NORMALIZATION AND CONVERGENCE OF ADAPTIVE IIR FILTERS

Markus Rupp
Institut für Netzwerk und Signaltheorie, TH Darmstadt, 6100 Darmstadt, FRG

ABSTRACT

In the last years several algorithms for adaptive IIR filters have been proposed. However, their practical usage involves considerations such as finding the global minimum, possible occurence of instability and uncertainty about the speed of convergence. This paper presents a generalization of these adaptive IIR algorithms. The algorithms can be classified into two groups: those which do not and those which do filter the adaptation error. For the first group a normalization is presented and convergence is proven. For the second group ideas for normalizations are presented and conditions for convergence are given. Special constraints for normalizations close the paper.

1. INTRODUCTION

Since the derivation of Feintuch's algorithm (RLMS) [3] there have been several suggestions for improving the behavior of adaptive IIR filters. Some ideas like Stearns' algorithm [10] and alternate filtering mode (AFM) [2] have been dropped. Others like series-parallel-filtering (SPLMS) [1], equation error formulation (EEFLMS) [9], bias remedy LMS (BRLMS) [6] and simplified hyperstable adaptive recursive filter (SHARF) [5] remain as candidates for further research. However, all of these algorithms are rather difficult to handle when dealing with applications involving speech signals.

Figure 1 depicts the situation of echo cancellation on a hybrid. The hybrid G has to decouple the near and far end speech. The near end speaker signal $u(k)$ has to be transmitted to the subscriber side whereas the far end speech signal $n(k)$ has to be transmitted to the near end loudspeaker. Since the hybrid is not an ideal device, an echo of the near end speech appears at the loudspeaker. This echo has to be estimated by a system identification of the hybrid. As illustrated in Figure 1,

$$e_o(k) = d(k) - \hat{y}(k) \tag{1.1a}$$

$$d(k) = n(k) + y(k) \tag{1.1b}$$

$$y(k) = \sum_{i=1}^{M_a} a_i\, y(k-i) + \sum_{j=0}^{M_b-1} b_j\, u(k-j)\,. \tag{1.1c}$$

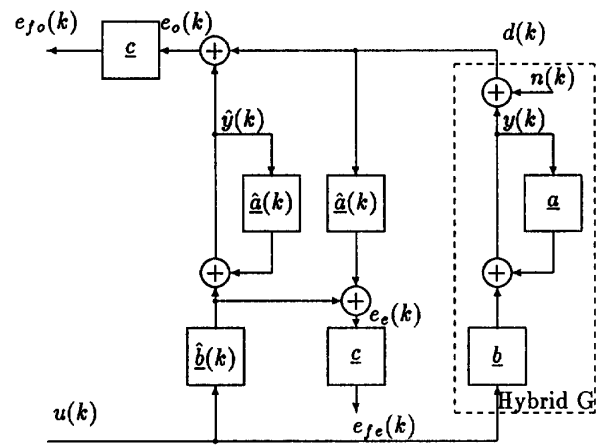

Figure 1: Adaptive filter structure for echo cancelling.

For a clearer description a vector notation is often used:

$$\underline{a}^T = [a_1, a_2, ..., a_{M_a}], \tag{1.2a}$$

$$\underline{b}^T = [b_0, b_1, ..., b_{M_b-1}], \tag{1.2b}$$

$$\underline{w}^T = [\underline{a}^T, \underline{b}^T], \tag{1.2c}$$

$$\underline{y}^T(k) = [y(k-1), y(k-2), ..., y(k-M_a)], \tag{1.2d}$$

$$\underline{u}^T(k) = [u(k), u(k-1), ..., u(k-M_b+1)], \tag{1.2e}$$

$$\underline{z}^T(k) = [\underline{y}^T(k), \underline{u}^T(k)]\,. \tag{1.2f}$$

The parameters $\underline{b}$ and $\underline{a}$ have been combined to form a new vector $\underline{w}$ of order $M = M_a + M_b$. Now the hybrid can easily be described as $d(k) = n(k) + \underline{z}^T(k)\underline{w}$. The problem consists of finding the parameters $\hat{a}_i$ for $i = 1..M_a$ and $\hat{b}_j$ for $j = 0..M_b - 1$ while observing the input signal $u(k)$ and the output signal $d(k)$ of the hybrid. In a very similar way the echo canceller can be described by an estimated parameter set:

$$\hat{\underline{a}}^T(k) = [\hat{a}_1(k), \hat{a}_2(k), ..., \hat{a}_{M_a}(k)], \tag{1.3a}$$

$$\hat{\underline{b}}^T(k) = [\hat{b}_0(k), \hat{b}_1(k), ..., \hat{b}_{M_b-1}(k)], \tag{1.3b}$$

$$\hat{\underline{w}}^T(k) = [\hat{\underline{a}}^T(k), \hat{\underline{b}}^T(k)]\,. \tag{1.3c}$$

The signal vector for calculating the estimated hybrid

0-7803-0946-4/93 $3.00

	gradient $\underline{\psi}(k)$					
e_a	z_o	z_e	z_c	z_{fo}	z_{fe}	z_{fc}
e_o	RLMS			Stearns'	AFM	
e_e		SPLMS	BRLMS			
e_c						
e_{fo}	SHARF			St.-McBr.		
e_{fe}						
e_{fc}						

Table 1: Possibilities for gradient-based algorithms.

output $\hat{y}(k)$ can be constructed by either using former estimated values $\hat{y}(k-m)$ for $m = 1..M_a$ or observed values $d(k)$:

$$\underline{\hat{y}}^T(k) = [\hat{y}(k-1), \hat{y}(k-2), ..., \hat{y}(k-M_a)], \quad (1.4a)$$

$$\underline{d}^T(k) = [d(k-1), d(k-2), ..., d(k-M_a)], \quad (1.4b)$$

$$\underline{z}_o^T(k) = [\underline{\hat{y}}^T(k), \underline{u}^T(k)], \quad (1.4c)$$

$$\underline{z}_e^T(k) = [\underline{d}^T(k), \underline{u}^T(k)]. \quad (1.4d)$$

Depending on the choice of the canceller's signal vector, an output error (subscript 'o') or an equation error (subscript 'e') method is used. Now, the formula for a gradient-based algorithm can be given:
$\underline{\hat{w}}(k+1) = \underline{\hat{w}}(k) + \mu(k)\, e_a(k)\, \underline{\psi}(k)$.
Here, an adaptation error $e_a(k)$ and a gradient term $\underline{\psi}^T(k) = [\underline{\psi}_a^T(k), \underline{\psi}_b^T(k)]$ are used. The algorithms can be described according to the choice of adaptation error and gradient used. A summary of the possibilities is given in Table 1. The subscript 'c' is used for a correction term, $\underline{z}_c(k) = \tau(k)\underline{z}_o(k) + (1-\tau(k))\underline{z}_e(k)$, ($\tau(k) \in [0,1]$) which is necessary for incorporating the BRLMS algorithm within this same description scheme. Although the Steiglitz-McBride [11] algorithm is usually of Newton type, a gradient algorithm is possible as well. The adaptation error $e_a(k)$ can be:
1) the output error, $e_o(k) = d(k) - \underline{\hat{w}}^T(k)\underline{z}_o(k)$,
2) the equation error, $e_e(k) = d(k) - \underline{\hat{w}}^T(k)\underline{z}_e(k)$,
3) a corrected error, $e_c(k) = d(k) - \underline{\hat{w}}^T(k)\underline{z}_c(k)$,
4) a filtered output error, $e_{fo}(k) = e_o(k) + \sum_{i=1}^{P} c_i\, e_o(k-i)$,
5) a filtered equation error, $e_{fe}(k) = e_e(k) + \sum_{i=1}^{P} c_i\, e_e(k-i)$,
6) a filtered corrected error, $e_{fc}(k) = e_c(k) + \sum_{i=0}^{P} c_i\, e_c(k-i)$.
In a very similar way the gradient term $\underline{\psi}(k)$ can be the vector with the estimated output signal $\underline{z}_o(k)$, the vector with the measured output signal $\underline{z}_e(k)$, $\underline{z}_c(k)$, a linear combination of $\underline{z}_o(k)$ and $\underline{z}_e(k)$, or a filtered version of all these three. Thus, 36 different algorithms are possible and these can be divided into two groups. The first group, in the first three rows, does not filter the adaptation error. The remaining algorithms in the second group all use a filtered adaptation error. Of the 36 different algorithms, only six (labelled in Table 1) have been thoroughly investigated.

2. NORMALIZATION FOR ALGORITHMS WITHOUT FILTERED ADAPTATION ERROR

Normalization used in the LMS algorithm has demonstrated superior properties [8]: the algorithm behaves independent of the input signal level, the convergence is independent of the input signals PDF and the convergence speed is increased. In the past analyses have fallen under two broad categories for the input sequences: one deterministic, the other stochastic. In this paper a combination of both leads to new results. The adaptation rule can be given as follows

$$\underline{\hat{w}}(k+1) = \underline{\hat{w}}(k) + \mu(k) e_a(k) \underline{z}_1(k) \quad (2.1a)$$

$$e_a(k) = d(k) - \underline{z}_2^T(k)\underline{\hat{w}}(k). \quad (2.1b)$$

Here, two different vectors $\underline{z}_1(k)$ and $\underline{z}_2(k)$ are used, either of which can be one of the six vectors decribed in the last section. A minimal error $e_m(k)$ is introduced $e_m(k) = d(k) - \underline{z}_2^T(k)\underline{w}$.
It is the lowest error when using the optimal solution $\underline{w}$ under the condition that $\underline{z}_2(k)$ is used instead of $\underline{z}_1(k)$. It is now possible to describe the weight-error vector $\underline{\epsilon}(k) = \underline{w} - \underline{\hat{w}}(k)$ as an inhomogeneous system of first order:

$$\underline{\epsilon}(k+1) = \left(\underline{I} - \mu(k)\underline{z}_1(k)\underline{z}_2^T(k)\right)\underline{\epsilon}(k) + \mu(k)\, e_m(k)\, \underline{z}_1(k). \quad (2.2)$$

A first hint for a normalization is revealed,
Normalization 1:

$$\mu(k) = \frac{\alpha}{\underline{z}_1^T(k)\underline{z}_2(k)}. \quad (2.3)$$

Obviously, the smaller the angle between the two vectors $\underline{z}_1(k)$ and $\underline{z}_2(k)$ is, the smaller the stepsize $\mu(k)$ will be. Examining only the homogeneous part of Eq. 2.2 and using Normalization 1 leads to:

$$\underline{\epsilon}(k+1) = \left(\underline{I} - \alpha\frac{\underline{z}_1(k)\underline{z}_2^T(k)}{\underline{z}_1^T(k)\underline{z}_2(k)}\right)\underline{\epsilon}(k). \quad (2.4)$$

One eigenvalue of the homogeneous equation equals $(1-\alpha)$, whereas the remaining $(M-1)$ eigenvalues equal one. Thus, a convergent algorithm is obtained for the normalized stepsize $\alpha \in [0,2]$.
Although the eigenvalues remain bounded, the inhomogeneous part may exceed all limits causing an increase in the weight-error vector as well. The squared L_2 norm of the perturbation in Eq. 2.2 with Normalization 1 is considered:

$$\alpha^2 e_m^2(k)\,\frac{\underline{z}_1^T(k)\underline{z}_1(k)}{\left(\underline{z}_1^T(k)\underline{z}_2(k)\right)^2} \geq \alpha^2 e_m^2(k)\,\frac{1}{\underline{z}_2^T(k)\underline{z}_2(k)}\,. \quad (2.5)$$

Therefore, the inhomogeneous part can exceed every limit which can cause instability of the algorithm.

To find a better norm a stochastic description is now used. Calculating the expectation of the quadratic L_2 norm of $\underline{\epsilon}(k)$ and diagonalization of the inner matrix: $\underline{I} - \mu(\underline{z}_1(k)\underline{z}_2^T(k) + \underline{z}_2(k)\underline{z}_1^T(k)) + \mu^2\underline{z}_2(k)\underline{z}_2^T(k)\underline{z}_1^T(k)\underline{z}_1(k)$ leads to two eigenvalues unequal to zero. Since there are $(M-2)$ vectors that are orthogonal to $\underline{z}_1(k)$ and $\underline{z}_2(k)$, the same number of eigenvalues equals one. The remaining two eigenvalues can be found with an eigenvector that is the linear combination of both $\underline{z}_1(k)$ and $\underline{z}_2(k)$: $\lambda_{1,2}(k) = \left\{1 - 2\mu\,\underline{z}_2^T(k)\underline{z}_1(k) + \mu^2\,\underline{z}_1^T(k)\underline{z}_1(k)\underline{z}_2^T(k)\underline{z}_2(k), 1\right\}$. Only one eigenvalue is unequal to one and is responsible for the convergence behavior. A normalization should ensure that this eigenvalue lies in the desired interval. It can be seen that the eigenvalue describes a parabola in the stepsize $\mu(k)$. Therefore, there exists a minimum value $\mu_{\min}(k)$ and a limit $\mu_{\lim}(k)$ corresponding to the minimal eigenvalue $\lambda_{\min}(k)$ and $\max(\lambda(k)) = 1$, respectively, for $\mu(k)$:

$\mu_{\min}(k) = \dfrac{\underline{z}_1^T(k)\underline{z}_2(k)}{\underline{z}_1^T(k)\underline{z}_1(k)\underline{z}_2^T(k)\underline{z}_2(k)} = \dfrac{\mu_{\lim}(k)}{2}$. This result leads to

Normalization 2:

$$\mu(k) = \frac{\alpha\,\underline{z}_1^T(k)\underline{z}_2(k)}{\underline{z}_1^T(k)\underline{z}_1(k)\underline{z}_2^T(k)\underline{z}_2(k)}\,. \quad (2.6)$$

Convergence is again guaranteed for $0 < \alpha < 2$. With the help of Schwarz' inequality it can be shown that Normalization 2 is always lower than or equal to Normalization 1 and thereby also fulfills the homogeneous system of Eq. 2.4. The squared L_2 norm of the inhomogeneous part is considered again

$$e_m^2(k)\frac{\left(\underline{z}_1^T(k)\underline{z}_2(k)\right)^2}{\underline{z}_1^T(k)\underline{z}_1(k)\left(\underline{z}_2^T(k)\underline{z}_2(k)\right)^2} \leq e_m^2(k)\frac{1}{\underline{z}_2^T(k)\underline{z}_2(k)}\,.$$

The various normalization terms for the known algorithms are listed in Table 2. Although Normalization 2 shows several advantages over Normalization 1, drawbacks exist as well. The algorithm may converge to a fixed point in the parameter space where the algorithm remains stable, whereas the filter is instable.

3. NORMALIZATION FOR ALGORITHMS WITH FILTERED ADAPTATION ERROR

For this group of algorithms the adaptation error $e_a(k)$ must be replaced by the filtered error $e_f(k)$, that is a linear combination of past $e_a(k)$ values, $e_f(k) = C[e_a(k)] = e_a(k) + \sum_{i=1}^{P} c_i e_a(k-i)$. Describing the situation with the weight-error vector as in the previous section and using Eq. 2.1b, a vector differential

Algorithm	Norm 1	Norm 2
RLMS	$\dfrac{1}{\underline{z}_o^T(k)\underline{z}_o(k)}$	$\dfrac{1}{\underline{z}_o^T(k)\underline{z}_o(k)}$
Stearns	$\dfrac{1}{\underline{z}_{fo}^T(k)\underline{z}_o(k)}$	$\dfrac{\underline{z}_{fo}^T(k)\underline{z}_o(k)}{\underline{z}_o^T(k)\underline{z}_o(k)\underline{z}_{fo}^T(k)\underline{z}_{fo}(k)}$
AFM	$\dfrac{1}{\underline{z}_{fe}^T(k)\underline{z}_o(k)}$	$\dfrac{\underline{z}_{fe}^T(k)\underline{z}_o(k)}{\underline{z}_o^T(k)\underline{z}_o(k)\underline{z}_{fe}^T(k)\underline{z}_{fe}(k)}$
SP-LMS	$\dfrac{1}{\underline{z}_e^T(k)\underline{z}_e(k)}$	$\dfrac{1}{\underline{z}_e^T(k)\underline{z}_e(k)}$
BRLMS	$\dfrac{1}{\underline{z}_c^T(k)\underline{z}_e(k)}$	$\dfrac{\underline{z}_c^T(k)\underline{z}_e(k)}{\underline{z}_e^T(k)\underline{z}_e(k)\underline{z}_c^T(k)\underline{z}_c(k)}$

Table 2: Possible normalizations for the well-known algorithms

system of order P is obtained:

$\underline{\epsilon}(k+1) = \underline{\epsilon}(k) + \mu\underline{z}_1(k)(e_m(k) - \underline{\epsilon}^T(k)\underline{z}_2(k)) + \mu\underline{z}_1(k)\sum_{i=1}^{P} c_i(e_m(k-i) - \underline{\epsilon}^T(k-i)\underline{z}_2(k-i))$.

In the next step the homogeneous solution of the case $(P = 1)$ for a system of order one is investigated:

$$\begin{bmatrix} \underline{\epsilon}(k+1) \\ \underline{\epsilon}(k) \end{bmatrix} = \begin{bmatrix} \underline{I} - \mu\underline{z}_1(k)\underline{z}_2^T(k) & -c_1\mu\underline{z}_1(k)\underline{z}_2^T(k-1) \\ \underline{I} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\epsilon}(k) \\ \underline{\epsilon}(k-1) \end{bmatrix}.$$

Using Normalization 1 the only eigenvalues unequal to one or zero are:

$$\lambda_{1,2} = \frac{1-\alpha}{2} \pm \sqrt{\left(\frac{1-\alpha}{2}\right)^2 - c_1\alpha\frac{\underline{z}_2^T(k-1)\underline{z}_1(k)}{\underline{z}_2^T(k)\underline{z}_1(k)}}\,. \quad (3.1)$$

The term $\frac{\underline{z}_2^T(k-1)\underline{z}_1(k)}{\underline{z}_2^T(k)\underline{z}_1(k)}$ can be assumed to lie mainly between zero and one depending on the correlation of the input process. If c_1 is sufficiently small, $\alpha \in [0, 2]$ guarantees convergence again. If c_1 is a high positive value the eigenvalues become complex. Their real part, however, remains in the stable region, and therefore, a stable oscillation occurs. To ensure the convergence for the inhomogeneous case as well, the choice of Normalization 2 instead of Normalization 1 is again necessary. If both vectors $\underline{z}_1(k)$ and $\underline{z}_2(k)$ are equal to $\underline{z}_o(k)$ the SHARF algorithm is obtained. For the special case $(c_1 = 0)$ the SHARF algorithm simplifies to Feintuch's algorithm. The effect of a filter constant $c_1 > 0$ is to decrease the real part of the eigenvalue and thus cause a higher convergence speed. Since an increasing imaginary part has no effect on the convergence of the algorithm, the choice of c_1 is not critical.

The case of higher system orders $P > 1$ can be handled in a similar way. To find the eigenvalues that are responsible for convergence, the following polynomial equation has to be solved $(c_0 = 1)$:

$\lambda^{P+1} - \lambda^P + \sum_{i=0}^{P} \lambda^{P-i} c_i \mu(k) \underline{z}_2^T(k-i) \underline{z}_1(k) = 0$.
It is to be expected that for larger delays i the expression $\underline{z}_2^T(k-i)\underline{z}_1(k)$ will become increasingly smaller, and therefore, the parts with a larger index lose importance and the first order system remains.

4. MEASUREMENT RESULTS

The most important algorithms in Table 1 have been implemented on a DSP56001. Measurements have taken place in a hybrid environment depicted in Figure 1 with $M_a = 3$ and $M_b = 9$ in which the echoes originate from the local speaker. The input signal on the local speaker side was a white Gaussian noise sequence. Table 3 shows the results. The echo return loss enhancement (ERLE) has been measured after 166ms and after the steady-state error had been reached. The column labelled "Afterburst" describes the ERLE after a burst from the subscriber side. The AFM algorithm with the normalization from [2] and with the new norm showed undesirable behavior. The convergence speed of the prefiltering algorithms (i.e. AFM, Stearns) is low and the reaction from the burst is slow. Even the steady-state error is bigger than those of the other algorithms. All other tested algorithms behaved very well, especially NSHARF and NRLMS showed good steady-state values. A 32-tap transversal filter with NLMS algorithm is given as a comparison.

Algo.	α	$ERLE_{166}$	$ERLE_{\infty}$	Afterburst
AFM	2^{-7}	10dB	17dB	17dB after 500ms
NAFM	0.25	4dB	13dB	13dB after 500ms
NStearns	1	11dB	15dB	18dB direct
NRLMS	1	22dB	23dB	23dB direct
NSHARF	1	22dB	23dB	23dB direct
NLMS	0.5	31dB	31dB	31dB direct

Table 3: Measured results with Normalization Rule 2

5. OTHER POSSIBILITIES FOR NORMALIZATION

In this section alternative normalizations are considered.

1) Splitting the gradient vector in its original components, (i.e. $\underline{u}(n), \underline{d}(n)$) allows to build an algorithm independent of the system gain q (i.e. $d(k) = n(k) + q\, y(k)$).

$$\underline{\hat{a}}(k+1) = \underline{\hat{a}}(k) + \frac{\alpha_a}{\underline{\psi}_a^T(k)\underline{\psi}_a(k)q^2}\, q e_a(k)\ \ \underline{\psi}_a(k) q,$$

$$q\underline{\hat{b}}(k+1) = q\underline{\hat{b}}(k) + \frac{\alpha_b}{\underline{\psi}_b^T(k)\underline{\psi}_b(k)}\, q e_a(k)\ \ \underline{\psi}_b(k).$$

With the Schwarz' inequality it is obvious that convergence is guaranteed for:

$$0 < \alpha_a + \alpha_b < 2.$$

A rule for optimal values of α_a and α_b has not yet been determined. Moreover, other rules are possible:

2) The concept of using several different stepsizes can be generalized. In the extreme case every one of the M parameters can obtain its own stepsize $\mu_i(k); i = 1..M$. The rule for Normalization 2 then reduces to:

$$\mu_i(k) = \alpha_i \frac{\underline{z}_{2i}^T(k)\underline{z}_{1i}(k)}{\underline{z}_{1i}^T(k)\underline{z}_{1i}(k)\underline{z}_{2i}^T(k)\underline{z}_{2i}(k)},$$

where the subscript i determines the i-th. component of the vectors. Again, a convergence bound is obtained:

$$0 < \sum_{i=1}^{M} \alpha_i < 2.$$

3) A squared L_1 norm can be useful in speech applications, since the typically low amplitudes of a speech signal are not markedly further lowered. Since $L_1 > L_2$, convergence is guaranteed as well.

References

[1] Maurice G. Bellanger, "Adaptive Digital Filters and Signal Analysis," Marcel Dekker, N. Y., 1986.

[2] Hong Fan, W. Kenneth Jenkins, "An Investigation of an Adaptive IIR Echo Canceller: Advantages and Problems," IEEE Trans. Acoust., Speech, Sig. Proc., vol. ASSP-36, no. 12, pp. 1819-1833, Dec. 1988.

[3] P. L. Feintuch, "An Adaptive Recursive LMS Filter," Proc. IEEE, vol. 64, no. 11, pp. 1622-1624, Nov. 1976.

[4] Sharlene Gee, Markus Rupp, "A Comparison of Adaptive IIR Echo Cancellers for Hybrids," Proc. ICASSP, Toronto, pp. 1541-1544, Mai 1991.

[5] Michael G. Larimore , John R. Treichler, C. Richard Johnson, "SHARF: An Algorithm for Adapting IIR Digital Filters," IEEE Trans. Acoust., Speech, Sig. Proc., vol. ASSP-28, no. 4, pp. 428-440, Aug. 1980.

[6] Ji-Nan Lin, Rolf Unbehauen, "Bias-Remedy Least Mean Square Equation Error Algorithm for IIR System Parameter Recursive Estimation," IEEE Trans. SP, vol. 40, no. 1, pp. 62-69, Jan. 1992.

[7] Markus Rupp, "Adaptive IIR Echo Cancellers for Hybrids using the Motorola 56001," in Proc. Eusipco Signal Processing V, Barcelona, pp. 1487-1490, 1990.

[8] Markus Rupp, "The Behavior of LMS and NLMS Alg. in the Presence of Spherically Invariant Processes," IEEE Trans. Sig. Proc., vol. 41, no. 3, Mar. 1993.

[9] John J. Shynk, "Adaptive IIR Filtering," IEEE ASSP Magazine, vol. 6, no. 2, Apr. 1989.

[10] Samuel D. Stearns, G. R. Elliot, "On Adaptive Recursive Filtering," Proc. 10th Asilomar Conf. on Circuits, Systems, and Computers, pp. 5-11, Nov. 1976.

[11] K. Steiglitz, L. E. McBride, "A Technique for the Identification of linear Systems," IEEE Trans. Autom. Control, vol. AC-10, pp. 461-464, 1965.