Mixture structure of kernel adaptive filters for improving the convergence characteristics

Kiyoshi Nishikawa and Hiroya Nakazato
Tokyo Metropolitan University, Tokyo, Japan
E-mail: knishikawa@m.ieice.org Tel: +81-42-585-8423

***Abstract*—In this paper, we propose a mixture structure of the linear and kernel adaptive fiilters for improving the convergence characteristics of the kernel normalized least mean square (KLMS) adaptive algorithm. The proposed method is based on the concept of the affine constrained mixture structure for the linear normalized LMS adaptive filters which uses the more than two adaptive filters concurrently. We derive the proposed structure, and its implementation method. We confirm the effectiveness of the proposed method through the computer simulations.**

I. Introduction

In this paper, we propose a mixture structure of a linear and a kernel adaptive filters which improve the convergence characteristics of the kernel adaptive filter.

Linear adaptive filters have been used in a variety of applications, e.g., echo or noise canceler, equalization in wireless communication channels, and so on. Besides, there are a lot of theoretical analyses on their behavior and characteristics[1], [2]. Recently, as an extension of the linear counterparts, kernel adaptive filters have been proposed that enable us to adaptively identify non-linear systems[3], and are expected to be used in applications such as non-linear channel equalization for improving the convergence properties[4].

Kernel adaptive filters are derived by applying the kernel method to linear adaptive filters, and several algorithms were proposed, i.e., the kernel recursive least squares (KRLS)[4], the kernel least mean square (KLMS)[5], the kernel normalized LMS (KNLMS)[6], the kernel ERLS-DCD[7] algorithms, and so forth. It is shown by those researches that the kernel algorithms could provide better convergence characteristics than the linear ones under some conditions[3], [6]. However, theoretical analysis of those kernel algorithms are not developed enough to predict the behavior of them in the actual applications.

In addition, the kernel algorithms require some settings which do not exist in their linear counterparts, namely, the kernel functions, the kernel parameters, the parameter for sparseness etc. The selection of them affect the performance of the algorithms and unsuitable selection would degrade the convergence characteristics of the kernel algorithms to almost same as those of linear algorithms as confirmed in this paper.

In this paper, we propose a mixture structure of a kernel and a linear adaptive filters for improving the convergence characteristics of the kernel adaptive filters. The proposed method could be regarded as an extension of the mixture structure for the linear adaptive filters[8], [9]. However, the proposed method employs a kernel and a linear adaptive filters instead of multiple linear filters. This feature enables us to identify wider class of systems including linear and non-linear ones using the proposed method. The outputs of the two filters will be mixed by the mixing equation. The ratio of mixing will be automatically adjusted by adaptively controlling the value of a parameter, the mixing parameter. We confirm the convergence characteristics of the proposed method when applied to non-linear system identification problems.

II. Preparation

Here, we briefly review the conventional kernel adaptive filters[3], [6].

A. Kernel method

First, the input signal $x(n)$ is transformed into a high-dimensional feature space F by the transformation $\Phi(x)$. Then, the output signal of the adaptive filter is expressed as

$$f\left(\boldsymbol{x}_n\right) = \Phi^T\left(\boldsymbol{x}_n\right)\boldsymbol{w}_n \tag{1}$$

where $\boldsymbol{w}_n$ and $\boldsymbol{x}_n$ are filter coefficient vector of the adaptive filter, and tap-input vector at time n respectively. Also, T shows the transpose of a vector as usual, and $\Phi^T(\boldsymbol{x})$ shows the transpose of the transformed vector. We defined $\boldsymbol{w}_n$ and $\boldsymbol{x}_n$ as

$$\boldsymbol{w}_n = [w_0, \ldots, w_{M-1}], \tag{2}$$

$$\boldsymbol{x}_n = [x(n), \ldots, x(n-M+1)] \tag{3}$$

where w_i, $x(n)$ and M show the i-th coefficient of the filter at time n, the input sample at n and the length of the filter respectively.

In order to apply the kernel method, we assume that the filter vector $\boldsymbol{w}_n$ can be expressed as a linear combination of m vectors $\Phi(\boldsymbol{y}_j)$ as

$$\boldsymbol{w}_n = \sum_{j=1}^{m} \alpha_j \Phi\left(\boldsymbol{y}_j\right). \tag{4}$$

The vectors $\boldsymbol{y}_j$ are subset of $\boldsymbol{x}_\ell$ ($\ell = 0, 1, \ldots, n-1$) and α_j is the weight corresponding to $\boldsymbol{y}_j$. Then, the output in (1) is expressed[3] as

$$f\left(\boldsymbol{x}_n\right) = \sum_{j=1}^{m} \left(\Phi^T\left(\boldsymbol{x}\right)\Phi\left(\boldsymbol{y}_j\right)\right)\alpha_j. \tag{5}$$

Let us define the vector $\boldsymbol{\alpha}_n$ as $\boldsymbol{\alpha}_n = [\alpha_1, \ldots, \alpha_m]^T$. In the kernel adaptive filter, $\boldsymbol{\alpha}_n$ is regarded as the coefficient vector of the adaptive filter instead of $\boldsymbol{w}_n$. A lot of algorithms are derived[6], [5], [10] to estimate the optimum $\boldsymbol{\alpha}$ based on the linear adaptive algorithms.

In those algorithms, the inner product $\Phi^T(\boldsymbol{x}(n))\Phi(\boldsymbol{y}_j)$ in Eq. (5) is obtained via the the kernel function. A kernel function $k(\cdot,\cdot)$ is given as

$$\forall \mathbf{a}, \mathbf{b} \in X \qquad \kappa(\boldsymbol{a}, \boldsymbol{b}) = \Phi^T(\mathbf{a})\Phi(\mathbf{b}) \tag{6}$$

and is used to calculate the inner product in the space F [3].

The Gaussian kernel defined as below is widely used in kernel adaptive filtering and, in this paper, we also assume that the Gaussian kernel is used:

$$\kappa(\boldsymbol{a}, \boldsymbol{b}) = \exp\left(-\zeta \|\boldsymbol{a} - \boldsymbol{b}\|^2\right) \tag{7}$$

where $\|\cdot\|$ show the Euclidean norm and ζ is the kernel parameter.

B. Kernel normalized LMS algorithm

The conventional linear adaptive algorithm could be used for updating the filter coefficients $\boldsymbol{\alpha}_n$, and several algorithms are proposed so far, e.g., the kernel RLS[4], the kernel LMS[5], the kernel NLMS[6], the kernel ERLS-DCD[7] and so forth.

Although the proposed method is not restricted with the selection of the adaptive algorithm, we will use the kernel normalized LMS (KNLMS) algorithm in the following. Here, we briefly review the algorithm[6].

First, we rewrite Eq. (5) as

$$f(\boldsymbol{x}_n) = \begin{bmatrix} \Phi(\boldsymbol{x}_n)^T \Phi(\mathbf{y}_1) \\ \Phi(\boldsymbol{x}_n)^T \Phi(\mathbf{y}_2) \\ \vdots \\ \Phi(\mathbf{x_n})^T \Phi(\mathbf{y}_m) \end{bmatrix}^T \boldsymbol{\alpha}_n = \begin{bmatrix} \kappa(\boldsymbol{x}_n, \boldsymbol{y}_1) \\ \kappa(\boldsymbol{x}_n, \boldsymbol{y}_2) \\ \vdots \\ \kappa(\boldsymbol{x}_n, \boldsymbol{y}_m) \end{bmatrix}^T \boldsymbol{\alpha}_n \tag{8}$$

$$= \boldsymbol{h}_n \boldsymbol{\alpha}_n \tag{9}$$

where we defined $\boldsymbol{h}_n$ as $\boldsymbol{h}_n = [\kappa(\boldsymbol{x}_n, \boldsymbol{y}_1), \ldots, \kappa(\boldsymbol{x}_n, \boldsymbol{y}_m)]^T$. Then, the filter $\boldsymbol{\alpha}_n$ can be updated using a linear adaptive algorithm by regarding $\boldsymbol{h}_n$ as the input vectors to $\boldsymbol{\alpha}_n$.

Here, we define the matrix D which is consisting of the vectors $[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m]$ as

$$\mathbf{D} = \begin{bmatrix} \boldsymbol{y}_1 & \ldots & \boldsymbol{y}_m \end{bmatrix} \tag{10}$$

and $\boldsymbol{D}$ is called the dictionary. The vectors stored in the dictionary $\boldsymbol{D}$ are m $(m \leq n)$ input vectors of the previous time, i.e., $\boldsymbol{x}_\ell$ where m is a variable determined by the algorithm below.

Let us denote $\boldsymbol{D}$ at time n by $\boldsymbol{D}_n$. Then, $\boldsymbol{D}_n$ and $\boldsymbol{h}_n$ are updated according to the following pseudo algorithm:

Initialization

$$\boldsymbol{D}_1 = \boldsymbol{y}_1 = \boldsymbol{x}_1$$
$$\boldsymbol{h}_1 = k(\boldsymbol{x}_1, \boldsymbol{y}_1)$$
$$\boldsymbol{\alpha}_1 = 0 \quad , \quad m = 1$$

for $n = 2, 3, \cdots$

if $\max_{j=1,\ldots,m} |k(\boldsymbol{x}_n, \boldsymbol{y}_j)| > \gamma_0$ (11)

$$\boldsymbol{D}_n = \boldsymbol{D}_{n-1}$$
$$\boldsymbol{h}_n = \begin{bmatrix} k(\boldsymbol{x}_n, \boldsymbol{y}_1) & \cdots & k(\boldsymbol{x}_n, \boldsymbol{y}_m) \end{bmatrix}^T \tag{12}$$

else

$$m = m + 1$$
$$\boldsymbol{D}_n = \boldsymbol{D}_{n-1} \cup \{\boldsymbol{x}_n\}$$
$$\boldsymbol{h}_n = \begin{bmatrix} k(\boldsymbol{x}_n, \boldsymbol{y}_1) & \cdots & k(\boldsymbol{x}_n, \boldsymbol{y}_m) \end{bmatrix}^T \tag{13}$$

end if

end for

In Eq. (11), γ_0 is a threshold in the range $0 < \gamma_0 < 1$ and its value is determined according to the sparseness of the signal[6]. According to the condition Eq. (11), $\boldsymbol{x}_n$ will be stored in $\boldsymbol{D}_n$ as a new training vector.

Then, the filter coefficients vector $\boldsymbol{\alpha}_n$ will be updated by the following equation:

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n + \mu_\alpha \frac{d(n) - \boldsymbol{h}_n \boldsymbol{\alpha}_n}{\delta + ||\boldsymbol{h}_n||^2} \boldsymbol{h}_n \tag{14}$$

where μ_α is the normalized step-size parameter in the range $0 < \mu_\alpha < 2$ and δ is a stabilized parameter to prevent the divergence of the algorithm when $||\boldsymbol{h}_n||^2$ is zero.

C. Convergence characteristics of kernel adaptive filters and selection of parameter

Although the kernel adaptive filters are expected to improve the convergence characteristics of the learning of the non-linear unknown systems, under some conditions, the convergence characteristics would be degraded compared to those of the linear filters. Here, we show some examples of such situations based on the computer simulations.

In Figs 1, 2, and 3, we show the comparison of the convergence characteristics of the linear NLMS and the kernel NLMS filters. In these simulations, we apply the adaptive filters to the system identification and forward prediction problems[6]. The conditions of the simulations are shown in Table I and Table II respectively.

From these results, we could see that the convergence characteristics of the kernel adaptive filters vary depending on the environments, and in some cases, the linear adaptive filters could provide comparative, or better in some sense (e.g., better initial convergence etc), characteristics with the kernel ones. Namely, in Fig 2, the KNLMS provides better characteristics with the filter length $M = 2$. However, on the other hand, by increasing M to $M = 5$, the NLMS provides comparative convergence characteristics with those of the KNLMS.

Other than the number of filter coefficients, there are two parameters in KNLMS which affects the convergence characteristics, i.e., the kernel parameter ζ of Eq. (7) and the threshold parameter γ_0 of Eq. (11). Although there are some theoretical analyses of the kernel adaptive filters[11], the effects of those parameters in the algorithms are not clearly analyzed. Besides, we could not see before applying the adaptive filters that if the unknown system could not be sufficiently modeled by the linear adaptive filters and would require kernel adaptation. Hence, we need an adjustable system which automatically select the appropriate adaptive filter according to the environments to broaden the applicable areas of kernel adaptive filters.

In this paper, we will consider a mixture structure of a kernel and a linear adaptive filters.

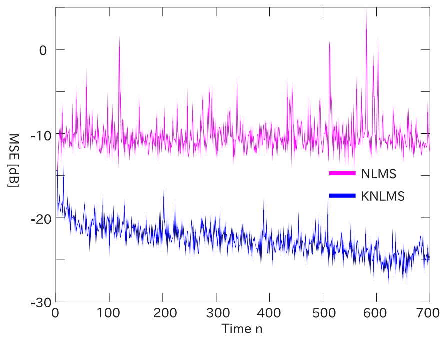

Fig. 1. Comparison of convergence characteristics of the NLMS, KNLMS

TABLE I

THE CONDITIONS OF THE COMPUTER SIMULATIONS. THE RESULTS ARE SHOWN IN FIG. 1. WE APPLIED THE ADAPTIVE FILTERS TO THE NON-LINEAR DYNAMICAL SYSTEM[6] WHOSE INPUT-OUTPUT RELATION IS GIVEN AS THE EQUATION BELOW. u_n WAS SAMPLED FROM A GAUSSIAN PROCESS OF ZERO-MEAN.

Filter length M	5
Desired System	$\begin{cases} lv_n = 1.1\exp(-\lvert v_{n-1}\rvert) + u_n \\ d_n = v_n^2 \end{cases}$
Kernel function	$\exp(-0.13\lvert\lvert \boldsymbol{a}, \boldsymbol{b}\rvert\rvert)$
γ_0 of Eq. (11)	0.90
Normalized step size	1.0 for NLMS and KNLMS
Additive noise	White Gaussian process of SNR=40[dB]
Ensemble average	30

III. PROPOSED METHOD

Here, we describe the proposed method. The proposed method is an extension of the mixture structure for the two or more linear adaptive filters[8], [9]. The proposed structure consists of a kernel and a linear adaptive filters.

A. Structure of the proposed method

In Fig. 4, the structure of the proposed method is shown. The structure is composed of two adaptive filters, i.e., a kernel and a linear adaptive filters. Note that, in this paper, we assume that the linear adaptive filters is updated using the normalized LMS algorithm, and on the other hand, the kernel using the kernel NLMS algorithm[6]. However, the proposed method does not require specific algorithms to be used, and the selection here is only for simplicity of the description.

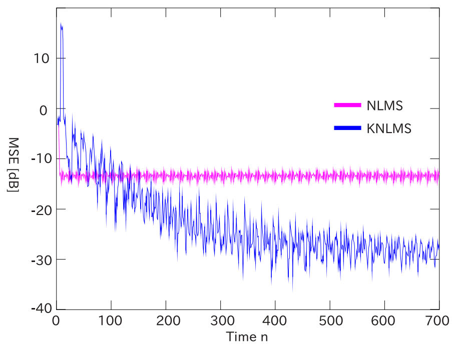

Fig. 2. Comparison of convergence characteristics of the NLMS, KNLMS applied to forward prediction problem[6]. The length of the adaptive filter was $M = 2$. The conditions of the simulation are shown in Table II.

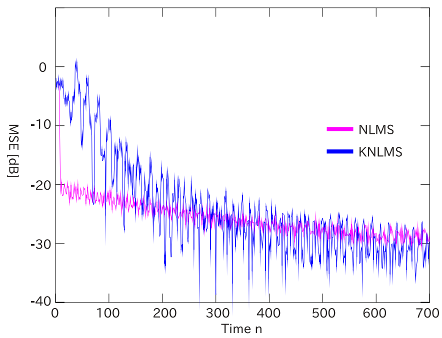

Fig. 3. Comparison of convergence characteristics of the NLMS, KNLMS applied to forward predication problem[6]. The conditions of the simulation are the same as those of Fig. 2 except the filter length M was $M = 5$, and are shown in Table II.

TABLE II

THE CONDITIONS OF THE COMPUTER SIMULATIONS. THE RESULTS ARE SHOWN IN FIG. 1. WE APPLIED THE ADAPTIVE FILTERS TO A LINEAR SYSTEM WHOSE COEFFICIENTS WERE DESIGNED USING THE REMEZ ALGORITHM AND THE INPUT SIGNAL WAS THE WHITE GAUSSIAN PROCESS WITH ZERO MEAN.

Filter length M	2 (Fig. 2) or 5 (Fig. 3)
Forward Prediction	$x_n = \left(0.8 - 0.5\exp\left(-x_{n-1}^{-2}\right)\right)x_{n-1} - \left(0.3 + 0.9\exp\left(-x_{n-1}^2\right)\right)x_{n-2} + 0.1\sin\left(x_{n-1}\pi\right)$
Kernel function	$\exp(-3.73\lvert\lvert \boldsymbol{a}, \boldsymbol{b}\rvert\rvert)$
γ_0 of Eq. (11)	0.90
Normalized step size	1.0 for NLMS and KNLMS
Additive noise	White Gaussian process of SNR=40[dB]
Ensemble average	30

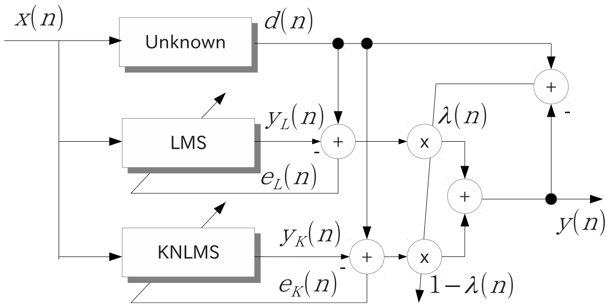

Fig. 4. Structure of the proposed method.

In this structure, there are two error signals $e_{\mathrm{L}}(n)$ and $e_{\mathrm{K}}(n)$, and are defined as

$$e_{\mathrm{L}}(n) = d(n) - y_{\mathrm{L}}(n) \tag{15}$$
$$e_{\mathrm{K}}(n) = d(n) - y_{\mathrm{K}}(n) \tag{16}$$

where $y_{\mathrm{L}}(n)$ and $y_{\mathrm{K}}(n)$ are the output of the linear and the kernel adaptive filters respectively. We use the subscripts L and K to indicate that the variable is of the linear or the kernel adaptive filters respectively.

Let us define the filter coefficients vectors as

$$\boldsymbol{w}_{\mathrm{L}} = [w_{L,0}(n), \ldots, w_{L,M-1}(n)], \tag{17}$$
$$\boldsymbol{w}_{\mathrm{K}} = [w_{K,0}(n), \ldots, w_{K,M-1}(n)]. \tag{18}$$

Besides, the output signals $y_{\mathrm{L}}(n)$ and $y_{\mathrm{K}}(n)$ are given as

$$y_{\mathrm{L}}(n) = \sum_{i=0}^{M-1} w_{L,i}(n)x(n-i) = \boldsymbol{w}_{\mathrm{L}}^T \boldsymbol{x}(n) \tag{19}$$
$$\begin{aligned} y_{\mathrm{K}}(n) &= \sum_{i=0}^{M-1} w_{K,i}(n)\Phi(x(n-i)) \\ &= \sum_{j=1}^{m} \kappa(\boldsymbol{x}, \boldsymbol{y}_j)\alpha_j = \boldsymbol{h}_n \boldsymbol{\alpha}_n \end{aligned} \tag{20}$$

respectively. Note that, for the kernel adaptive filters, we estimate the vector $\boldsymbol{\alpha}_n$ instead of $\boldsymbol{w}_{\mathrm{K}}$.

B. The mixture structure

As depicted in Fig. 4, the output signals $y_{\mathrm{L}}(n)$ and $y_{\mathrm{K}}(n)$ are mixed according to the mixture equation to produce the overall output $y(n)$ of the system.

For producing $y(n)$, we select here the following simplest equation[8] as

$$y(n) = \lambda(n)y_{\mathrm{L}}(n) + [1 - \lambda(n)]y_{\mathrm{K}}(n). \tag{21}$$

where $\lambda(n)$ is called as the mixing parameter, and its value should be in the range $0 \le \lambda(n) \le 1$s. By adjusting the value of the mixing parameter the principal filter will be selected, namely, by letting $\lambda(n)$ as $\lambda(n) = 0$, the kernel filter will be selected, and $\lambda(n) = 1$ the linear filter will do.

According to the applied environments, we should adaptively adjust the value of the mixing parameter $\lambda(n)$. By comparing the equations (19) and (20), we could see that the length of the coefficients of the filters $\boldsymbol{w}_{\mathrm{L}}$ and $\boldsymbol{\alpha}_n$ are different, and more over, the length of $\boldsymbol{\alpha}_n$ would increase as n. Therefore, we can not use the update equation based on the estimation of the variation of the filter coefficients.

We consider, hence, the update of $\lambda(n)$ based on the outputs of the filters. In this paper, we propose the following equation to update the parameter $\lambda(n)$:

$$\lambda(n) = \lambda(n-1) + \mu_\lambda[y_{\mathrm{L}}(n) - y_{\mathrm{K}}(n)] \tag{22}$$

where μ_λ is the step size parameter to control the convergence of $\lambda(n)$. Moreover, we limit the value of $\lambda(n)$ in the range $\{0 \le \lambda(n) \le 1\}$ as

$$\text{if } \lambda(n) > 1: \qquad \lambda(n) = 1 \tag{23}$$
$$\text{else if} \lambda(n) < 0: \qquad \lambda(n) = 0. \tag{24}$$

Besides, there is a freedom of selecting the initial value of $\lambda(0)$. We suggest to choose $\lambda(0)$ to be $\lambda(0) = 1$ so that the linear filter will be selected at the first stage of the adaptation because the performance of the linear filter is more controllable than the kernel one so that we can predict the initial behavior of the system. Note that, however, in our simulations, we confirmed the performance of the proposed method does not depend on the selection of the initial value.

IV. Simulation Results

Finally, we show the results of computer simulations to confirm the validity of the proposed method.

A. System identification

First, we show the results of applying the adaptive filters to the system identification problems. The conditions of the simulations are shown in Table III and the results are shown in Fig. 5. We compared the NLMS, the KNLMS, and the proposed method in terms of the mean squared errors (MSEs).

From the figure, we can confirm that the proposed method successfully select the KNLMS algorithm which provides the better convergence characteristics under the conditions. Note that we confirmed that the characteristics were not affected by the selection of the initial value of $\lambda(0)$.

TABLE III
THE CONDITIONS OF THE COMPUTER SIMULATIONS SHOWN IN FIG. 5. u_n WAS A GAUSSIAN DISTRIBUTION OF ZERO-MEAN.

Filter length M	5
System Identification	$\begin{cases} lv_n = 1.1\exp(-\lvert v_{n-1}\rvert) + u_n \\ d_n = v_n^2 \end{cases}$
Kernel function	$\exp(-0.13\lvert\lvert \boldsymbol{a}, \boldsymbol{b}\rvert\rvert)$
γ_0 of Eq. (11)	0.90
Normalized step size	1.0 for NLMS and KNLMS
Additive noise	White Gaussian process of SNR=40[dB]
Ensemble average	30

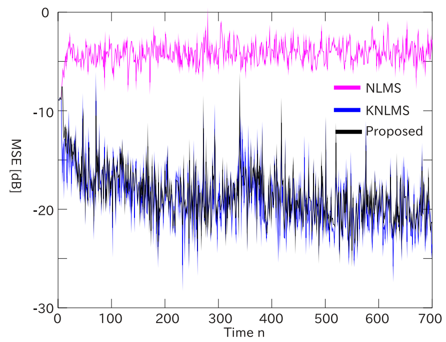

Fig. 5. Comparison of convergence characteristics of the NLMS, the KNLMS, and the proposed algorithms.

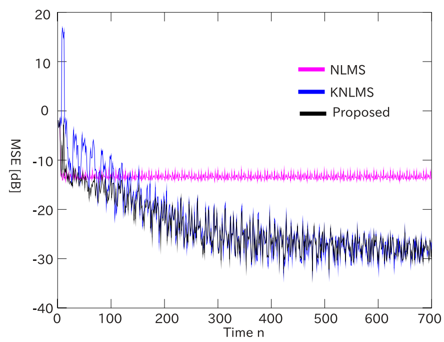

Fig. 6. Comparison of convergence characteristics of the NLMS, conventional KNLMS, and the proposed algorithms. The proposed method improved the initial convergence of the KNLMS and simultaneously maintain the lower excess MSE.

B. Forward prediction

Next, we show the results of forward prediction. The conditions of the simulations are shown in Table IV. As in the previous simulation, we compared the convergence characteristics of the NLMS, the KNLMS, and the proposed method in terms of the MSEs. The results are shown in Fig. 6.

TABLE IV
THE CONDITIONS OF THE COMPUTER SIMULATIONS SHOWN IN FIG. 6.

Filter length M	2
Desired System	$$x_n = \left(0.8 - 0.5\exp\left(-x_{n-1}^{-2}\right)\right)x_{n-1} - \left(0.3 + 0.9\exp\left(-x_{n-1}^{2}\right)\right)x_{n-2} + 0.1\sin\left(x_{n-1}\pi\right)$$
Kernel function	$\exp(-3.73\|\|\boldsymbol{a},\boldsymbol{b}\|\|)$
γ_0 of Eq. (11)	0.90
Normalized step size	1.0 for NLMS and KNLMS
Additive noise	White Gaussian process of SNR=40[dB]
Ensemble average	30

From the figure, we could confirm that the proposed method could select the correct output signal, and therefore, the convergence characteristics would be better than that of the solo use of the NLMS, or the KNLMS. Namely, at the initial state, the output of the NLMS was selected to improve the characteristics, and then switch to that of the KNLMS gradually as n increases.

C. Non-stationary forward prediction

Finally, we applied the proposed method to a non-stationary environment to investigate its tracking property. The conditions of the simulation are shown in Table V and the results are shown in Fig. 7. Note that we changed the equation to produce the signal at time $n = 100$ to simulate the non-stationary environments.

By comparing the convergence characterstics of the linear and the kernel filters, we notice that the linear filter provides better characteristics in some time interval, and in the other, the kernel one does.

Under these conditions, we could see that the proposed method could select the filter which provides better MSE at each time n. Hence, it provides lower MSE in all the time interval in this simulation.

TABLE V
THE CONDITIONS OF THE COMPUTER SIMULATIONS SHOWN IN FIG. 7.

Filter length M	2
Forward prediction	Time: $n < 100$ $$x_n = \left(-0.5\exp\left(-x_{n-1}^{-2}\right)\right)x_{n-1} - \left(0.9\exp\left(-x_{n-1}^{2}\right)\right)x_{n-2} + 0.5\sin\left(x_{n-1}\pi\right)$$ Time: $n >= 100$ $$x_n = \left(0.8 - 0.5\exp\left(-x_{n-1}^{-2}\right)\right)x_{n-1} - \left(0.3 + 0.9\exp\left(-x_{n-1}^{2}\right)\right)x_{n-2} + 0.1\sin\left(x_{n-1}\pi\right)$$
Kernel function	$\exp(-3.73\|\|\boldsymbol{a},\boldsymbol{b}\|\|)$
γ_0 of Eq. (11)	0.90
Normalized step size	1.0 for NLMS and KNLMS
Additive noise	White Gaussian process of SNR=40[dB]
Ensemble average	30

V. CONCLUSION

In this paper, we proposed a mixture structure of a kernel adaptive filters for improving the convergence characteristics. The proposed method is consisting of a kernel and a linear adaptive filters. Their outputs are combined by a mixture equation to produce the output of the whole system. By adaptively adjusting the value of the mixture parameter of the equation, the system will select the appropriate filter to generate better convergence characteristics. Through the computer simulations, we showed that, by using the proposed method, we can improve the convergence characteristics of the kernel adaptive filter both in stationary and non-stationary environments.

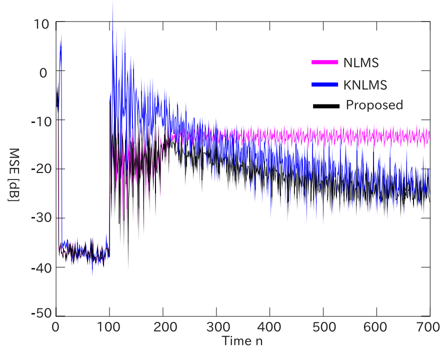

Fig. 7. Comparison of convergence characteristics of the NLMS, the KNLMS, and the proposed algorithms. The proposed method improved the characteristics of the KNLMS between $n = 100$ to 200 where the NLMS outperforms the conventional KNLMS.

REFERENCES

[1] A. H. Sayed, *Adaptive Filters*. John Wiley & Sons, 2008.

[2] A. H. Sayed, *Fundamentals of Adaptive Filtering*. John Wiley & Sons, 2003.

[3] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*. Wiley, 2010.

[4] Y. Engel, S. Mannor, and R. Meir, "The Kernel Recursive Least-Squares Algorithm," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2275–2285, Aug. 2004.

[5] W. Liu, P. P. Pokharel, and J. C. Principe, "The Kernel Least-Mean-Square Algorithm," *IEEE Transactions on Signal Processing*, vol. 56, pp. 543–554, Feb. 2008.

[6] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online Prediction of Time Series Data With Kernels," *IEEE Transactions on Signal Processing*, vol. 57, pp. 1058–1067, Mar. 2009.

[7] Y. Ogawa and K. Nishikawa, "A Kernel Adaptive Filter based on ERLS-DCD Algorithm," in *Proc. of Intl Tech. Conf. Circuits Systems, Computer, Communications 2011*, (Gyeongju), 2011.

[8] J. Arenas-Garcia, A. Figueiras-Vidal, and A. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Transactions on Signal Processing*, vol. 54, pp. 1078–1090, Mar. 2006.

[9] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, "Steady-State MSE Performance Analysis of Mixture Approaches to Adaptive Filtering," *IEEE Transactions on Signal Processing*, vol. 58, pp. 4050–4063, Aug. 2010.

[10] W. Liu, I. M. Park, Y. Wang, and J. C. Principe, "Extended Kernel Recursive Least Squares Algorithm," *IEEE Transactions on Signal Processing*, vol. 57, pp. 3801–3814, Oct. 2009.

[11] W. D. Parreira, J. C. M. Bermudez, C. Richard, and J.-Y. Tourneret, "Stochastic Behavior Analysis of the Gaussian Kernel Least-Mean-Square Algorithm," *IEEE Transactions on Signal Processing*, vol. 60, pp. 2208–2222, May 2012.