Interpretation and Convergence Analysis of Two Recently Introduced Adaptive Filter Algorithms (FEDS/RAMP)

M. Shams Esfand Abadi, John H. Husøy, and A. Mahlooji Far

***Abstract*—Fast Euclidean Direction Search (FEDS) and Recursive Adaptive Matching Pursuit (RAMP) are two recently introduced algorithms for adaptive filtering characterized by low computational complexity, good convergence, and numerical robustness. While conceived from quite different perspectives, we point out, that both algorithms are closely related and can be interpreted as different variants of 1) A matching pursuit procedure applied to a particular over-determined equation set, 2) A constrained Least Squares (LS) optimization problem, and 3) A Gauss-Seidel like iterative solution procedure applied to a normal equation. Both FEDS and RAMP have been demonstrated experimentally to possess excellent convergence behavior in several application scenarios. However, a tool for predicting the convergence of these algorithms based on second order statistics is lacking. This paper provides such a tool to study the convergence analysis of these adaptive filter algorithms. This tool relies on energy conservation arguments and doses not restrict to assume specific models for the regression data. Finally, we demonstrate through simulations that these results are useful in predicting adaptive filter performance.**

***Index Terms*—Adaptive filter, fast Euclidean direction search (FEDS), recursive adaptive matching pursuit (RAMP), convergence analysis.**

NOMENCLATURE

$\lvert\cdot\rvert$	Norm of a scalar.
$\lVert\cdot\rVert$	Euclidean norm of a vector.
$\lVert\underline{t}\rVert_{\Sigma}^2$	Σ-weighted Euclidean norm of a column vector $\underline{t}$ defined as $\underline{t}^T\Sigma\underline{t}$.
$vec(T)$	Creating an $M^2\times 1$ column vector $\underline{t}$ through stacking the columns of the $M\times M$ matrix T.
$vec(\underline{t})$	Creating an $M\times M$ matrix T from the $M^2\times 1$ column vector $\underline{t}$.
$A\otimes B$	Kronecker product of matrices A and B.
$Tr(.)$	Trace of a matrix.
$(.)^T$	Transpose of a vector or a matrix.
$E\{.\}$	Expectation operator.
$\oplus$	Modulo operator.

Manuscript received August 1, 2006; revised May 21, 2007.

M. Shams Esfand Abadi is with the Department of Electrical Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran. (e-mail: mshams@srttu.edu).

John H. Husøy is with the Department of Electrical and Computer Engineering, University of Stavanger, Stavanger, Norway. (e-mail: john.h.husoy@uis.no).

A. Mahlooji Far is with the Department of Electrical Engineering, Tarbiat Modarres University, Tehran, Iran (e-mail: mahlooji@modares.ac.ir).

Publisher Item Identifier S 1682-0053(08)1542

I. INTRODUCTION

Fast Euclidean Direction Search (FEDS) [1] and Recursive Adaptive Matching Pursuit (RAMP) [2] are two recently introduced algorithms for adaptive filtering characterized by low computational complexity, good convergence, and numerical robustness [3]. Both algorithms have been demonstrated to be useful in several applications [1], [2], [4]-[8].

These algorithms were originally derived from very different perspectives: The FEDS algorithm was formulated [1] as simplified conjugate gradient adaptive filter in which the search directions were restricted to the Euclidian directions $\underline{g}_i=[0,..,0,1,0,..,0]^T$, where the 1 appears in the i'th position. On the other hand, the RAMP algorithm was motivated [9], [2] through the application of a *matching pursuit* (MP) procedure to the iterative solution of the over determined equation set.

$$X^T(n)\underline{h}(n)=\underline{d}(n) \tag{1}$$

where

$$\underline{h}(n)=\left[h_0(n),h_1(n),\ldots,h_{M-1}(n)\right]^T \tag{2}$$

is the column vector of M filter coefficients at time n,

$$X^T(n)=\left[\underline{\tilde{x}}_0(n),\underline{\tilde{x}}_1(n),\ldots,\underline{\tilde{x}}_{M-1}(n)\right] \tag{3}$$

with these columns defined through

$$\underline{\tilde{x}}_j(n)=\left[x(n-j),x(n-j-1),\ldots,x(n-j-L+1)\right]^T \tag{4}$$

with $x(n)$ denoting the input signal to the adaptive filter. The selection of $L>M$ determines the memory of the adaptive algorithm. Note that the columns of $X(n)$, denoted by $\underline{x}_i(n)$ for $i=0,1,\ldots,L-1$, are

$$\underline{x}_i(n)=\left[x(n-i),x(n-i-1),\ldots,x(n-i-M+1)\right]^T \tag{5}$$

The vector $\underline{d}(n)$ of desired signal samples is given by

$$\underline{d}(n)=\left[d(n),d(n-1),\ldots,d(n-L+1)\right]^T \tag{6}$$

Given these seemingly differences in the conception of FEDS and RAMP it is important and interesting to observe that the algorithms are closely related. In the first part of the paper, we summarize and expand results of our previous research [9], [10], [2] by first interpreting the algorithms as different variants of (a) a *matching pursuit* procedure [11] applied to a particular over- determined equation set, (b) a constrained Least Squares (LS) optimization problem, and finally (c) a Gauss-Seidel like iterative solution procedure applied to a normal equation. Based on this, the exact relationship between FEDS and RAMP is identified.

While strong empirical evidence support the claims about good convergence behavior of both FEDS and RAMP, a tool for predicting the convergence of these algorithms based on second order statistics is lacking. In the case of transient analysis, or convergence analysis, important recent contributions are the analysis of data normalized adaptive algorithms [12] (for example the Normalized Least Mean SHVquare (NLMS) algorithm), the (family of) Affine Projection Algorithm(s) (APA) [13], [14] and Data-Reusing adaptive algorithms [15] where excellent agreement between theoretically obtained results and simulations are obtained. The convergence analysis of these algorithms relies on energy conservation arguments and does not restrict the regressors to assume specific models for the regression data.

We also presented in [16], a general framework to convergence analysis of adaptive filters. In the next section the interpretations of FEDS and RAMP algorithms are presented. In the third part of the paper, we provide a tool to study the convergence analysis of FEDS and RAMP algorithms based on energy conservation arguments. Following this, we present the expressions for the learning curve, the excess mean square error and the mean square coefficient deviation. We conclude the paper by showing a comprehensive set of simulations supporting the validity of our approach.

II. Interpretations of FEDS/RAMP

In the interpretations to be presented here, we shall use the objective function

$$\left\|\underline{d}(n)-X^T(n)\underline{h}(n)\right\|^2 \tag{2}$$

whose minimum with respect to $\underline{h}(n)$ is sought. A direct solution to this problem, for each time index n, leads to a sliding window *Recursive Least Squares* (RLS) algorithm. By slightly modifying the definition of the $X^T(n)$ and $\underline{d}(n)$, we can alternatively obtain an *exponentially weighted* RLS algorithm. Both implementation alternatives and numerical properties of such algorithms have been extensively studied [17], [18]. To summarize, such algorithms are characterized by high computational complexity and numerical stability problems related to the need to recursively compute the inverse of an estimated autocorrelation matrix. It is a common experience that the more efficient the implementation of an RLS algorithm is, the more severe are the attendant numerical problems, see for example page 128 of [19].

Both FEDS and RAMP can be viewed as attempts at devising low complexity adaptive filter algorithms with good numerical properties. For both algorithms this is done by filter coefficient update equations *updating only one element of the filter vector at a time*. At each time instant, n, we can perform one or more such updates. The number of such single coefficient updates performed at each time instant is denoted by P. With this objective function of (7) and a constraint on its recursive minimization given by the restriction that only one element of $\underline{h}(n)$ is to be updated at a time, we can adopt several viewpoints as detailed below. For clarity of presentation, we shall now use notation $\underline{h}^{(prev)}$ and $\underline{h}^{(new)}$ to denote the filter vector before and after an update. Three possible , and in retrospect, equivalent points of view related to the problem as stated above are as follows.

A. Matching Pursuit

Given the filter vector $\underline{h}^{(prev)}$ before an update, the residual vector associated with (7) is [2], [9], [11]

$$\underline{e}_a(n)=\underline{d}(n)-X^T(n)\underline{h}^{(prev)} \tag{8}$$

We may interpret $\underline{e}_a(n)$ as the error associated with the approximation of $\underline{d}(n)$ as a linear combination of columns of $X^T(n)$ where the elements of $\underline{h}^{(prev)}$ are the associated weighting coefficients. Updating one element of $\underline{h}^{(prev)}$, say, corresponds to adding Δ_i times column no. i of $X^T(n)$, i.e. $\underline{\tilde{x}}_i(n)$, to the current approximation of $\underline{d}(n)$. To obtain maximum improvement to the approximation of $\underline{d}(n)$, it is obvious that $\Delta_i\,\underline{\tilde{x}}_i(n)$ must be selected as the best possible approximation to $\underline{e}_a(n)$ onto each unit length column $\underline{\tilde{x}}_i(n)/\left\|\underline{\tilde{x}}_i(n)\right\|$, $i=0,1,\ldots,M-1$ and identifying the index j corresponding to the maximum projection. In other words

$$j=\arg\max_i\left|\underline{e}_a^T(n)\underline{\tilde{x}}_i(n)\right/\left\|\underline{\tilde{x}}_i(n)\right\| \tag{9}$$

The corresponding approximation to $\underline{e}_a(n)$ is the projection in the direction of the selected vector $\underline{\tilde{x}}_j(n)$, i.e.

$$\underline{e}_a^T(n)\frac{\underline{\tilde{x}}_j(n)}{\left\|\underline{\tilde{x}}_j(n)\right\|}\cdot\frac{\underline{\tilde{x}}_j(n)}{\left\|\underline{\tilde{x}}_j(n)\right\|}= \underline{e}_a^T(n)\underline{\tilde{x}}_j(n)\left\|\underline{\tilde{x}}_j(n)\right\|^{-2}\underline{\tilde{x}}_j(n) \tag{10}$$

Based on this we identify j given above as the index of the element of $\underline{h}^{(prev)}$ to be updated and the corresponding update equation as

$$h_j^{(new)}=h_j^{(prev)}+\underline{e}_a^T(n)\underline{\tilde{x}}_j(n)\left\|\underline{\tilde{x}}_j(n)\right\|^{-2}. \tag{11}$$

The procedure described above directly corresponds to the application of one step pf a *Basic Matching Pursuit* (BMP) procedure to the problem of approximating vector $\underline{d}(n)$ by a linear combination of dictionary vectors $\left\{\underline{\tilde{x}}_0(n),\underline{\tilde{x}}_1(n),\ldots,\underline{\tilde{x}}_{M-1}(n)\right\}$ [20].

B. Constrained LS

Directly attacking (7) with a postulated update equation given by [11]

$$\underline{h}^{(new)}=\underline{h}^{(prev)}+\Delta_j\,\underline{1}_j \tag{12}$$

where $\underline{1}_j$ denotes a vector of all zeros except element j which is a 1, and minimizing with respect to Δ_j gives the same update equation as (11). In selecting the index of the filter coefficient to update, j, this is selected as the index for which the update equation gives the maximum reduction of the residual norm. Not surprisingly, j is to be selected as in (9).

C. Gauss-Seidel

The Gauss-Seidel [2] iterative scheme for the solution of sets of linear equations $A\underline{x}=\underline{b}$ is given by the following element updates for $j=0,1,\ldots,M-1$ [21]

$$x_j^{(new)} = x_j^{(prev)} + \frac{1}{a_{j,j}}\left\{b_j - \sum_{l=0}^{M-1} a_{j,l} x_l^{(prev)}\right\} \qquad (13)$$

where the indexed quantities have obvious definitions. Applying this to normal equation associated with (7), i.e.

$$X(n)X^T(n)\underline{h}(n) = X(n)\underline{d}(n) \qquad (14)$$

the element update equations for $j = 0,1,\ldots,M-1$ are given by

$$\begin{aligned} h_j^{(new)} &= h_j^{(prew)} + \frac{\underline{\tilde{x}}_j^T(n)}{\left\|\underline{\tilde{x}}_j(n)\right\|^2}\left\{\underline{d}(n) - \sum_{l=0}^{M-1} h_l^{(prev)}\underline{\tilde{x}}_l(n)\right\} \\ &= h_j^{(prev)} + \frac{\underline{\tilde{x}}_j^T(n)}{\left\|\underline{\tilde{x}}_j(n)\right\|^2}\left\{\underline{d}(n) - X^T(n)\underline{h}^{(prev)}\right\} \end{aligned} \qquad (15)$$

In view of (8) we see that this update equation is exactly the same as (11). Again the question as the selection of j, the index of the element of $\underline{h}^{(prev)}$ to update, surfaces. In direct application of the Gauss-Seidel procedure it is common practice to start with $j = 0$ and increment j by one counted modulo M before each update computation. It is also possible to select j, before each update computation, such that the update computation has the *maximum residual reducing effect*. This variant of the Gauss-Seidel method is referred to as the *Southwell method*. This method enjoyed some popularity among civil engineers in conjunction with stress computations before the use of computers in the solution of linear equations became widespread. Again, it should come as no surprise that the maximum residual reducing selection of the coefficient to update is given by (9).

In summary, with three different lines of thought, we have established the update equation

$$h_{j(n)}^{(new)} = h_{j(n)}^{(prev)} + \frac{\underline{\tilde{x}}_{j(n)}^T(n)}{\left\|\underline{\tilde{x}}_{j(n)}(n)\right\|^2}\cdot\left\{\underline{d}(n) - X(n)\underline{h}^{(prev)}\right\} \qquad (16)$$

when coefficient $j(n)$ has been identified as the one to update. We have identified two ways of selecting $j(n)$: (a) incrementing $j(n)$ sequentially by n modulo M $(n \oplus M)$, and (b) selecting $j(n)$ in a such a way as to maximally reduce the residual of the corresponding update computation.

The former selection in conjunction with (16) results in the RAMP algorithm. It is convenient to write the update equation as

$$\underline{h}^{(new)} = \underline{h}^{(prev)} + \frac{1}{\left\|\underline{\tilde{x}}_{j(n)}\right\|^2} i_{j(n)} X(n) \times \{\underline{d}(n) - X^{(T)}(n)\underline{h}^{(prev)}\} \qquad (17)$$

where $i_{j(n)}$ is the $M \times M$ matrix with a 1 in position $(j(n), j(n))$ and zeros in all other positions, i.e. a matrix leaving row no. $j(n)$ of the matrix with which it is premultiplied and zeroing out all the other rows. If more than one coefficient update is to be performed for each time instant, n, we substitute notation $j(n)$ by $j_l(n)$ with $l = 0,1,\ldots,P-1$ where P is the number of updates to perform at each sample time.

We close this section by pointing out that efficient implementations of FEDS/RAMP are available. For exponentially weighted and sliding window versions, it is well known that implementations having a multiplicative complexity given by $(5+P)M$ can be devised [2]. If we use a block exponentially weighted version [1], implementations with a multiplicative complexity of $(3+P)M$ are possible.

III. INTERPRETATIONS OF FEDS/RAMP

The transient behavior of any adaptive filtering algorithm is clearly of interest. The analysis strategy adopted below is based on [14] in which the performance of the APA was analyzed. However the scope of the present analysis is to study the convergence behavior of FEDS and RAMP algorithms.

The convergence behavior of an adaptive filter algorithm is determined by the evolution of the expected squared a priori error with time n, i.e. $E\{e_a^2(n)\}$. The a priori error is defined as

$$e_a(n) = \underline{x}^T(n)\left[\underline{h}_t - \underline{h}(n)\right] \qquad (19)$$

where $\underline{h}_t$ is the unknown filter vector we are trying to estimate. Defining the coefficient deviation vector $\underline{\varepsilon}(n) = \underline{h}_t - \underline{h}(n)$, we have $e_a(n) = \underline{x}^T(n)\underline{\varepsilon}(n)$. This implies that

$$E\{e_a^2(n)\} = E\{\underline{\varepsilon}^T(n)R\,\underline{\varepsilon}(n)\} = E\{\left\|\underline{\varepsilon}(n)\right\|_R^2\} \qquad (20)$$

where again the definition of the autocorrelation matrix $R = E\{\underline{x}(n)\underline{x}^T(n)\}$ has been used. Thus, to find the learning curve, we need to find $E\{\left\|\underline{\varepsilon}(n)\right\|_R^2\}$ as a function of n.

Indeed we can find a recursion for $E\{\left\|\underline{\varepsilon}(n)\right\|_\Sigma^2\}$, where Σ is some positive definite symmetric matrix of dimension commensurate with that of $\underline{\varepsilon}(n)$. Assuming a linear model for the desired signal, $d(n)$, given by

$$d(n) = \underline{x}^T(n)\underline{h}_t + v(n) \qquad (21)$$

which we prefer to express as

$$\underline{d}(n) = X^T(n)\underline{h}_t + \underline{v}(n) \qquad (22)$$

where $\underline{v}(n)$ is measurement noise assumed to be zero mean, white, Gaussian, and independent of the input signal matrix $X(n)$. Now we prefer to replace $\underline{h}^{(new)}$ and $\underline{h}^{(prev)}$ by $\underline{h}(n+1)$ and $\underline{h}(n)$, respectively. Therefore we can write (17) in the form of (23)

$$\underline{h}(n+1) = \underline{h}(n) + C(n)X(n)\underline{e}(n) \qquad (23)$$

where

$$C(n) = \frac{1}{\left\|\underline{\tilde{x}}_{j(n)}(n)\right\|^2} i_{j(n)} \qquad (24)$$

and

$$\underline{e}(n) = \underline{d}(n) - X^T(n)\underline{h}(n) \qquad (25)$$

is the output estimation error vector. If $P \succ 1$, we get somewhat more involved expression for $C(n)$. Finding the general expression for $C(n)$, when $P \succ 1$, is explained in detail in Appendix.

Using the $\underline{\varepsilon}(n) = \underline{h}_t - \underline{h}(n)$, we obtain from (23),

$$\underline{\varepsilon}(n+1) = \underline{\varepsilon}(n) - C(n)X(n)\underline{e}(n) \qquad (26)$$

from (25) and (22), the output estimation error vector $\underline{e}(n)$ can be stated as

$$\underline{e}(n) = X^T(n)\underline{\varepsilon}(n) + \underline{v}(n) \tag{27}$$

substitute (27) in (26), we obtain

$$\underline{\varepsilon}(n+1) = \underline{\varepsilon}(n) - C(n)X(n)(X^T(n)\underline{\varepsilon}(n) + \underline{v}(n)) \tag{28}$$

now taking the Σ-weighted norm from the both sides of (28)

$$\|\underline{\varepsilon}(n+1)\|_{\Sigma}^2 = \|\underline{\varepsilon}(n)\|_{\Sigma'}^2 + \underline{v}^T(n)X^{\Sigma}(n)\underline{v}(n) + \{\textit{Cross terms involving one instance of } \underline{v}(n)\} \tag{29}$$

where

$$\Sigma' = \Sigma - \Sigma C(n)X(n)X^T(n) - X(n)X^T(n)C^T(n)\Sigma + X(n)X^{\Sigma}(n)X^T(n) \tag{30}$$

and

$$X^{\Sigma}(n) = X^T(n)C^T(n)\Sigma C(n)X(n) \tag{31}$$

taking the expectation from both sides of (29)

$$E\{\|\underline{\varepsilon}(n+1)\|_{\Sigma}^2\} = E\{\|\underline{\varepsilon}(n)\|_{\Sigma'}^2\} + E\{\underline{v}^T(n)X^{\Sigma}(n)\underline{v}(n)\} \tag{32}$$

we obtain the time evolution of the weight-error variance. The expectation of $\|\underline{\varepsilon}(n)\|_{\Sigma'}^2$ is difficult to calculate because of dependency of Σ' on $C(n)$, $X(n)$, and of $\underline{\varepsilon}(n)$ on prior regressors. To solve this problem we need to use the following independent assumptions [14].

1) The matrix sequence $X(n)$ is independent and identically distributed.
2) $\underline{\varepsilon}(n)$ is independent of $C(n)X(n)X^T(n)$.

using these independence assumptions, the final result is

$$E\{\|\underline{\varepsilon}(n+1)\|_{\Sigma}^2\} = E\{\|\underline{\varepsilon}(n)\|_{\Sigma'}^2\} + E\{\underline{v}^T(n)X^{\Sigma}(n)\underline{v}(n)\} \tag{33}$$

where now Σ' is

$$\Sigma' = \Sigma - \Sigma E\{C(n)X(n)X^T(n)\} - E\{X(n)X^T(n)C^T(n)\}\Sigma + E\{X(n)X^{\Sigma}(n)X^T(n)\} \tag{34}$$

Looking only at the second term of the right hand side of (33) we have

$$E\{\underline{v}^T(n)X^{\Sigma}(n)\underline{v}(n)\} = E\{Tr(\underline{v}(n)\underline{v}^T(n)X^{\Sigma}(n))\} = Tr(E\{\underline{v}(n)\underline{v}^T(n)\}E\{X^{\Sigma}(n)\}) \tag{35}$$

since $E\{\underline{v}(n)\underline{v}^T(n)\} = \sigma_v^2 I$, (33) can be stated as

$$E\{\|\underline{\varepsilon}(n+1)\|_{\Sigma}^2\} = E\{\|\underline{\varepsilon}(n)\|_{\Sigma'}^2\} + \sigma_v^2 Tr(E\{X^{\Sigma}(n)\}) \tag{36}$$

Now using the *vec* operation in the both sides of (34)

$$vec(\Sigma') = vec(\Sigma) - vec(\Sigma E\{C(n)X(n)X^T(n)\}) - vec(E\{X(n)X^T(n)C^T(n)\}\Sigma) + vec(E\{X(n)X^{\Sigma}(n)X^T(n)\}) \tag{37}$$

since, in general $vec(P\Sigma Q) = (Q^T \otimes P)vec(\Sigma)$, we find that (37) can be written as

$$\underline{\sigma}' = \underline{\sigma} - (E\{X(n)X^T(n)C^T(n)\} \otimes I).\underline{\sigma} - (I \otimes E\{X(n)X^T(n)C^T(n)\}).\underline{\sigma} + (E\{(X(n)X^T(n)C^T(n)) \otimes (X(n)X^T(n)C^T(n))\}).\underline{\sigma} \tag{38}$$

where $\underline{\sigma}' = vec(\Sigma')$ and $\underline{\sigma} = vec(\Sigma)$. With definition of the $M^2 \times M^2$ matrix G

$$G = I - E\{X(n)X^T(n)C^T(n)\} \otimes I - I \otimes E\{X(n)X^T(n)C^T(n)\} + E\{(X(n)X^T(n)C^T(n)) \otimes (X(n)X^T(n)C^T(n))\} \tag{39}$$

Equation (38) can be stated as

$$\underline{\sigma}' = G.\underline{\sigma} \tag{40}$$

Looking at the second term of the right hand side of (36)

$$Tr(E\{X^{\Sigma}(n)\}) = Tr(E\{C(n)X(n)X^T(n)C^T(n)\}.\Sigma) \tag{41}$$

and defining $\underline{\gamma}$ through

$$\underline{\gamma} = vec(E\{C(n)X(n)X^T(n)C^T(n)\}), \tag{42}$$

then

$$Tr(E\{C(n)X(n)X^T(n)C^T(n)\}) = \underline{\gamma}^T.\underline{\sigma} \quad . \tag{43}$$

With the above considerations, the recursion of (36) can now be stated as

$$E\{\|\underline{\varepsilon}(n+1)\|_{\underline{\sigma}}^2\} = E\{\|\underline{\varepsilon}(n)\|_{G\underline{\sigma}}^2\} + \sigma_v^2 \underline{\gamma}^T \underline{\sigma} \quad . \tag{44}$$

Focusing again on the learning curve, we substitute R for Σ, define $\underline{r} = vec(R)$, and find

$$E\{e_a^2(n)\} = E\{\|\underline{\varepsilon}(n)\|_{\underline{r}}^2\} = E\{\|\underline{\varepsilon}(0)\|_{G^n\underline{r}}^2\} + \sigma_v^2 \underline{\gamma}^T \{I + G + \ldots + G^{n-1}\}\underline{r} \tag{45}$$

This expression is easy to compute recursively once we have estimates for G and R. Such estimates are easily obtained from a *single realization* of the signals involved in the adaptive filter. From this recursion, we will be able to evaluate excess mean square error (EMSE), when n goes to infinity

$$EMSE = \sigma_v^2 \underline{\gamma}^T (I - G)^{-1}\underline{r} \tag{46}$$

and the mean square coefficient deviation (MSD) is given by

$$MSD = \sigma_v^2 \underline{\gamma}^T (I - G)^{-1} vec(I) \tag{47}$$

since $e(n) = \underline{x}^T(n)\underline{\varepsilon}(n) + v(n)$, we obtain that

$$E\{e^2(n)\} = E\{e_a^2(n)\} + \sigma_v^2 \tag{48}$$

According to (48), the expression for the steady state mean square error (MSE) is given by

$$MSE = \sigma_v^2 \underline{\gamma}^T (I - G)^{-1}\underline{r} + \sigma_v^2 \tag{49}$$

IV. Simulation Results

In a system identification setup we applied an input signal $x(n)$ generated through $x(n) = \rho x(n-1) + w(n)$ which is a first order autoregressive signal. $w(n)$ can be either zero mean white Gaussian signal or a zero mean uniformly distributed random sequence between -1 and 1. In the first case, we set $\rho = 0.9$, and in the second case, $\rho = 0.5$. With $\rho = 0.9$ and $\rho = 0.5$, a highly colored and a somewhat colored signal will be generated respectively. The unknown filter $\underline{h}_t$ was selected at a random length 8 vector and a initial filter $\underline{h}(0)$ was set to the zero vector.

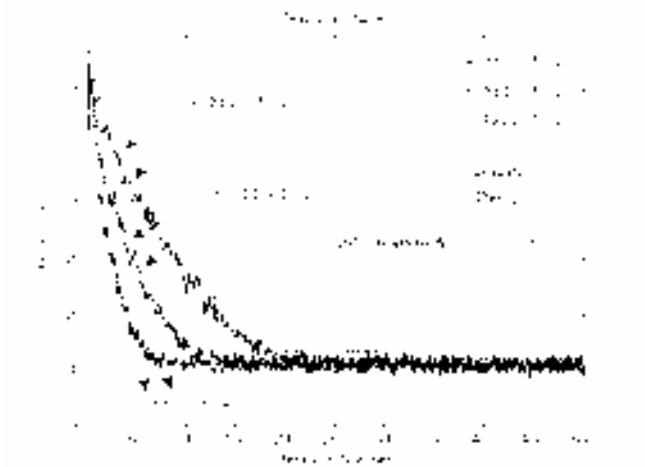

Fig. 1. Simulated and theoretical learning curves for FEDS algorithm with $L = 32$ and various selections for P. Highly colored ($\rho = 0.9$) Gaussian input.

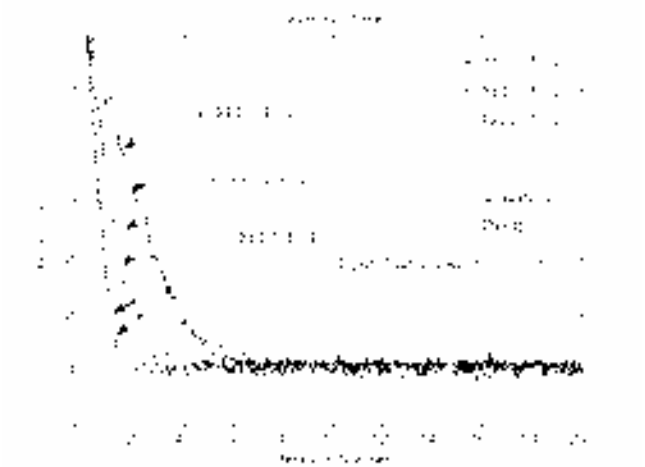

Fig. 4. Simulated and theoretical learning curves for FEDS algorithm with $L = 64$ and various selections for P. Somewhat colored ($\rho = 0.5$) uniformly distributed input.

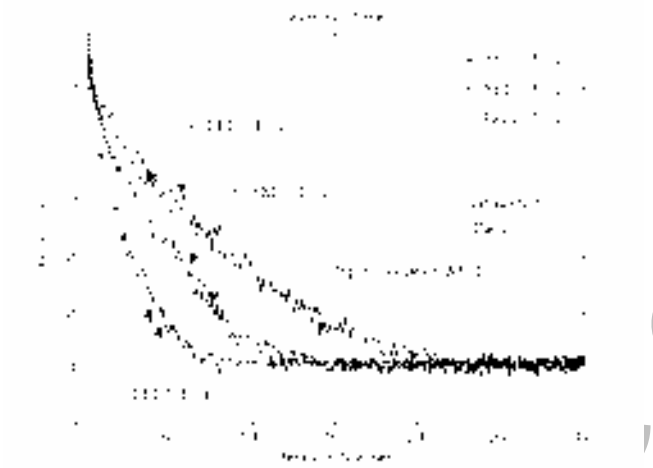

Fig. 2. Simulated and theoretical learning curves for FEDS algorithm with $L = 64$ and various selections for P. Highly colored ($\rho = 0.9$) Gaussian input.

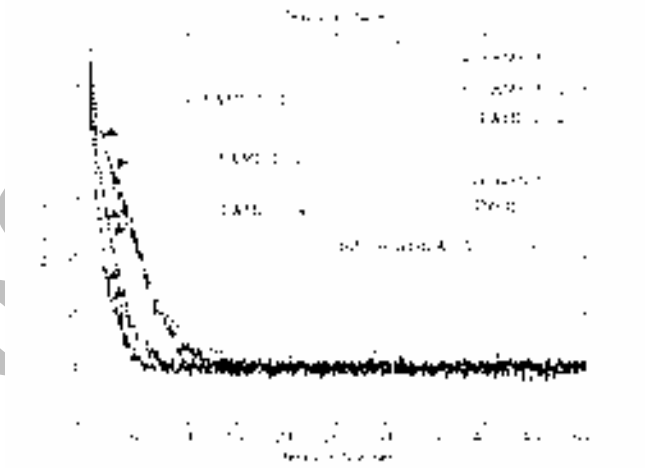

Fig. 5. Simulated and theoretical learning curves for RAMP algorithm with $L = 32$ and various selections for P. Highly colored ($\rho = 0.9$) Gaussian input.

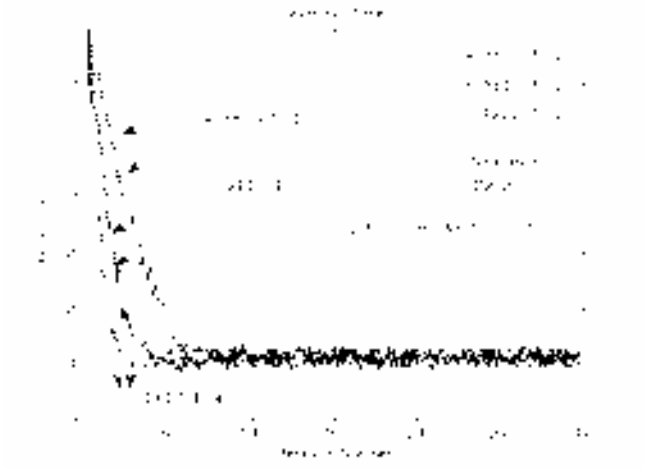

Fig. 3. Simulated and theoretical learning curves for FEDS algorithm with $L = 32$ and various selections for P. Somewhat colored ($\rho = 0.5$) uniformly distributed input.

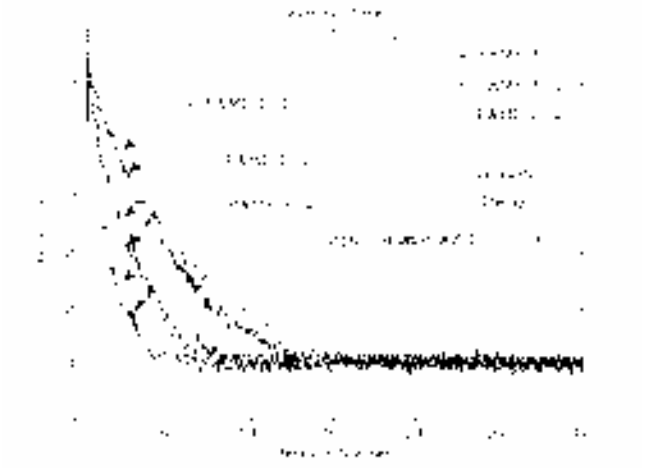

Fig. 6. Simulated and theoretical learning curves for RAMP algorithm with $L = 64$ and various selections for P. Highly colored ($\rho = 0.9$) Gaussian input.

The assumed filter lengths of the FEDS/RAMP algorithms were also set to 8 and the window length was set to $L = 32$ and in the other experiment was set to $L = 64$. White measurement noise, $v(n)$, uncorrelated with $x(n)$ and with $\sigma_v^2 = 10^{-3}$ was added to the noise free desired signal generated through $d(n) = \underline{h}_t^T \underline{x}(n)$.

The simulated learning curves were obtained by averaging the estimation error over 200 independent realizations. The theoretical learning curves obtained by adding a fixed value, $\sigma_v^2 = 10^{-3}$, to $E\{e_a^2(n)\}$ found using the recursion of (45). Simulated and computed learning curves for the FEDS/RAMP for various selections of P

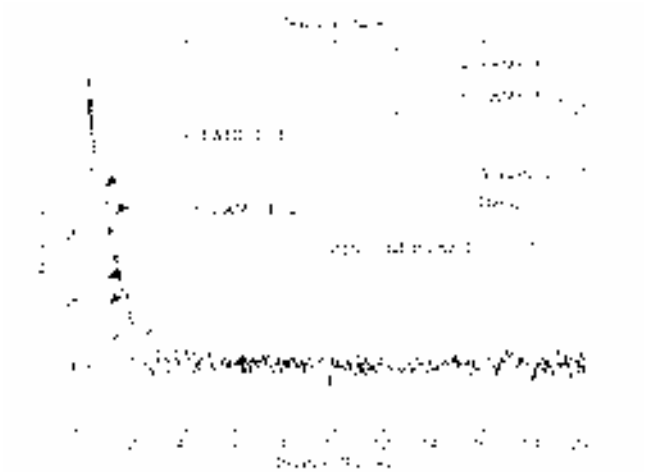

Fig. 7. Simulated and theoretical learning curves for RAMP algorithm with $L = 32$ and various selections for P. Somewhat colored ($\rho = 0.5$) uniformly distributed input.

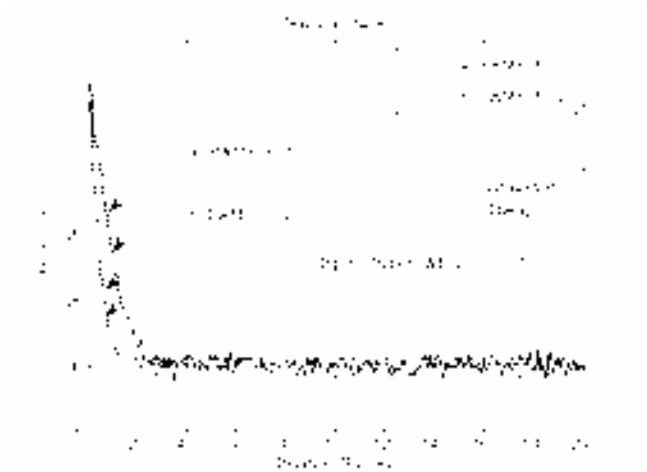

Fig. 8. Simulated and theoretical learning curves for RAMP algorithm with $L = 64$ and various selections for P. Somewhat colored ($\rho = 0.5$) uniformly distributed input.

and window length L and for two input signals are given in Figs. 1-8. As is readily observed, we have reasonably good agreement between actual and computed learning curves in all figures. Thus, our claim regarding the usefulness of (45) is justified.

V. Conclusions

We have presented three alternative interpretations of the recently introduced FEDS and RAMP adaptive filter algorithms showing that the two algorithms are essentially the same except for the way in which the index of the element of the filter vector to be updated is determined. In FEDS the coefficients are updated in a sequential manner, whereas in RAMP the coefficient update resulting in the maximum reduction in the residual norm is determined prior to each update. Interestingly, this difference corresponds directly to the difference between a "straight" Gauss-Seidel approach and the Southwell method to the iterative solution of linear equation sets. For both algorithms, we have also presented a convergence analysis resulting in a recursion for theoretically computing the learning curve. Following this we provided the expressions for the learning curve, the excess mean square error and the mean square coefficient deviation. The value of this recursion as a tool in predicting the convergence speed of the algorithms has been demonstrated.

Appendix

In this appendix, we explain how can we find the general expression for $C(n)$, when $P \succ 1$. Initialize $\underline{h}$ at new time instant.

$$\underline{h}^{[0]}(n+1) = \underline{h}^{[p]}(n) \tag{50}$$

Therefore, the output error vector can be stated as

$$\underline{e}_0(n) = \underline{d}(n) - X^T(n)\underline{h}^{[0]}(n+1) \tag{51}$$

When $P \succ 1$, the following algorithm is performed during each new time instant n,

For $l = 0,1,\ldots,P-1$

$$j_l(n) = \arg\max_i \left| e_l^T(n)\underline{x}_i(n) \right| . \left\| \underline{x}_i(n) \right\|^{-1} \quad (RAMP)$$

$$j_l(n) = [nP + l] \oplus M \quad (FEDS)$$

$$\underline{h}^{[l+1]}(n+1) = \underline{h}^{[l]}(n+1) + i_{j_l(n)} \cdot \left\| \underline{x}_{j_l(n)}(n) \right\|^{-2} \times X(n)\underline{e}_l(n)$$

$$\underline{e}_{l+1}(n) = \left\{ I - X^T(n) C_{j_l(n)}(n) X(n) \right\} \underline{e}_l(n)$$

End

where $C_{j_l(n)}(n)$ in the last row of this algorithm is

$$C_{j_l(n)}(n) = i_{j_l(n)} \cdot \left\| \underline{x}_{j_l(n)}(n) \right\|^{-2} . \tag{52}$$

The detail of the last row in this algorithm can be explained from the following relations

$$\begin{aligned} &\underline{e}_{l+1}(n) = \underline{d}(n) - X^T(n)\underline{h}^{[l+1]}(n+1) = \\ &\underline{d}(n) - X^T(n)\left[\underline{h}^{[l]}(n+1) + C_{j_l(n)}(n) X(n)\underline{e}_l(n) \right] = \\ &\left\{ I - X^T(n) C_{j_l(n)}(n) X(n) \right\} \underline{e}_l(n) \end{aligned} \tag{53}$$

where $\underline{e}_l(n) = \underline{d}(n) - X^T(n)\underline{h}^{[l]}(n+1)$.

From the above considerations, we can write the following relation between iteration $P-1$ and P

$$\begin{aligned} &\underline{h}^{[p]}(n+1) = \underline{h}^{[p-1]}(n+1) + \\ &i_{j_{p-1}(n)} \cdot \left\| \underline{x}_{j_{p-1}(n)}(n) \right\|^{-2} X(n)\underline{e}_{p-1}(n) \end{aligned} \tag{54}$$

which with the definition

$$C_l(n) = i_{j_l(n)} \cdot \left\| \underline{x}_{j_l(n)}(n) \right\|^{-2} \tag{55}$$

gives

$$\begin{aligned} \underline{h}^{[p]}(n+1) = \underline{h}^{[p-1]}(n+1) \\ + C_{[p-1]}(n) X(n)\underline{e}_{[p-1]}(n) \end{aligned} \tag{56}$$

Now by using on (56), we find the following relation between iteration P and 0

$$\underline{h}^{[p]}(n+1) = \underline{h}^{[0]}(n+1) + \sum_{i=0}^{p-1} C_i(n) X(n)\underline{e}_i(n) . \tag{57}$$

But from (53), $\underline{e}_i(n)$ can be stated as

$$\underline{e}_i(n) = \prod_{l=0}^{i-1}\{I - X^T(n)C_l(n)X(n)\}\underline{e}_0(n). \qquad (58)$$

Finally, the update equation from iteration $n+1$ to n and with P iteration in each new time instant n is

$$\underline{h}(n+1) = \underline{h}(n) + \sum_{i=0}^{P-1}\{C_i(n)X(n)\times \prod_{l=0}^{i-1}\{I - X^T(n)C_l(n)X(n)\}\}\underline{e}(n) \qquad (59)$$

Comparing (59) with (23), we find that to evaluate the performance of FEDS and RAMP algorithms, when $P \succ 1$, we need to modify (59) to find the matrix $C(n)$. Looking again at (59), the update equation can be written as in the form of (60)

$$\underline{h}(n+1) = \underline{h}(n) + \sum_{i=o}^{p-1}\{C_i(n) \times \prod_{l=0}^{i-1}\{I - X^T(n)C_l(n)X(n)\}\}.X(n)\underline{e}(n) \qquad (60)$$

Therefore, when $P \succ 1$, the matrix $C(n)$ can be found from (61).

$$C(n) = \sum_{i=0}^{p-1}C_i(n).\prod_{l=0}^{i-1}[I - X(n)X^T(n)C_l(n)] \qquad (61)$$

ACKNOWLEDGMENT

We would like to express our thanks to ITRC for their financial support of this project (TMU 85-12-85).

REFERENCES

[1] T. Bose and G. F. Xu, "The euclidean direction search algorithm adaptive filtering," *IEICE Trans, Fundamentals*, vol. E85-A, no. 3, pp. 532–539, Mar. 2002.

[2] J. H. Husøy, "RAMP: An adaptive filter with links to matching pursuits and iterative linear equation solver," in *Proc. ISCAS*, vol. 4, pp. 381-384, Bangkok, Thailand, May 2003.

[3] J. H. Husøy, "A comparative study of some simplified RLS-type algorithms," in *Proc. First Int. Symp. on Control, Communications and Signal Processing*, pp. 705-708, Hammamet, Tunisia, Mar. 2004.

[4] J. H. Husøy, J. Eilevstjønn, T. Eftestøl, S. O. Aase, H. Myklebust, and P. A. Steen, "Removal of cardiopulmonary resuscitation artifacts from human ECG using an efficient matching pursuit-like algorithm," *IEEE Trans. on Biomedical Engineering*, vol. 49, no. 11, pp. 1287-1298, Nov. 2002.

[5] M. S. E. Abadi and J. H. Husøy, "Channel equalization using recursive adaptive matching pursuit algorithm," in *Proc. Iranian Conf. on Electrical Engineering*, pp. 265-268, Zanjan, Iran, May 2005.

[6] M. S. E. Abadi, A. M. Far, S. Daneshvar, and M. Lotfizad, "Recursive adaptive matching pursuit in noise cancellation for speech enhancement," in *Proc. 2nd Int. Conf. on Information and Knowledge Technology*, Tehran, Iran, Jun. 2005.

[7] M. S. E. Abadi, A. M. Far, E. Kabir, and R. Ebrahimpour, "Image restoration using two dimensional fast Euclidean direction search based adaptive algorithm," in *Proc. 4th IEEE Int. Workshop WSTST'05*, pp. 182-191, Muroran, Japan, May 2005.

[8] M. S. E. Abadi and A. M. Far, "Two dimensional recursive adaptive matching pursuit filter," in *Proc. 11th Int. CSI (Computer Society of Iran) Computer Conference*, pp. 240-246, Tehran, Iran, Jan. 2006.

[9] J. Ommundsen and J. H. Husøy, "An adaptive filter based on matching pursuits," in *Proc. NORSIG*, pp. 40-44, Trondheim, Norway, Oct. 2001.

[10] J. H. Husøy and J. Ommundsen, "A novel algorithm for adaptive filters based on optimum selective update of coefficients," in *Proc. Applied Electronics*, pp.114-121, Plzen, Czech Republic, Sep. 2001.

[11] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397-3415, Dec. 1993.

[12] T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Trans. Signal Processing*, vol. 51, no. 3, pp. 639-652, Mar. 2003.

[13] H.-C. Shin and A. H. Sayed, "Transient behavior of affine projection algorithms," in *Proc. Int. Conf. Acoustic. Speech. Signal Proc.*, vol. 4, pp. 353-356, Hong Kong, Apr. 2003.

[14] H.-C. Shin and A. H. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans. Signal Processing*, vol. 52, pp. 90-102, Jan. 2004.

[15] H.-C. Shin, W. J. Song, and A. H. Sayed, "Mean-square performance of data-reusing adaptive algorithms," *IEEE Signal Processing Letters*, vol. 12, pp. 851-854, Dec. 2005.

[16] J. H. Husøy and M. S. E. Abadi, "Transient analysis of adaptive filters using a general framework," *Automatika, Journal for control, Measurement, Electronics, Computing and Communications*, vol. 45, pp. 121-127, 2004.

[17] S. Haykin, *Adaptive Filter Theory*, Fourth ed. Upper Saddle River, NJ. USA: Prentice Hall, 2002.

[18] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, 2003.

[19] J. R. Treichler, C. R. Johnson, and M. G. Larimore, *Theory and Design of Adaptive Filters. Upper Saddle River*, NJ: Prentice Hall, 2001.

[20] S. F. Cotter, J. Adler, B. D. Rao, and K. Kreutz-Delgado, "Forward sequential algorithms for best basis selection," *IEE Proc. Vis. Image, Signal Processing*, vol. 146, no. 5, pp. 235-244, Oct. 1999.

[21] Y. Saad, Iterative *Methods for Sparse Linear Systems*, PWS Publishing, 1996.

M. Shams Esfand Abadi was was born in Tehran, Iran, on September 18, 1978. He received the B.S. degree in electrical engineering from Mazandaran University, Mazandaran, Iran and the M.S. degree in electrical engineering from Tarbiat Modarres University, Tehran, Iran in 2000 and 2002 respectively, and the Ph.D. degree in biomedical engineering from Tarbiat Modarres University, Tehran, Iran in 2007. Since 2004 he has been with the department of electrical engineering, Shahid Rajaee Teacher Training University, Tehran, Iran. During the fall of 2003, spring 2005, and again in the spring of 2007, he was a visiting scholar with the Signal Processing Group at the University of Stavanger, Norway. His research interests include digital filter theory and adaptive signal processing algorithms.

John H. Husøy received the M.Sc. and Ph.D. degrees in electrical engineering from the Norwegian University of Science and Technology. In his early career he was involved in hardware and software development in various positions in several companies in Canada and Norway. Since 1992 he has been a Professor with the Department of Electrical and Computer Engineering, University of Stavanger, Norway. His research interests include adaptive algorithms, digital filtering, signal representations, image compression, bioelectrical signal processing, and image analysis.

A. Mahlooji Far was born in Qom, Iran, on July 1, 1965. He received the B.S. degree in electrical engineering from Tehran University, Tehran, Iran and the M.S. degree in electrical engineering from Sharif University, Tehran, Iran in 1988 and 199, respectively, and the Ph.D. degree in biomedical engineering from the University of UMIST, Manchester, UK. Since 1997 he has been an Associate Professor with the department of electrical engineering, Tarbiat Modarres University, Tehran, Iran. His research interests include digital signal processing, medical imaging and image analysis.