Fast Fully Adaptive Processing: A Multistage STAP Approach

O. Saleh*, R. S. Adve**, and R. J. Riddolls***
*Research in Motion, 305 Philip Street, Waterloo, Ontario, Canada
**Department of Elec. and Comp. Eng., University of Toronto, Toronto, Ontario, Canada
***Defence Research and Development Canada-Ottawa, Ottawa, Ontario, Canada
email: oliver.saleh@gmail.com, rsadve@comm.utoronto.ca, Ryan.Riddolls@drdc-rddc.gc.ca

***Abstract*—Due to the associated computational load and required statistically homogeneous training, fully optimal space-time adaptive processing (STAP) is well-accepted to be impractical. Previous work has addressed these issues by reducing the adaptive degrees of freedom (DoF). In this paper we introduce a new multistage STAP approach that significantly reduces required sample support, and computational complexity, while still processing all available DoF. The multistage "fast fully adaptive" (FFA) scheme draws inspiration from the butterfly structure of the Fast Fourier Transform (FFT). The reduction in complexity and required sample support rival currently available reduced DoF algorithms. We test the efficacy of this algorithm using measured high frequency surface wave radar data.**

I. Introduction

Space-time adaptive processing (STAP) remains the most effective means of detecting a weak target buried within significantly stronger interference. Consider an adaptive phased-array radar with N spatial channels and M pulses in a coherent pulse interval (CPI). The fully adaptive matched filter (AMF) approach computes an adaptive weight for each of the NM adaptive degrees of freedom (DoF) based on the $NM \times NM$ interference covariance matrix. This matrix must be estimated using available training measurements. It is now well accepted that there are two important problems with implementing the AMF in practice [1]:

1) Obtaining the adaptive weights requires the solution of a $NM \times NM$ matrix equation in real time for each range cell of interest. Given reasonable values for N and M this task is almost impossible.

2) Obtaining a reasonably accurate estimate of the interference covariance matrix with NM DoF, such that the output signal-to-interference-plus-noise ratio (SINR) is within 3dB of the optimum, requires at least $2NM$ statistically homogeneous secondary data samples [2]. Given that these secondary data samples are obtained from range cells surrounding the one under test, this requirement can rarely be met in practice.

To address these issues researchers have developed techniques with lower complexity and fewer adaptive DoF. Some popular low complexity approaches include the Joint Domain Localized (JDL) algorithm [3], [4], the Parametric Adaptive Matched Filter (PAMF) [5] and the $\Sigma\Delta$ algorithm [6]. The common feature that all these methods share is that the total number of adaptive DoF are reduced to meet the constraints on the available secondary data, which inevitably leads to most of the originally available adaptive DoF being "wasted". The JDL method, for example, adaptively processes data within a relatively small localized processing region (LPR), after performing a non-adaptive transformation of the space-time data to the angle-Doppler domain. Reducing the adaptive DoF yields corresponding reductions in the sample support required and computational load, but at the same time impacts performance and reduces target discrimination.

This paper proposes an alternative approach that exploits all available degrees of freedom while simultaneously reducing computational complexity and required sample support. This multistage adaptive processing technique, referred to here as the Fast Fully Adaptive (FFA) approach, draws its inspiration from the butterfly structure of the Fast Fourier Transform (FFT). Essentially, the FFA approach sub-divides an $N \times M$ space-time data matrix into several sub-matrices of smaller dimensions, and then uses the AMF within each such sub-matrix to compute an intermediate statistic. The key idea underlying the FFA approach is that the *outputs* from each stage form the data matrix of the subsequent stage. This process of partitioning the newly formed data matrix, followed by adaptively processing each resulting partition, is repeated until the original $N \times M$ data matrix is reduced to a single final statistic (whose magnitude can be compared against a chosen threshold to determine if a target is present or absent at the range cell under test). Hence, as with the FFT algorithm, the FFA achieves lower complexity via a divide-and-conquer approach. A distinct advantage the FFA approach has over other conventional low-complexity STAP methods, such as the JDL, is that *all* the adaptive DoF are used at every stage.

In this paper we develop the FFA algorithm and compare the performance of the AMF (when possible), FFA and JDL algorithms using both simulated and *measured* high frequency surface wave radar (HFSWR) data obtained by Defense Research and Development Canada (DRDC) [7].

II. System Model and Fully Adaptive STAP

We develop the FFA algorithm in the context of an equi-spaced linear array of N isotropic point sensors. This choice is largely for ease of exposition and is not essential to the workings of the algorithm. The radar transmits a pulse in a chosen direction ϕ_t, referenced to broadside, searching for potential targets in this direction. In turn, the array receives returns from potential targets and other interference sources. The return signal is sampled L times with each sample corresponding to a range cell. This process is repeated M times within a CPI forming a $N \times M \times L$ data cube. For each range bin, the received data can be stored in a length NM vector which is a sum of the contributions from external interference sources, thermal noise, and possibly a target. This vector can be written as

$$\mathbf{x} = \xi \mathbf{v}(\phi_t, f_t) + \mathbf{n}, \tag{1}$$

where $\mathbf{n}$ is the vector of all interference and noise sources, ξ is the target amplitude, $\mathbf{v}$ is the space-time steering vector corresponding to a target at look angle ϕ_t and look Doppler frequency f_t. This steering vector can be written in terms of a spatial steering vector $\mathbf{a}(\phi_t)$ and a temporal steering vector $\mathbf{b}(f_t)$ [1],

$$\begin{aligned} \mathbf{v}(\phi_t, f_t) &= \mathbf{b}(f_t) \otimes \mathbf{a}(\phi_t), & (2) \\ \mathbf{a}(\phi_t) &= \begin{bmatrix} 1 & z_s & z_s^2 & \ldots & z_s^{(N-1)} \end{bmatrix}^T, & (3) \\ \mathbf{b}(f_t) &= \begin{bmatrix} 1 & z_t & z_t^2 & \ldots & z_t^{(M-1)} \end{bmatrix}^T, & (4) \\ z_s &= e^{j2\pi f_s}; \qquad z_t = e^{j2\pi f_t / f_R}, & (5) \end{aligned}$$

where $\otimes$ represents the Kronecker product of two vectors, ^{T} the transpose operator, $f_s = (d/\lambda)\sin\phi_t$ the normalized spatial frequency, λ the wavelength of operation and f_R the pulse repetition frequency (PRF). A linear processor uses a weight vector $\mathbf{w}$ to form a decision statistic Λ, i.e.,

$$\begin{aligned} y &= \mathbf{w}^H \mathbf{x}, & (6) \\ \Lambda &= |y|^2 = \left|\mathbf{w}^H \mathbf{x}\right|^2 \underset{H_0}{\overset{H_1}{\gtrless}} \Lambda_0, & (7) \end{aligned}$$

where Λ_0 is a threshold and H_0 and H_1 represent the target-absent and present hypotheses respectively. The optimal weight vector, in the minimum mean-squared error sense, is given by $\mathbf{w} = \mathbf{R}^{-1}\mathbf{v}$ where $\mathbf{R}$ is the covariance matrix of the interference, $\mathbf{R} = \mathrm{E}\left[\mathbf{n}\mathbf{n}^H\right]$, and ^{H} denotes the Hermitian of a matrix. In practice, the interference covariance matrix is unknown and must be estimated using training data. In STAP, the covariance matrix corresponding to the primary range cell (the cell under test) is estimated using data from K secondary range cells as $\widehat{\mathbf{R}} = \frac{1}{K}\sum_{r=1}^{K} \mathbf{x}_k \mathbf{x}_k^H$, where $\mathbf{x}_k$ is the space-time data snapshot at the k^{th} secondary range cell. As mentioned earlier, the fundamental reason this processor is impractical is that an adequate number of secondary cells are rarely available in practice.

III. Fast Fully Adaptive Processing

In this section we will propose an alternative STAP approach that exploits all available degrees of freedom while simultaneously reducing computational complexity and required sample support. This multistage adaptive processing scheme draws its inspiration from the butterfly structure of the FFT. Essentially, the FFA approach sub-divides an $N \times M$ data matrix into several sub-matrices of smaller dimensions, and then uses the AMF within each such sub-matrix to compute an intermediate statistic.

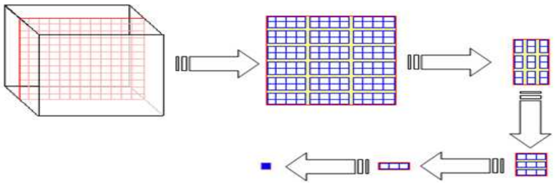

Fig. 1. The multistage representation of the FFA method

The key idea behind the FFA approach lies in that the outputs from each successive stage can be combined *adaptively* to form the data matrix of the subsequent stage. This process of repartitioning the newly formed data matrix, followed by adaptively processing each resulting partition, is repeated until the original $N \times M$ data matrix is reduced to a single final statistic whose magnitude is compared against a certain threshold to determine if a target is present or absent at the range cell under test. The FFA scheme is of relatively low complexity, with the distinct advantage that the entire data matrix is adaptively processed at every stage.

A. Regular FFA

In this section we focus on the development of the FFA approach in its most intuitive form; the "regular" FFA illustrated in Fig. 1. We adopt a divide and conquer strategy that partitions the $N \times M$ space time snapshot into $(t_s \times t_t)$ smaller spatio-temporal matrices of dimensions $N' \times M'$, where $N' = N/t_s$ and $M' = M/t_t$ are chosen corresponding to the available training data. Importantly, $N' \ll N$ and $M' \ll M$. We then apply the AMF algorithm on each of these partitions which results in a new $t_s \times t_t$ matrix whose entries are composed of *the complex output statistics, using Eqn.* (6)*, of the corresponding AMF processes*.

The weights computed by the AMF within each partition are scaled to retain the steering vector structure across the resulting spatial and temporal dimensions. As a result the new $t_s \times t_t$ matrix of residual spatio-temporal outputs, from the first stage of processing, forms a space-time matrix containing the same target, but with a steering vector whose inter-element and inter-pulse phase shift has been modified. Note that this is largely for convenience; as long the impact of the processing of the original steering vector is tracked, performance is not affected. Also modified by the first stage of processing is the interference, which has been suppressed in each partition by the AMF, yielding an "attenuated" residual interference matrix in the forthcoming processing stage.

The resulting $t_s \times t_t$ data-matrix is again repartitioned (not necessarily in the same way as the original space-time snapshot) and each partition is processed by the AMF yielding the next stage of spatio-temporal outputs. This procedure is repeated until a final statistic is obtained.

For convenience, we formalize this approach for the case with the same partition size at all levels for a linear array of point sensors. We begin by reformatting the data and steering vectors, $\mathbf{x}$ and $\mathbf{v}$, as $N \times M$ matrices $\mathbf{X}$ and $\mathbf{V}$ respectively. These matrices are partitioned into $t_s \times t_t$ sub-matrices each of size $N' \times M'$. Denote the n^{th} spatial and m^{th} temporal partition of the space-time snapshot and steering matrix as $\mathbf{X}_{nm}^{(0)}$ and $\mathbf{V}_{nm}^{(0)}$, $n = 1, 2, \ldots, t_s$ and $m = 1, 2, \ldots, t_t$ respectively. The superscript $^{(0)}$ specifies that we are currently processing the starting (zeroth) stage in the tree-like structure.

The $(n,m)^{th}$ steering matrix partition, $\mathbf{V}_{nm}^{(0)}$, is related to first steering matrix partition, $\mathbf{V}_{11}^{(0)}$ as follows:

$$\mathbf{V}_{nm}^{(0)} = z_s^{(n-1)N'} z_t^{(m-1)M'} \mathbf{V}_{11}^{(0)}. \tag{8}$$

The AMF is used within each partition. Consider the first partition, with data matrix $\mathbf{X}_{11}^{(0)}$ of size $N' \times M'$. The sample support required to estimate the relevant interference covariance matrix, $\mathbf{R}_{11}^{(0)}$, is reduced from $2NM$ to approximately $2N'M'$, with corresponding reductions in the computational load to solve the resulting matrix equation. The weight vector for the first partition is given by $\mathbf{w}_{11}^{(0)} = (\widehat{\mathbf{R}}_{11}^{(0)})^{-1}\mathbf{v}_{11}^{(0)}$, where $\mathbf{x}_{11}^{(0)}$ and $\mathbf{v}_{11}^{(0)}$ are the $N'M' \times 1$ vectorized forms of $\mathbf{X}_{11}^{(0)}$ and $\mathbf{V}_{11}^{(0)}$ respectively. The intermediate statistic, for the *next* stage, corresponding to this first partition is given by

$$y_{11}^{(1)} = \frac{(\mathbf{w}_{11}^{(0)})^H \mathbf{x}_{11}^{(0)}}{(\mathbf{w}_{11}^{(0)})^H \mathbf{v}_{11}^{(0)}} = \frac{(\mathbf{w}_{11}^{(0)})^H (\mathbf{n}_{I11}^{(0)})}{(\mathbf{w}_{11}^{(0)})^H \mathbf{v}_{11}^{(0)}} + \xi \frac{(\mathbf{w}_{11}^{(0)})^H (\mathbf{v}_{11}^{(0)})}{(\mathbf{w}_{11}^{(0)})^H \mathbf{v}_{11}^{(0)}}, \tag{9}$$

$$= n_{I11}^{(1)} + \xi, \tag{10}$$

where $\mathbf{n}_{I11}^{(0)}$ is the colored noise component in the first partition which reduces to $n_{I11}^{(1)}$ in the next stage. The target amplitude, ξ, remains unchanged. In a similar fashion we can compute the optimal weight vector and intermediate statistic for the $(n,m)^{th}$ partition using $\mathbf{w}_{nm}^{(0)} = (\widehat{\mathbf{R}}_{nm}^{(0)})^{-1}\mathbf{v}_{nm}^{(0)}$ to get

$$y_{nm}^{(1)} = \frac{(\mathbf{w}_{nm}^{(0)})^H \mathbf{x}_{nm}^{(0)}}{(\mathbf{w}_{nm}^{(0)})^H \mathbf{v}_{11}^{(0)}} = \frac{(\mathbf{w}_{nm}^{(0)})^H \mathbf{n}_{Inm}^{(0)}}{(\mathbf{w}_{nm}^{(0)})^H \mathbf{v}_{11}^{(0)}} + \xi \frac{(\mathbf{w}_{nm}^{(0)})^H \mathbf{v}_{nm}^{(0)}}{(\mathbf{w}_{nm}^{(0)})^H \mathbf{v}_{11}^{(0)}}, \tag{11}$$

$$= n_{Inm}^{(1)} + \xi z_s^{(n-1)N'} z_t^{(m-1)M'}, \tag{12}$$

where we made use of the relation in Eqn. (8) between the $(n,m)^{th}$ steering matrix partition and the first steering matrix partition. Note that the normalization in Eqn. (11) uses the steering vector corresponding to the first partition.

From Eqn. (12) in stage one, the phase shift of the target component between the first and $(n,m)^{\text{th}}$ partition is given by $z_s^{(n-1)N'} z_t^{(m-1)M'}$, i.e., using Eqn. (5) the equivalent phase shifts in stage one are given by

$$z_s^{(1)} = z_s^{N'}, \qquad z_t^{(1)} = z_t^{M'}. \tag{13}$$

Using Eqns. (10)-(13), the second stage comprises a $t_s \times t_t$ data matrix containing a target with the same amplitude but with *new temporal and spatial phase shifts,* $z_t^{(1)}$ and $z_s^{(1)}$ respectively. The algorithm iterates the partitioning and processing until a single final statistic is obtained. At each stage, the target steering matrix changes as per Eqn. (13).

Block Diagonal Interpretation: The expected advantages of the FFA are clear: the use of the divide-and-conquer approach allows for *all* DoF to be used while significantly reducing both the sample support requirements and computation load. Each adaptive problem is of size $N'M'$. However, it is important to note that the FFA scheme *does not* lead to an equivalent model of the fully-optimal AMF which solves for all NM DoF simultaneously. As a result, if adequate sample support were available, some performance degradation is expected. However, for practical scenarios where sample support is scarce, the fully optimal AMF is not implementable and the FFA becomes a strong practical alternative.

The FFA algorithm may be interpreted as a block-diagonal approximation to the fully-optimal AMF. The AMF weights are given by $\mathbf{w} = \widehat{\mathbf{R}}^{-1}\mathbf{v}$ where $\widehat{\mathbf{R}}$ is matrix of size $NM \times NM$. On the other hand, the FFA processes blocks of size $N'M' \times N'M'$ on the *diagonal* of $\widehat{\mathbf{R}}$. (This interpretation requires some re-indexing of the space-time data vector. This re-indexing is not presented here.) The loss in performance of the FFA, compared to the optimal AMF, is due to the neglecting of the off-diagonal blocks within $\widehat{\mathbf{R}}$.

B. Unequal and Overlapping Partitions

The FFA algorithm, as described above, assumes partitions of the same size. This restrictive choice is largely for ease of exposition and not fundamental to the algorithm. Simple variants of the FFA scheme are unequal and, even, overlapping partitions. For example, in the case of unequal partitions, one could use P_s spatial and P_t temporal partitions such that the i^{th} partition comprises $N_i^0, i = 0, \ldots, P_s - 1$ spatial and $M_i^0, i = 0, \ldots, P_t - 1$ temporal DoF. At each stage, the processing remains as described above for equal partitions except that the steering vector/matrix has to be carefully calculated for each stage. For example, in creating stage 1, the $(n,m)^{\text{th}}$ entry in the $P_s \times P_t$ steering matrix is given by

$$z_s^{S_n^0} z_t^{S_m^0},$$

where $S_n^0 = \sum_{i=0}^{n} N_i^0$ and $S_m^0 = \sum_{i=1}^{m} M_i^0$. This entry assumes that the $(0,0)^{\text{th}}$ entry is set to 1.

The steering vector in stage one, therefore, corresponds to an *unequally* spaced array with P_s elements and P_t unequal temporal spacings within a CPI. The spacings are defined by the partition sizes, N_0^i and M_0^i , in the spatial and temporal domains respectively. Subsequent stages are formed in a similar manner. The FFA algorithm with unequal partitioning can therefore be used for any values of N and M.

C. Randomized FFA

The previous versions of the FFA scheme, described above, use partitions of the space-time data matrix, i.e., contiguous entries in the matrix. From the block-diagonal interpretation described above, this translates to focusing exclusively on the diagonal blocks of the $NM \times NM$ overall space-time covariance matrix. These blocks may be of different sizes (unequal partitions) or overlap (overlapping partitions), but are restricted to the diagonal.

In this section we illustrate a more interesting variation: a *randomized* partitioning. This variant has a few key advantages: (i) it addresses the absence of a systematic and efficient procedure to determine the optimal set of spatio-temporal partitions that maximizes performance of the algorithm. A computer search for the "best" sequence becomes impossible for the case of large values of N and M with numerous factors; (ii) using the block diagonal interpretation, the FFA scheme focuses exclusively on the diagonal blocks of the space-time covariance matrix. However, a scheme that accounts for these neglected portions of this matrix would clearly improve performance; (iii) the fully optimal AMF essentially forms a coherent weighted average of the random interference component, thereby reducing its impact. Being able to do this repeatedly would improve performance.

The FFA algorithm is not limited to any specific size, or location, of partition. In fact, there is no need to restrict choices to rectangular partitions. As long as the process keeps track of the steering vector at each stage, the AMF can be applied to *any* subset of the space-time data vector. The key to the randomized FFA algorithm is taking many *random* subsets of the data vector. The resulting statistics can be grouped into a new data vector for the next stage of processing; furthermore this process can be repeated as many times as necessary.

1) Given the available training data and computation resources, choose N_{DoF}, the maximum number of adaptive DoF that can be processed. Also, vectorize the space-time data and steering matrices.
2) *Randomly* interleave (rearrange) the data vector and apply the same interleaver to the steering vector.
3) Choose blocks of length N_{DoF} from within the interleaved vectors and processes these blocks using the AMF. For example, in the zeroth stage, there would be approximately NM/N_{DoF} blocks.
4) The output statistic of each block forms the data and steering vectors for the following processing stage. Repeat steps 2 and 3 until a single "final" *complex* statistic is obtained.
5) *Repeat* steps 2-4 as many times as computationally feasible to form multiple "final" statistics that can be grouped to form a new data and steering vector. Repeat steps 2 and 3 the truly final statistic is obtained.

There are two unusual aspects to the randomized FFA algorithm described above: the first is the use of random subsets of the space-time data vector. Essentially, the randomized FFA algorithm allows use of random sub-matrices within the larger $NM \times NM$ interference covariance matrix. The second aspect is that the DoF can be *reused* as many times as necessary by re-interleaving and taking different random subsets.

IV. Numerical Evaluation

In this section we present results of simulations used to test the efficacy of the FFA approaches. The simulations use measured HFSWR data. We compare the performance of the FFA schemes against the nonadaptive filter matched to the space-time steering vector and JDL algorithms. The JDL algorithm is chosen for its relatively low complexity and sample support requirements as representative of the class of reduced rank algorithms.

The HFSWR data was measured using a system based on the east coast of Canada. The radar, operated by Defense Research and Development Canada (DRDC) at Cape Race on the Canadian East Coast [7], comprises $N = 16$ channels, $M = 4096$ pulses, and $L = 270$ range cells. The radar operating frequency is 3.1 MHz and the first range cell corresponds to 62.75km with each range cell covering 1.5 km. The 4096 pulses use a pulse repetition frequency (PRF) of 15.625 Hz. The inter-element distance of the uniform linear array is d = 33.33 m. The examples here use the data set measured on 25^{th} March 2002 at 03.02.57 am. Of the 270 range cells only the last 93 include ionospheric clutter.

The plots are based on the modified sample matrix inversion (MSMI) statistic [8], defined for a weight vector $\mathbf{w}$, primary data vector $\mathbf{x}$ and steering vector $\mathbf{v}$ as:

$$\eta_{\text{MSMI}} = \frac{|\mathbf{w}^H\mathbf{x}|^2}{\mathbf{w}^H\mathbf{v}} \tag{14}$$

A. Proxy for Probability of Detection

Example 1: This example develops a test for the measured ionospheric HFSWR clutter data analogous to probability of detection (P_D) plots used with simulated data. Using a measured data set does not allow for independent realizations to form a P_D plot. In this example we add a target-like signal to a single range cell and measure the ratio of the MSMI statistic in this primary range to the maximum MSMI statistic in the target-free cells, essentially the difference on the dB scale, denoted as ΔMSMI. The target Doppler is set at 0.18 Hz. Negative values of ΔMSMI indicate false alarms. This process is repeated for all 93 range cells with ionospheric clutter and the results averaged. The parameters used are:

- JDL: 3 angle bins, 3 Doppler bins, $1/(N\sqrt{2})$ angle spacing and $1/(M\sqrt{2})$ Doppler spacing.
- Regular FFA: spatial partitioning sequence $= [2, 2, 2, 2]$ and temporal partitioning sequence $= [4, 4, 16, 16]$.
- Overlapping FFA: $d_M = [2, 8, 8, 2, 16]$, $d_N = [2, 2, 2, 2, 1]$, $s_M = [2, 8, 8, 2, 16]$ and $s_N = [2, 2, 2, 2, 1]$.
- Randomized FFA: $N_{\text{DoF}} = 16$, N_{iter} = 1, i.e., iterations over random interleavings are not used.

Here, d_N (d_M) is the length of a spatial (temporal) partition and s_N (s_M) the separation between spatial (temporal) partitions, at a certain depth of the FFA tree. Figure 2 plots the results for the JDL and FFA methods. It should be emphasized that the abscissa here is the amplitude of the injected target in dB and not a signal-to-noise ratio. This is because the noise level in the data is unknown as is the scale factors used in the measurements. In this case of *measured* data, the superiority of the FFA-based schemes is clear. The JDL scheme is not able to reliably detect all target amplitudes considered. The randomized FFA scheme clearly shows the best performance

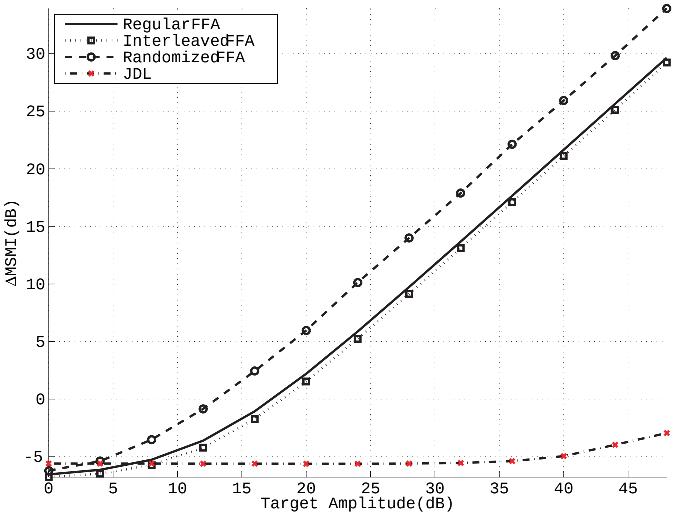

Fig. 2. ΔMSMI versus target amplitude for the JDL and FFA algorithms using $K = 93$ secondary samples.

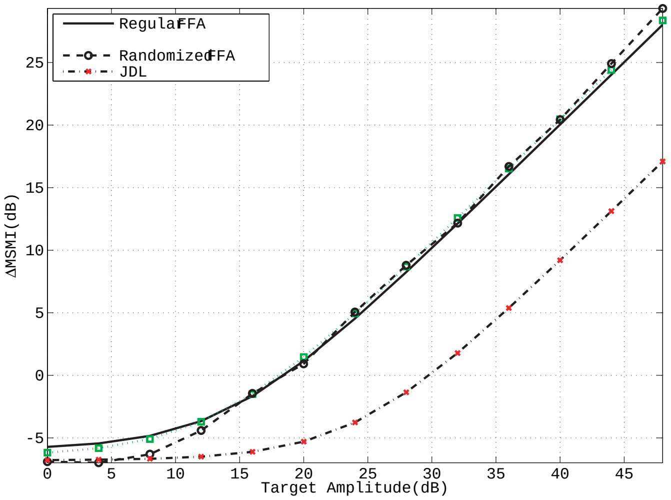

Fig. 3. ΔMSMI versus target amplitude for the JDL, and FFA algorithms reduced sample support scenarios ($K = 20$).

with about a 4.5 dB gap from the regular and overlapping FFA schemes which show similar performance.

Example 2: The previous example used all 93 ionospheric range cells to estimate the interference covariance matrix; the final example in this section focuses on a reduced sample support with $K = 20$.

The algorithm parameters utilized in this simulation are:

- JDL: 3 angle bins, 3 Doppler bins, $1/(N\sqrt{2})$ angle spacing and $1/(M\sqrt{2})$ Doppler spacing.
- Regular FFA: spatial partitioning sequence $= [2, 2, 2, 2, 1, 1]$, temporal partitioning sequence $= [4, 4, 4, 4, 4, 4]$.
- Overlapping FFA: $d_M = [4, 3, 3, 5, 4, 4, 2, 10, 8]$, $d_N = [2, 2, 2, 2, 2, 2, 3, 1, 1]$, $s_M = [2, 2, 2, 2, 2, 2, 2, 3, 1]$ and $s_N = [2, 1, 1, 1, 1, 1, 1, 1, 1]$.
- Randomized FFA: $N_{\mathrm{DoF}} = 10$ in the first two stages followed by $N_{\mathrm{DoF}} = 9$ in the following stages. $N_{\mathrm{iter}} = 1$, i.e., DoF are not reused.

The results of this simulation are shown in Figure 3. There are several interesting points to note from this figure; while the performance of all three FFA schemes has worsened due to the reduced sample support, the performance of the JDL method has improved significantly. While this may seem counter intuitive, we attribute the behavior of the JDL algorithm to the non-homogeneity of the ionospheric clutter across wider range spans. Over shorter spans such as 20 range cells, the ionospheric clutter does maintain a certain degree of homogeneity and as a result leads to a more accurate estimate of the error covariance for the JDL, and, in turn, to an improvement in performance. Interestingly, the FFA methods seem to behave in the opposite manner, i.e., they seems to be less sensitive to the non-homogeneity of the ionospheric clutter than JDL, and are capable of better exploiting the available sample support. As this plot reveals, the regular and overlapping FFA methods enter the linear region at a target amplitude of approximately 15dB. The randomized FFA enters its linear region at approximately 7dB, while JDL enters the linear region at about 23dB. In the high SNR region all three FFA schemes show similar performance for the reduced sample support scenario, and outperform JDL by approximately 12.5dB. Note that in this case, the randomized FFA scheme does not reuse adaptive DoF.

Figures 2 and 3 allow some speculation as to the source of the stability of the FFA algorithms in non-homogeneous clutter scenarios. While each individual "small" AMF implementation within the larger FFA scheme is severely impacted by clutter non-homogeneities, the figures suggest that the several combinations and recombinations average out this impact.

B. *MSMI vs Range*

In previous publications using measured data, e.g., [4], a popular approach to algorithm testing is plots of the detection statistic versus range. In this section we inject realistic targets occupying multiple range cells (due to the range resolution of the radar and the chosen sampling rate) at specific ranges into the data cube, and attempt to detect the injected targets using the algorithms under test. This test is most relevant to measured data and we focus here on the measured ionospheric clutter data sets. As in the examples above, the figure of merit is the ratio of the MSMI statistic at the target range cell to the maximum statistic at other range cells or the difference between these two statistics on the dB scale.

The injected target has an absolute amplitude of 45dB, an amplitude at which non-adaptive processing cannot detect the target. As before, the target response is obtained via a separate data set with a high-SNR target. The measured target response shows a spread over three range cells. The target is injected in range in 223, in the heart of the ionospheric clutter region, at an angle of $35°$[1], and at a Doppler of 0.18Hz.

[1] A preliminary analysis of the measured HFSWR datacubes indicated the presence of a high power external interference source at an angle of 35° and spread across all the ranges of interest.

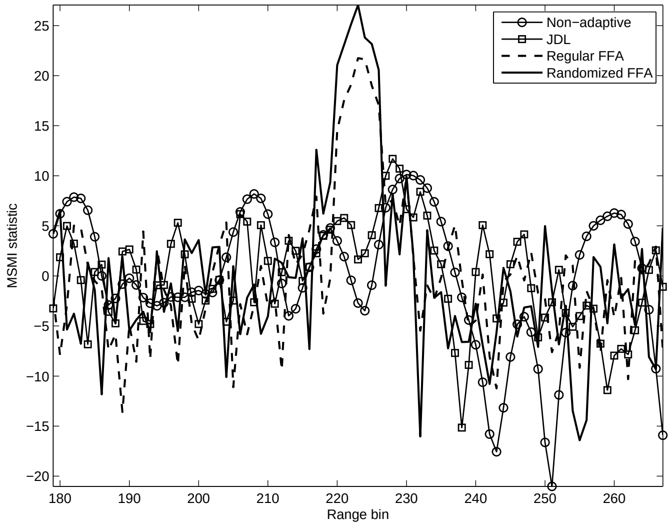

Fig. 4. MSMI vs Range plots for the Nonadaptive, JDL, regular FFA and randomized FFA schemes.

The parameters used are:

- JDL: 3 angle bins, 3 Doppler bins, $1/(N\sqrt{2})$ angle spacing and $1/(M\sqrt{2})$ Doppler spacing.
- Regular FFA: spatial partitioning sequence $= [2, 2, 2, 2]$ and temporal partitioning sequence $= [4, 4, 16, 16]$.
- Overlapping FFA: $d_M = [2, 8, 8, 2, 16]$, $d_N = [2, 2, 2, 2, 1]$, $s_M = [2, 8, 8, 2, 16]$ and $s_N = [2, 2, 2, 2, 1]$.
- Randomized FFA: $N_{\mathrm{DoF}} = 16$, N_{iter} = 1, i.e., iterations over random interleavings are not used.

The example uses 30 range cells for secondary sample support.

Figure 4 plots the results of using four processing schemes; the figure plots the MSMI statistic for non-adaptive matched filtering, the JDL scheme (representing low DoF methods available in the literature), and the two schemes developed here: the regular FFA and the randomized FFA algorithms. The regular FFA scheme uses partition sizes of $N' = 2, M' = 4$ at each stage while the randomized FFA uses 10 DoF at each AMF operation.

As is clear from the figure, for these target parameters, both the non-adaptive and JDL schemes are unable to pick out the target. The improved performance using the FFA schemes is clear. The target for both FFA schemes rises far above the surrounding clutter range cells. The range spread in the statistic is due to the radar range resolution. It is worth noting that the improved performance over the JDL algorithm is achieved with significantly higher computational complexity due to the multiple AMF processes that must be executed.

V. Conclusions

In this paper we introduced the Fast Fully Adaptive (FFA) approach which uses a divide-and-conquer strategy to significantly reduce the computational complexity and sample support requirements of the fully-adaptive STAP scheme. The key idea underlying the schemes presented is to adaptively combine the intermediate statistics output from each individual smaller AMF process by accounting for the impact each stage has on the space-time steering vector.

In this paper the performance of the FFA schemes was tested using measured HFSWR data. In the case of non-homogeneous ionospheric clutter, the FFA scheme far outperforms the JDL scheme. Our explanation is that the FFA process combines the results of multiple smaller AMF processes thereby averaging out the non-homogeneities.

The improved performance of the FFA schemes arises at the cost of increased computational complexity. This is because of the multiple, albeit smaller, AMF processes that must be executed. One could envision a parallel implementation of these processes to reduce the computational time per look range cell. It is important to note that the fundamental limiting factor in STAP is the limited available training and the FFA scheme was designed to address this fundamental problem.

The work here could be extended in several directions. There has not been any attempt here to optimize the parameters of the FFA schemes. In this paper the FFA schemes were developed in the context of the fully-adaptive AMF algorithm. In severely non-homogeneous scenarios, researchers have developed direct data domain approaches [9], [10] which are generally very computationally intensive. The FFA scheme could be combined with these approaches to improve their computational complexity. This is especially true for systems with large values of N and/or M.

References

[1] J. Ward, “Space-Time Adaptive Processing for Airborne Radar,” MIT Linclon Laboratory, Tech. Rep. F19628-95-C-0002, December 1994.

[2] I. S. Reed, J. Mallett, and L. Brennan, “Rapid Convergence Rate in Adaptive Arrays,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 10, no. 6, pp. 853–863, Nov. 1974.

[3] A. Jaffer, M. Baker, W. Ballance, and J. Staub, “Adaptive space-time processing techniques for airborne radars,” Hughes Aircraft Company,Fullerton,CA, Tech. Rep. F3060289-D-0028, July 1991.

[4] R. S. Adve, T. B. Hale, and M. C. Wicks, “Joint Domain Localized Adaptive Processing in Homogeneous and Non-homogeneous Environments. Part I: Homogeneous Environments,” *IEE Proceedings on Radar Sonar and Navigation*, vol. 147, no. 2, pp. 57–65, April 2000.

[5] J. Roman, M. Rangaswamy, D. Davis, Q. Zhang, B.Himed, and J. Michels, “Parametric Adaptive Matched Filter for Airborne Radar Applications,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, No.2, pp. 677–692, 2000.

[6] E. Yang, J. Chun, and R. Adve, “A Hybrid D3-Sigma Delta STAP Algorithm in Non-Homogeneous Clutter,” *IEEE International Radar Conf.*, October 2007.

[7] H. C. Chan, “Characterization of Ionospheric Clutter in HF Surface-Wave Radar,” Defense Research Devlopment Canada - Ottawa, Tech. Rep. DRDC Ottawa TR 2003-114, September 2003.

[8] E. Kelly, “An adaptive detection algorithm,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22, pp. 115–127, July 1986.

[9] T. K. Sarkar and N. Sangruji, “An Adaptive Nulling System for a Narrow-Band Signal with a Look-Direction Constraint Utilizing the Conjugate Gradient Method,” *IEEE Transactions on Antennas and Propagation*, vol. 37, no. 7, pp. 940–944, July 1989.

[10] R. S. Adve, T. B. Hale, and M. C. Wicks, “Joint Domain Localized Adaptive Processing in Homogeneous and Non-homogeneous Environments. Part II: Non-Homogeneous Environments,” *IEE Proceedings on Radar Sonar and Navigation*, vol. 147, no. 2, pp. 66–73, April 2000.