

가변 망각인자를 사용한 커널 RLS 알고리즘

임준석*, 편용국^o

Kernel RLS Algorithm Using Variable Forgetting Factor

Jun-Seok Lim*, Yong-Guk Pyeon^o

요약

최근 KRLS-T라고 하는 커널 기반의 RLS 알고리즘이 제안되었다. 이 알고리즘은 비정재 환경에서 베이저안 이론에 기반한 망각 기저를 사용하고 있다. 이런 망각 기저의 핵심은 고정 망각인자를 사용하는 것이다. 그러나 실제 환경에서는 시스템의 비정재 정도까지도 시변인 경우가 많다. 이 경우 고정 망각인자로는 시스템의 바른 추정이 불가하다. 본 논문에서는 가변 망각인자를 사용한 KRLS-T를 제안한다. 그리고 모의 실험에서 가변 망각인자를 가진 KRLS-T가 시변 시스템의 추정에 더 효과적임을 보인다.

Key Words : RLS, KRLS, KRLS-T, variable forgetting factor

ABSTRACT

In a recent work, kernel recursive least-squares tracker (KRLS-T) algorithm has been proposed. It is capable of tracking in non-stationary environments using a forgetting mechanism built on a Bayesian framework. The forgetting mechanism in KRLS-T is implemented by a fixed forgetting factor. In practice, however, we frequently meet that the fixed forgetting factor cannot handle time-varying system effectively. In this paper we propose a new KRLS-T with a variable forgetting factor. Experimental results show that proposed algorithm can handle time-varying system more effectively than the KRLS-T.

1. 서론

RLS(Recursive Least-Squares) 알고리즘은 신호처리 분야에서 가장 일반적인 적응 필터 중의 하나이다^[1-5]. 이것은 반복적인 방법으로 정재 시스템의 최소자승 선형 예측을 계산해낸다. 표준 RLS는 망각인자를 사용하여 비정재 시스템으로 그 응용을 확장할 수 있다.

최근 커널을 사용한 적응 필터 알고리즘이 활발하게 연구되고 있다^[6,7]. SVM (Support Vector Machine)과 커널 주성분 분석 (KPCA) 등이 성공적

인 커널 사용 적응 필터 예들이다^[8,9]. 커널을 사용하는 것은 커널에 의해서 데이터가 고차원의 커널 힐버트 공간으로 투영되는 것을 이용하여 비선형 문제를 해결할 수 있는 가능성을 갖고 있어서 주목 받고 있다. 이런 알고리즘 중에 RLS의 커널 버전인 KRLS(Kernel Recursive Least-Squares)가 있다. 당초 KRLS은 정재신호에 대해서 사용하도록 제안되었다. 그리고 신호처리, 통신 및 패턴 인식 등의 분야에서 성공적으로 사용되고 있다^[6,7]. KRLS에 시변 신호 같은 비정재 신호에 대응하게 하기 위해서는 추가 정보가 필요하다. KRLS를 시변 신호에 대응하기 위해서

* First Author : Sejong University, Dept. of Electrical Engineering, jslim@sejong.ac.kr, 종신회원

^o Corresponding Author : GangWon Provincial College, Dept. of Information and Communication, pyk12@naver.com, 정회원
논문번호 : KICS2015-06-178, Received June 9, 2015; Revised August 12, 2015; Accepted August 20, 2015

제안된 알고리즘에는 이동형 고정 창 커널 RLS (sliding window KRLS, SW-KRLS)와 커널형 확장 RLS (kernel-based EX-KRLS)가 있다^{7,10)}.

참고문헌^{11,12)}에 확률론적 접근법을 사용해서 KRLS를 시변 신호에 대응하게 한 알고리즘이 발표되었다. 이 알고리즘은 새로운 형식의 망각인자를 KRLS에 적용하였다. KRLS-T라고 불리는 이 방법은 가우시안 과정 시계열 분석(Gaussian Process Regression, GP Regression)에 뿌리를 둔 접근법을 사용한다. 당초 가우시안 과정 시계열 분석에 뿌리를 둔 방법들은 배치 처리(batch process)에 사용할 수 있도록 개발되었다. 반면에 KRLS는 새 데이터가 들어올 때마다 처리를 하는 온라인 처리를 염두하고 개발된 알고리즘이다. KRLS로 모든 데이터를 차례로 다 처리하고 나면, 그 결과는 모든 데이터를 배치 작업을 해서 만들어진 가우시안 과정 시계열 분석의 결과와 같다. 이런 등가 결과는 참고문헌¹¹⁻¹⁴⁾에 나와 있다. 따라서 KRLS는 온라인 가우시안 과정 시계열 분석이라고 할 수 있다¹¹⁻¹³⁾.

KRLS-T 알고리즘은 고정된 망각인자를 사용한다. 고정된 망각인자는 불필요한 정보를 버리는 범위가 고정되어 있다. 이 같은 성질은 시스템의 시변성에 대한 정보가 충분하고, 또 그 성질이 추정 중에 변하지 않는다면 고정된 망각인자를 사용하여 우수한 추정을 얻는 것이 가능하다. 그러나, 시스템 시변성 자체가 수시로 변한다면 하나의 고정된 망각인자로 전체 시간 동안 우수한 추정을 하는 것이 불가능하다. 따라서 상황에 맞게 망각인자를 알맞게 변경하는 것이 바람직하다. 이를 가변 망각인자라고 한다¹⁵⁻¹⁸⁾.

선형 추정법 분야에서는 망각인자를 변경하는 여러 방법이 제안되어 있다¹⁵⁻¹⁸⁾. 그러나 이들 방법은 망각인자를 구하기 위하여 행렬을 미분과 그의 역행렬이 포함되어 있어서, 추정 벡터 차수의 자승 이상의 계산량이 추가된다. 따라서 KRLS-T에 직접 적용하기에는 심각히 고려해야할 점이 존재한다. 즉, KRLS는 커널을 사용하여 입력 벡터를 다른 공간으로 매핑하는 비선형성을 포함하고 있어서 행렬 미분을 포함하는 알고리즘을 시도할 경우 기존 선형 RLS에서 보다 복잡도가 매우 커질 것이 예상된다. 본 논문에서는 행렬 미분을 사용하지 않아 계산량에 부담을 최소화 한 가변 망각인자 방법을 채용한 KRLS-T를 제안한다. 그리고 이 가변 망각인자를 사용한 새로운 KRLS-T가 시변 환경에서 잘 동작한다는 것을 보이기 위하여 전체 추정 기간 중에 시스템이 변경되는 경우를 상정하여 모의 실험을 한다. 뿐만 아니라 임펄스 잡음 상황

이 주는 추정 시스템의 갑작스런 외란에 의한 시변 상황에도 제안한 알고리즘이 적용 가능성도 보인다.

II. 본 론

2.1 가우시안 과정 정리 (Gaussian Process (GP))

본 장에서는 GP regression에 관한 간단히 설명을 한다¹⁵⁾. 입력력 쌍의 집합 $D_i = x_i, y_{i=1}^n$ 이 i.i.d. (independent identical distributed)일 때, Regression 이 하는 일은 새로운 입력벡터 X_* 가 주어졌을 때 해당하는 출력 y_* 을 예측하는 것이다. GP regression은 출력을 내재함수(latent function) $f(X_i)$ 와 부가 잡음 ε_i 의 합으로 모델한다.

$$y_i = f(X_i) + \varepsilon_i \quad (1)$$

여기서 $f(X) = N(0, k(X, X'))$ 이고, $k(X, X')$ 는 상호상관 함수이다. $\varepsilon_i = N(0, \sigma^2)$.

출력 $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ 과 이에 대응하는 latent 함수 벡터 $\mathbf{f} = [f(x_1), f(x_2), \dots, f(x_n)]^T$ 와의 joint 분포는 다음과 같다¹⁵⁾.

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f} \end{bmatrix} = \left(0, \begin{bmatrix} \mathbf{K} + \sigma^2 \mathbf{I} & \mathbf{K} \\ \mathbf{K} & \mathbf{K} \end{bmatrix} \right) \quad (2)$$

\mathbf{K} 는 $[K]_{ij} = k(X_i, X_j)$ 인 행렬이다. 그리고 \mathbf{I} 는 \mathbf{K} 와 같은 사이즈의 단위 행렬이다. 출력 행렬 \mathbf{y} 가 주어졌을 때 latent 함수 벡터에 대한 사후 조건 확률 분포함수는

$$\begin{aligned} p(\mathbf{f}|\mathbf{y}) &= N(\mathbf{f} | \mathbf{K}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, \\ &\quad \mathbf{K} - \mathbf{K}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}) \\ &= N(\mathbf{f} | \boldsymbol{\mu}, \boldsymbol{\Sigma}), \end{aligned} \quad (3)$$

이다. 위 확률 분포 파라미터를 계산할 때 제일 많은 계산량이 드는 부분은 크기가 $n \times n$ 인 행렬 $(\mathbf{K} + \sigma^2 \mathbf{I})$ 의 역행렬을 계산할 때이다. 이 경우 $O(n^3)$ 정도의 복잡도를 갖는다. 위 확률분포의 각 파라미터들의 차원은 새로운 입력 샘플이 도래할 때마다 점진적으로 늘어난다. 따라서 계산 복잡도는 시간에 따라서 급격하게 늘어나게 된다.

출력 행렬 \mathbf{y} 가 주어졌을 때 latent 함수 벡터에 대

한 조건 확률 분포함수가 $N(\mathbf{f}|\boldsymbol{\mu}, \Sigma)$ 로 주어졌을 때, 새로운 출력에 대한 조건 확률 분포함수는 다음과 같다^[19].

$$p(y_*|\mathbf{y}) = \int p(y_*|f_*)p(f_*|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}df_* \quad (4)$$

$$= N(y_*|\mathbf{q}^T\boldsymbol{\mu}, \sigma^2 + \gamma^2 + \mathbf{q}^T\Sigma\mathbf{q}),$$

여기서, $\mathbf{q} = \mathbf{K}^{-1}\mathbf{k}_*$ 이고 $\gamma^2 = k_{**} - \mathbf{k}_*^T\mathbf{K}^{-1}\mathbf{k}_*$ 이다. \mathbf{k}_* 는 요소 값 $k(\mathbf{X}_i, \mathbf{X}_*)$ 들로 이루어진 벡터이다. 그리고 $k_{**} = k(\mathbf{X}_*, \mathbf{X}_*)$ 이다.

위 식에 $p(\mathbf{f}|\mathbf{y}) = N(\mathbf{f}|\boldsymbol{\mu}, \Sigma)$ 를 대입하여 다시 계산하면 아래와 같은 확률 밀도 함수를 얻을 수 있다.

$$p(y_*|\mathbf{x}_*, \mathbf{y}) = N(y_*|\mu_{GP^*}, \sigma_{GP^*}^2), \quad (5a)$$

$$\mu_y = \mathbf{k}_*^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, \quad (5b)$$

$$\sigma_y^2 = \sigma^2 + k_{**} - \mathbf{k}_*^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_*. \quad (5c)$$

2.2 Gaussian Process(GP)의 순차적 갱신 방법

많은 신호처리 응용에서 데이터가 연속으로 도래한다. 따라서 추정 알고리즘은 매번 새로운 데이터가 도래할 때마다 추정 값을 갱신하는 것이 바람직하다. 서론에서도 언급했듯이 GP는 비실시간 방식인 배치 형식으로 사용할 것을 전제로 만들어졌다. 따라서 근본적인 변경 없이 연속으로 도래하는 데이터에 대응하여 GP의 통계 파라미터를 계산하려면 차원이 점점 더 커지는 행렬, $(\mathbf{K} + \sigma^2\mathbf{I})$, 전체에 대해서 역행렬을 구하는 계산량이 많이 드는 불편함을 감수하여야 한다. 따라서 이전 샘플까지의 통계 파라미터 행렬 결과를 충분히 이용하는 방식으로 접근하여 계산량을 줄이는 방식이 필요하다. 이를 on-line GP라고 칭한다^[20].

만약 n 번째 샘플까지 사용하여 추정이 끝난 상태에서 새로이 $n+1$ 번째 샘플, \mathbf{x}_{n+1} , 을 받았다면, 부분 행렬의 역행렬 계산법과 Woodbury 등가를 이용하여 다음과 같이 순차적으로 $n+1$ 번째 출력 샘플, y_{n+1} , 에 관한 확률 밀도 함수의 파라미터를 구하기 위해서 다음과 같은 계산량이 적은 갱신식을 사용할 수 있다. 즉, n 번째 샘플에 대한 행렬 $(\mathbf{K} + \sigma^2\mathbf{I})$ 를 $\mathbf{C}_n = (\mathbf{K}_n + \sigma^2\mathbf{I}_n)$ 이라 할 때,

$$\mathbf{C}_{n+1}^{-1} = \begin{bmatrix} \mathbf{C}_n^{-1} + \mathbf{C}_n^{-1}\mathbf{k}_{n+1}\mathbf{k}_{n+1}^T\mathbf{C}_n^{-1}/\sigma_{yn+1}^2 & \mathbf{C}_n^{-1}\mathbf{k}_{n+1}/\sigma_{yn+1}^2 \\ -\mathbf{k}_{n+1}^T\mathbf{C}_n^{-1}/\sigma_{yn+1}^2 & 1/\sigma_{yn+1}^2 \end{bmatrix} \quad (6)$$

$\sigma_{yn+1}^2 = \sigma^2 + k_{n+1n+1} - \mathbf{k}_{n+1}^T(\mathbf{K}_{n+1} + \sigma^2\mathbf{I}_{n+1})^{-1}\mathbf{k}_{n+1}$ 이고,

$\mathbf{k}_{n+1} = [k(\mathbf{x}, \mathbf{x}_{n+1}) \ k(\mathbf{x}_2, \mathbf{x}_{n+1}) \ \dots \ k(\mathbf{x}_{n+1}, \mathbf{x}_{n+1})]$ 이다. 식(1)에서 보면 출력 신호 y 의 평균과 그의 해당하는 latent 함수 $f(\mathbf{x})$ 의 평균은 같다고 할 수 있다. 이를 토대로 n 번째까지 확보한 입력 샘플들에 대한 latent 함수 $f(\mathbf{x})$ 들을 벡터로 묶은 새로운 벡터에 대한 평균도 식(5b)를 확장 이용하여 다음과 같이 에 대한 구할 수 있다.

$$\boldsymbol{\mu}_n = \mathbf{K}_n^T\mathbf{C}_n^{-1}\mathbf{y}_n. \quad (7)$$

이는 식(3)에서도 다시 확인 할 수 있다. 또한 식(3)의 공분산 행렬을 다시 n 번째까지 확보한 입력 샘플들에 대한 latent 함수 들을 벡터로 묶은 새로운 벡터에 대한 것을 다시 생각하면 다음과 같다.

$$\Sigma_n = \mathbf{K}_n - \mathbf{K}_n\mathbf{C}_n^{-1}\mathbf{K}_n. \quad (8)$$

또 이 경우도 새로운 데이터 쌍 이 도래했을 경우 새로운 평균과 공분산 행렬의 갱신식을 다음과 같이 쓸 수 있다.

$$\boldsymbol{\mu}_{n+1} = \begin{bmatrix} \boldsymbol{\mu}_n \\ \mu_{f(\mathbf{x}_{n+1})} \end{bmatrix} + \frac{\mu_{f(\mathbf{x}_{n+1})} - y_{n+1}}{\sigma_{yn+1}^2} \begin{bmatrix} \mathbf{h}_{n+1} \\ \sigma_{f(\mathbf{x}_{n+1})} \end{bmatrix}. \quad (9)$$

$$\Sigma_{n+1} = \begin{bmatrix} \Sigma_n & \mathbf{h}_{n+1} \\ \mathbf{h}_{n+1}^T & \sigma_{f(\mathbf{x}_{n+1})}^2 \end{bmatrix} - \frac{1}{\sigma_{yn+1}^2} \begin{bmatrix} \mathbf{h}_{n+1} \\ \sigma_{f(\mathbf{x}_{n+1})}^2 \end{bmatrix} \begin{bmatrix} \mathbf{h}_{n+1} \\ \sigma_{f(\mathbf{x}_{n+1})}^2 \end{bmatrix}^T. \quad (10)$$

여기서

$$\mathbf{h}_{n+1} = \Sigma_n\mathbf{K}_n^{-1}\mathbf{k}_{n+1} = (\mathbf{I}_n - \mathbf{K}_n\mathbf{C}_n^{-1})\mathbf{k}_{n+1}$$

2.3 시변 신호추정을 위한 망각인자 도입을 위한 기존의 방법

식(9)와 같은 갱신식은 잘 알려진 KRLS와 등가임이 알려져 있다. 따라서 KRLS는 GP regression이 제공하는 평균과 공분산 중에서 평균만을 추정하는 것임도 알 수 있다. 기존 KRLS는 시불변 시스템을 위한 on-line 방법이었다. 즉 latent 함수가 시불변이라는 가정을 한 것이다. 그러나 많은 신호처리 분야의 신호가 시변인 성질을 갖는 경우가 많다. 이를 위해서 몇몇 KRLS의 변형이 생겼다. 이런 방법들 중에는 extended RLS를 도입한 EX-KRLS^[7]와 사용하는 데이터 구역을 현재 샘플을 중심으로 제한된 구역만 사

용하도록 하는 sliding-window KRLS^[10]가 대표적인 예가 될 수 있다. On-line GP에도 시변 신호를 위한 적응성을 도입하기 위해서 망각인자를 다음과 같이 도입하였다.

$$\mu \leftarrow \sqrt{\lambda} \mu. \tag{11a}$$

$$\Sigma \leftarrow \lambda \Sigma + (1 - \lambda) K. \tag{11b}$$

위와 같은 망각인자를 도입한 On-line GP 방법을 KRLS-T(KRLS Tracker)라고 알려져 있다^[21].

III. 시변 신호 추정을 위한 성능 향상 방법들

KRLS-T 알고리즘은 고정된 망각인자를 사용한다. 고정된 망각인자는 망각을 하는 범위가 제한되어 있다. 이는 시스템의 시변성에 대한 정보가 충분하고, 또 그 성질이 변하지 않는다면 고정된 망각인자를 적용하는 것이 가능하지만, 시스템에 대한 정보가 충분하지 않으면 알맞은 망각인자를 미리 정하기 어렵다. 뿐만 아니라 시스템 시변성 자체가 수시로 변한다면 망각인자를 변한 시스템 특성에 알맞게 변경하는 것이 바람직하다. 이를 가변 망각인자라고 한다^[15-18]. 선형 추정법에서는 이 망각인자를 변경하는 여러 방법이 제안되어 있다^[15-18]. 그러나 이들 방법은 망각인자에 대해서 행렬을 미분하는 것이 포함되어 있어서 KRLS-T에 직접 적용하기는 어렵다. 본 논문에서는 행렬 미분을 사용하지 않는 방법들 중에서 KRLS-T에 적용 가능한 방법들을 골라 적용한다.

방법1^[22]
$$\lambda(n+1) = \lambda_{\min} + (1 - \lambda_{\min}) \times 2^{L(n)},$$

$$L(n) = -\text{NINT}(\rho 1e^2(n)). \tag{12}$$

여기서 $e(n) = y_n - \hat{y}_n$, $\text{NINT}()$ 는 가까운 정수로 매핑하는 함수이고, $\rho 1$ 는 조절상수이다. 본 방법은 추정 오차의 크기가 커지면 망각인자가 λ_{\min} 에 가깝게 되고 오차가 작아지면 망각인자가 1에 가깝게 된다.

방법2^[23]
$$\lambda(n+1) = [1 - 1/N(n)],$$

$$N(n) = \frac{\sigma_e^2 N_{\max}}{Z(n)}, Z(n) = \frac{1}{Q} \sum_{q=0}^{Q-1} |e(n-q)|^2. \tag{13}$$

이 방법은 처음부터 현재까지 추정 오차의 분산값과 가까운 과거부터 현재까지 추정 오차의 분산값의 비가 1에 가까우면 망각인자를 1에 가깝게 하고 비가

1보다 크면 망각인자 값이 1보다 작게 된다. 따라서 변화가 거의 없는 경우 망각인자가 1에 가깝게 되고 변화가 있는 경우 망각인자를 1보다 작은 값으로 줄이게 된다.

방법3 위의 방법2 중 망각인자를 계산하는 것을 변경한 방법이다.

$$\lambda(n+1) = 1/1 + e^{-\rho 3 N(n)},$$

$$N(n) = \frac{\sigma_e^2 N_{\max}}{Z(n)}, Z(n) = \frac{1}{Q} \sum_{q=0}^{Q-1} |e(n-q)|^2, \tag{14}$$

여기서 $\rho 3$ 는 조절상수이다. 위 세 가지 방법을 식(11a)와 식(11b)에 적용하면 측정된 출력값과 추정된 출력값을 차이로부터 시스템의 시변성을 유추하고 그로부터 망각인자를 조절하여 추정값과 측정값 간의 차이를 줄이도록 할 수 있다. 이 방법은 방법2를 조금 수정한 것으로써 방법2에서 채택한 망각인자 변화 방식을 sigmoid 함수에 적용하여 망각인자 가변을 좀더 자연스럽게 하려고 한 방법이다. 이 방법에서는 $\rho 3$ 상수에 의해서 변화 속도를 조절할 수 있다. 다음 표1에는 제안된 가변 망각인자를 적용한 KRLS-T에 대한 알고리즘을 정리하였다.

표 1. 알고리즘 정리
Table 1. Algorithm summary

<p>Initialization: $\mu_1, \Sigma_1, K_1, Q_1 = K_1^{-1}, \lambda_1, \lambda_{\max}, \lambda_{\min}$, Selction of Kernel type.</p> <p>for n=1,2, ...</p> <p>$\mu_n \leftarrow \sqrt{\lambda_n} \mu_n.$</p> <p>$\Sigma_n \leftarrow \lambda_n \Sigma_n + (1 - \lambda_n) K_n.$</p> <p>use a new input data vector \mathbf{x}_{n+1},</p> <p>$\mathbf{k}_{n+1} = [k(\mathbf{x}, \mathbf{x}_{n+1}) \ k(\mathbf{x}_2, \mathbf{x}_{n+1}) \dots k(\mathbf{x}_{n+1}, \mathbf{x}_{n+1})]$</p> <p>$k_{n+1} = k(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}).$</p> <p>$\mathbf{q}_{n+1} = Q_n \mathbf{k}_{n+1}.$</p> <p>$\mathbf{h}_{n+1} = \Sigma_n \mathbf{q}_{n+1}.$</p> <p>$\sigma_{f(\mathbf{x}_{n+1})}^2 = k_{n+1, n+1}$ $\quad - \mathbf{k}_{n+1}^T (\mathbf{K}_{n+1} + \sigma^2 \mathbf{I}_{n+1})^{-1} \mathbf{k}_{n+1}.$</p> <p>$\sigma_{y_{n+1}}^2 = \sigma^2 + \sigma_{f(\mathbf{x}_{n+1})}^2.$</p> <p>$\hat{y}_{n+1} = \mathbf{k}_{n+1}^T \mu_n.$</p> <p>measure a new output $y_{n+1}.$</p> <p>$\mu_{n+1} = \begin{bmatrix} \mu_n \\ \hat{y}_{n+1} \end{bmatrix} + \frac{y_{n+1} - \hat{y}_{n+1}}{\sigma_{y_{n+1}}^2} \begin{bmatrix} \mathbf{h}_{n+1} \\ \sigma_{f(\mathbf{x}_{n+1})}^2 \end{bmatrix}.$</p>
--

```


$$\Sigma_{n+1} = \begin{bmatrix} \Sigma_n & \mathbf{h}_{n+1} \\ \mathbf{h}_{n+1}^T & \sigma_{f(\mathbf{x}_{n+1})}^2 \end{bmatrix} - \frac{1}{\sigma_{y_{n+1}}^2} \begin{bmatrix} \mathbf{h}_{n+1} \\ \sigma_{f(\mathbf{x}_{n+1})}^2 \end{bmatrix} \begin{bmatrix} \mathbf{h}_{n+1} \\ \sigma_{f(\mathbf{x}_{n+1})}^2 \end{bmatrix}^T$$


$$e(n) = y_{n+1} - \hat{y}_{n+1}$$

method1:

$$L(n) = -\text{NINT}(\rho_1 e^2(n)).$$


$$\lambda_{n+1} = \lambda_{\min} + (1 - \lambda_{\min}) \times 2^{L(n)}.$$

method2:

$$N(n) = \frac{\sigma_e^2 N_{\max}}{Z(n)}, Z(n) = \frac{1}{Q} \sum_{l=0}^{Q-1} |e(n-l)|^2$$


$$\lambda_{n+1} = [1 - 1/N(n)]. N_{\max} = 1/(1 - \lambda_{\max}).$$

where  $\sigma_e^2$  is the variation of the estimation error and  $W$  is the length of local estimation window.
method3:

$$N(n) = \frac{\sigma_e^2 N_{\max}}{Z(n)}, Z(n) = \frac{1}{Q} \sum_{q=0}^{Q-1} |e(n-q)|^2$$


$$\lambda_{n+1} = 1/1 + e^{-\rho_3 N(n)}.$$

the other coefficients are the same as those from method2.

if  $\lambda_{n+1} > \lambda_{\max}$ ,

$$\lambda_{n+1} = \lambda_{\max}.$$

else if  $\lambda_{n+1} > \lambda_{\min}$ 

$$\lambda_{n+1} = \lambda_{\min}.$$

endif

for_end

```

IV. 실험 및 평가

본 장에서는 제안된 가변 망각인자를 사용한 새로운 KRLS-T를 두 가지 대표적인 환경에서 실험하고 가변 망각인자의 유용성을 보인다. 두 가지 중 하나는 시변 비선형 채널 환경에서의 신호 추정 실험이고 또 하나는 첫 번째 실험에서 사용한 시변 비선형 채널 환경과 임펄스성 잡음이 함께 존재하는 신호 환경에서 신호를 추정하는 실험이다.

4.1 시변 비선형 환경에서 신호 추정 실험

비교실험을 위해서 시변 비선형 채널을 만들어 추정 실험을 하였다. 시변 비선형 채널은 선형 채널 뒤에 비메모리 비선형 함수가 있는 것으로 모델 하였다. 그리고 시변성을 주기 위해서 선형 채널 부분을 500

번째 샘플에서 B1 채널에서 B2 채널로 전환하였다.

$$B1 = [1, 0.8668, -0.4764, 0.2070],$$

$$B2 = [1, -0.8326, 0.6656, -0.7153],$$

비선형 함수 : $\tanh(\cdot)$.

앞 장에서 제안한 방법 1,2,3.에는 필요한 상수가 있다. 방법1에는 ρ_1 , 방법2에는 Q , 방법3에는 Q 와 ρ_3 들이 있다. 여기서 방법2와 방법3에 있는 Q 은 시간 변화 양상을 따라가기 위한 데이터 창 길이 이다. 이 길이는 망각인자에 의해 생기는 유효 데이터 길이로 부터 정할 수 있다. 본 실험에서는 방법2와 3에서 가변 망각인자의 변화 범위를 최소 0.7 최대 0.999로 제한하여 사용한다. 따라서 방법2와 방법3의 Q 와 N_{\max} 값은 최소 망각인자와 최대 망각인자로부터 $Q = 1/(1 - 0.7)$, $N_{\max} = 1/(1 - 0.999)$ 로 각각 설정하여 사용할 수 있다. 그리고 나머지 상수들은 표2와 같이 여러 시도를 통해서 좋은 것을 취하도록 한다. 표2는 KRLS-T의 망각인자와 방법1의 ρ_1 , 방법3의 ρ_3 들을 각각 정하기 위해서 전체 2000 샘플 중에서 채널이 갑자기 변하기 전인 1~500 샘플 구간 중 300번째 샘플과 499번째 샘플에서의 MSE (Mean Square Error)를 취하고 또 채널이 변한 후 1000번째 샘플과

표 2. MSE 비교를 통한 상수 선정 표
Table 2. Table of constant selection by comparing MSE.

		300 th sample	499 th sample	1000 th sample	last sample
KRLS-T	$\lambda=0.995$	0.0161	0.0159	0.0203	0.0210
	$\lambda=0.999$	0.0106	0.0077	0.0166	0.0110
	$\lambda=0.9995$	0.0078	0.0063	0.0255	0.0163
method1	$\rho_1=1$	0.0108	0.0073	0.0146	0.0106
	$\rho_1=2$	0.0106	0.0080	0.0140	0.0106
	$\rho_1=2.5$	0.0101	0.0074	0.0124	0.0099
	$\rho_1=3$	0.0100	0.0075	0.0130	0.0091
method3	$\rho_3=0.05$	0.0111	0.0075	0.0154	0.0106
	$\rho_3=0.1$	0.0093	0.0077	0.0167	0.0104
	$\rho_3=0.15$	0.0105	0.0080	0.0166	0.0108
	$\rho_3=0.2$	0.0112	0.0078	0.0167	0.0107
	$\rho_3=0.25$	0.0099	0.0074	0.0153	0.0091
	$\rho_3=0.3$	0.0100	0.0073	0.0150	0.0102
	$\rho_3=0.35$	0.0111	0.0081	0.0163	0.0102

마지막 샘플에서의 MSE를 성능 지표로 삼았다.

표2를 보면 KRLS-T의 망각인자는 망각인자가 클수록 1~500 샘플 구간에서의 성능이 좋으나 변화가 발생한 이후 구간을 보면 망각인자가 큰 것이 더 좋지는 않은 것을 알 수 있다. 이는 유효 구간 길이가 너무 길어져서 생기는 문제라고 생각할 수 있다. 방법1에서는 ρ_1 가 2.5나 3에서 상대적으로 좋은 결과가 나왔다. 방법3에서는 ρ_3 가 0.25일 때 제일 좋은 결과가 나왔다. 따라서 그 결과로부터 KRLS-T의 망각인자를 0.999를 선택하였다. 그리고 방법1의 ρ_1 는 3로 설정하였다. 방법3의 ρ_3 는 0.25로 설정하였다.

다음 그림은 3가지 가변 망각인자 방법을 적용한 결과들과 고정 망각인자를 사용하는 기존 KRLS-T를 적용한 결과들을 서로 비교한 그림이다.

그림 1을 보면 고정 망각인자를 사용하는 KRLS-T의 제일 좋은 결과인 그림 1(a)를 기준으로 하여 그림 1(b)~(d)를 평가 할 때, 세 가지 방법 모두 채널이 갑자기 변한 후에도 상대적으로 빠른 적응을 함을 보였다. 이는 표2에서 1000번째와 마지막 샘플의 MSE 값을 통해서도 알 수 있다. 3가지 방법 중에서는 방법1

과 방법3이 좀 더 빨리 수렴하고 있어서, 방법2에 비해서 상대적으로 좀 좋은 결과를 보인다.

4.2 임펄스성 잡음이 존재하는 시변 비선형 환경에서 신호 추정 실험

임펄스성 잡음은 추정기에 예외적으로 큰 오차를 일으키고 그 영향을 추후에도 오랫동안 영향을 미치는 경우가 비일비재하다. 따라서 임펄스성 잡음의 영향을 감소시킬 필요 있다. 본 실험에서는 가변 망각인자가 예외적으로 큰 오차가 일어났을 때 효과적으로 작동하여 빠르게 신호 추정기를 정상화 시킴을 보인다. 이를 위해서 실험 1에서 사용한 시변 비선형 채널을 그대로 사용하고 여기에 임펄스성 잡음 환경을 위하여 300번째 샘플과 1300번째 샘플에 임펄스 잡음을 추가하였다.

KRLS-T에서 임펄스 부가 효과를 보기 위해서 그림 2에서 임펄스 부가 전과 후를 비교하여 보았다. 그림 2 (a)는 임펄스 부가 전의 MSE 성능 곡선이다. 이에 반해서 그림 2(b)는 300번째와 1300번째 샘플에 임펄스를 부가한 경우이다. 그림에서 보는 바와 같이 300번째 임펄스 이후 추정 성능이 급격히 저하 되어

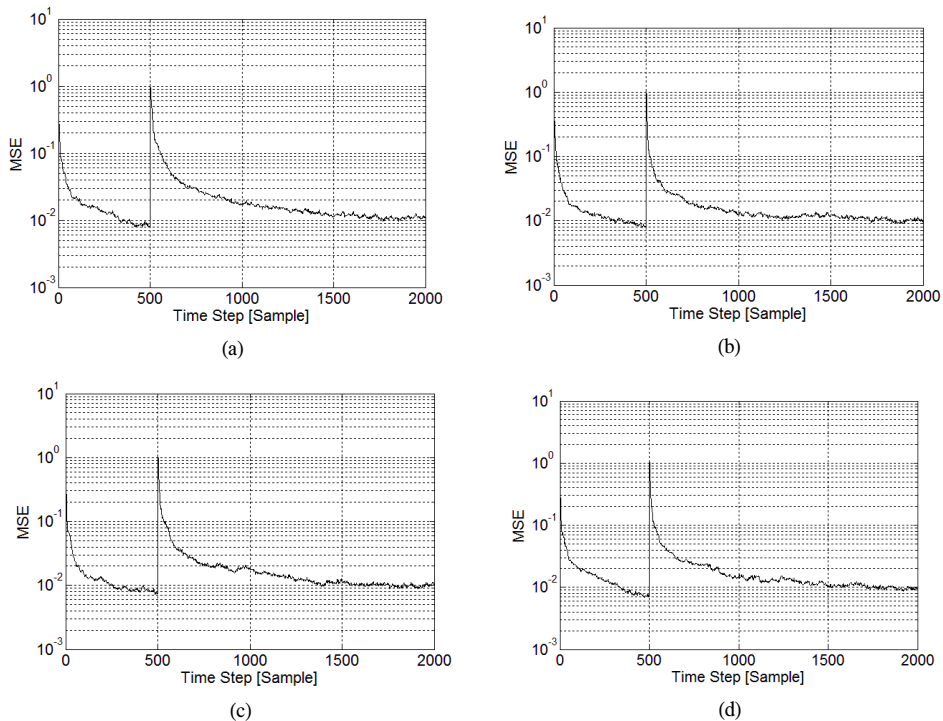
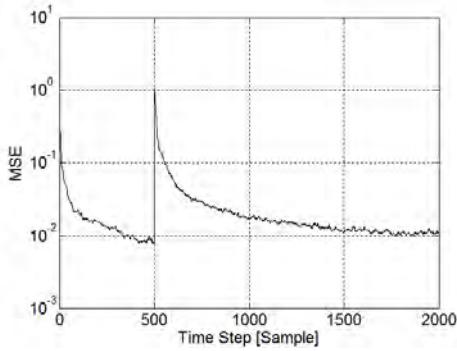
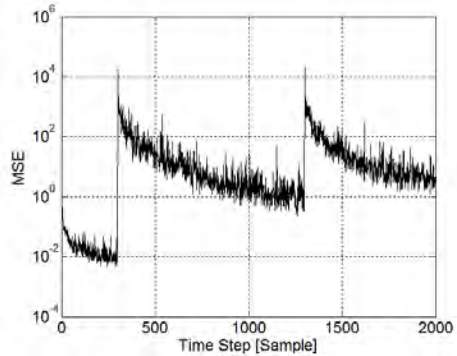


그림 1. 수렴성 비교 (a) KRLS Tracker (b) 방법1을 적용한 KRLS Tracker (c) 방법2을 적용한 KRLS Tracker (d) 방법3을 적용한 KRLS Tracker

Fig. 1. Compare convergence (a) KRLS Tracker (b) Applying the method 1 KRLS Tracker (c) Applying the method 2 KRLS Tracker (d) Applying the method 3 KRLS Tracker



(a)



(b)

그림 2. KRLS의 수렴성 비교 (a) 가우시안 부가 잡음 상황에서 KRLS Tracker의 추정 성능 (b) 임펄스 잡음 상황에서 KRLS Tracker의 추정 성능.

Fig. 2. Comparison of convergence KRLS (a) In addition KRLS Tracker estimation performance of Gaussian noise conditions (b) Estimated performance of impulse noise situation in KRLS Tracker.

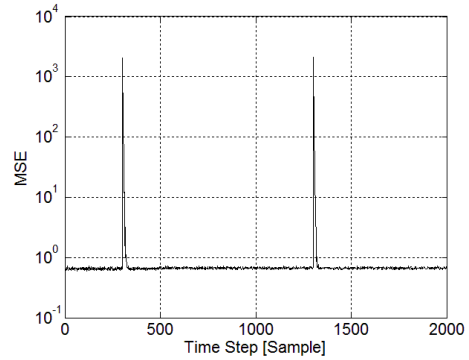
복원 되지 않는 것을 볼 수 있다, 이는 망각인자의 유효 길이가 길어서 임펄스 영향을 배제하지 못한 결과이다.

그림 3에는 제안된 3가지 알고리즘들을 임펄스 환경 하에서 수행하여 얻은 결과를 나란히 제시하였다. 그림 3(a)는 방법1을 임펄스 환경에서 수행한 결과이다. 이 결과를 보면 그림 2(b)와 같이 임펄스 이후 추정 성능이 완전히 없어지지 않지만 추정 성능이 매우 저하된 것을 확인 할 수 있다. 그러나 그림 3(b)와 그림 3(c)를 보면 임펄스 후에도 빨리 정상적인 추정을 하는 것을 확인할 수 있다.

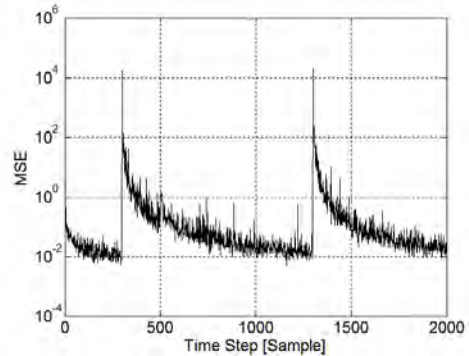
그림 2의 결과와 그림 3의 결과를 서로 비교하여 보면 제안한 알고리즘으로 보완할 경우 임펄스 잡음에 강한 결과를 얻을 수 있음을 알 수 있다. 특히 방법2와 방법3이 임펄스 환경에서 강인함을 알 수 있다.

실험 1과 실험2를 종합하면 제안한 방법 중 3번 방

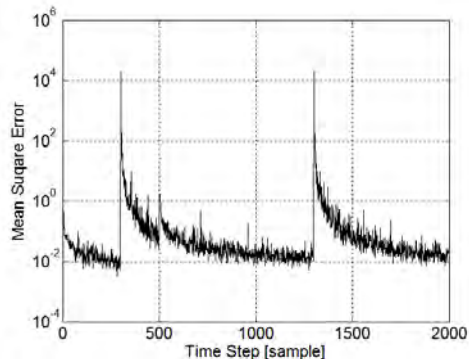
법이 시변 환경과 임펄스 환경 모두에서 상대적으로 더 우수한 추정 성능을 발휘함을 알 수 있다.



(a)



(b)



(c)

그림 3. 제안한 방법들의 수렴성 비교 (a) 임펄스 잡음 상황에서 방법1을 적용한 KRLS Tracker (b) 임펄스 잡음 상황에서 방법2를 적용한 KRLS Tracker (c) 임펄스 잡음 상황에서 방법3을 적용한 KRLS Tracker

Fig. 3. Compare convergence of the proposed method (a) How to apply one in impulse noise conditions KRLS Tracker (b)Method 2 is applied to the impulse noise situation in KRLS Tracker (c) Method 3 is applied to the impulse noise situation in KRLS Tracker

V. 결 론

본 논문에서는 가우시안 과정 기반의 추정법 중 하나인 KRLS-T 알고리즘의 시변 추정 성능 향상을 위한 방법 3가지를 제안하였다. 또 시변 채널 환경에서 모의 실험을 통하여 제안된 방법이 기존의 방법의 성능을 향상 시킴을 보였다. 또 임펄스가 추정 성능에 미치는 악영향을 효과적으로 줄일 수 있음도 보이기 위하여 시변 채널 상에 임펄스가 존재할 경우에도 실험을 하였고 그 경우에도 제안한 방법이 KRLS Tracker 알고리즘을 임펄스 잡음 환경에 강인하게 함을 확인하였다.

References

- [1] S. Haykin, *Adaptive Filter Theory*, 4th Ed., Prentice Hall, 2001.
- [2] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley-IEEE Press, 2003.
- [3] M. Choi and S. Lee, "Comparison study of channel estimation algorithm for 4S maritime communications," *J. KICS*, vol. 38C, no. 3, pp. 288-295, Mar. 2013.
- [4] J. S. Lim and Y. G. Pyeon, "Time delay estimation using LASSO (Least Absolute Selection and Shrinkage Operator)," *J. KICS*, vol. 39B, no. 10, pp. 715-721, Dec. 2014.
- [5] C. Hwang and K. Kim, "Doppler frequency estimation for time-varying underwater acoustic communication channel," *J. KICS*, vol. 40, no. 01, pp. 187-192, Jan. 2015.
- [6] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275-2285, Aug. 2004.
- [7] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, Wiley, 2010.
- [8] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag Inc., 1995.
- [9] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, 2001.
- [10] S. V. Vaerenbergh, J. Via, and I. Santamaria, "A sliding-window kernel RLS algorithm and its application to nonlinear channel identification," in *Proc. IEEE ICASSP*, pp. 789-792, Toulouse, France, May 2006.
- [11] M. Lazaro-Gredilla, S. V. Vaerenbergh, and I. Santamaria, "A Bayesian approach to tracking with kernel recursive least-squares," in *Proc. IEEE Int. Wksp. MLSP*, pp. 1-6, Beijing, China, Sept. 2011.
- [12] S. V. Vaerenbergh, M. Lazaro-Gredilla, and I. Santamaria, "Kernel recursive least-squares tracker for time varying regression," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 23, no. 8, pp. 1313-1326, Aug. 2012.
- [13] L. Csato and M. Opper, "Sparse online Gaussian processes," *Neural Computation*, vol. 14, no. 3, pp. 641-668, 2002.
- [14] W. Liu, I. Park, and J. C. Principe, "An information theoretic approach of designing sparse kernel adaptive filters," *IEEE Trans. Neural Netw.*, vol. 20, no. 12, pp. 1950-1961, Dec. 2009.
- [15] S. Song, J. S. Lim, S. Baek, and K. M. Sung, "Gauss Newton variable forgetting factor recursive least squares for time varying parameter tracking," *Electron. Lett.*, vol. 36, no. 11, pp. 988-990, Nov. 2000.
- [16] S. Song, J. S. Lim, S. J. Baek, and K. M. Sung, "Variable forgetting factor linear least square algorithm for frequency selective fading channel estimation," *IEEE Trans. Veh. Technol.*, vol. 51, no. 3, pp. 613-616, May 2002.
- [17] J. S. Lim, S. Lee, and H. Pang, "Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to nonstationary system estimations," *Neural Comput. Appl.*, vol. 22, no. 3, pp. 569-576, May 2013.
- [18] S. Lee, J. S. Lim, and K. M. Sung, "Low-complexity VFF-RLS algorithm using normalization technique," *J. Acoust. Soc. Kr.*, vol. 29, no. 1, pp. 18-23, Jan. 2010.
- [19] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*,

MIT Press, 2006.

- [20] L. Csato and M. Opper, *Sparse representation for gaussian process models*, in *Neural Inf. Process. Syst.* 13, pp. 444-450, Cambridge, MA: MIT Press, 2001.
- [21] S. V. Vaerenbergh, M. Lazaro-Gredilla, and I. Santamaria, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1313-1326, Aug. 2012.
- [22] D. J. Park, B. E. Jun, and J. H. Kim, "Fast tracking RLS algorithm using novel variable forgetting factor with unity zone," *Electron. Lett.*, vol. 27, no. 23, pp. 2150-2151, Nov. 1991.
- [23] A. K. Kohli and A. Rai, "Numeric variable forgetting factor RLS algorithm for second-order volterra filtering," *Circuits Syst. Signal Process*, vol. 32, no. 1, pp. 223-232, Feb. 2013.

임 준 석 (Jun-Seok Lim)



1986년 2월 : 서울대학교 전자공학과 학사 졸업
 1988년 2월 : 서울대학교 전자공학과 석사 졸업
 1996년 8월 : 서울대학교 전자공학과 박사 졸업
 1996년 7월~1997년 10월 : LG 종합기술원

1998년 3월~현재 : 세종대학교 전자공학과 교수
 <관심분야> 신호처리

편 용 국 (Yong-Guk Pyeon)



1993년 2월 : 강원대학교 전자공학과 학사 졸업
 1996년 2월 : 관동대학교 전자공학과 석사 졸업
 2004년 2월 : 세종대학교 전자공학과 박사 졸업
 2004년 9월~현재 : 강원도립대학 정보통신과 조교수

<관심 분야> 신호 처리