이다. 계수벡터들은 직접적으로 구하지 않고 후치대입법(back substitution)을 적용하여 구한다. 이 과정의 계산량도  $O(N^2)$ 이어서 일반적인 QR 분해법을 이용한 알고리즘은 계산량이  $O(N^2)$ 이다. 입력 벡터들의 시간천이성 성질을 이용하면 QR 분해에서 생성되는 삼각행렬의 요소를 순환시키는데 계산량을 절약할 수 있다. 이 성질을 이용한 알고리즘은  $O(N)$ 을 가지고 있다. 이것은 일부의 오차만을 관심을 갖는 분야에서는 효율적인 알고리즘이 될 수 있다 [7]. 참고문헌[7-9]에서 QR 분해에서 생성되는 상 삼각행렬  $R$ 을 근사화 가정을 통하여 주 대각 요소만으로 대각화하는 근사법을 사용하여  $O(N)$ 인 알고리즘을 제안하고 있다. 이 알고리즘의 유도는 Householder 변환법과 Givens

\* 건양대학교 전자정보공학과, \*\* 군산대학교 전파공학과(stefano@kunsan.ac.kr, Corresponding Author)  
접수일자 : 2010년 5월 6일, 수정완료일자 : 2010년 6월 7일, 최종게재확정일자 : 2010년 6월 25일

회전법을 적용한 예가 나타나있다.

본 논문에서도 위의 근사화 과정을 사용한다. 2.2절에서 이 근사화를 이용하여 Givens 회전 할 때 발생하는 sine 파라미터와 cosine 파라미터의 값을 이용한 D-QR-RLS 알고리즘을 제안한다. D-QR-RLS 알고리즘의 계산량은  $O(N)$ 이다.

2.3절에서는 제안된 알고리즘의 성능을 평가하기 위하여 D-QR-RLS의 안정도와 수렴특성에 대하여 고찰한다. III절에서 컴퓨터 모의실험은 고정 FIR 시스템과 시변 계수를 갖는 FIR 시스템의 시스템 식별문제에 D-QR-RLS 알고리즘을 적용 그 성능을 평가하였다. 마지막을 결론 및 고찰을 수록하였다.

## II. 주요 알고리즘

### 2.1 QRD-RLS

입력  $\mathbf{x}(n)$ 에 대하여 출력과 원하는 응답의 차이를 오차로 정의한다.

$$e(n) = d(n) - \mathbf{x}_M^T(n) \mathbf{w}(n) \quad (1)$$

최소자승 면에서 위의 오차를 최소화하기 위해서 다음의 비용함수를 최소화해야한다.

$$\xi(n) = \sum_{i=0}^n \lambda^{n-i} |e(i)|^2 \quad (2)$$

여기서  $\lambda$ 는 망각계수(forgetting factor)로  $0 < \lambda \leq 1$ 로 정의된다.

식 (2)를 다음과 같은 행렬 형태로 쓸 수 있다.

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{A}_M(n) \mathbf{w}(n) \quad (3)$$

$$\xi(n) = \|\mathbf{Q}(n) \Lambda^{1/2}(n) \mathbf{e}(n)\|^2 \quad (4)$$

$\mathbf{Q}(n) \Lambda^{1/2}(n) \mathbf{e}(n)$ 은 다음과 같이 쓸 수 있다.

$$\mathbf{Q}(n) \Lambda^{1/2}(n) \mathbf{e}(n) = \begin{matrix} \mathbf{Q}(n) \Lambda^{1/2}(n) \mathbf{d}(n) \\ - \mathbf{Q}(n) \Lambda^{1/2}(n) \mathbf{A}_M(n) \mathbf{w}(n) \end{matrix} \quad (5)$$

여기서  $\mathbf{Q}(n)$ 은 직교행렬이다.

식 (6)에서  $\mathbf{x}_M^H(n)$ 을 Givens 회전시키기 위해 M개의 회전행렬이 필요하다.

$$\mathbf{Q}_M \cdots \mathbf{Q}_1 \begin{bmatrix} \lambda^{1/2} \mathbf{R}(n-1) \\ \mathbf{x}_M^H(n) \end{bmatrix} = \begin{bmatrix} \mathbf{R}(n) \\ \mathbf{O} \end{bmatrix} \quad (6)$$

$$\mathbf{Q}_M \cdots \mathbf{Q}_1 \begin{bmatrix} \lambda^{1/2} \mathbf{p}(n-1) \\ d^*(n) \end{bmatrix} = \begin{bmatrix} \mathbf{p}(n) \\ e(n) \end{bmatrix} \quad (7)$$

위의 식에서 계수벡터는 다음식으로 구할 수 있다.

$$\mathbf{w}_R(n) = \mathbf{R}^{-1}(n) \mathbf{p}(n) \quad (8)$$

계수벡터의 계산은 back substitution으로 구하거나 역

QR(inverse QR) 방법으로 구할 수 있다.

### 2.2 D-QR-RLS 알고리즘

고속 D-QR-RLS 방법을 유도한 방법은 참고문헌 [7-9]과 유사하다. 위의 식 (6)과 식 (7)를 하나의 방정식으로 표현하자.

$$\mathbf{Q}(n) \begin{bmatrix} \lambda^{1/2} \mathbf{R}(n-1) & | & \lambda^{1/2} \mathbf{p}(n-1) \\ \mathbf{x}_M^H(n) & | & d(n) \end{bmatrix} = \begin{bmatrix} \mathbf{R}(n) & | & \mathbf{p}(n) \\ \mathbf{O} & | & e(n) \end{bmatrix} \quad (9)$$

식 (9)로부터 다음과 같이 계수벡터를 구할 수 있다.

$$\mathbf{R}(n) \widehat{\mathbf{w}}_R(n) = \mathbf{p}(n) \quad (10)$$

상 삼각 행렬  $\mathbf{R}(n)$ 과  $\widehat{\mathbf{w}}_R(n)$ 을 행렬 곱셈한 후

$\mathbf{R}(n)$ 의 주 대각 요소 항으로 정리하면 다음과 같은 식을 얻을 수 있다. 이것은 back substitution과 같다.

$$\begin{aligned} r_{1,1}(n)w_1 &= p_1 - r_{1,2}(n)w_2 - \cdots - r_{1,M}(n)w_M \\ r_{2,2}(n)w_2 &= p_2(n) - r_{2,3}(n)w_3 - \cdots - r_{2,M}(n)w_M \\ r_{3,3}(n)w_3 &= p_3(n) - \cdots - r_{3,M}(n)w_M \\ &\vdots \\ r_{M,M}(n)w_M &= p_M(n) \end{aligned} \quad (11)$$

본 논문에서는 위의 방법을 방법과 다른 방법을 사용하여 계수벡터의 순환을 얻는다. 순환과정은 다음과 같다.

$$r_{i,i}(n) = \sqrt{[\lambda^{1/2} r_{i,i}(n-1)]^2 + [x_i^{(i-1)}(n)]^2} \quad (12)$$

$$c_i(n) = \frac{\lambda^{1/2} r_{i,i}(n-1)}{r_{i,i}(n)} \quad (13)$$

$$s_i(n) = \frac{x_i^{(i-1)}(n)}{r_{i,i}(n)} \quad (14)$$

본 논문의 핵심은 식 (12), 식 (13) 및 식 (14)이다.

위의 과정에서 식 (12)는 벡터  $[\lambda^{1/2} r_{i,i}(n-1), x_i^{(i-1)}(n)]$ 의 크기를 나타내고 있다.

$c_i(n)$ 은  $\lambda^{1/2} r_{i,i}(n-1)$ 의 단위벡터를 나타내고  $s_i(n)$ 은  $x_i^{(i-1)}(n)$ 의 단위 벡터를 나타낸다.  $s_i(n)$ 은 입력 신호 벡터들의 단위벡터를 생성하게 된다. 그러므로 계수 벡터는 간단히 오차를  $s_i(n)$ 에 투영(projection) 시킴으로 구할 수 있다. 계수벡터의 순환식은 LMS 형태로 다음과 같이 구한다.

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mathbf{s}(n) a_{(N)}(n) \tilde{e}(n) \quad (15)$$

표 1. 제안된 D-QR-RLS 알고리즘

초기값	$a_{(0)} = 1.0, r_{i,i}(0) = \delta$
	for $n=1, 2, 3, \dots, Ndata$
	$\tilde{e}(n) = d(n) - \mathbf{x}^H(n) \mathbf{w}(n-1)$
	for $i=1, 2, 3, 4, \dots, N$

$$r_{i,i}(n) = \sqrt{\lambda [r_{i,i}(n-1)]^2 + [a_{(i-1)} x_i(n)]^2}$$

$$c_{i,i}(n) = \frac{\lambda^{1/2} r_{i,i}(n-1)}{r_{i,i}(n)}$$

$$s_i(n) = \frac{a_{(i-1)}(n) x_i(n)}{r_{i,i}(n)}$$

$$a_{(i)}(n) = c_{i,i}(n) a_{(i-1)}(n)$$

end %i loop  
for i=1,2,3, ...,N

$$w_i(n+1) = w_i(n) + a_{(N)}(n) s_i(n) \tilde{e}(n)$$

end %i loop  
end %n loop

알고리즘의 순환과정에서 사전 예측 오차(priori estimation error)가 얻어지므로 사후 예측 오차는 다음과 같이 구한다.

$$a_{(N)}(n) = \prod_{i=1}^N c_{i,i}(n) \quad (16)$$

이상의 결과를 가지고 표 1.의 D-QR-LS 알고리즘을 제안한다. 연산량은 곱셈 7N, 나눗셈 2N 및 제곱근 N번이다. 즉  $O(N)$ 인 알고리즘이다.

### 2.3 D-QR-RLS 알고리즘의 안정도 고찰

D-QR-RLS 알고리즘이 평균면에서 최적 계수벡터에 수렴하는가를 판단하여 보자. 먼저 계수벡터를 정리하자.

$$w_i(n+1) = w_i(n) + \frac{a_{(N)}(n) a_{(i-1)}(n)}{r_{i,i}(n)} x_i(n) \tilde{e}(n)$$

$$= w_i(n) + M_i(n) x_i(n) \tilde{e}(n) \quad (17)$$

최적 계수벡터  $w_o^i$ 와 예측된  $w_i(n)$ 과의 계수 오차 벡터를 다음과 같이 정의하자.

$$\varepsilon_i(n) = w_o^i - w_i(n) \quad (18)$$

원하는 응답은 다음과 같다.

$$d(n) = \mathbf{w}_o^H \mathbf{x}(n) + v(n) \quad (19)$$

여기서  $\mathbf{w}_o$ 은 미지의 시스템 파라미터이고  $v(n)$ 은 부가잡음이다. 그러면 식 (17), 식 (18) 및 식 (19)에서 다음의 식을 얻을 수 있다.

$$\varepsilon_i(n) = \varepsilon_i(n-1) - M_i(n) x_i(n) \tilde{e}(n) \quad (20)$$

여기서 오차는 다음과 같다.

$$\tilde{e}(n) = \mathbf{x}^H(n) \boldsymbol{\varepsilon}(n-1) + v(n) \quad (21)$$

식 (20)에 식 (21)을 대입 정리한 후 기대값을 취하면 다음과 같다.

$$E[\varepsilon_i(n)] = E[ [1 - M_i(n) x_i(n) \mathbf{x}^H(n)] \varepsilon_i(n-1) - E[M_i(n) x_i(n) v(n)] ] \quad (22)$$

여기서  $\mathbf{x}(n)$ ,  $v(n)$ 이 상관관계가 없다고 가정하면 다음의 수식을 얻는다.

$$E[\varepsilon_i(n)] = E[ [1 - M_i(n) x_i(n) \mathbf{x}^H(n)] \varepsilon_i(n-1) ] \approx [1 - E[M_i(n)] E[x_i(n) \mathbf{x}^H(n)]] E[\varepsilon_i(n-1)] \quad (23)$$

$M(n)$ 의  $i$ 번째 요소를 먼저 계산하자.

$$E[M_i(n)] \approx \frac{E[a_{(N)}(n) a_{(i-1)}(n)]}{E[r_{i,i}(n)]} \quad (24)$$

$$E[a_{(N)}] = \prod_{k=1}^N E[c_{k,k}(n)] = \lambda^{N/2} \quad (25)$$

$$E[c_{i,i}(n)] = \lambda^{1/2} \frac{E[r_{i,i}(n-1)]}{E[r_{i,i}(n)]} \approx \lambda^{1/2} \quad (26)$$

$$E[r_{i,i}^2(n)] = \frac{\lambda \sigma_{x_i}^2}{(1-\lambda)} \quad (27)$$

식 (24)의 분자항을 정리하면 다음과 같다.

$$E[a_{(N)}(n) a_{(i-1)}(n)] \approx \lambda^{(N+i)/2} \quad (28)$$

식 (28)과 식 (29)를 식 (26)에 대입하면 다음 식을 얻는다.

$$E[M_i(n)] \approx \frac{\lambda^{(N+i)/2}}{\sqrt{\frac{\lambda \sigma_{x_i}^2}{1-\lambda}}} \approx \frac{\beta_i}{\sqrt{\sigma_{x_i}^2}} \quad (29)$$

여기서  $\beta_i$ 는 다음과 같다.

$$\beta_i = \lambda^{N/2} \sqrt{1-\lambda} \quad (30)$$

$\beta_i$ 는 망각계수(forgetting factor)만으로 이루어진 함수이다.

식(21)에서  $E[x_i(n) \mathbf{x}^H(n)] \approx E[x_i^2(n)]$ 이라 가정하자. 그리고 다음을 계산하자.

$$1 - E[M_i(n)] E[x_i(n) \mathbf{x}^H(n)] \approx 1 - E[M_i(n)] E[x_i^2(n)] \approx 1 - \beta_i \sqrt{\sigma_{x_i}^2} \quad (31)$$

그리고  $1 - \beta_i \sqrt{\sigma_{x_i}^2}$ 는 1보다는 작고 1에 근접한 양의 실수를 가지게 된다. 또한 식 (31)은 입력 신호의 분산  $\sqrt{\sigma_{x_i}^2}$ 에 비례하게 된다. 계수 오차 벡터는 다음과 같이 된다.

$$E[\varepsilon_i(n)] = (1 - \beta_i \sqrt{\sigma_{x_i}^2})^n E[\varepsilon_i(0)] \quad (32)$$

전체적으로 평균면에서 다음과 같이 최적계수 벡터에 수렴할 것이다.

$$E[\varepsilon_i(n)] = (1 - \beta \sqrt{\sigma_{x_i}^2})^n E[\varepsilon_i(0)] \text{ for } i=1, 2, \dots, N \quad (33)$$

또한 식 (30)의  $\beta_i$ 를 식 (33)의 괄호 안에 대입하면 다음 식을 얻는다.

$$1 - \beta \sqrt{\sigma_{x_i}^2} = 1 - \lambda^{N^2} \sqrt{1 - \lambda} \sqrt{\sigma_{x_i}^2} \quad (34)$$

식 (34)에서 알 수 있듯이 망각계수가  $\lambda=1$  또는  $\lambda=0$  인 경우  $\beta=0$ 이 되므로 이 경우 알고리즘의 성능은 나쁠 것으로 판단된다. 그러므로 망각계수 범위는 다음과 같다.

$$0 < \lambda < 1 \quad (35)$$

앙상블 평균 학습 곡선(ensemble average learning curve)에 대하여 개략적으로 살펴보자.

$$\begin{aligned} \tilde{e}(n) &= d(n) - \mathbf{x}^H(n) (\mathbf{w}_o - \boldsymbol{\varepsilon}(n-1)) \\ &= e_o(n) + \boldsymbol{\varepsilon}^H(n-1) \mathbf{x}(n) \end{aligned} \quad (36)$$

여기서  $e_o(n)$ 은 측정잡음이고 입력 신호 벡터  $\mathbf{x}(n)$ 과 독립(independent)이고 평균은 0, 분산은  $\sigma_o^2$ 을 갖는다. 앙상블 평균의 학습곡선을 다음 식과 같이 정의하자.

$$\begin{aligned} \mathcal{J}(n) = E[|\tilde{e}(n)|^2] &= E[e_o(n)^2] + E[\boldsymbol{\varepsilon}^H(n-1) \mathbf{x}(n) \boldsymbol{\varepsilon}(n-1)] \\ &\quad + E[\boldsymbol{\varepsilon}^H(n-1) \mathbf{x}(n) e_o^*(n)] \\ &\quad + E[\boldsymbol{\varepsilon}^H(n-1) \mathbf{x}(n) \mathbf{x}^H(n) \boldsymbol{\varepsilon}(n-1)] \end{aligned} \quad (37)$$

입력 신호들이 uncorrelated 되어 있다고 가정하면 다음과 같이 정리될 수 있다.

$$\begin{aligned} \mathcal{J}(n) &= \sigma_o^2 + \sum_{i=1}^N [E[x_i^2(n)] E[(w_i(n-1) - w_o^i)^2]] \\ &\approx \sigma_o^2 + \sigma_{x_i}^2 \sum_{i=1}^N E[(w_i(n-1) - w_o^i)^2] \end{aligned} \quad (38)$$

식 (36)에 식 (31)의 결과를 이용하면 식 (38)은 다음과 같이 됨을 추론 할 수 있다.

$$\mathcal{J}(n) \approx \sigma_o^2, \text{ for } n \rightarrow \infty \quad (39)$$

### III. 모의 실험 및 검토

랜덤잡음  $u(n)$ 을 식 (40)에 통과시켜 FIR 시스템의 출력  $y(n)$ 를 얻었다.

$$\begin{aligned} y(n) &= u(n) + 0.5u(n-1) + 1.5u(n-2) - 2.0u(n-3) \\ &\quad + 2.0u(n-4) + v(n) \end{aligned} \quad (40)$$

$u(n)$ 과  $v(n)$ 는 분산이 1이고 평균이 0인 가우시안 잡음이다. 실험에서 측정잡음  $v(n)$ 는  $u(n)$ 에 대하여 SN비를 -60dB, -30dB 및 -10dB로 하였다. 그림 1에는 S/N별로 D-QR-RLS 알고리즘의 학습곡선을 나타내었다. 그림에서 알 수 있듯이 각 측정 잡음의 S/N값에 수렴함을 알 수 있다.

그림 1의 빨강색 그래프들은 식 (38)을 컴퓨터 시뮬레이션하여 나타낸 그림이다. 식 (38)은 완전히 유도된 식은 아니지만 전체적으로 학습 곡선의 모양을 잘 나타내 있다. 그림 2는 S/N 비를 -10dB, -30dB, -60dB로 하여 얻은 계수 벡터를 모두 나타낸 그림이다. 각 계수는 최적에 값에 수렴함을 알 수 있다. 그림 1과 그림 2에서 고정 계수를 갖는 FIR 시스템 식별을 잘 수행함을 알 수 있다. 그림 3은 제안한 D-QR-RLS 알고리즘, 일반적인 QRD-RLS 알고리즘 및 참고문헌 [7-9]의 TA-QR 또는 A-QR-RLS 알고리즘의 학습곡선을 나타낸 것이다. 참고문헌 [7-9]에서 QR 방법과 Householder 변환법에 의한 방법을 내포하고 있다. 그리고 시뮬레이션에서 Volterra와 ARMA로 주로 시뮬레이션 하였는데 참고문헌 [7-9]의 결과와 유추하여 보길 바란다. 수렴 특성을 조절할 수 있는 파라미터가 없이 유도되었기 때문에  $\lambda$ 는 1을 나타낸다. QRD-RLS 알고리즘에서  $\lambda=0.9$ 일 때는 수렴특성이 좋지 않고  $\lambda=0.99999 \approx 1.0$ 일 때 성능이 좋다. 그리고 참고문헌 [7-9]의 방법은 본 논문에서 제안한 방법 보다 성능이 좋지 못하다.

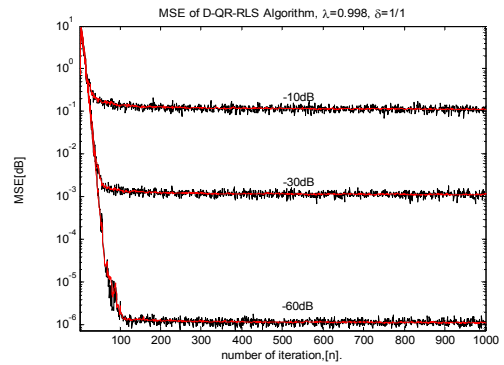


그림 1. 제안한 알고리즘의 학습곡선.

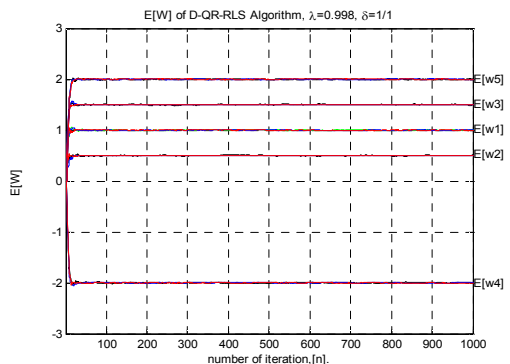


그림 2. 제안한 알고리즘 계수들의 궤적.

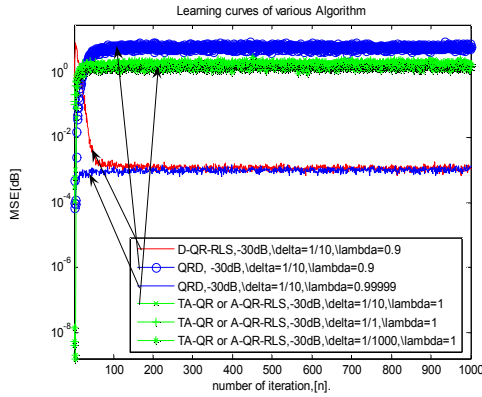


그림 3. 여러 알고리즘의 학습곡선들.

#### IV. 결론

본 논문에서 입력 신호들이 서로 상관관계가 없다고 가정하는 근사화 과정을 이용한 적응 알고리즘을 제안하였다. 알고리즘의 유도는 QR 분해과정에서 생성되는 sine 파라미터가 입력 신호가 생성하는 공간의 단위벡터인 것을 발견하여 이것을 계수 순환식에 이용한 D-QR-RLS 알고리즘을 제안하였다. 제안한 알고리즘의 계산량은  $O(N)$ 으로 고속 알고리즘이다. 또한 제안한 알고리즘의 안정도와 수렴 특성에 대하여 이론적으로 고찰하였다. 그 결과 D-QR-RLS 알고리즘은 최적 계수벡터에 수렴함을 알 수 있었다. 앞으로 본 알고리즘을 이용하여 비선형 적응 신호처리 문제와 ARMA등에 대한 응용을 더욱 연구해 볼 필요성이 있다.

#### 참고 문헌

[1] M. T. Morf, T. Kailath and L. Ljung, "Fast algorithm for recursive identification," in proc., 1976 Congress Decision Control, Clearwater Beach, FL, pp. 916-921, 1976.

[2] J. M. Cioffi and T. Kailath, "Fast RLS Transversal filters for adaptive filtering," IEEE Trans., Acoust., Speech, Signal Processing, vol. ASSP-32, pp. 304-337, 1984.

[3] S. Ljung and L. Ljung, "Error propagation properties of recursive least squares adaptive algorithms," Automatica, vol. 21, pp.157-167, 1985.

[4] D.T.M. Slock and T. Kailath, "Numerically stable fast recursive least squares transversal filters," Proc., IEEE int. Conf. Acoust., Speech, Signal Processing, pp. 1369-1372, 1988

[5] W. M. Gentleman and H. T. Kung, "Matrix triangularization by systolic array," in Proc. SPIE int. Soc., Opt. Eng., vol. 298. pp. 298-303, 1981

[6] J. G. McWhiter, "Recursive least squares minimization using a systolic array," in Proc. SPIE int. Soc., Opt. Eng., vol. 431. pp. 105-112, 1983

[7] Z. S. Liu, "QR methods of  $O(N)$  complexity in adaptive parameter estimation," IEEE Trans, Signal Processing, vol. 43, pp. 720-729, 1995.

[8] Z. S. Liu and J. Li, "A QR-based least mean squares algorithm for adaptive parameter estimation," IEEE Trans. Circuits Syst., vol. 45, pp. 321-329, Mar. 1998.

[9] S. C. Chan, X. X. Yang, "Improved approximate QR-LS algorithms for adaptive filtering," IEEE Transactions on Circuits and Systems-II: Express Briefs, vol. 51, no. 1, pp. 29 - 39, Jan 2004.

#### 저자

조해성(Hae-Seong Cho) 정회원



1994년 2월 : 전북대학교  
전자공학과 (공학사)  
1996년 2월 : 전북대학교  
전자공학과 (공학석사)  
2001년 2월 : 전북대학교 전자공학과  
(공학박사)

2009년 현재 : 건양대학교 전자정보공학과 교수

<관심분야> 통신 자원할당 기술, 센서 네트워크, Ad-Hoc Network, BcN

조주필(Ju-Phil cho) 정회원



1992년 : 전북대학교 정보통신공학과  
졸업.  
1994년 : 전북대학교 전자공학과 석사  
2001년 : 전북대학교 전자공학과  
공학박사.  
2000년~2005년 : ETRI 이동통신연구단  
선임연구원.

2005년~현재 : 군산대학교 전파공학과 부교수.

2006년~2007년 : ETRI 초빙연구원.

<관심분야> 통신신호처리, 적응신호처리, 4세대 이동통신기술, Wibro 요소기술, Cognitive Radio, UWB, WPAN