Performance Analysis of Mixture Approaches and Tracking Performance of Adaptive Filter using Adaptive Neural Network

A.Vijayalakshmi
Associate Professor, ECE
Vardhaman College of Engineering, Hyderabad, INDIA

D.Spoorthi
M.Tech Student, ECE
Vardhaman college of Engineering, Hyderabad, INDIA

ABSTRACT
This paper mainly concentrates on different mixture structures which include affine and convex combinations of several parallel running adaptive filters. The mixture structures are investigated using their final MSE values and the tracking of the nonlinear system is done using an ANN model that updates the filter weights using nonlinear learning strategies(it uses stochastic gradient descent to update the filter weights based on MSE's of mixture structures).the mixture structures greatly improve the convergence and performance of the of the constituent filters compared to traditional adaptive methods. The mixture structures employed in this paper can be applied to the constituent filters that employ different adaptation algorithms. We describe an adaptive neural network model that updates the weights of the filter using nonlinear methods.

General Terms
Adaptive filter, mean square error ,correlation.

Keywords
Mixtures structures, affine, convex, adaptive filter, artificial neural network, adaptive neural network , tracking.

1. INTRODUCTION
This paper mainly concentrates on unknown system identification property of adaptive filter and convergence rate of mixture approaches. The filter performance is effected by the step size, filter order and the choice of algorithm made. Recently there has been an interest in combination structures, where the output of several constituent filters are combined using different mixing structures for improved performance of the overall system. In [7]we have presented a combination of two least-mean square(LMS) with different learning rates that was effective at combining fast convergence low residual adjustment, the parallel running adaptive branches are considered as alternative hypothesis about data model as well as different diversity sources, which can be used to achieve better performance than individual filter. The intelligent combining approaches may be well suited for wide variety of adaptive filtering applications which specially the application that involve non-stationary data environment(the applications that has lack of apriori and posteriori data) we consider the mixture approaches that are convex(the weights are Constrained to be nonnegative and sum up to one) and affine combination(where the linear mixture weights are constrained to sum up to one)[1],[2],[4] .the objective here is to combine different adaptive filters(that use different algorithms and have different learning rates) running in parallel on a given task with a goal of achieving better performance or at least as good as the best constituent algorithm for all sequences .this is achieved by exploiting the time-dependent nature of the best choice of the constituent filter. An alternative algorithm that adapts the convex combination using stochastic gradient method (stochastic gradient descent is used to update filter weights) was studied in [4] Although the analysis given for the convex combination of two filters was generic [2], the results were then specialized to the case of two LMS filters with different learning rates: one with a comparably smaller and the other with a comparably larger learning rate, were combined. Hence, the combination approaches enjoyed fast converge in the start of the adaptation and smaller excess MSE at the steadystate.

The tracking performance of the adaptive filter was measured using an adaptive neural network[8].The tracking of a nonlinear system using a linear model is difficult, since artificial neural networks are inherently nonlinear models, ANN-based filtering methods are useful for signals that are inherently nonlinear. An ANN model with hidden layer is able to approximate any nonlinear system theoretically that is realizable.[7]-[11].the performance of the ANN model is limited by the number of hidden units and the learning strategies employed .the ANN model employed in this paper has one input unit, one output unit and 10 hidden units. The ANN model updates the filter weights according to the MSE values and tracks the unknown system accurately compared to that of a normal adaptive filtering model.

The organization of this paper is as follows: section2 includes the model description (i.e. the calculation of excess MSE's and minimization of MSE.section 3 consist of different combination structures i.e., convex and affine combinations. Section4 includes the adaptive methods to update the weights of mixture structures.section4 consist of an ANN model that is meant for tracking the unknown system (nonlinear system identification).section5 consists of simulation results and conclude the paper by producing the results obtained by the simulations of section5.

Table1: Notations

Row and column vectors	Bold phase lower case letters(ex: β)
matrices	Bold phase capital letters(ex:**R**)
β:β	Denotes the i^{th} individual entry
β^t	Transpose of β
a: $\|a\|$	Represents absolute value of integer
$\rho_i(R)$	Represents the Eigen values sorted in descending order

THE MODEL

This model aims at converting the input signal to the expected output with minimum possible error. Adaptive tuning helps in changing the filter coefficients inline with the input signal, thereby reducing the gap between the observed output and the output predicted by our model.

The model has two parts

i. The first part consists of estimating the desired signal using 'm' parallel running adaptive algorithms.

ii. The second part consists of the combination methods used and analysing the tracking performance of the filter.

Objective:
The main objective is to minimize MSE using different combination methods.

$$e_{o,i}(t) = d(t) - \beta_o^T u_i(t)$$

Where $e_{o,i}(t)$ represents the error, the filter coefficients are

$$\beta_{o,i} = R_i^{-1} p_i$$

The optimal value or the "clean" part of the desired signal will be

$$\hat{d}_i(t) = \beta_i^T x(t) - [\beta_{o,i} - \beta_i(t)]^T x(t)$$

And the output of the model is

$$y_i = f(x_i, \beta_{o,i})$$

EXAMPLE:

Consider an example where a linear model with two coefficients is to be found (here we consider the system without noise)

$$y_i = \beta_o + \beta_1 x_i$$

The optimal values of β_o and β_1 are to be found in order to minimize the MSE.

$$e_i = (y_i - \hat{y}_i)$$
$$e_i^2 = (y_i - (\beta_o + \beta_1 x_i))^2$$

Let $E^2 = \sum e_i^2$

Hence $$\frac{1}{n}\sum e_i^2 = MSE \quad (1)$$

From (1) we have,

$$\frac{\partial MSE}{\partial \beta_1} = 0 \quad (2)$$

$$\frac{\partial MSE}{\partial \beta_0} = 0 \quad (3)$$

Hence by solving the equations (2) and (3) we get the optimal values of β_o and β_1.

The second part of the model is the combination of the constituent filters or the mixing stage.

$$\hat{d}(t) = \beta^T(t) y(t) \quad (4)$$

Where $y(t) = [\hat{d}_1(t), ...\hat{d}_m(t)]$ similar to the constituent algorithm ,we consider only the combination stage in the output. Hence the final estimation error is given by

$$e(t) = d(t) - \hat{d}(t)$$

With the above definitions, the autocorrelation matrix for input of the combinati*on* stage is $R(t) = E[y(t)y^T(t)]$, $R \equiv \lim_{t\to\infty} R(t)$ and the cross-correlation Vector is given by $P(t) = E[y(t)d(t)], p \equiv \lim_{t\to\infty} P(t)$, when limit exist.to calculate R,we observe that for any filter pairs *i ,j:*

$$\lim_{t\to\infty} E[g_i(t) - e_{a,i}(t)][g_i(t) - e_{a,j}(t)]$$
$$= E[g_i(t)g_j(t)] + \lim_{t\to\infty} E[e_{a,i}(t)e_{a,j}(t)]$$
$$= \sigma_{g,ij}^2 + J_{ex,ij}$$

$\sigma_{g,ij}^2 = \beta_{0,i}^T E[x_i(t)x_j^T(t)]\beta_{0,j}$ wherein we use a separation assumption similar to the one used in[8]

$E[g_i(t)e_{a,ij}(t)] = E[g_i(t)]E[e_{a,ij}(t)]$ for all i,j in the limit as $t \to \infty$ .with this result by orthogonality we have

$$R = \begin{bmatrix} J_{ex,1} + \sigma_{g,11}^2 .....J_{ex,1m} + \sigma_{g,1m}^2 \\ ..... \\ J_{ex,m} + \sigma_{g,m1}^2 .....J_{ex,mm} + \sigma_{g,mm}^2 \end{bmatrix}$$

=G+J

Hence the steady-state autocorrelation vector, since $\lim_{t\to\infty} E[d(t)\{g_i(t)-e_{a,i}(t)\}]=\sigma^2_{g,ii}$,or by orthogonality we have

$$P=\begin{bmatrix}\sigma^2_{g,11}\\ .\\ .\\ \sigma^2_{g,mm}\end{bmatrix}$$

Hence from the above basic model, we observe that different filters have different autocorrelation matrices and cross-correlation vectors. with the above basic model ,we consider two different combination structures that are constrained affine and convex combinations. we analyse the tracking performance and MSE of the achieved by the structures by formulating the minimum MSE (as shown in the example) levels corresponding to the optimal combination weights for each structure.

2. THE MIXTURE STRUCTURES

Here we consider different methods to combine the constituent filters that use different algorithms that provide an optimal solution. The different combinations we use are affine combination and convex combination. the derivations in this section are generic and can be used for constituent filters of any order and any algorithm.

2.1 Affine combination

The system under investigation is shown in fig.2 .here the filter uses one LMS and one RLS filter with different step sizes

$$\beta_i(t+1)=\beta_i(t)-\mu_i e_i(t)x_i(t) \quad (5)$$

Where $e_i(t)=d(t)-\beta_i^T(t)x(t)$

$$d(t)=e_0(t)+\beta_0^T x(t)$$

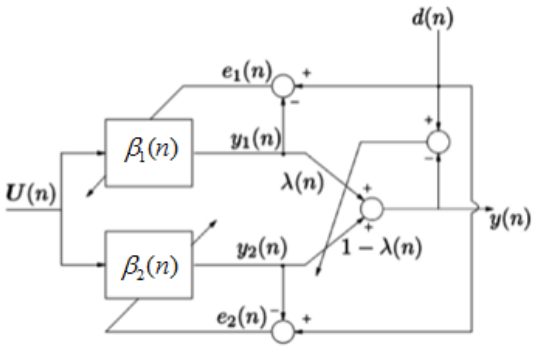

Fig1.combination of two adaptive filters

The output of the two filters is combined as shown in in fig: as

$$y(t)=\alpha_{aff}(t)y_1(t)+[1-\alpha_{aff}(t)]y_2(t) \quad (6)$$

The MMSE of the affine combination can be solved by

$$\min_\beta\left\{\sigma_d^2-p^T R^{-1}p+(\beta-\beta_o)^T R(\beta-\beta_o)\right\} \quad (7)$$

The optimal affine combination weights are given by

$$\beta_o^a=\beta_o-\frac{(1^T\beta_o-1)}{1^T R^{-1}1}R^{-1}1$$

for this optimal affine combination, the MSE is given by

$$J^a_{\min}=\sigma_d^2-p^T R^{-1}p+\frac{(1^T\beta_o-1)^2}{1^T R^{-1}1}$$

2.2 Convex combination

The adaptive scheme is studied from an optimal convex combiner viewpoint

$$y(t)=\alpha_{cvx}(t)y_1(t)+[1-\alpha_{cvx}(t)]y_2(t)$$

Where $\lambda_{cvx}(t)$ is the optimal convex mixing combiner. The MMSE of this combination is given as

$$\min_\beta\left\{\sigma_d^2-p^T R^{-1}p+(\beta-\beta_o)^T R(\beta-\beta_o)\right\} \quad (9)$$

Subject to $1^T\beta=1$ and $\beta^{(i)}>0, i=1,.....,m$ where $\beta=[\beta^{(1)},.....,\beta^{(m)}]^T$ i.e, we have a

quadratic minimization problem which is the intersection plane of the affinity constraints and nonnegative orthant the cost function in (9) can be written as

$$J^c=J_{\min}+\|\beta-\beta_0\|_R^2$$

Which is in terms of the weighted norm of($\beta-\beta_0$).therefore, ignoring the constant term $J_{\min}$ the optimization problem in (9) as

$$\min_w\|\beta-\beta_0\|_R^2 \quad (10)$$

Which is the projection of β_0 to the simplex with respect to the weighted norm. We can further Write

$$J^c = J^a_{\min} + \|\beta - \beta_0\|^2_R - 2\frac{(1^T\beta_0 - 1)}{1^T R^{-1}1}(\beta - \beta_0^a)^T 1$$

The constant term $J^a_{\min}$ can be ignored and we can reformulate the optimization problem(10) Corresponding to the best convex coefficient as

$$\min_w \|\beta - \beta_0^a\|^2_R$$

Which is the projection of β_0^a to the unit simplex with respect to the weighted norm.

- If $\beta_0^a \in \Delta, i.e., \beta_0^a$ consists only nonnegative elements, then $\beta_o^c = \beta_0^a$;
- Otherwise, β_o^c is at the boundary of the constraint set Δ.

When there are two adaptive branches to be combined, then the projection is

$$\beta_o^c = \begin{cases} \beta_0^a & \beta_0^a \geq 0 \\ [0 \quad 1]^T & \beta_0^{a(1)} < 0 \\ [1 \quad 0]^T & \beta_0^{a(2)} < 0 \end{cases} \qquad (11)$$

Where $\beta \geq 0$ means all the entries of vector is nonnegative. As example, we present the case given in the first line of(11) in fig.3(a) and 3(b) represents the second and third entries of (11).with the level set cost function as the ellipse segment corresponding to the minimum achievable cost. The excess MSE corresponding to convex combination approach, relative to affine combination approach, is given by

$$J^c_{\min} - J^a_{\min} = \begin{cases} 0 & \beta_o^a \geq 0 \\ w_o^{a(1)\,2}\zeta & \beta_0^{a(1)} < 0 \\ w_o^{a(2)\,2}\zeta & \beta_0^{a(2)} < 0 \end{cases}$$

Where $\xi = R_{11} + R_{22} - 2R_{12}$ and R_{ij} is the element in row i and column jj.

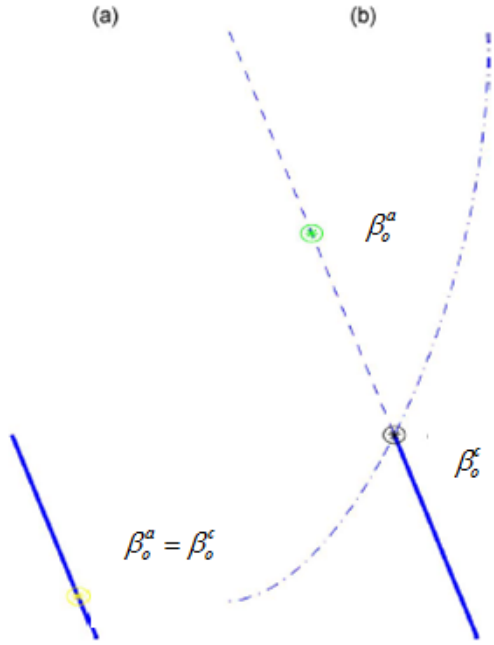

Fig2.Optimal mixture weights for affine versus convex combination of two adaptive filters

3. THE ADAPTIVE METHODS TO UPDATE AFFINE AND CONVEX COMBINATIONS

3.1 The LMS approach

When weights are constrained to be affine, we can use the following parameterization involving m-1 unconstrained weights $\beta^{(i)}(t) = z^{(i)}(t), i = 1, ..., m-1$

$$\beta^{(m)}(t) = 1 - \sum_{i=1}^{m-1} z^{(i)}(t)$$

Here, the m-1 dimensional vector $z(t) = [z^{(1)}(t), ..., z^{(m-1)}(t)]^T$ is the unconstrained weight vector. Hence, we transformed the constrained optimization problem into an unconstrained quadratic optimization problem. we note that when we combine just two filters, this update corresponds to stochastic gradient update given in[4].observing that $e(t) = d(t) - \beta^T(t)y(t)$ and if we z(t) as our weight vector, we have

$$e(t) = [d(t) - \hat{d}_m(t)] - z^T(t)\vec{\delta}(t)$$

Where

$\delta(t) = [\hat{d}_1(t) - \hat{d}_m(t), ..., \hat{d}_{m-1}(t) - \hat{d}_m(t)]^T$

.hence we have an adaptive filter problem with $[d(t) - \hat{d}_m(t)]$ as the desired signal and $\delta(t)$ as the input vector.

Since we have transformed affine constrained weights $\beta(t)$ into unconstrained weights z(t),we apply LMS update directly on z(t).
The update to minimize the variance of e(t) is given by

$$z(t+1) = z(t) - \frac{1}{2}\mu\nabla_z e^2(t)$$

$$= z(t) + \mu e(t)\delta(t)$$

3.2Adapting mixture weights using the RLS update

We apply the RLS update on z(t)using the desired signal $[d(t) - \hat{d}_m(t)]$ and the input vector $\delta(t)$ .the RLS update for the combination weights are given as

$$z(t+1) = z(t) + \Lambda^{-1}(t+1)e(t)\delta(t)$$

With $e(t) = d(t) - \hat{d}_m(t) - z^T(t)\delta(t)$ and

$$\Lambda^{-1}(t+1) = \lambda[\Lambda^{-1}(t) - \frac{\lambda^{-1}\Lambda^{-1}(t)\delta(t)\delta^T(t)\Lambda^{-1}(t)}{1+\lambda^{-1}\delta^T(t)\Lambda^{-1}(t)\delta(t)}]$$

Where $0< \lambda(t) <1$ is the forgetting factor, $\Lambda(t) = \sum_{l=1}^{t}\lambda^{t-1}\delta(l)\delta^T(l) + \lambda^t \in I$ is the estimated correlation matrix, $\Lambda(0) = \in I, \in$ is a small positive number.

4. TRACKING ANALYSIS USING ADAPTIVE NEURAL NETWORK

The original idea of applying artifical neural network (ANN) was to imitate the way the human brain processes information. For our purpose, ANN will simply be regarded as a convenient way to model the nonlinear input-output mapping of the process.

4.1 Identification of nonlinear system:

The structure for identification of nonlinear system is shown in fig:3 and requires the error between the output of the neural network $y_{NN}(t)$ and the output of the unknown system $y(t)$ .in this case the cost function can be minimized as

$$\bar{J} = \frac{1}{2}\sum_k (y(t) - y_{NN}(t)) \qquad (12)$$

The output of the neural network can be modelled by

$$y_{NN}(t) = f_{NN}(x(t-1), y(t-1))$$

Where $f_{NN}(.)$ represents the transfer function of the neural network which replaces the nolinear system $x(t-k), y(t-k)$ are vectors contain k delayed elements of x and y respectively starting from (t-1) i.e.,

$$x(t-k) = |x(t-1), x(t-2), ..., x(t-k)|^T ,$$

$$y(t-k) = |y(t-1), y(t-2), ..., y(t-k)|^T$$

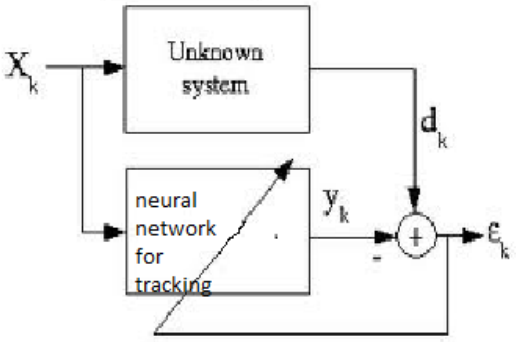

Fig3.identification of nonlinear system using ANNs

In this work, a two layered feed forward neural network architecture is been adopted to track the unknown system. The neural network structure of the neural network model has a two layer perceptron system and one output variable $y_{nn}()$.
The neural network model has the following input/output mapping relationship:

$$y_{nn}(t) = \sum_{j=1}^{N}\beta_j^0\rho_j(\beta_j^x x(t-1) + \beta_j^y y(k-1) + b_j) + b$$

Where ρ_j is the activation function of j^{th} in the hidden layer;

β_j^x represents the weight vector for the j^{th} neuron with respect to the inputs stored in $x(t-1)$;

β_j^y represents the weight vector for the j^{th} neuron with respect to the inputs stored in $y(k-1)$;

b_j represents the bias function for j^{th} neuron in hidden layer

β_j^0 represents the weight for output layer corresponding to j^{th} neuron from hidden layer;

B represent the bias function for the output layer;

To determine the mathematical model of the unknown system, weights are updated by the cost function(12).the weights can be recursively adjusted to minimize the cost function $\bar{j}(t)$ to its minimum value by gradient descent method. The weights are updated using:

$$\beta(t+1) = \beta(t) - \mu \frac{\partial \bar{J}(t)}{\partial \beta(t)}$$

Where mu is the positive learning rate

5. SIMULATIONS

In this section we demonstrate the performance of mixture approaches through simulation using stationary data. The combination structures provide improved performance over constituent filters.

The first set of experiments involves different combination structures which include constrained affine and convex combinations. To observe the accurateness of the results under different input parameters and different learning rates $\mu = [0.0008, 0.0010, 0.500, 0.100]$.We select the constituent algorithms as linear filters of different orders using both LMS and RLS updates to train their weight vectors $\beta_i(t) \in IR^i$.the input vectors that are fed to the unknown system model and the constituent filters are generated using the data model $d_i(t) = \beta_i^T(t) x_i(t)$.In fig(5) we plot weights and the error plots of different constituent filters with different learning rates. The simulation results show that the mixture

approach gives a better convergence rate over constituent filters.

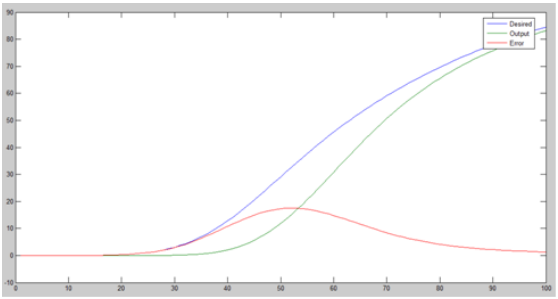

Fig5.filter weights of the combination structure

In Fig5. x-axis represents the time and y-axis represents the filter weights of the convex combination structure.

The second set of experiments include training of a neural network with the input vectors calculated above as inputs to the neural network and the unknown system and the outputs obtained by the constituent filters as targets to the neural network. We train the neural network for tracking the unknown system and calculating the MSE at each iteration. Neural network plots the MSE for the constituent filters, the gradient plot (stochastic gradient descent by which the weights are updated during tracking) and the regression plot for different learning rates of the constituent filters.

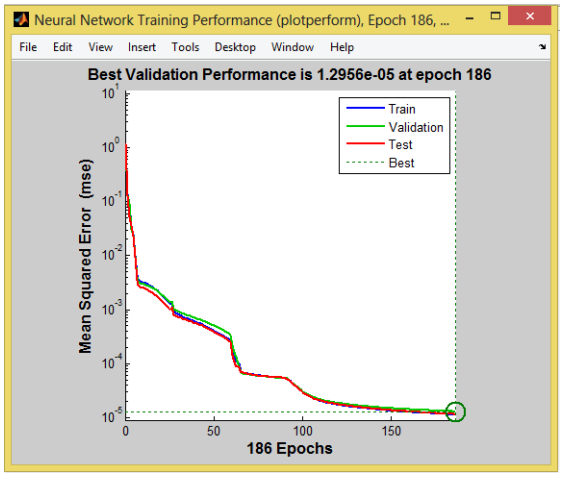

Fig.6 the mean square error plot

Fig6.shows the MSE and performance of the tracking system(the neural network).the x-axis represents the number of iterations performed and the y-axis gives the MSE value at each iteration and the tracking performance is estimated based on value of the MSE.

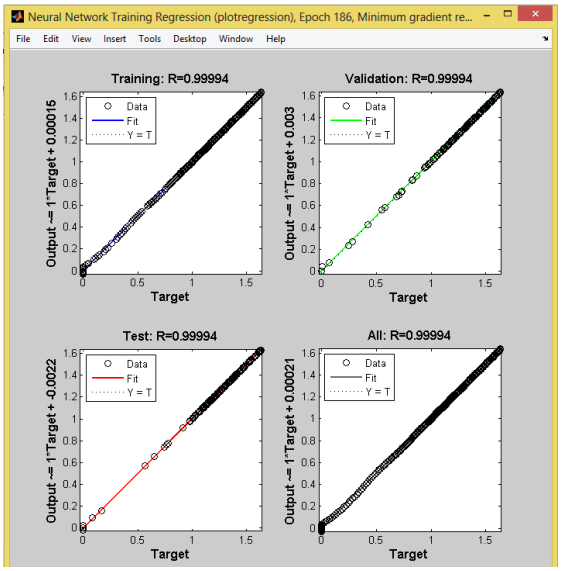

Fig7.regression plot of the tracking system

Fig7.shows the regression analysis of the tracking system for different learning rates.

6. CONCLUSION AND FUTURE SCOPE

In this paper, we investigated adaptive linear mixture approaches in terms of their final MSE in the steady state for stationary and non stationary environments. Our analysis is

generic such that the mixtures can be constructed based on several different adaptive filters each having a different adaptation method, structure or length. We demonstrated the performance gains when we use unconstrained affine and convex combination weights, and provided adaptive methods to achieve these results. We show that by using these mixture approaches, we can greatly improve upon the performance of the constituent filters by exploiting the cross-correlation information between the constituent filter outputs and biasing the combination weights toward zero for low SNR.

7. REFERENCES

[1] Suleyman Serdar Kozat, Member, IEEE, Alper Tunga Erdogan, Member, IEEE, Andrew C. Singer, Fellow, IEEE, and Ali H. Sayed, Fellow, IEEE,"steday state MSE performance analysis of mixture approaches to adaptive filtering",IEEETRANSACTIONS ON SIGNALPROCESSING VOL.,58,AUGUST 2010

[2] A. C. Singer and M. Feder, "Universal linear prediction by model order weighting," IEEE Trans. Signal Process., vol. 47, no. 10, pp. 2685–2699, Oct. 1999.

[3] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," IEEE Trans. Signal Process., vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[4] Neil J. Bershad, Fellow, IEEE, José Carlos M. Bermudez, Senior Member, IEEE, and Jean-Yves Tourneret, Member, IEEEAn "Affine Combination of Two LMS Adaptive Filters—Transient Mean-Square Analysis"IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 56, NO. 5, MAY 2008

[5] E. Eweda, "Comparison of RLS, LMS and sign algorithms for tracking randomly time-varying channels," IEEE Trans. Signal Process., vol. 42, no. 11, pp. 2937–2944, Nov. 1994.

[6] W. Zhuang, "RLS algorithm with variable forgetting factor for decision feedback equalizer over time-variant fading channels," Wireless Per- sonal Commun., vol. 8, pp. 15–29, 1998.

[7] A. H. Sayed, Fundamentals of Adaptive Filtering. New York: Wiley, 2003.

[8] Dejan Kihas, ˇZeljko M. Djurovi´c, Branko D. Kovaˇcevi´ c"Adaptive Filtering based on Recurrent Neural Networks"JOURNAL OF AUTOMATIC CONTROL,UNIVERSITY OF BELGRADE,VOL. 13(1):13-24, 2003 c

[9] S. S. Kozat, A. C. Singer, and G. Zeitler, "Universal piecewise linearprediction via context trees," *IEEE Trans. Signal Process.*, vol. 55, no.7, pp. 3730–3745, Jul. 2007.

[10] V. Vovk, "Competitive on-line statistics," *Int. Stat. Rev.*, vol. 69, pp.213–248, 2001.

[11] N. Littlestone and M. K.Warmuth, "The weighted majority algorithm,"*Inf. Comput.*, vol. 108, no. 2, pp. 212–261, 1994.

[12] D. Haussler and J. Kivinen, "Additive versus exponentiated gradientupdates for linear prediction," *J. Inf. Comput.*, vol. 132, no. 1, pp. 1–64,1997.