Abstract: This paper introduces a dynamically regularized fast recursive least squares (DR-FRLS) adaptive filtering algorithm. Numerically stabilized FRLS algorithms exhibit reliable and fast convergence with low complexity even when the excitation signal is highly self-correlated. FRLS still suffers from instability, however, when the condition number of the implicit excitation sample covariance matrix is very high. DR-FLRS, overcomes this problem with a regularization process which only increases the computational complexity by 50%. The benefits of regularization include: 1) the ability to use small forgetting factors resulting in improved tracking ability and 2) better convergence over the standard regularization technique of noise injection. Also, DR-FLRS allows the degree of regularization to be modified quickly without restarting the algorithm.

1. Introduction

One of the more promising classes of adaptive filtering algorithms for acoustic echo cancellation is fast recursive least squares (FRLS). These exhibit reasonably low computational complexity combined with fast convergence even when the excitation signal is highly colored. Versions with improved numerical stability have appeared in the past few years[3][4] yet, even these suffer from instability which occurs when the excitation signal’s sample covariance matrix, \( R_n \), is poorly conditioned. This situation may arise from the excitation signal’s actual statistics (it may be highly self-correlated) or from the use of an insufficiently long data window in \( R_n \)’s estimation. Since RLS and FRLS rely on the explicit or implicit inversion (respectively) of \( R_n \), both become unduly susceptible to system measurement noise and/or numerical errors from finite precision arithmetic when \( R_n \) has some small eigenvalues.

Regularization is a common technique used in least squares methods whereby a matrix such as \( \delta I_N \) is added to \( R_n \) prior to inversion. Here, \( \delta \) is a small positive number and \( I_N \) is the N dimensional identity matrix. This establishes \( \delta \) the lower bound for the minimum eigenvalue of the resulting matrix, stabilizing the solution (if \( \delta \) is big enough) at the price of biasing the least squares solution slightly.

Here, we introduce a technique for regularizing the \( R_n \) inverse in such a way that the \( O(N) \) complexity of the FRLS algorithms may be retained. Moreover, the degree of regularization (the size of \( \delta \)) may be changed in real-time without restarting the adaptive filter, resulting in a dynamically regularized FRLS (DR-FRLS) adaptive filtering algorithm.

2. Regularization Refresh

RLS and FRLS are efficient ways of implementing the following algorithm,

\[
\begin{align*}
\epsilon_{0,n} &= d_n - x_n^T h_{n-1} \\
h_n &= h_{n-1} + R_n^{-1} x_n \epsilon_{0,n}
\end{align*}
\]

In acoustic echo cancellation parlance, the scalars, vectors, and matrix of (1) and (2) are defined as follows:

- \( n \) is the sample index.
- \( d_n \) is the desired signal. It consists of both the room response (the echo) and any other background acoustic signal, \( v_n \), picked up by the microphone.
- \( x_n \) is the excitation signal and is assumed to equal 0 for \( n < 0 \).
- \( \tilde{x}_n \) is the N-length excitation vector,
\[ \tilde{x}_n = [x_n, x_{n-1}, ..., x_{N-1}]^T. \]
- \( h_n \) is the N-length adaptive filter coefficient vector,
\[ h_n = [h_{1,n}, h_{2,n}, ..., h_{N,n}]^T. \]
- \( \epsilon_{0,n} \) is the a priori error, or residual echo.
- \( R_n \) is the N-by-N sample covariance matrix of \( x_n \).

Various windows can be applied to the data used to estimate \( R_n \). The exponential window is popular since it allows rank one updating from sample period to sample period. Specifically,

\[
R_n = \lambda^{n+1} \delta_0 D_\lambda + \sum_{i=0}^{n} \lambda^i \delta_{n-i} \tilde{x}_n^T
\]

where, \( \lambda \) is the forgetting factor selected within the range, \( 0 < \lambda < 1 \). Also, we have used,

\[
D_\lambda = \text{diag} \{ \lambda^{N-1}, \lambda^{N-2}, ..., \lambda, 1 \}
\]

and

\[
R_{-1} = \delta_0 D_\lambda.
\]

Exploitation of the rank-one update of \( R_n \) of equation (6) led to RLS from classical least squares methods and the exploitation of the shift invariant nature of \( \tilde{x}_n \) led to FRLS from RLS[1][2].

With \( R_n \) as defined in (5) through (8), \( \delta_0 D_\lambda \) serves to initially regularize the inverse in (2), but according to the first term in (5) its effect diminishes with time. Adding an appropriately scaled version of \( \delta I_N \) to \( R_n \) prior to inversion each sample period would indeed regularize the least squares solution, but that would require an additional rank \( N \) update each sample period, eliminating the computational benefit of the rank-one update in (6). An alternative, is to add an approximation to \( \delta I_N \), \( D_\lambda \) which itself is updated each sample period with a rank-one update matrix constructed from the outer product of a shift invariant vector. Then, (2) can be modified to
\[ h_n = h_{n-1} + R^{-1}\lambda_n e_{0,n} \]  

(9)

where

\[ R_{t,n} = D_n + R_n. \]  

(10)

With both, \( D_n \) and \( R_n \) being maintained by rank-one updates, \( R_{t,n} \) requires a rank-two update. This will increase the computational complexity somewhat over those algorithms using only \( R_n \), but with the benefit of regularization.

It is desirable that \( D_n \) be constructed recursively, using the outer product of a vector composed of a shift invariant signal such that the eigenvalues of \( D_n \) are updated, or refreshed as often as possible. Accordingly, let us define the vector

\[ p_n = [0, 0, ..., 0, 1, 0, ..., 0]^T \]  

(11)

where all elements in the vector are zero except for a one in position \( 1 + \lfloor n \rfloor_{mod N} \). We can then define the regularization update to be

\[ D_n = \sum_{i=0}^{n} \lambda_i \phi_i \xi_n^2 p_{n-i} p_{n-i}^T \]  

(12)

\[ = \lambda D_{n-1} + \phi_n \xi_n^2 p_n p_n^T \]  

(13)

with \( D_1 = 0 \) as the initial condition. In (12) and (13) \( \phi_n \) takes on values of \( \pm 1 \). Furthermore, we restrict the sample periods that \( \phi_n \) and \( \xi_n \) may change values to those where the \( p_n \) vector has its only non-zero value in the first position. This guarantees that \( \phi_n \xi_n p_n \) is shift invariant, a necessary property for the derivation of fast algorithms. Signals \( \phi_n \) and \( \xi_n \) control the size of the regularization in the adaptive filter. Signal \( \phi_n \) determines whether \( \xi_n \) will slightly inflate or deflate the regularization matrix.

If \( \phi_n \) and \( \xi_n \) are fixed, then the \( i^{th} \) diagonal element of \( D_n \) will reach a steady state of

\[ d_{i,n} = \phi_n \xi_n^2 \frac{\lambda^{n-i+1}_{mod N}}{1-\lambda} \]  

(14)

Equation (14) shows that the regularization provided by the \( i^{th} \) diagonal element of \( D_n \) varies periodically due to the periodic nature of the regularization update. The shape of the variation is the familiar "saw-tooth" pattern and the degree of variance is from \( \lambda^{-1} \) to 1. The parameter \( \lambda \) must be chosen close enough to one to obtain a reasonably steady amount of regularization.

Another obvious tact, would be to update \( D_n \) with a rank two update with the second dyadic product producing a regularization update 180 degrees out of phase with the first. This would decrease the variation in the \( d_{i,n} \)'s at the price of additional computational complexity. This could be useful in situations where extremely small forgetting factors are used.

3. Dynamically Regularized FTF

We now turn to the derivation of a dynamically regularized fast transversal filter (FTF) algorithm (an FRLS algorithm) which also incorporates numerical stabilization. The rank-two update of \( R_{t,n} \) can alternately be viewed as two rank one updates, where an intermediate regularization refreshed covariance matrix \( R_{t,n} \) is defined as,

\[ R_{t,n} = \lambda R_{t,n-1} + \phi_n \xi_n^2 p_n p_n^T \]  

(15)

and then, the data updated covariance matrix is

\[ R_{t,n} = R_{p,n} + \xi_n^2 p_n p_n^T. \]  

(16)

Using (16) and the matrix inversion lemma, which states that

\[ (A + BC)^{-1} = A^{-1} - A^{-1}B(1 + CA^{-1}B)^{-1}CA^{-1} \]  

(17)

we can express the data a posteriori kalman gain vector, as

\[ k_{1,n,t} = \frac{R^{-1}_{p,n} \xi_n}{(R_{p,n}^2 - R_{p,n}^2 \xi_n^2 (1 + 2 \xi_n R_{p,n}^2 \xi_n)^{-1} \xi_n^2 R_{p,n}^2 \xi_n).} \]  

(18)

Now, define the data a priori kalman gain vector as

\[ \hat{k}_{0,n,t} = \frac{R^{-1}_{p,n} \xi_n}{(1 + \xi_n R_{p,n}^2 \xi_n)^{-1}}. \]  

(19)

and the data likelihood variable as

\[ \mu_{n,t} = (1 + \hat{k}_{0,n,t}^2 + \xi_n R_{p,n} \xi_n). \]  

(20)

Then, after some manipulation, (18) becomes,

\[ k_{1,n,t} = \hat{k}_{0,n,t} \xi_n. \]  

(21)

Using (21) in (9) the coefficient update becomes,

\[ h_n = h_{n-1} + \hat{k}_{1,n,t} \xi_n e_{0,n}. \]  

(22)

The remaining derivations all focus on the stable computation of \( \hat{k}_{0,n,t} \) and \( \mu_{n,t} \) from sample period to sample period. As we shall see, the computation of the data kalman gain vectors and data likelihood variables (as is common, we will define two of each) are predicated on the existence of those of the previous sample period together with the current sample period’s regularization prediction vectors and regularization prediction error energies associated with the regularization refresh update covariance matrix, \( R_{p,n} \). These regularization prediction vectors and prediction error energies are calculated from their predecessors and the regularization kalman gain vectors and regularization likelihood ratios which in turn are calculated from their predecessors and the previous sample period’s data prediction vectors and their prediction error energies, completing the cycle.

We begin by invoking the order update identity for \( R_{p,n}^{-1} \).

\[ R_{p,n}^{-1} = \begin{bmatrix} 0 & 0^T \{R_{p,n}^{-2} \}^T \end{bmatrix} + E_{a_{p,n}^a b_{p,n}^b} \]  

(23)

and

\[ R_{p,n}^{-1} = \begin{bmatrix} R_{p,n}^{-1} & 0 \{0, 0^T \} \end{bmatrix} + E_{b_{p,n}^b} \]  

(24)

Where the following definitions are in order:

- \( \tilde{R}_{p,n} \) is the upper right hand (N-1)-by-(N-1) sub-matrix of \( R_{p,n} \).
- \( a_{p,n}^a \) and \( b_{p,n}^b \) are the optimal forward and backward linear prediction vectors for \( R_{p,n} \) and \( E_{a_{p,n}^a} \) and \( E_{b_{p,n}^b} \) are their corresponding prediction error energies. Together, these form the relationships,

\[ R_{p,n} a_{p,n}^a = E_{a_{p,n}^a} \mu \]  

(25)

and

\[ R_{p,n} b_{p,n}^b = E_{b_{p,n}^b} \nu \]  

(26)

where \( \mu \) and \( \nu \) are the pinning vectors,

\[ \mu = [1, 0, 0, ..., 0]^T \]  

\[ \nu = [0, 0, ..., 0, 1]^T. \]  

(27a,b)

Post multiplying (23) and (24) by \( \xi_n \), it can be shown that
\[ k_{0,1,n} = \begin{bmatrix} 0 \\ k_{0,1,n-1} \end{bmatrix} + E_{e,n}^{-1} e_{0,1,n} a_{p,n} \]  

(28)

and

\[ \bar{k}_{0,1,n} = k_{0,1,n} - E_{b,n}^{-1} e_{0,1,n} b_{p,n} \]  

(29)

where,

\[ \bar{k}_{0,1,n} = \left[ \begin{array}{c} \bar{R}_{n}^{-1} \\ 0 \end{array} \right] \]  

(30)

and the a priori forward and backward data prediction errors are,

\[ e_{0,1,n,a} = \bar{y}_{n}^{T} a_{p,n}, \quad e_{0,1,n,b} = \bar{y}_{n}^{T} b_{p,n}. \]  

(30a,b)

Looking at the first and last elements of the vectors in (28) and (29) shows that

\[ k_{0,1,n} = E_{e,n}^{-1} e_{0,1,n,a}, \quad k_{0,1,n,N} = E_{b,n}^{-1} e_{0,1,n,b}. \]  

(31a,b)

The data likelihood variable can be found by premultiplying (28) and (29) by \( \bar{y}_{n} \) and adding 1 to both sides, then from (20) we see that

\[ \mu_{n}^{-1} = \mu_{n-1}^{-1} + E_{e,n}^{-1} e_{0,1,n,a} \]  

(32)

and

\[ \bar{\mu}_{n}^{-1} = \mu_{n-1}^{-1} + E_{b,n}^{-1} e_{0,1,n,b} \]  

(33)

where

\[ \bar{\mu}_{n} = \left( 1 + \bar{k}_{0,1,n}^{T} \bar{y}_{n} \right)^{-1}. \]  

(34)

Just as we derived (28), (29), (31a) and (31b), from the order update identity of \( R_{n}^{-1} \), we can also derive the following relationships,

\[ k_{1,1,n} = \left[ \begin{array}{c} 0 \\ k_{1,1,n-1} \end{array} \right] + E_{a,n}^{-1} a_{1,1,n} a_{1,1,n} \]  

(35)

and

\[ \bar{k}_{1,1,n} = k_{1,1,n} - E_{b,n}^{-1} e_{1,1,n} b_{1,1,n} \]  

(36)

where,

\[ R_{1,n} a_{1,n} = E_{a,n}^{-1} a_{1,n} u_{n}, \quad R_{1,n} b_{1,n} = E_{b,n}^{-1} b_{1,n} \]  

(37a,b)

\[ \bar{k}_{1,1,n} = \left[ \begin{array}{c} \bar{R}_{n}^{-1} \\ 0 \end{array} \right] \]  

(38)

and the a posteriori forward and backward data prediction errors are,

\[ e_{1,1,n,a} = \bar{y}_{n}^{T} a_{1,1,n}, \quad e_{1,1,n,b} = \bar{y}_{n}^{T} b_{1,1,n}. \]  

(39a,b)

Looking at the first and last elements of the vectors in (35) and (36) shows that

\[ k_{1,1,n} = E_{a,n}^{-1} e_{1,1,n,a}, \quad k_{1,1,n,N} = E_{b,n}^{-1} e_{1,1,n,b}. \]  

(40a,b)

Post multiply (16) by \( a_{p,0} \) and then premultiply by \( R_{1,1}^{-1} \), then using (25), it can be shown that

\[ a_{p,n} = E_{e,n}^{-1} a_{e,n} + k_{1,1,n} e_{0,1,a,n}. \]  

(41)

Observing the first elements of the vectors of (41) and using (31a) and (21) we see that

\[ E_{e,n}^{-1} a_{e,n} + k_{1,1,n} e_{0,1,a,n} = k_{0,1,n}^{-1} E_{e,n}^{-1} E_{a,n}^{-1} k_{0,1,n}^{-1} \mu_{n}. \]  

(42)

Multiplying both sides of (42) by \( E_{e,n}^{-1} \), we can use this result in (41) and then invoke (21), (31a), (40a), and (35) to show that

\[ a_{1,n} = a_{p,n} - [0, k_{1,1,n-1}^{T} e_{0,1,n,a} \]  

(43)

\[ = a_{p,n} - [0, k_{1,1,n-1}^{T} e_{0,1,n,a} \]  

(44)

\[ = a_{p,n} - [0, k_{1,1,n-1}^{T} e_{1,1,n,b} \]  

(45)

where in (44) we have used an easily derived identity, \( k_{1,1,n} = \bar{\mu}_{n} k_{0,1,n} \), and in (45) we have used the fact that

\[ e_{1,1,n,a} = \bar{\mu}_{n-1} e_{0,1,n,a} \]  

(46)

which can be seen by premultiplying (44) by \( \bar{y}^{T} \) and using (34).

Relations for the data backward predictor and data backward prediction error energies can be derived in a manner similar to that which led to equations (41) through (46) with the results:

\[ E_{b,n} = E_{b,n}^{-1} + e_{0,1,n,a} e_{1,1,n,b} \]  

(47)

\[ b_{1,n} = b_{1,n}^{-1} + [k_{1,1,n-1}^{T} e_{1,1,n,b} \]  

(48)

\[ e_{1,1,n,b} = \bar{\mu}_{n-1} e_{0,1,n,b} \]  

(49)

The data related variable updates are summarized in steps 20) through 38) in the DR-FLRS algorithm of Table 2. The regularization related variable definitions and updates that are shown in steps 1) through 19) can be derived in a manner analogous to the data updates. In both of these sections of the algorithm we have used error feedback for stabilization[4] and specifically, in steps 39) and 40), the likelihood variable estimates are stabilized using the multi-channel, multi-experiment method of Stock and Kailath[5]. In steps 19) and 38) the likelihood variables serve as rescue variables. If rescue is necessary, the initialization procedure shown in Table 1 should be executed with the exception of the zeroing of the data vector \( y_{n} \) which serves as the data vector in the joint process extension as shown in steps 41) through 43). Essentially, \( y_{n} \) is simply a copy of \( y_{n} \) except during restarts when \( y_{n} \) is zeroed and \( y_{n} \) is not.

**Table 1: DR-FLRS Initialization**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Initial Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_{1,n} )</td>
<td>( = u )</td>
<td>( b_{1,n} = \delta )</td>
</tr>
<tr>
<td>( E_{b,n}^{-1} )</td>
<td>( = )</td>
<td>( E_{a,n}^{-1} = \bar{\mu}_{n-1} )</td>
</tr>
<tr>
<td>( \mu_{n}^{-1} )</td>
<td>( = 1 )</td>
<td>( \mu_{n}, \bar{\mu}_{n} = 1 )</td>
</tr>
<tr>
<td>( k_{0,1,n}^{-1} )</td>
<td>( = 0 )</td>
<td>( K_{1} = 1.5, K_{2} = 2.5, K_{3} = 0, K_{4} = 0 )</td>
</tr>
</tbody>
</table>

The steps of DR-FLRS which require significant computational complexity, N multiplies, are so noted on the right side of Table 2. The total complexity is 12N multiplies, only 50% more than stabilized FTF[4]. The sparse nature of \( n_{p} \) makes steps 1) and 5) trivial to compute, reducing the overall complexity slightly. We also note that for the purpose of clarity, we have distinguished the forward and backward prediction vectors (and their associated prediction error energies) of the regularization and data related updates. For the purpose of implementation, it is only necessary to have one set of forward and backward predictors and prediction error energies.
Table 2: DR-FRLS

<table>
<thead>
<tr>
<th>Regularization Related Updates:</th>
<th>Multiplies</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) $e_{0,p,n,a} = a_n^T x_n - \xi_p^T \xi_p$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2) $k_{0,p,n-1} = E_{x_n,k_{0,p,n-1}}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3) $\xi_p = [0, \xi_p^T k_0 p, \ldots 1]^T + \xi_{p-1}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4) $\mu_{0,n} = \mu_{0,n} + \phi_n e_{0,p,n} k_{0,p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5) $\bar{e}<em>{0,p,a,b} = b</em>{0,a,b} + \xi_n^T \xi_n$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6) $k_{0,p,n} = \lambda \xi_{p-1} E_{x_n,k_{0,p,n}}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7) extract $k_{0,p,n}$ from $k_{0,p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8) $\varphi_{p,n} = \lambda \xi_{p-1} e_{0,p,n,b} k_{0,p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9) $\varphi_{1,p,n} = K_1^{(0)} e_{0,p,n} + (1 - K_1) e_{0,p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10) $\varphi_{0,p,n,N} = K_{0,p,n,N} + (1 - K_1) k_{0,p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11) $(\xi_{p,n}^T 0)^T = k_{0,p,n} - \xi_{p,n}^T k_{0,p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12) $\mu_{p,n} = \mu_{p,n} - (\varphi_n e_{0,p,n,b} k_{0,p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13) $\mu_{p,n} = (\mu_{p,n} - \mu_{p,n-1} e_{0,p,n})$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14) $a_{p,n} = \varphi_{p,n} - \mu_{p,n} e_{0,p,n,b} k_{0,p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15) $E_{a,n} = \lambda \xi_{p-1} e_{0,p,n} - (1 - K_1)$ $k_{0,p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16) $e_{0,p,n,b}^{(j)} = \mu_{p,n} e_{0,p,n,b}^{(j)}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17) $b_{p,n} = b_{p,n-1} - \varphi_n [k_{0,p,n} 0] \xi_{p,n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18) $E_{b,p,n} = \lambda E_{b,n} + (1 - K_1) \varphi_n e_{0,p,n,b}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19) if $\mu_{p,n} &gt; 1$ restart regularization updates</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data Related Updates

20) $e_{0,0,0,a} = x_n^T a_n$ | | |
21) $k_{0,0,1} = E_{x_n,k_{0,0,1}}$ | | |
22) $k_{0,0,1} = [0, k_{0,0,1}]^T + a_{0,0,1} k_{0,0,1}$ | | |
23) $\mu_{0,1} = \mu_{0,1} + \phi_{0,0,1} k_{0,0,1}$ | | |
24) $e_{0,0,1} = b_{0,0,1} x_n$ | | |
25) $k_{0,0,1,N} = E_{0,0,1}$ | | |
26) extract $k_{0,0,1,N}$ from $k_{0,0,1}$ | | |
27) $e_{0,0,1} = E_{x_n,k_{0,0,1}}$ | | |
28) $e_{0,0,1} = K_0^{(0)} e_{0,0,1} + (1 - K_0) e_{0,0,1}$ | | |
29) $k_{0,0,1,N} = K_0^{(0)} k_{0,0,1,N} + (1 - K_1) k_{0,0,1,N}$ | | |
30) $(\xi_{0,0,1}^T 0)^T = k_{0,0,1} - \xi_{0,0,1}^T k_{0,0,1}$ | | |
31) $\mu_{0,1} = \mu_{0,1} - (\varphi_n e_{0,0,1}$ | | |
32) $e_{0,0,1} = e_{0,0,1} - \mu_{0,1} e_{0,0,1}$ | | |
33) $a_{0,1} = a_{0,1} - (\xi_{0,0,1}^T 0)^T e_{0,0,1}$ | | |
34) $E_{0,0,1} = E_{x_n,k_{0,0,1}}$ | | |
35) $e_{0,0,1} = \xi_{0,0,1} e_{0,0,1}$ | | |
36) $b_{0,1} = \mu_{0,1} e_{0,0,1}$ | | |
37) $E_{b,0,1} = E_{b,0,1} + \mu_{0,1} e_{0,0,1}$ | | |
38) if $\mu_{0,1} > 1$ restart data updates | | |

Likelihood Variable Stabilization:

39) if $n$ is odd: $\bar{\mu}_{p,n} = \lambda N^{-1} \bar{\mu}_{p,n} / E_{p,n}$ | | |
40) if $n$ is even: $\bar{\mu}_{p,n} = \lambda N^{-1} \bar{\mu}_{p,n} / E_{p,n}$ | | |

Joint Process Extension:

41) $e_{0,n} = a_n^T E_{x_n,E_{0,n}/E_{x_n,n}}$ | | |
42) $e_{1,n} = e_{0,n} \mu_{1,n}$ | | |
43) $b_{1,n} = b_{n-1} + k_{0,0,1,n} e_{1,n}$ | | |

Total Complexity: 12N

4. A Simulation

In Figure 1 the convergence of the coefficient error (in dB) is shown for DR-FRLS and another common regularization approach called noise injection. In noise injection, a white noise signal is simply added to $x_n$ in the formulation of $\xi_n$ (but not $\lambda_n$) thus, whitening somewhat the sample covariance matrix. In the simulation of Figure 1 the excitation signal was a 5 second speech signal, the variance of the noise injection signal was $4.5\sigma_n^2 / N$. N was 1000, $\lambda = (3N - 1) / 3N$, $E_{x_n-1} = \delta = 8.4\sigma_n^2$ and the echo-signal to background-noise ratio was 30 dB. The DR-FRLS simulation used the same values and in addition, $\xi_n^2 = 3.8\sigma_n^2$ and $\phi_0 = 1$. The figure shows that DR-FRLS converges faster and to a lower final error level than noise injection.

5. Acknowledgement

The author would like to thank Dr. Juergen Cezanne of AT&T Bell Laboratories for many useful discussions.

![Figure 1. Convergence of DR-FRLS versus FRLS with Noise Injection](image)

REFERENCES


