

超安定性 適應 循環 필터의 收斂性 改善

Improved Convergency Characteristics of the Hyperstable Adaptive Recursive Filter

신 윤 기*
(Yoon-Ki Shin*)

요 약

시스템의 고급화에 따라 기존의 고정 시스템만으로는 보다 고차원의 만족할 만한 결과를 얻을 수 없었으며, 이에 따라 주위 환경에 스스로 적응하면서 기능을 수행할 수 있는 적응 시스템이 활발히 연구되어왔으며 그중 가장 대표적인 분야가 적응 필터로 볼 수 있다. 그러나 적응 순환 필터는 적응 비순환 필터에 비해 상대적으로 매우 낮은 차수로도 충분한 기능을 수행할 수 있다는 큰 장점이 있지만, 안정성(수렴성)면에서는 매우 어려운 점이 많았다. 그 후 V.M. Popov의 초안정 이론을 이용하여 수렴성이 확실히 보장되는 초안정성 적응 순환 필터 알고리즘이 제안되었다. 본 연구에서는 이 알고리즘의 수렴특성을 더욱 개선시키는 문제에 대해 연구하였다.

ABSTRACT

Fixed systems are limited in their performances to achieve the more complicated and higher level operations. Accordingly adaptive system, which adjusts itself in accordance with the time-varying environments, has been introduced to camouflage the deficiency of fixed systems in varying environment, and adaptive filter is the outstanding fields in adaptive system. Adaptive recursive filter is far more efficient in that it can perform the signal processing with relatively lower filter order, but there remains severe problem in stability(convergency). On the basis of hyperstability introduced by V.M. Popov, a hyperstable new adaptive recursive filter is introduced which is theoretically guaranteed in stability. In this paper a more stable algorithm for adaptive recursive filter is devised.

I. 서 론

적응 순환 필터는 적응 비순환 필터에 비해 매우 낮은 차수로도 그 이상의 성능을 발휘할 수 있으므로 많은 연구가 진행되어 오고 있다. 적응 순환 필터에는 크게 방정식 오차 방법과 출력 오차 방법의 두 가지 機作이 있다. 방정식 오차 방법은 적응 필터의 극점과 영점을 적응 비순환 필터의 원리로 각각 따로 적응시키기 때문에 안정성면에서는 문제가 없지만 한정된 경우를 제외하고는 실제의 신호 처리 시스템으로 사용하기에는 불가능한 경우가 많다. 이러한 점을 보완할 수 있는 방법이 출력 오차를 이용하는 출력 오차 방법의 적응 순환 필터가 있다.

그러나 출력 오차 방법은 방정식 오차 방법에 비해 실제 신호처리 시스템으로 사용할 수 있을 뿐 아니라, 잡음에 대한 내구성도 상대적으로 강하지만 적응 과정에서

극점이 단위원 밖으로 벗어날 수 있으며, 그 결과 필터의 수렴성에 심각한 문제가 야기될 수 있다. 이를 해결하기 위하여 여러 가지 알고리즘들이 제시되었지만, 모두가 적응 이득을 매우 낮게 설정하여 필터의 수렴 과정에서 안정성을 확보하고자 하는 의도이므로 수렴 속도가 매우 낮을 뿐 아니라, 확실한 수렴 보장이 증명되지 않고 있다.

이러한 문제를 해결하기 위하여 V.M. Popov의 비선형 시스템의 안정성을 위한 초안정성 이론¹⁾을 바탕으로 적응 순환 필터 알고리즘이 제시되었다^{2, 3, 4, 5, 6)}. 이 알고리즘의 특징은 출력 오차를 적절한 필터를 사용하여 처리한 다음 처리된 출력 오차를 이용하여 극점과 영점을 적응시켜나간다는 점이다. 이 필터를 오차 완화 필터(error smoothing filter) 혹은 보상기(compensator)라 부른다(이하 보상기로 통일한다). 그러나 이 유형의 알고리즘 역시 심각한 문제점이 있는 바, 그것은 바로 보상기를 결정하기 위해서는 적응 필터의 극점에 대한 일련의 사전 지식이 요구된다는 사실이다. 그러나 알려지지 않은 적응 필터의 극점에 대한 사전 지식이 요구됨으로써, 미지의 극

*대전대학교 전자공학과
접수일자: 1997년 6월 16일

같은 식의 해를 구하는 적응 필터의 시스템 식별(system identification)기능면에서 보면 식으로 표현된다. 즉 실제 상황에서는 사용하지가 거의 불가능하다.

초안정을 이용한 유형의 적응 순환 필터에서의 이러한 문제를 해결하기 위하여 보상기 자체 역시 적응 기작에 의해 최적치로 유도해 나가는 방법이 제시되었다¹⁾. 이러한 보상기를 적응 보상기라 한다. 이 방법은 수렴성이 보정됨으로써, 적응 이득을 적정 수준으로 증가시켜 원하는 수준의 수렴 속도를 얻을 수 있다.

본 연구에서는 적응 보상기를 채용한 초안정성 적응 순환 필터(Adaptively Compensated Hyperstable Adaptive Recursive Filter, 이하 ACHARF라 약한다)의 수렴 특성을 좀더 개선시킬 수 있는 방법을 설명하고 이에 따른 알고리즘을 소개하기로 한다.

II. 초안정적 적응 순환 필터

그림 1은 출력 오차 방법을 바탕으로 하는 적응 순환 필터의 구성도이다. 여기서 $A_0(z)$, $B_0(z)$ 에 의해 각각 극점과 영점이 결정되는 시스템은 미지의 시스템으로써 기준 시스템이라 한다. 그림 1에서 $A(z)$, $B(z)$ 로 구성된 적응 필터의 목적은 미지의 기준 시스템의 입력 $x(k)$ 와 출력 $d(k)$ 를 바탕으로 하여 기준 시스템에 대한 특성, 예를 들면 $A_0(z)$, $B_0(z)$ 의 계수 등을 추정해 내는 것이다. 이러한 일련의 과정은 모두 입력 $x(k)$ 와 기준 시스템의 출력 $d(k)$ 를 관찰 자료로 하여 신호가 진행됨에 따라 시간을 두고 적응적으로 유추해 내는 작업인 것이다. 즉 적응 필터의 출력 $y(k)$ 가 몇번의 적응 후에 기준 시스템의 출력 $d(k)$ 에 최소 차승의 의미에서 가까워 지느냐가 중요하다. 이 과정에서 적응 필터의 성능에 대한 판단 기준이 세워지며 그것이 바로 안정성과 수렴 속도인 것이다. 기준 시스템의 출력 $d(k)$ 는 적응 필터의 입장에서는 따라가야 할 기준이 되므로 원하는 출력이라 하며, 일각간의 오차로써 $d(k)$ 를 추적해나가는 $y(k)$ 를 추정 출력이라 한다. 미지의 기준 시스템과 적응 필터는 $x(k)$ 를 공통으로 입력 신호로 사용하므로 추정 출력 $y(k)$ 가 원하는 출력 $d(k)$ 를 추적하는 정도가 바로 $A(z)$, $B(z)$ 가 각각 $A_0(z)$, $B_0(z)$ 를 추적하는 정도로 볼 수 있다.

그림 1의 적응 순환 필터에서 가장 문제가 되는 것은 적응 과정에서 $1+A(z)$ 의 영점, 즉 적응 필터의 극점의 분포이다. 만일 어떤 적응 순간에 $1+A(z)$ 의 영점이 단위 원 밖으로 벗어나 자리하게 되면 그 이후의 추정 출력 $y(k)$ 에 발산 가능성이 생기게 되며, 이러한 과정이 반복 되면 결국 $y(k)$ 는 쉽게 발산하게 된다. 방정식 오차 방법의 적응 순환 필터에서는 원하는 출력 $d(k)$ 에 근거하여 $A(z)$ 의 계수가 수정되는 것과는 달리, 출력 오차 방법은 추정 출력 $y(k)$ 를 근거로 $A(k)$ 의 계수가 수정되므로 $y(k)$ 의 발산은 $A(z)$ 계수의 발산을 유도하며 $A(z)$ 계수의 발산은 다시 추정 출력 $y(k)$ 를 더욱 발산시키는 상승 작용

을 일으키므로 적응 필터는 쉽게 안정성을 잃게 되는 것이다.

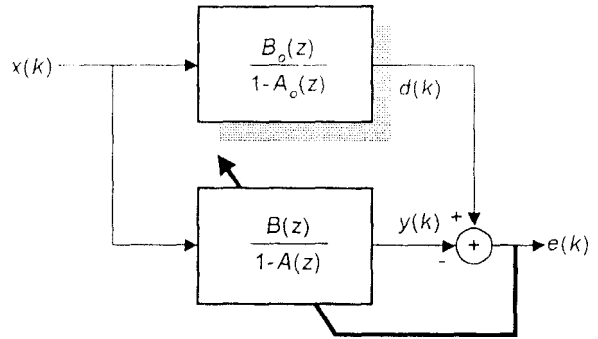


그림 1. 적응 순환 필터의 구성
Fig. 1. Structure of Adaptive Recursive Filter

그림 2는 그림 1의 구조가 가지는 안정성 문제를 해결하기 위하여, 출력 오차 신호(이하 오차 신호)를 보상기로서 필터링한 다음 사용함으로써 전체 적응 필터가 초안정성을 만족시킬 수 있도록 한 구조이다. 특히 고정 보상기에서의 문제점을 피하기 위하여 보상기 계수 자체도 알고리즘에 의해 최적치로 적용될 수 있도록 한 구조이다. 잡음 $n(k)$ 는 측정시에 발생하는 측정 잡음이나 그 외의 여러 요인에 의해 발생하는 모든 잡음을 하나의 잡음으로 등가화한 것이다. 그림 2에서 원하는 출력 $d(k)$ 는

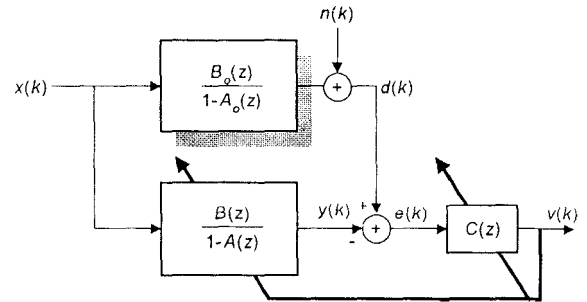


그림 2. 적응 보상기를 채용한 초안정적 적응 순환 필터의 구성
Fig. 2. Hyperstable Adaptive Recursive Filter employing Adaptive Compensator

$$d(k) = \sum_{i=1}^{N_1} a_{opt,i} d(k-i) + \sum_{j=0}^{N_2} b_{opt,j} x(k-j) = \Theta_{opt}^T \Phi_{opt}(k) \tag{1}$$

로 표현할 수 있으며, 여기서 벡터 Θ_{opt} , $\Phi_{opt}(k)$ 는 각각 파라미터 벡터 및 데이터 벡터로서

$$\Theta_{opt}^T = [a_{opt,1}, a_{opt,2}, \dots, a_{opt,N_1}, b_{opt,0}, b_{opt,1}, \dots, b_{opt,N_2}] \tag{2}$$

$$\Phi_{opt}(k) = [d(k-1), d(k-2), \dots, d(k-N_p), x(k), x(k-1), \dots, x(k-N_2)]^T \tag{3}$$

로 정의된다. 다음으로 적응 필터의 추정 출력 $y(k)$ 는

$$y(k) = \sum_{i=1}^{N_1} a_i(k)y(k-i) + \sum_{j=0}^{N_2} b_j(k)x(k-j) = \Theta(k)^T \Phi(k) \quad (4)$$

로 표현할 수 있으며, 여기서 벡터 $\Theta(k)$, $\Phi(k)$ 는 각각

$$\Theta(k)^T = [a_1(k), a_2(k), \dots, a_{N_1}(k), b_0(k), b_1(k), \dots, b_{N_2}(k)] \quad (5)$$

$$\Phi(k) = [y(k-1), y(k-2), \dots, y(k-N_p), x(k), x(k-1), \dots, x(k-N_x)]^T \quad (6)$$

로서 Θ_{opt} , $\Phi_{opt}(k)$ 에 대한 적응 순간 k 에서의 추정 벡터들이다. 여기서 다음과 같이 또 다른 한가지의 추정 출력 $y^*(k)$ 및 데이터 벡터 $\Phi^*(k)$ 를 정의한다.

$$y^*(k) = \sum_{i=1}^{N_1} a_i(k+1)y^*(k-i) + \sum_{j=0}^{N_2} b_j(k+1)x(k-j) = \Theta(k+1)^T \Phi^*(k) \quad (7)$$

$$\Phi^*(k) = [y^*(k-1), y^*(k-2), \dots, y^*(k-N_p), x(k), x(k-1), \dots, x(k-N_x)]^T \quad (8)$$

식 (4)의 추정 출력 $y(k)$ 는 k 번째 적응 순간에서의 파라미터 추정치인 $\{a_i(k), i=1, \dots, N_p\}$ 와 $\{b_j(k), j=1, \dots, N_x\}$ 에 의해 추정된 출력인데 비해 $y^*(k)$ 는 식 (7)이 나타내는 바와 같이 k 번째 적응 순간에서, 다음 단계인 $k+1$ 번째에서 사후에 추정될 예정인 $\{a_i(k+1), i=1, \dots, N_p\}$ 와 $\{b_j(k+1), j=1, \dots, N_x\}$ 에 의해 추정된 출력으로서 사후 추정 출력(a-posteriori estimated output)이라 한다. 이제 위의 두 추정 출력을 근간으로 다음과 같은 4가지의 출력 오차를 정의한다.

$$e(k) = d(k) - y(k) = d(k) - \Theta(k)^T \Phi(k) \quad (9)$$

$$e^*(k) = d(k) - y^*(k) = d(k) - \Theta(k+1)^T \Phi^*(k) \quad (10)$$

$$e^o(k) = d(k) - \Theta(k-1)^T \Phi(k) \quad (11)$$

$$e^{*o}(k) = d(k) - \Theta(k)^T \Phi^*(k) \quad (12)$$

여기서 $e(k)$, $e^*(k)$ 는 k 번째 적응에서의 추정 오차 및 사후 추정 오차를 의미하며, $e^o(k)$, $e^{*o}(k)$ 는 각각 $e^*(k)$, $e^*(k)$ 에 비해 한 단계전에 추정된 파라미터를 이용한 오차로서 각각 $e(k)$, $e^*(k)$ 에 대한 사전 추정 오차(a-priori estimation error)라 한다. 이상의 네가지 오차를 적응 보상기를 통하여 처리한 다음과 같은 네가지의 처리된 오차(processed error)를 정의한다.

$$v(k) = e(k) + \sum_{l=1}^{N_1} c_l(k)e(k-l) \quad (13)$$

$$v^*(k) = e^*(k) + \sum_{l=1}^{N_1} c_l(k+1)e^*(k-l) \quad (14)$$

$$v^o(k) = e^o(k) + \sum_{l=1}^{N_1} c_l(k-l)e^o(k-l) \quad (15)$$

$$\eta^*(k) = e^{*o}(k) + \sum_{l=1}^{N_1} c_l(k)e^*(k-l) \quad (16)$$

식 (13)에서 (16)의 $v(k)$, $v^*(k)$, $\eta(k)$, $\eta^*(k)$ 를 벡터 형태로 간략히 표현기 위하여 식 (5)로부터 (8)에 걸쳐 정의된 추정 파라미터 벡터와 추정 데이터 벡터에 대한 연장된 형태의 벡터를 각각 다음과 같이 정의한다.

$$\Theta_e(k)^T = [a_1(k), a_2(k), \dots, a_{N_1}(k), b_0(k), b_1(k), \dots, b_{N_2}(k), -c_1(k), -c_2(k), \dots, -c_{N_1}(k)] \quad (17)$$

$$= [\Theta(k)^T, -c_1(k), -c_2(k), \dots, -c_{N_1}(k)]$$

$$\Phi_e(k) = [y(k-1), y(k-2), \dots, y(k-N_p), x(k), x(k-1), \dots, x(k-N_x), e(k-1), e(k-2), \dots, e(k-N_p)]^T \quad (18)$$

$$= [\Phi(k), e(k-1), e(k-2), \dots, e(k-N_p)]^T$$

$$\Phi_e^*(k) = [y^*(k-1), y^*(k-2), \dots, y^*(k-N_p), x(k), x(k-1), \dots, x(k-N_x), e^*(k-1), e^*(k-2), \dots, e^*(k-N_p)]^T \quad (19)$$

$$= [\Phi^*(k), e^*(k-1), e^*(k-2), \dots, e^*(k-N_p)]^T$$

위의 식 (17), (18) 및 (19)를 이용하면 식 (13)에서 (16)에 걸쳐 정의된 처리된 오차들은 각기 다음과 같이 정리된 벡터 형태로 표현된다.

$$v(k) = e(k) + \sum_{l=1}^{N_1} c_l(k)e(k-l) = d(k) - \Theta_e(k)^T \Phi_e(k) \quad (20)$$

$$v^*(k) = e^*(k) + \sum_{l=1}^{N_1} c_l(k+1)e^*(k-l) = d(k) - \Theta_e(k+1)^T \Phi_e^*(k) \quad (21)$$

$$\eta(k) = e^o(k) + \sum_{l=1}^{N_1} c_l(k-1)e(k-l) = d(k) - \Theta_e(k-1)^T \Phi_e(k) \quad (22)$$

$$\eta^*(k) = e^{*o}(k) + \sum_{l=1}^{N_1} c_l(k)e^*(k-l) = d(k) - \Theta_e(k)^T \Phi_e^*(k) \quad (23)$$

그림 2에서 적응 보정기 $C(z)$ 가 $A(z)$ 가 되면 그림 2의 적응 필터는 항상 초안정성을 만족할 수 있으므로, $C(z)$ 의 계수인 $\{c_l(k), l=1, \dots, N_p\}$ 는 각각 $\{-a_i(k), i=1, \dots, N_p\}$ 로 수렴시켜야 한다^[7]. 따라서 식 (17)의 파라미터 벡터 $\Theta_e(k)$ 에 대한 최적치를 $\Theta_{e, opt}(k)$ 라 하면

$$\begin{aligned} \Theta_{e, opt}^T &= [a_{opt,1}, a_{opt,2}, \dots, a_{opt,N_p}, b_{opt,0}, b_{opt,1}, \dots, b_{opt,N_p}, \\ &\quad a_{opt,1}, a_{opt,2}, \dots, a_{opt,N_p}] \quad (24) \\ &= [\Theta_{opt}^T, a_{opt,1}, a_{opt,2}, \dots, a_{opt,N_p}] \end{aligned}$$

로 주어짐을 알 수 있다. 즉 적합한 알고리즘으로써 $\Theta_e(k)$ 를 $\Theta_{e, opt}(k)$ 로 수렴시키면 $\{c_l(k), l=1, \dots, N_p\}$ 가 $\{-a_i(k), i=1, \dots, N_p\}$ 로 수렴하여 적응 필터의 초안정성이 만족되게 할 수 있는 것이다. 이상의 사항들을 바탕으로 하여 본 논문에서는 다음과 같은 Cost 함수를 정의하고 이를 최소화하는 알고리즘을 유도하기로 한다.

$$\begin{aligned} J(\Theta_e(k)) &= v^*(k-1)^2 + v(k)^2 + [\Theta_e(k) \\ &\quad - \Theta_e(k-1)]^T M_e^{-1} [\Theta_e(k) - \Theta_e(k-1)] \quad (25) \end{aligned}$$

식 (25)에서 행렬 M_e 는 적응 이득을 나타내는 행렬이며 아래와 같이 대각선 행렬로 정의된다.

$$M_e = \text{diag} \{ \mu_1, \mu_2, \dots, \mu_{N_p}, \rho_0, \rho_1, \dots, \rho_{N_p}, \gamma_1, \gamma_2, \dots, \gamma_{N_p} \} \quad (26)$$

$J(\Theta_e(k))$ 를 최소화하는 $\Theta_e(k)$ 는 다음과 같이 구할 수 있다.

$$\begin{aligned} \frac{\partial J(\Theta_e(k))}{\partial \Theta_e(k)} &= 2v^*(k-1) \frac{\partial v^*(k-1)}{\partial \Theta_e(k)} + 2v(k) \frac{\partial v(k)}{\partial \Theta_e(k)} \\ &\quad + 2M_e^{-1} \{ \Theta_e(k) - \Theta_e(k-1) \} \quad (27) \\ &= 0 \end{aligned}$$

식 (27)로부터

$$\Theta_e(k) = \Theta_e(k-1) + M_e \{ v^*(k-1) \Phi_e^*(k-1) + v(k) \Phi_e(k) \} \quad (28)$$

의 순환식을 얻을 수 있다. 식 (28)은 풀어서 다음과 같이 표현할 수 있다.

$$\begin{aligned} a_i(k) &= a_i(k-1) + \mu_i y^*(k-i-1) v^*(k-1) \\ &\quad + \mu_i y(k-i) v(k) \quad (29-a) \end{aligned}$$

$$\begin{aligned} b_j(k) &= b_j(k-1) + \rho_j x(k-j-1) v^*(k-1) \\ &\quad + \mu_j x(k-j) v(k) \quad (29-b) \end{aligned}$$

$$\begin{aligned} c_l(k) &= c_l(k-1) - \gamma_l e^*(k-l-1) v^*(k-1) \\ &\quad + \gamma_l e(k-l) v(k) \quad (29-c) \end{aligned}$$

식 (29)에서 $v^*(k-1)$ 과 $v(k)$ 는 모두 $\{a_i(k)\}, \{b_j(k)\}, \{c_l(k)\}$ 들의 함수이므로 k 번째 순간에서의 파라미터 값들을 구하기 위해서는 역행렬의 계산이 요구됨을 알 수 있다. 이를 피하기 위하여 다음과 같은 과정을 거친다. 먼저 $v^*(k-1)$ 는

$$\begin{aligned} v^*(k-1) &= e^*(k-1) + \sum_{l=1}^{N_p} c_l(k) e^*(k-l-1) \\ &= \sum_{l=1}^{N_p} a_{opt,l} x(k-l-1) + \sum_{j=0}^{N_p} b_{opt,j} x(k-j-1) \\ &\quad - \sum_{l=1}^{N_p} a_l(k) y(k-l-1) - \sum_{j=0}^{N_p} b_j(k) x(k-j-1) \\ &\quad + \sum_{l=1}^{N_p} c_l(k) e^*(k-l-1) \quad (30) \end{aligned}$$

와 같이 표현할 수 있으며, 위의 식 (30)의 $\{a_i(k)\}, \{b_j(k)\}, \{c_l(k)\}$ 대신 식 (29-a), (29-b) 및 (29-c)를 각각 대입하여 정리하면 다음과 같은 식을 얻을 수 있다.

$$\begin{aligned} & \left(1 + \sum_{l=1}^{N_p} \mu_l y^*(k-l-1)^2 + \sum_{j=0}^{N_p} \rho_j x(k-j-1)^2 \right. \\ & \quad \left. + \sum_{l=1}^{N_p} \gamma_l e^*(k-l-1)^2 \right) v^*(k-1) \\ &= \eta^*(k-1) - \left\{ \sum_{l=1}^{N_p} \mu_l y^*(k-i-1) y(k-i) \right. \\ & \quad \left. + \sum_{j=0}^{N_p} \rho_j x(k-j-1) x(k-j) \right. \\ & \quad \left. + \sum_{l=1}^{N_p} \gamma_l e^*(k-l-1) e(k-l) \right\} v(k) \quad (31) \end{aligned}$$

마찬가지 과정을 거쳐 $v(k)$ 역시

$$\begin{aligned} v(k) &= e(k) + \sum_{l=1}^{N_p} c_l(k) e(k-l) \\ &= \sum_{l=1}^{N_p} a_{opt,l} x(k-l) + \sum_{j=0}^{N_p} b_{opt,j} x(k-j) \\ &\quad - \sum_{l=1}^{N_p} a_l(k) y(k-l) - \sum_{j=0}^{N_p} b_j(k) x(k-j) \\ &\quad + \sum_{l=1}^{N_p} c_l(k) e(k-l) \quad (32) \end{aligned}$$

와 같이 표현되므로 식 (32)의 $\{a_i(k)\}, \{b_j(k)\}, \{c_l(k)\}$ 대신 식 (29-a), (29-b) 및 (29-c)를 각각 대입하여 정리하면 다음과 같은 식을 얻을 수 있다.

$$\begin{aligned} & \left(1 + \sum_{l=1}^{N_p} \mu_l y(k-l)^2 + \sum_{j=0}^{N_p} \rho_j x(k-j)^2 \right. \\ & \quad \left. + \sum_{l=1}^{N_p} \gamma_l e(k-l)^2 \right) v(k) \\ &= \eta(k) - \left\{ \sum_{l=1}^{N_p} \mu_l y^*(k-i-1) y(k-i) \right. \quad (33) \end{aligned}$$

$$\begin{aligned}
 & + \sum_{j=0}^{N_1} \rho_j x(k-j-1)x(k-j) \\
 & + \sum_{l=1}^{N_2} \gamma_l e^*(k-l-1)e(k-l))v^*(k-1)
 \end{aligned}$$

식 (31)과 (33)을 간단히 표현하기 위하여 다음과 같이 스칼라 변수 $A(k)$, $A^*(k-1)$, $C(k)$ 및 $D(k)$ 를 정의한다.

$$\begin{aligned}
 A(k) = & 1 + \sum_{i=1}^{N_1} \mu_i v^*(k-i)^2 + \sum_{j=0}^{N_1} \rho_j x(k-j)^2 \\
 & + \sum_{l=1}^{N_2} \gamma_l e^*(k-l)^2
 \end{aligned} \quad (34)$$

$$\begin{aligned}
 A^*(k-1) = & 1 + \sum_{i=1}^{N_1} \mu_i v^*(k-i-1)^2 \\
 & + \sum_{j=0}^{N_1} \rho_j x(k-j-1)^2 + \sum_{l=1}^{N_2} \gamma_l e^*(k-l-1)^2
 \end{aligned} \quad (35)$$

$$\begin{aligned}
 C(k) = & \sum_{i=1}^{N_1} \mu_i v^*(k-i-1)y(k-i) \\
 & + \sum_{j=0}^{N_1} \rho_j x(k-j-1)x(k-j) \\
 & + \sum_{l=1}^{N_2} \gamma_l e^*(k-l-1)e(k-l)
 \end{aligned} \quad (36)$$

$$D(k) = A(k)A^*(k-1) - C(k)^2 \geq 1 \quad (37)$$

식 (34)~(37)을 이용하여 식 (31)과 (33)은

$$\begin{bmatrix} A^*(k-1) & C(k) \\ C(k) & A(k) \end{bmatrix} \begin{bmatrix} v^*(k-1) \\ v(k) \end{bmatrix} = \begin{bmatrix} \eta^*(k-1) \\ \eta(k) \end{bmatrix} \quad (38)$$

와 같이 표현할 수 있다. 식 (38)에서 우변의 $\eta^*(k-1)$ 와 $\eta(k)$ 는 모두 $\{a_i(k-1)\}$, $\{b_j(k-1)\}$, $\{c_l(k-1)\}$ 의 함수이며, 스칼라 변수 $A(k)$, $A^*(k-1)$, $C(k)$ 및 $D(k)$ 는 모두 k 번째 순간에서 주어진 데이터들로서 이루어지므로 매우 복잡한 역행렬 계산을 거치지 않고서도 식 (38)로부터 $v^*(k-1)$ 와 $v(k)$ 를 쉽게 구할 수 있게 되는 것이다. 더구나 식 (38)에서 $A(k)$, $A^*(k-1)$, $C(k)$ 및 $C(k)$ 로 구성된 행렬은 식 (37)의 성질에 의해 determinant가 항상 1보다 큼을 알 수 있다. 식 (38)에서 구한 $v^*(k-1)$ 와 $v(k)$ 를 식 (29)에 대입하면 식 (29)에 대한 causal한 순환식을 쉽게 얻을 수 있다. 이상의 과정을 알고리즘으로 나타내면 다음과 같다.

제안된 알고리즘:

k 번째 적용 순간에서 주어진 $\{a_i(k-1)\}$, $\{b_j(k-1)\}$, $\{c_l(k-1)\}$ 로부터

- 오차 신호를 구한다.

$$\begin{aligned}
 e^o(k) &= d(k) - \Theta(k-1)^T \Phi(k) \\
 e^{*o}(k-1) &= d(k-1) - \Theta(k-1)^T \Phi^*(k-1)
 \end{aligned}$$

- 사전 처리된 오차 신호를 구한다.

$$\eta(k) = e^o(k) + \sum_{l=1}^{N_1} c_l(k-1)e(k-l)$$

$$\eta^*(k-1) = e^{*o}(k-1) + \sum_{l=1}^{N_2} c_l(k-1)e^*(k-l-1)$$

- 스칼라 변수 $A(k)$, $A^*(k-1)$, $C(k)$ 및 $D(k)$ 를 구한다.

$$\begin{aligned}
 A(k) = & 1 + \sum_{i=1}^{N_1} \mu_i v^*(k-i)^2 + \sum_{j=0}^{N_1} \rho_j x(k-j)^2 \\
 & + \sum_{l=1}^{N_2} \gamma_l e^*(k-l)^2
 \end{aligned}$$

$$\begin{aligned}
 A^*(k-1) = & 1 + \sum_{i=1}^{N_1} \mu_i v^*(k-i-1)^2 \\
 & + \sum_{j=0}^{N_1} \rho_j x(k-j-1)^2 + \sum_{l=1}^{N_2} \gamma_l e^*(k-l-1)^2
 \end{aligned}$$

$$\begin{aligned}
 C(k) = & \sum_{i=1}^{N_1} \mu_i v^*(k-i-1)y(k-i) \\
 & + \sum_{j=0}^{N_1} \rho_j x(k-j-1)x(k-j) \\
 & + \sum_{l=1}^{N_2} \gamma_l e^*(k-l-1)e(k-l)
 \end{aligned}$$

$$D(k) = A(k)A^*(k-1) - C(k)^2 \geq 1$$

- 처리된 오차 신호를 구한다.

$$\begin{bmatrix} v^*(k-1) \\ v(k) \end{bmatrix} = \frac{1}{D(k)} \begin{bmatrix} A(k) & -C(k) \\ -C(k) & A^*(k-1) \end{bmatrix} \begin{bmatrix} \eta^*(k-1) \\ \eta(k) \end{bmatrix}$$

- 파라미터를 적응시킨다.

$$\begin{aligned}
 a_i(k) = & a_i(k-1) + \mu_i v^*(k-i-1)v^*(k-1) \\
 & + \mu_i x(k-i)v(k), \quad 1 \leq i \leq N_p
 \end{aligned}$$

$$\begin{aligned}
 b_j(k) = & b_j(k-1) + \rho_j x(k-j-1)v^*(k-1) \\
 & + \mu_j x(k-j)v(k), \quad 0 \leq j \leq N_z
 \end{aligned}$$

$$\begin{aligned}
 c_l(k) = & c_l(k-1) - \gamma_l e^*(k-l-1)v^*(k-1) \\
 & + \gamma_l e(k-l)v(k), \quad 1 \leq l \leq N_p
 \end{aligned}$$

참고로 논문^[6]에서 제안한 ACHARF 알고리즘을 소개 하면 다음과 같다.

ACHARF 알고리즘^[6]:

$$\begin{aligned}
 a_i(k) = & a_i(k-1) + \frac{\mu_i}{1 + \Phi_p(k)^T M_p \Phi_p(k)} \eta(k)v(k-i) \\
 & , 1 \leq i \leq N_p
 \end{aligned}$$

$$\begin{aligned}
 b_j(k) = & b_j(k-1) + \frac{\rho_j}{1 + \Phi_p(k)^T M_p \Phi_p(k)} \eta(k)x(k-j) \\
 & , 0 \leq j \leq N_z
 \end{aligned}$$

$$\begin{aligned}
 c_l(k) = & c_l(k-1) + \frac{\gamma_l}{1 + \Phi_p(k)^T M_p \Phi_p(k)} \eta(k)e(k-l) \\
 & , 1 \leq l \leq N_p
 \end{aligned}$$

$$\eta(k) = e^o(k) + \sum_{l=1}^{N_1} c_l(k-1)e(k-l)$$

III. 제안 알고리즘의 수렴성 고찰

제안된 알고리즘의 수렴성을 살펴보기로 한다. 먼저 식 (9)의 $e(k)$ 는

$$\begin{aligned} e(k) &= d(k) - y(k) \\ &= \Theta_{opt}^T \Phi_{opt}(k) - \Theta(k)^T \Phi(k) \\ &= \Theta_{opt}^T \Phi(k) - \Theta(k)^T \Phi(k) + \{\Theta_{opt}^T \Phi_{opt}(k) - \Theta_{opt}^T \Phi(k)\} \\ &= \{\Theta_{opt}^T - \Theta(k)^T\} \Phi(k) + \Theta_{opt}^T \{\Phi_{opt}(k) - \Phi(k)\} \end{aligned} \quad (39)$$

와 같이 표현할 수 있으며, 여기서 다음과 같이 새로운 벡터

$$\mathbf{a} = [a_1, a_2, \dots, a_{N_p}]^T \quad (40)$$

$$\mathbf{e}_k = [e(k-1), e(k-2), \dots, e(k-N_p)]^T \quad (41)$$

를 정의하면 식 (39)는

$$e(k) = \{\Theta_{opt}^T - \Theta(k)^T\} \Phi(k) + \mathbf{a}^T \mathbf{e}_k \quad (42)$$

로 표현된다. 마찬가지로 식 (10)의 $e^*(k)$ 역시

$$\begin{aligned} e^*(k) &= d(k) - y^*(k) \\ &= \Theta_{opt}^T \Phi_{opt}(k) - \Theta(k+1)^T \Phi^*(k) \\ &= \Theta_{opt}^T \Phi^*(k) - \Theta(k+1)^T \Phi^*(k) \\ &\quad + \{\Theta_{opt}^T \Phi_{opt}(k) - \Theta_{opt}^T \Phi^*(k)\} \\ &= \{\Theta_{opt}^T - \Theta(k+1)^T\} \Phi^*(k) + \Theta_{opt}^T \{\Phi_{opt}(k) - \Phi^*(k)\} \end{aligned} \quad (43)$$

로 표현되며, 여기서 또한 새로운 벡터

$$\mathbf{e}^*_k = [e^*(k-1), e^*(k-2), \dots, e^*(k-N_p)]^T \quad (44)$$

를 정의하면 식 (43)은

$$e^*(k) = \{\Theta_{opt}^T - \Theta(k+1)^T\} \Phi^*(k) + \mathbf{a}^T \mathbf{e}^*_k \quad (45)$$

로 표현할 수 있다. 이상의 식 (42)와 (45)를 이용하여 식 (20)과 (21)의 $\nu(k)$ 와 $\nu^*(k)$ 는 각각 다음과 같이 표현할 수 있다.

$$\begin{aligned} \nu(k) &= e(k) + \sum_{l=1}^{N_s} c_l(k) e(k-l) \\ &= \mathbf{a}^T \mathbf{e}_k + \mathbf{c}_k^T \mathbf{e}_k + \{\Theta_{opt} - \Theta(k)\}^T \Phi(k) \\ &= \{\Theta_{c,opt} - \Theta_c(k)\}^T \Phi_c(k) \end{aligned} \quad (46)$$

$$\nu^*(k) = e^*(k) + \sum_{l=1}^{N_s} c_l(k+1) e^*(k-l)$$

$$\begin{aligned} &= \mathbf{a}^T \mathbf{e}^*_k + \mathbf{c}_{k+1}^T \mathbf{e}^*_k \\ &\quad + \{\Theta_{opt} - \Theta(k+1)\}^T \Phi^*(k) \\ &= \{\Theta_{c,opt} - \Theta_c(k+1)\}^T \Phi_c^*(k) \end{aligned} \quad (47)$$

식 (46)과 (47)에서 파라미터 오차 벡터를

$$\mathbf{r}_c(k) = \Theta_{c,opt} - \Theta_c(k) \quad (48)$$

와 같이 정의함과 동시에, 다음과 같이 두 개의 스칼라

$$\mathbf{w}(k) = -\mathbf{r}_c(k)^T \Phi_c(k) = -\overline{\mathbf{w}}(k) \quad (49)$$

$$\mathbf{w}^*(k) = -\mathbf{r}_c(k+1)^T \Phi_c^*(k) = -\overline{\mathbf{w}}^*(k) \quad (50)$$

를 정의하면 식 (46)과 (47)은

$$\nu(k) = -\mathbf{r}_c(k)^T \Phi_c(k) = \mathbf{w}(k) = -\overline{\mathbf{w}}(k) \quad (51)$$

$$\nu^*(k) = -\mathbf{r}_c(k+1)^T \Phi_c^*(k) = \mathbf{w}^*(k) = -\overline{\mathbf{w}}^*(k) \quad (52)$$

와 같이 표현된다. 파라미터 오차 벡터 $\mathbf{r}_c(k)$ 는 최적 파라미터 벡터 $\Theta_{c,opt}$ 와 추정 파라미터 벡터 $\Theta_c(k)$ 와의 차이를 나타내므로 시간 k 에 따른 $\mathbf{r}_c(k)$ 의 상태가 바로 적응 필터의 수렴 상황을 나타내는 중요한 지표가 된다. 식 (28)의 양변으로부터 $\Theta_{c,opt}$ 를 빼면 $\mathbf{r}_c(k)$ 에 대한

$$\mathbf{r}_c(k+1) = \mathbf{r}_c(k) + M_c(\nu(k) \Phi^*(k) + \nu(k+1) \Phi_c(k+1)) \quad (53)$$

와 같은 순환식을 얻을 수 있다. 이제 식 (51), (52) 및 (53)으로부터

$$\begin{aligned} \overline{\mathbf{w}}^*(k) &= \mathbf{r}_c(k+1)^T \Phi_c^*(k) = \mathbf{r}_c(k)^T \Phi_c^*(k) \\ &\quad + \Phi_c^*(k)^T M_c \Phi_c^*(k) \nu^*(k) \\ &\quad + \Phi_c(k+1)^T M_c \Phi_c^*(k) \nu(k+1) \end{aligned} \quad (54)$$

$$\begin{aligned} \overline{\mathbf{w}}(k+1) &= \mathbf{r}_c(k+1)^T \Phi_c(k+1) \\ &= \mathbf{r}_c(k)^T \Phi_c(k+1) \\ &\quad + \Phi_c^*(k)^T M_c \Phi_c(k+1) \nu^*(k) \\ &\quad + \Phi_c(k+1)^T M_c \Phi_c(k+1) \nu(k+1) \end{aligned} \quad (55)$$

의 두 식을 얻을 수 있으며 식 (53), (54) 및 (55)를 상태 방정식으로 표시하면

$$\begin{aligned} [\mathbf{r}_c(k+1)] &= \mathbf{I}[\mathbf{r}_c(k)] + M_c[\Phi_c^*(k), \Phi_c(k+1)] \\ &\quad \begin{bmatrix} \nu^*(k) \\ \nu(k+1) \end{bmatrix} \end{aligned} \quad (56-a)$$

$$\begin{aligned} \begin{bmatrix} \overline{w^*(k)} \\ \overline{w(k+1)} \end{bmatrix} &= \begin{bmatrix} \Phi_o^*(k)^T \\ \Phi_o(k+1)^T \end{bmatrix} r_o(k) \\ &+ \begin{bmatrix} \Phi_o^*(k)^T M_o \Phi_o^*(k) \\ \Phi_o(k+1)^T M_o \Phi_o(k) \\ \Phi_o^*(k)^T M_o \Phi_o(k+1) \\ \Phi_o(k+1)^T M_o \Phi_o(k+1) \end{bmatrix} \begin{bmatrix} v^*(k) \\ v(k+1) \end{bmatrix} \end{aligned} \quad (56-b)$$

와 같이 나타낼 수 있으며, 이를 시스템적으로 도시하면 그림 3과 같이 폐환이 있는 非線型 時變 시스템으로 나타낼 수 있다. 식 (56)과 그림 3에서 처리된 오차 신호 $v^*(k)$ 와 $v(k)$ 가 0으로 수렴하는지를 조사하기 위해 다음의 lemma를 이용한다.

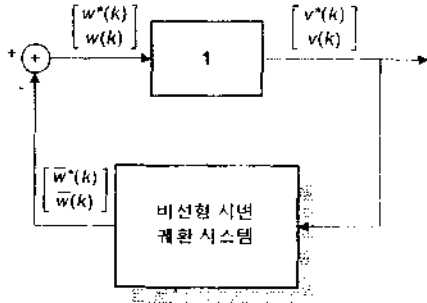


그림 2. $v^*(k)$, $v(k)$ 와 $w^*(k)$, $w(k)$ 에 대한 폐환 시스템
Fig. 3. Feedback system representation for $v^*(k)$, $v(k)$ and $w^*(k)$, $w(k)$

lemma :

아래와 같은 線形 時變離散 시스템(discrete-time linear time-varying system)이 있다고 하자.

$$x_{k+1} = A_k x_k + B_k u_k \quad (57-a)$$

$$v_k = C_k x_k + J_k u_k \quad (57-b)$$

- x_k : 상태 벡터($n \times 1$)
- u_k : 입력 벡터($m \times 1$)
- v_k : 출력 벡터($m \times 1$)

만일 아래의 조건

$$A_k^T P_{k+1} A_k - P_k = -Q_k \quad (58-a)$$

$$C_k - B_k^T P_{k+1} A_k = S_k^T \quad (58-b)$$

$$J_k + J_k^T - B_k^T P_{k+1} B_k = R_k \quad (58-c)$$

을 만족하는 時變 對稱행렬들인 P_k , Q_k , R_k 및 時變 행렬 S_k 가 존재한다면, 입력 벡터와 출력 벡터의 內積인 $v_k^T u_k$ 의 합은 다음과 같이 표시할 수 있다.

$$\begin{aligned} \sum_{k=K_0}^{K_1} v_k^T u_k &= \frac{1}{2} x_{K_1}^T P_{K_1}^{-1} x_{K_1} - \frac{1}{2} x_{K_0}^T P_{K_0}^{-1} x_{K_0} \\ &+ \frac{1}{2} \sum_{k=K_0}^{K_1} (x_k^T Q_k x_k + 2u_k^T S_k x_k + u_k^T R_k u_k) \end{aligned} \quad (59)$$

이 (59) 상태방정식의 두 변수를 분리 lemma에 적용된 변수들과 각각 다음과 같이 대응시킬 수 있다.

$$x_{k+1} = r_o(k+1), \quad u_k = \begin{bmatrix} v^*(k) \\ v(k+1) \end{bmatrix}, \quad v_k = \begin{bmatrix} \overline{w^*(k)} \\ \overline{w(k+1)} \end{bmatrix}$$

$$A_k = 1, \quad B_k = M_o [\Phi_o^*(k), \Phi_o(k+1)], \quad C_k = \begin{bmatrix} \Phi_o^*(k)^T \\ \Phi_o(k)^T \end{bmatrix}$$

$$J_k = \begin{bmatrix} \Phi_o^*(k)^T M_o \Phi_o^*(k) & \Phi_o^*(k)^T M_o \Phi_o(k+1) \\ \Phi_o(k)^T M_o \Phi_o(k+1) & \Phi_o(k+1)^T M_o \Phi_o(k+1) \end{bmatrix}$$

또한 $P_{k+1} = P_k = \dots = M_o^{-1}$ 로 하면 각각 다음과 같이 R_k , Q_k , S_k 를 얻을 수 있다.

$$Q_k = 0, \quad S_k = C_k - B_k^T P_{k+1} A_k = 0,$$

$$J_k + J_k^T - B_k^T P_{k+1} B_k = 2J_k - J_k = J_k = R_k$$

이상에서 얻은 식들을 식 (59)에 대입하면 다음과 같은 결과식을 구할 수 있다.

$$\begin{aligned} \sum_{k=K_0}^{K_1} \{ \overline{w^*(k)} v^*(k) + \overline{w(k+1)} v(k+1) \} \\ = \frac{1}{2} r_o(K_1+1)^T M_o^{-1} r_o(K_1+1) - \frac{1}{2} r_o(0)^T M_o^{-1} r_o(0) \\ + \frac{1}{2} \sum_{k=K_0}^{K_1} [v^*(k), v(k+1)] J_k \begin{bmatrix} v^*(k) \\ v(k+1) \end{bmatrix} \\ \geq -\frac{1}{2} r_o(0)^T M_o^{-1} r_o(0) \\ = -\gamma_o^2 \end{aligned} \quad (60)$$

($\because J_k$ is positive definite matrix 이므로)

식 (51)과 (52)로부터 $w^*(k) = v^*(k)$ 및 $w(k) = v(k)$ 이므로 식 (60)은 최종적으로 다음과 같이 표현된다.

$$\sum_{k=K_0}^{K_1} \{ v^*(k)^2 + v(k+1)^2 \} \leq \frac{1}{2} r_o(0)^T M_o^{-1} r_o(0) = \gamma_o^2 \quad (61)$$

즉 $K_1 \rightarrow \infty$ 가 됨에 따라 $v^*(k)$ 와 $v(k)$ 는 모두 0으로 수렴하게 된다.

IV. 전산 시뮬레이션

본 논문에서 제시한 알고리즘과 ACHARF 알고리즘의 수렴 상태를 비교하고자 한다. 본 논문의 제시된 알고리즘과 ACHARF 알고리즘 공히 적용 이득은 $\mu_1 = \mu_2 = \dots =$

$\alpha_1 = \mu, \alpha_2 = \dots = \alpha_N = \rho, Y_1 = Y_2 = \dots = Y_N, Y$ 로 한다. 오차 신호이다. 여기서 k 가인 $D(k) = |\Theta_{opt} - \Theta_{est}(k)|$ 이므로 정규화 오차 값을 수렴 상태를 판별하는 기준으로 사용한다.

(1) 시뮬레이션 1

표 1의 시스템의 진동 함수는 표 1과 같이 설정하였으며, 오차 값을 $n(k) = 0$ 으로 하였다.

표 1. 시뮬레이션 1의 환경

기준 시스템	$A(z) = 1.7z^{-1} - 0.7225z^{-2}, B(z) = 0.25$ $\sigma_x = 1.0$
잡음	$n(k) = 0$
적용 이득	ACHARF: $\mu = \rho = 0.01, \gamma = 0.001$ 제안 알고리즘: $\mu = \rho = 0.01, \gamma = 0.001$

두 알고리즘의 수렴 상태는 그림 4와 같다. 본 논문의 알고리즘이 ACHARF 알고리즘에 비해 같은 크기의 적용 이득하에서 약 2배 정도의 수렴 속도를 보임을 알 수 있다.

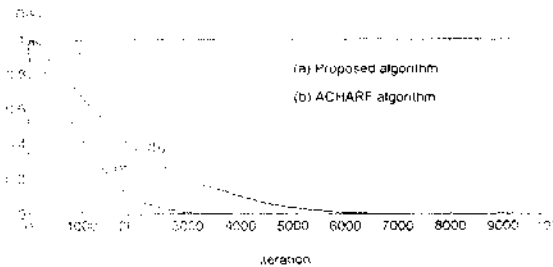


그림 4. 시뮬레이션 1의 결과.
Fig. 4. Result of simulation 1.

(2) 시뮬레이션 2

시뮬레이션 1과 같은 기준 시스템을 사용하였으며, 잡음 $n(k)$ 는 진폭이 1인 랜덤 신호로 하였으며, 두 알고리즘이 같은 수렴 속도를 갖는 조건에서 정상 상태에서의 편차적 수렴 정도를 알아보기 위해 ACHARF 알고리즘의 적용 이득을 본 논문의 알고리즘 보다 2배로 하였으며 (표 2) 그 결과는 그림 5와 같다. 동일한 수렴 속도에서는 본 논문의 알고리즘이 ACHARF 알고리즘에 비해 정상 상태에서의 편차적 수렴 정도가 낮은 것을 알 수 있다.

표 2. 시뮬레이션 2의 환경

기준 시스템	$A(z) = 1.7z^{-1} - 0.7225z^{-2}, B(z) = 0.25$ $\sigma_x = 1.0$
잡음	$n(k): \sigma_n = 1.0$
적용 이득	ACHARF: $\mu = \rho = 0.01, \gamma = 0.001$ 제안 알고리즘: $\mu = \rho = 0.02, \gamma = 0.002$

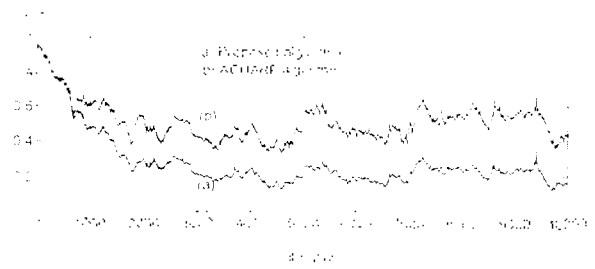


그림 5. 시뮬레이션 2의 결과.
Fig. 5. Result of simulation 2.

(3) 시뮬레이션 3

표 3과 같은 환경에서 시뮬레이션을 수행하였으며, 두 알고리즘의 수렴 속도를 같게한 상태에서 정상 상태에서의 편차적 수렴 정도를 알아 보았다. 이를 위해 ACHARF 알고리즘의 적용 이득을 10배로 하였다. 그 결과는 그림 6과 같다. 두 알고리즘의 수렴 속도가 동일한 상황에서는 본 논문의 알고리즘이 정상 상태에서 훨씬 작은 편차를 보임을 알 수 있다.

표 3. 시뮬레이션 3의 환경

기준 시스템	$A(z) = 2.4z^{-1} - 2.678z^{-2} + 2.355z^{-3}$ $- 2.481z^{-4} + 2.0537z^{-5} - 0.7865z^{-6}$ $B(z) = 0.25$ $\sigma_x = 1.0$
잡음	$n(k): \sigma_n = 10^{-2}$
적용 이득	ACHARF: $\mu = \rho = 0.3, \gamma = 0.03$ 제안 알고리즘: $\mu = \rho = 3, \gamma = 0.3$

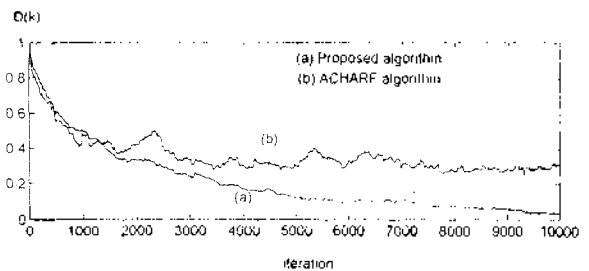


그림 6. 시뮬레이션 3의 결과.
Fig. 6. Result of simulation 3.

V. 결 론

적용 순환 필터에서 가장 문제가 되는 안정성을 보장 하기 위해 초안정성을 이용한 알고리즘을 제시하였다. 이 알고리즘은 기존의 초안정성 적용 순환 필터인 ACHARF 알고리즘에 비해, 동일한 조건하에서는 약 2배 이상으로 수렴 속도가 개선되었으며, 같은 수렴 속도하에서는 정상 상태에서의 편차적 수렴 정도가 2배 이상 작음을 확인 하였다. 특히 적용 순환 필터의 극점에 대한 수렴 편차는

전체 시스템에 매우 민감한 영향을 주기 때문에 정상 상태에서 편차적 수렴이 적다는 사실은 적응 순환 필터에서 매우 중요하다고 생각된다. 전체 알고리즘의 계산량은 ACHARF 알고리즘에 비해 약 2배 정도이나, 적응 순환 필터의 자수는 일반적으로 매우 낮기 때문에 이 계산량의 증가된 실제 상황에서는 문제가 되지 않을 것으로 판단된다.

이상으로 본 연구에서 제시한 알고리즘은 초안정성을 바탕으로 하였으므로 수렴성이 보장된 뿐 만아니라, 상대적으로 빠른 수렴 속도를 얻을 수 있으며 정상 상태에서의 편차적 수렴 정도도 상대적으로 낮은 유용한 알고리즘으로 판단된다.

참 고 문 헌

1. V.M. Popov, *Hyperstability of Automatic Control Systems*, Springer, New York, 1973.
2. M.G. Larimore, J.R. Trichler and C.R. Johnson, Jr., "SHARF: an algorithm for adapting IIR digital filters." *IEEE Trans. Acoust. Speech Signal Processing* vol. ASSP-28, p.428, August 1980.
3. C.R. Johnson, Jr., "A Convergency proof for a hyperstable adaptive recursive filter," *IEEE Trans. Information Theory*, vol. IT-2 no.6, November 1978.
4. C.R. Johnson, Jr., M.G. Larimore, J.R. Treichler and B.D. O. Anderson, "SHARF convergence properties," *IEEE Trans. Acoust. Speech Signal Processing* vol. ASSP-29, p. 659, June 1981.
5. H. Fan, W.K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. Circuits Syst.*, vol. CAS-33, no.10, October 1986.
6. J.R. Treichler, C.R. Johnson, Jr., M.G. Larimore, *Theory and Design of Adaptive Filters*, Texas Instruments, Inc., 1987, p.113-156.
7. 윤병우, 신윤기, "適應 補償器를 채용한 超安定性 適應 循環 필터." 1990년 3월 電子工學會論文誌 第27卷 第3號, p. 145-155.

▲신 윤 기(Yoon Ki Shin)

1957년 1월 2일생



1979년 2월: 서울대학교 전자공학과 졸업

1982년 2월: 서울대학교 전자공학과 석사학위 취득

1987년 2월: 서울대학교 전자공학과 박사학위 취득

1985년 3월~1992년 2월: 부산대학교 전자공학과 교수로 근무

1992년 3월~현재: 대전대학교 전자공학과 교수로 재직중
※주관심분야: 신호처리, 적응필티이론, 수중통신 등임.