AN EFFICIENT RLS (RECURSIVE-LEAST-SQUARES) DATA-DRIVEN ECHO CANCELLER FOR... (U) STANFORD UNIV CA INFORMATION SYSTEMS LAB J M CIOFFI ET AL. JUN 85 AFOSR-TR-85-0762

F49620-79-C-0058
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963-1
Computationally efficient Recursive-Least-Squares (RLS) procedures are presented specifically for the adaptive adjustment of the Data-Driven Echo Cancellers (DDECs) that are used in voiceband full-duplex data transmission. The methods are shown to yield very short learning times for the DDEC while they also simultaneously reduce computational requirements to below those required for other least-square procedures, such as those recently proposed by Salz (1983). The new methods can be used with any training sequence over any number of iterations, unlike any of the previous fast-converging methods. The methods are based upon the Fast Transversal Filter (FTF) RLS adaptive filtering algorithms that were independently introduced by the authors of this paper; however, several special features of the DDEC are introduced and exploited to further reduce computation to the levels that would be required for slower-converging stochastic-gradient solutions. Several trade-offs between computation, memory, learning-time and performance are also illuminated for the new initialization.
AN EFFICIENT, RLS, DATA-DRIVEN ECHO CANCELLER
FOR FAST INITIALIZATION OF FULL-DUPLEX DATA TRANSMISSION

John M. Cioffi
IBM Research Laboratory
San Jose, CA 95193

T. Kailath
Information Systems Laboratory
Stanford, CA 94305

ABSTRACT
Computationally efficient Recursive-Least-Squares (RLS) procedures are presented specifically for the adaptive adjustment of the Data-Driven Echo Cancellers (DDECs) that are used in voiceband full-duplex data transmission. The methods are shown to yield very short learning times for the DDEC while they also simultaneously reduce computational requirements to below those required for other least-square procedures, such as those recently proposed by Salz (1983). The new methods can be used with any training sequence over any number of iterations, unlike any of the previous fast-converging methods. The methods are based upon the Fast Transversal Filter (FTF) RLS adaptive filtering algorithms that were independently introduced by the authors of this paper; however, several special features of the DDEC are introduced and exploited to further reduce computational to the levels that would be required for slower-converging stochastic-gradient solutions.

Several trade-offs between computation, memory, learning-time, and performance are also illuminated for the new initialization.

1. INTRODUCTION
Echo cancellers were suggested for use in 2-wire full-duplex data transmission by Koll and Weinstein [1] in 1973. Additionally, much other work concerning the data echo canceller has appeared in [2-12.19]. Of particular concern in this paper is Mueller's Data-Driven Echo C canceller (DDEC) [6]. The use of stochastic-gradient LMS (Least-Mean-Square) algorithms in the DDEC has led to unacceptably long training periods for the full-duplex modem [2,11,12]. In this paper, we critically investigate the use of the Fast-Transversal-Filter (FTF) Recursive-Least-Squares (RLS) adaptive algorithms [13,14] in the DDEC to substantially reduce the necessary training period.

Because the transmitted data sequence is usually "whitened" through scrambling prior to entering the transmitter and DDEC, it was originally believed that the use of RLS adaptive algorithms would have to led to no improvement in the convergence time of the DDEC in comparison to stochastic-gradient techniques. However, Farrow [13], Honig [11] and Salz [12] verified a significant convergence improvement (about a factor of 5, see [11]) of the RLS (or closely related) methods when the double-taking data signal was intentionally silenced during the initial training phase of the data transmission. However, there are several drawbacks to the "Salz-Farrow" (SF) method in [11,12,15]. Most of these are the result of the SF method's absolute necessity for the training sequence to be "pseudorandom" with very special autocorrelation properties and with a period equal to the number of coefficients (order) of the echo canceller, which limits both the permissible order (to, say, 7, 15, 31, 63, 127, 255, 511,..., 2n-1) and the performance of the RLS DDEC.

This paper introduces FTF solutions that require less computation than the SF method, permit training of the echo canceller with any known training sequence of any length (long enough to converge), and which converge as fast or faster than the SF method. The freedom of choice in training sequence can also result in a factor 2 or more improvement (reduction) in learning time to get to the same echo canceller performance level as the SF echo canceller. We specifically investigate many interesting trade-offs between computational requirements and the performance of the echo canceller.

The new method's much-less-restrictive or arbitrary choice of training sequence permits use of sequences that are "white" (autocorrelation matrix is a diagonal), such as those of Miliewski [17]. The use of such a sequence can lead to as much as a 3dB advantage over the pseudorandom sequence. Also, the prewindowed FTF solutions do not require "priming" the echo channel with N inputs, before computation can begin, as is necessary in the SF method, and are numerically stable over the initialization time period (using the soft-constraint initialization of [13,14]).

A possible disadvantage, however, of the new method is its requirement of more read-only memory than the SF method, if one wishes to keep computation to an absolute minimum. This extra memory is used to pre-store certain quantities of the RLS algorithms that are solely a function of the known training sequence. Since practical experience dictates that the cost of read-only memory, in comparison to the cost of the other signal-processing functions that appear in high-speed modems, is low; this possible disadvantage is minimal.

Section 2 reviews and analyzes the RLS DDEC. Section 3 introduces and discusses the new recursive initialization procedures. Finally, Section 4 is a brief conclusion. A longer version of this paper appears in [16].

2. RLS AND THE DDEC
This section briefly reviews the DDEC and the application of RLS methods to it.

2.1 Definitions and Terminology
The near-end transmitted data signal is defined as

\[ u_1(t) \triangleq \text{Re} \left( \sum x(kT_s)p(t-kT_s)e^{j\omega_Tt} \right) \]

where the inphase and quadrature data symbols are the real and imaginary parts of \( x(kT_s) \), respectively. The carrier frequency is \( \omega_c/2\pi \), the baseband pulse shaping is \( p(t) \), and \( 1/T_s \) is the symbol rate. Also, "Re" denotes the real part of a complex number. \( u_1(t) \) is the real part of analytic signal, \( U_1(t) \).

\[ U_1(t) = \sum x(kT_s)p(t-kT_s)e^{j\omega_Tt} \]

The impulse response of the combined hybrid and channel path is \( h(t) \). The hybrid output, \( d(t) \), is the sum of the echo, and the uncorrelated double-taking data signal and channel noise \( u_2(t) \).

\[ d(t) = h(t) + u_1(t) + u_2(t) \]

where * denotes continuous-time convolution.

\[ h(t) = \text{Re} \left( h_{BB}(t)e^{j\omega_Tt} \right) \]

where \( h_{BB}(t) \) is the baseband-equivalent [18] echo path for \( h(t) \).

Approved for public release, distribution unlimited.
Then \( d(t) \) becomes (see [18])
\[
d(t) = \text{Re} \left\{ \sum_k x(kT_s)g(t-kT_s) e^{jk\omega_s t} \right\} + u_2(t) \quad 2-5
\]
where \( g(t) \)
\[
g(t) = \frac{1}{2} p(t) - h_{eg}(t) \quad 2-6
\]
The echo estimate is \( \hat{d}(t) \), and the error signal (double-talker estimate) is
\[
e(t) = d(t) - \hat{d}(t) \quad 2-7
\]
The Minimum Mean-Square-Error (MMSE) of \( e(t) \) is
\[
\hat{\mu} = \text{min} \ E[e(t)^2] = \hat{c}_2 \quad 2-8
\]
where \( \hat{\mu} \) becomes
\[
\hat{\mu} = \text{Re} \left\{ \sum_k x(kT_s)g(t-kT_s) e^{jk\omega_s t} \right\} \quad 2-9
\]
assuming a linear-time-invariant estimator followed by a modulator. \( E[\cdot] \) denotes expectation.

2.2 The Data-Driven Echo Canceller (DDEC)

The DDEC appears in Figure 1. The adaptive transversal filter acts at a tap-spacing, \( T_s/1 \), that is sufficiently short to satisfy the Sampling Theorem for the entire passband transmitted signal \( u_1(t) \), where \( l \) is integer. The continuous \( U_1(t) \) is rewritten
\[
U_1(t) = \sum_k x(kT_s)e^{jk\omega_s t}g(t-kT_s) e^{jk\omega_s (t-kT_s)} \quad 2-10a
\]
\[
= \sum_k \bar{x}(kT_s) \bar{g}(t-kT_s) \quad 2-10b
\]
where \( \bar{x}(kT_s) = x(kT_s) e^{jk\omega_s t} \),
\[
\text{and} \quad \bar{g}(t) = g(t) e^{j\omega_s t} \quad 2-12
\]
as discussed in [2]. In conventional echo-cancellation schemes [2], the designer carefully chooses \( \omega_s \) and \( T_s \) so that the rotation of the data symbols in Equation 2-10a is trivial (typically 270°). Thus, we now drop the tilde on \( \bar{x}(kT_s) \). In the ensuing results, the DDEC then synthesizes \( \bar{g}(t) \) at rate \( T_s/1 \). Since \( d(t) \) is real, \( \hat{\mu} \) should also be real, or
\[
\hat{\mu} = W_{N,k}^R \text{Re} \left\{ \bar{X}_N(k) \right\} - W_{N,k}^Q \text{Im} \left\{ \bar{X}_N(k) \right\} \quad 2-13
\]
where \( \text{"Im"} \) denotes imaginary part, \( W_{N,k}^R \) and \( W_{N,k}^Q \) are the real and imaginary parts of the adaptive transversal filter (complex \( k \times N \)-row vector) that estimates \( \bar{g}(t) \). \( N \) is the order of \( \bar{g}(t) \) or number of spanned symbol periods in \( \bar{g}(t) \) the DDEC, and \( \bar{X}_N(k) \) is the column vector \((N \times 1)\) corresponding to the last \( N \) DDEC inputs at the sampling rate \( 1/T_s \).
\[
\bar{X}_N(k) \triangleq [x(kT_s)]_0 \cdots 0 \quad \text{and} \quad x((k-1)[N+1]T_s) \quad 2-14
\]
where a superscript of \( T \) denotes transpose.

2.3 Subcancellers

Detailed analyses of the use of subcancellers appear in [2,3,5]. The essential structural simplification arises, because \( N \) (not \( N^2 \)) taps in the transversal filters contribute to \( d(T) \) at any sampling instant (see zeros in 2-14). The structure is equivalent to \( l \) sub-echo cancellers or "subcancellers" that independently act to estimate the \( l \) phases (per symbol period) of the desired echo-contaminated output. We add the new observation that the same inputs appear in each subcanceller, and the majority of computation in the FTF (or any fast-RLS) algorithms, which depends only on these inputs, need only be performed once for the group of subcancellers, even when the training sequence is unknown. Since 24 in practical voiceband modems for full-duplex data communications, this leads to large computational and storage savings.

2.4 The Application of RLS to the DDEC

The RLS DDEC chooses \( W_{N,k}^R = \Re W_{N,k}^R + j \Im W_{N,k}^Q \) to minimize the (for the \( k \)-th subcanceller)
\[
e^2_N(k) = \sum_{m=0}^{k} (x(mT_s) + iT_s) + W_{N,k}^R X_N(mT_s) + 2-15
\]
\[
+ W_{N,k}^Q X_N^*(mT_s) \quad 2-16
\]
respectively, and \( T_s \triangleq T_s/1 \). Minimization of Equation 2-15 directly requires the two-channel (real) FTF algorithms of [13-14]. However, a single-channel FTF algorithm can be used (during training) by staggering the inphase and quadrature data sequences (setting either or both to zero at appropriate time instants). The single-channel is less costly to implement than the two-channel. More about specific implementational comparisons appear in Section 3.

The solution to 2-15, when the DDEC stagggers the starting sequence, is
\[
W_{N,k}^R = \left( \sum_{m=0}^{k} d(mT_s) + iT_s X_N(mT_s) \right)^T \quad 2-17a
\]
\[
\left( \sum_{m=0}^{k} X_N^*(mT_s) X_N^*(mT_s)^T \right)^{-1} \quad 2-17b
\]
where \( k \geq N \), and \( k \) is the number of learning iterations for the subcanceller. (The reason for \( k+N \) in Equation 2-17b is
discussed in Section 3, and has to do with the aforementioned staggering. No generality or performance is lost if one by choosing \( X_{N}(mT_{s}) \) such that
\[
X_{N}^{*}(mT_{s} + (k + N)T_{s}) = X_{N}^{*}(mT_{s}) \quad 0 \leq m \leq k - 1.
\]
Thus, for all 2l subcancellers (\( f \) in phase and \( l \) quadrature), one need only invert the same matrix
\[
R_{N}(k) = \sum_{m=0}^{k} X_{N}^{*}(mT_{s})X_{N}(mT_{s})^{*}. \tag{2-19}
\]

The FTF methods invert this matrix only implicitly to obtain an equivalent set (much less storage) of parameters (\( C_{N,k} \) filters or "Kalman Gain," see [16]). All of the computation in the equivalent of the inversions can be off line since the training sequence \( X_{N}(kT_{s}) \) is known beforehand. No useful off-line computation can be performed in the SF method, which also is restricted to the use of pseudorandom training sequences.

2.5 Performance Analysis of RLS DDEC
We assume that \( NT_{s} \) exceeds the nonzero time extent of \( \tilde{g}(t) \). Under this common assumption, the RLS estimates of \( W_{N,k}^{R} \) and \( W_{N,k}^{Q} \) are unbiased after \( k \geq N \) iterations (see [13,14]), that is
\[
E[W_{N,k}^{R}] = Re\left[\tilde{g}(1T_{s})g^{*}(k - N + 1)T_{s} + i\tilde{g}(1T_{s})T_{s}\right], \tag{2-20a}
\]
irrespective of the double-talker's presence. \( g_{i} \) is the impulse response for the \( i \)th echo subchannel. This unbiased estimator property is not exhibited by stochastic-gradient solutions until much later time. The (RLS) covariance matrix ([13,14]) for either \( W_{N,k}^{R} \) or \( W_{N,k}^{Q} \) is (for \( k \geq N \) and white channel noise)
\[
\text{cov}[W_{N,k}^{R}] = \text{cov}[W_{N,k}^{Q}] = R_{N}^{-1}(k)\sigma_{a}^{2}. \tag{2-21}
\]
where \( \sigma_{a}^{2} = E[u_{a}(t)^{2}] \). \tag{2-22}
and \( R_{N}(k) \) is given in 2-19. Thus, the RLS solution is near optimum after \( N \) iterations only if \( \sigma_{a}^{2} \) is small. One achieves small \( \sigma_{a}^{2} \) by intentionally silencing the double-talking data signal during training [12]. This leaves \( \sigma_{a}^{2} \) equal to the residual channel-nose power level, which is typically much smaller than the power levels of the other signals in the problem. However, the stochastic-gradient methods will still be far from optimum because of the biased mean of \( W_{N,k}^{R} \) or \( W_{N,k}^{Q} \) after only \( N \) iterations. They take about 5-10 times longer [11,16].

Figure 2 simulates a situation typical of 4800 bps first iteration of the RLS covariance matrix ([13,14]) for procedures for the RLS DDEC. (See Tables 1-3).

3. RLS DDEC ALGORITHM COMPARISON
This section lists and compares the various initialization procedures for the RLS DDEC. (See Tables 1-3).

3.1 New FFT Solutions for DDEC Initialization
An important component in assessing performance and learning time of the RLS DDEC is the data window for the sum-of-squares-errors criterion (equation 2-15). In the DDEC, essentially two windowing cases are of interest: the prewindowed case and the Growing-Memory Covariance (GMC-"unwindowed") case (see [13,14]). The prewindowed FFT solutions assume that all data before the very first iteration is zero. The more general GMC case allows this data to take arbitrary values. The GMC method is only necessary for the DDEC if one desires the autocorrelation matrix to assume some exact, preprescribed form on the \( N \)th iteration of the initialization, such as is in the SF method [12]. This fixing of the autocorrelation matrix mandates the "priming" of the echo channel with approximately \( N \) nonzero data values prior to the first iteration of the algorithm, which adds an additional \( N \) (2N in multichannel or QAM case) delay (in suppressing both inphase and quadrature sequences) to the learning time. In the prewindowed solution, there is absolutely no need for this priming, thus leading to a reduction in learning time. Both experimentally and analytically, the elimination of priming is not a significant drawback for the prewindowed algorithm.

Another important component, in terms of learning time and computation, of the DDEC initialization is the choice of a single-channel or a multichannel solution. The staggered single-channel solution requires one-half the computation of the multichannel solution, but can lead to an extra \( N \) units of delay in the prewindowed case. Specifically, the proposed staggered single-channel solution first transmits and trains upon the inphase echo channel \( W_{N,k}^{R} \) and \( W_{N,k}^{Q} \) with \( \hat{e}(t) \), while simultaneously zeroing (suppressing) the quadrature training sequence. Then \( N \) symbol periods of delay are inserted before the quadrature sequences follow to clear the echo channel. The third and final step is to now transmit only the quadrature training sequence (usually the same sequence, see Equation 2-17b), while suppressing inphase signals. Since both inphase
and quadrature estimation is separate in time, a single channel algorithm is used twice, once for inphase training, and once for quadrature training.

The prewindowed single-channel (staggered) FTF algorithm is then \((k \geq N-1)\).

**Prewindowed (Single-Channel) FTF (i = 0, ..., 1)**

\[ R_k^1(m) = d(mT_k + iT_k') + W_{N,m-1}R_{N}^{k}(mT_k) \quad 3-1a \]

\[ W_{N,m}^1 = R_{N,m-1}^1 + \varepsilon_{N}^1(m)C_{N,m} \quad 3-1b \]

\[ \varepsilon_{N}^1(m) = 0 \quad 3-1c \]

\[ W_{N,m}^2 = W_{N,m-1}^2 + \varepsilon_{N}^2(m)C_{N,m} \quad 3-1d \]

while the GMC (unwindowed) version uses Covariance (Single-Channel) FTF (i = 0, ..., 1)

\[ \varepsilon_{N}^1(m) = d(mT_k + iT_k') + W_{N,m-1}^1X_N^{k}(mT_k) \quad 3-2a \]

\[ W_{N,m}^1 = R_{N,m-1}^1 + \varepsilon_{N}^1(m)C_{N,m} \quad 3-2b \]

\[ \varepsilon_{N}^2(m) = W_{N,m-1}^2 + \varepsilon_{N}^1(m-1)C_{N,m} \quad 3-2c \]

\[ W_{N,m}^2 = W_{N,m-1}^2 + \varepsilon_{N}^2(m)C_{N,m} \quad 3-2d \]

The filter \(C_{N,m}\) is computed from the known training sequence beforehand and stored for \(0 \leq m \leq k\), (see [16]). In general, in the prewindowed initialization, at least one-half of the total coefficients of \(C_{N,m}\) are always zero, leading to a reduction in both computation and storage in comparison to the covariance case, in which no such simplification generally arises. The covariance algorithm can use a training sequence that is exactly white over \(N\) iterations.

The multichannel algorithms determine the inphase and quadrature responses simultaneously. The multichannel prewindowed algorithm is \((k \geq N-1)\).

**Prewindowed (Multichannel) FTF (i = 0, ..., 1)**

\[ R_k^1(m) = d(mT_k + iT_k') + W_{N,m-1}^{k}R_{N}^{k}(mT_k) \quad 3-3a \]

\[ W_{N,m}^1 = \varepsilon_{N}^{k}(m)C_{N,m} \quad 3-3b \]

while the covariance case is Covariance (Multichannel) FTF (i = 0, ..., 1)

\[ \varepsilon_{N}^{k}(m) = d(mT_k + iT_k') + W_{N,m-1}R_{N}^{k}(mT_k) \quad 3-4a \]

\[ W_{N,m}^1 = \varepsilon_{N}^{k}(m)C_{N,m} \quad 3-4b \]

Table 1 compares the algorithms in Equations 3-1 through 3-4.

In Table 1, the single-channel (staggered) prewindowed FTF (Equations 3-1a, b, c, d) has the lowest computational requirements. The operation of this particular method was verified in Figure 2. Table 2 is a specific comparison for the conditions of Figure 2.

In Figure 2, we chose the starting sequence arbitrarily (actually pseudorandom sequence of length \(>65,000\)) to test the true echo-channel impulse response \(w\) known for the simulation, and we computed and plotted the quantity of Equations 2-23 and 2-24 (norm tap error vector). There, one sees that the DDEC converges in about \(k=22\) iterations at \(T_k\) (or 88 iterations at \(T_7\) as is shown in Figure 2), and the choice of training sequence is not critical. We have used \(k=100\) iterations at rate \(T_k\) (and 100 iterations of channel clearing between inphase and quadrature) to illustrate another advantage of our approach that the FTF solutions can be propagated for any number of iterations to further “fine-tune” the solution. Furthermore, one uses the formula for excess error from Section 2.2 of [13] to obtain

\[ \text{excess MSE} = \sigma^2_k(1 - \gamma_N(k)) \quad 3-5 \]

where

\[ \gamma_N(k) = \frac{1 - X_N^T(k)R_{N}^{-1}(k)X_N(k)}{X_N^T(k)R_{N}^{-1}(k)X_N(k)} \quad 3-6 \]

The worst (maximum of 3-5) that the echo canceller can do at any time \(k\geq N-1\) is

\[ \text{excess MSE} = \sigma^2_2 \quad 3-7 \]

In this case, the worst possible RLS MSE after echo cancellation is thus

\[ \text{MSE} = 2\sigma^2_2 \quad 3-8 \]

This worst possible performance of prewindowed RLS (which is nevertheless a dramatic improvement over stochastic-gradient methods) is achieved by the pseudorandom-trained SF method or the exactly white-trained GMC FTF method when \(T=N\). Thus, any other training sequence for the FTF performs at least as well under the excess MSE measure.
The random-access-memory (RAM) requirements of all
the various algorithms above are about the same, 2N1+N RAM
storage locations. However, the proposed FTF methods of this
chapter also require a significant amount of read only memory
(ROM) if one desires to store the quantity \( C_{N,n} \) over the
initialization interval rather than compute it on line. Just how
much storage depends upon the window and also upon the
number of channels (single- or multichannel). The storage
requirements appear in Table 3, in general and under the
conditions of Figure 3 (N=22, \( N=4 \)). There one determines that
the ROM requirements are not substantial by modern modem
standards, especially when one considers that many kilobytes of
software code are usually now found in commercial
microprocessor-controlled modem products.

4. SUMMARY

Computationally efficient Recursive-Least-Squares (RLS)
procedures have been presented specifically for the adaptive
adjustment of the Data-Driven Echo Cancellers (DDECs) that
are used in high speed full-duplex data transmission over
two-wire telephone lines. The methods have been shown to
yield very short learning times for the DDEC while they also are
shown to reduce computational requirements simultaneously to
levels below those that are required by the most efficient
existing RLS (SF) method [12]. During the initialization period,
the new numerically stable methods significantly outperform
slower-learning stochastic-gradient (LMS) solutions while also
requiring no more computational operations than these same
LMS solutions.

The new methods can be used with any training sequence
over any number of iterations. The new methods are
applications of the Fast Transversal (FTF) RLS adaptive
filtering algorithms of [13-14]. However, we additionally
exploit several special features of the DDEC to dramatically
reduce computation below the levels that would have been
required for a straightforward use of these FTF algorithms.
Several tradeoffs between computation, memory, learning-time
and performance have been illustrated. The results of this paper
can now be used to design cost-effective high-performance
DDEC’s for full-duplex data communications with acceptable
"start-up".

REFERENCES

Data Transmission Over a Two-Wire Circuit," Trans. on
4800 bps Full-Duplex Data Transmission," BSTJ, 1/85.
Full-Duplex DDD Modem," using Intersymbol
"Echo-Cancellation Techniques for Full-Duplex Data
Transmission on Two-Wire Lines," Proc NTC'76, Dallas.
for Full-Duplex Data Transmission on Two-Wire
on Comm., 9/76, pp. 957-967.
768-781.
Snijders, "Echo Cancellation in a Full-Duplex Data
Transmission System with Bipolar Encoding, Proc
NTC'76, Dallas.

---

Table 3

DDEC STORAGE COMPARISON FOR TRAINING

\( N=22 \) (bit)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RAM</th>
<th>ROM</th>
<th>R/O ( \approx )</th>
<th>ROM (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>900</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>BIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Exact FTF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
</tbody>
</table>

Prewindowed Solutions

---

Table 4

DDEC STORAGE COMPARISON FOR PRE-WINDOWING

\( N=22 \) (bit)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RAM</th>
<th>ROM</th>
<th>R/O ( \approx )</th>
<th>ROM (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>900</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>BIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Exact FTF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
</tbody>
</table>

Coresets (Crosswoven) Solutions

---

Table 5

DDEC STORAGE COMPARISON FOR CROSS-WOVERING

\( N=22 \) (bit)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RAM</th>
<th>ROM</th>
<th>R/O ( \approx )</th>
<th>ROM (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>900</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>BIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Exact FTF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
</tbody>
</table>

Coresets (Crosswoven) Solutions

---

Table 6

DDEC STORAGE COMPARISON FOR CROSS-WOVERING

\( N=22 \) (bit)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RAM</th>
<th>ROM</th>
<th>R/O ( \approx )</th>
<th>ROM (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>900</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>BIP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Exact FTF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malachowski Single-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
<tr>
<td>Malachowski Data-channel</td>
<td>2N8</td>
<td>2N8</td>
<td>208</td>
<td>2</td>
</tr>
</tbody>
</table>