Defence Research and Development Canada
Recherche et développement pour la défense Canada

CAN UNCLASSIFIED

DRDC | RDDC
technologysciencetechnologie

Fast Fully Adaptive Processing: A Multistage STAP Approach

Oliver Saleh
University of Toronto

Maryam Ravan
Cyberonics

Ryan Riddolls
DRDC – Ottawa Research Centre

Raviraj Adve
University of Toronto

IEEE Transactions on Aerospace and Electronic Systems
VOL. 52, NO. 5

Date of Publication from External Publisher: October 2016

Defence Research and Development Canada
External Literature (P)
DRDC-RDDC-2017-P121
December 2017

Canada

IMPORTANT INFORMATIVE STATEMENTS

Fast Fully Adaptive Processing: A Multistage STAP Approach

OLIVER SALEH
University of Toronto
Toronto, ON, Canada

MARYAM RAVAN
Cyberonics
Houston, TX, USA

RYAN RIDDOLLS
Defence Research and Development Canada
Ottawa, ON, Canada

RAVIRAJ ADVE
University of Toronto
Toronto, ON, Canada

Due to the need for adequate statistically homogeneous training, full-dimensional space-time adaptive processing (STAP) is well accepted to be impractical. Several previous works have addressed this issue by reducing the adaptive degrees of freedom (DoF), in turn reducing the required training. In this paper, we introduce a new multistage STAP approach that significantly reduces the required sample support while still processing all available DoF. The multistage fast fully adaptive (FFA) scheme draws inspiration from the butterfly structure of the fast Fourier transform. It uses a "divide-and-conquer" approach by creating several smaller STAP problems but then combines the outputs of each problem adaptively as well. The reduction in required sample support rivals currently available reduced DoF algorithms. We also develop three variants of the algorithm, including one that uses random subdivisions of the original STAP problem. We test the efficacy of the algorithms developed via simulations based on simulated airborne radar data and measured high-frequency surface wave radar data. The results show that for simulated homogeneous data, the performance of the FFA approaches is comparable to that of available STAP algorithms; however, with measured data, the FFA approach provides significantly better performance.

Manuscript received May 20, 2015; revised November 30, 2015, February 2, 2016; released for publication May 20, 2016.

DOI. No. 10.1109/TAES.2016.150366.

Refereeing of this contribution was handled by M. Rangaswamy.

Authors' address: O. Saleh, R. Adve, University of Toronto, Electrical and Computer Engineering, 10 King's College Road, Toronto, ON M5S 3G4, Canada; M. Ravan, Cyberonics, 100 Cyberonics Boulevard, Houston, TX 77058; R. Riddolls, Defence Research and Development Canada, Ottawa, 3701 Carling Avenue, Ottawa, ON K1A 0Z4, Canada. Corresponding author is R. Adve, E-mail: (rsadve@comm.utoronto.ca).

I. INTRODUCTION

Space-time adaptive processing (STAP) remains the most effective means to detect targets masked by far stronger interference. The optimal approach for target detection, if the interference statistics are Gaussian and known a priori, is the matched filter for colored noise. However, these statistics are usually unknown and must be estimated from available measurements. In practice, therefore, the adaptive matched filter (AMF) first estimates the interference statistics using training samples [1, 2].

Consider a phased-array radar system with N spatial channels and M pulses in a coherent pulse interval (CPI). Each range cell, therefore, corresponds to an $N \times M$ data matrix (equivalently, a length-NM data vector). The AMF solution or fully adaptive STAP, for each range-angle-Doppler cell under test, computes an adaptive weight for each of the NM adaptive degrees of freedom (DoF) based on the $NM \times NM$ estimated interference covariance matrix. There are, however, two significant problems with implementing the AMF. First, obtaining the adaptive weights requires the solution of an $NM \times NM$ matrix equation for each range cell and angle-Doppler of interest, a prohibitive computation load for practical values of N and M. Second, and fundamentally, obtaining a reasonably accurate estimate of the interference covariance matrix requires on order of $2NM$ statistically homogeneous training data samples [3]. Given that training samples are usually obtained from target-free range cells surrounding the range cell under test, this requirement can almost never be met in practice. This paper, for the most part, addresses the second—and fundamental—concern.

To address these issues, researchers have developed techniques that require less training data by using fewer adaptive DoF. Indeed, there is a vast literature on STAP, the algorithms, and their relative merits; we direct the reader to [2] and the references therein. Some well-established reduced-rank approaches include the joint domain localized (JDL) algorithm [4, 5]; the parametric adaptive matched filter (PAMF) [6], including a knowledge-aided version [7]; the $\Sigma\Delta$ algorithm [8, 9]; and newer approaches with antenna-pulse selection [10] or amplitude and phase estimation [11].

In these schemes, most of the originally available adaptive DoF are "discarded." The JDL method, for example, adaptively processes data within a relatively small localized processing region (LPR) after a nonadaptive transformation of the space-time data to the angle-Doppler domain. Reducing the adaptive DoF in this manner yields corresponding reductions in required sample support and computation load. These gains are achieved at the cost of performance in terms of discrimination of the target from interference; the best STAP algorithms minimize this loss of performance.

This paper proposes an alternative algorithm that exploits all available DoF while significantly reducing the required sample support. The approach proposed here is to subdivide the $N \times M$ data matrix into several submatrices

of smaller dimensions and then use the AMF within each such submatrix to compute an intermediate statistic. The size of the subdivisions is determined by the available training. The key innovation lies in using these intermediate statistics to form a new *data matrix*; the entries in this data matrix are to be repartitioned, and each partition is to be processed *adaptively* in a subsequent stage. This process of partitioning, adaptively processing each partition, and forming a new data matrix is repeated until the original $N \times M$ data matrix is reduced to a single final test statistic. In this regard, this multistage approach, referred to here as the fast fully adaptive (FFA) algorithm, draws inspiration from the butterfly diagram of the fast Fourier transform (FFT). Importantly, in the final reckoning, the FFA process assigns an independent adaptive weight to each of the original *NM* DoF.

The closest cousin to the FFA algorithm, as described above, is to use spatial smoothing (forward-backward smoothing) with subapertures formed from the spatiotemporal data matrix [12, 13]. In both approaches, the data matrix is partitioned using a chosen subdivision. The key difference is that, in the FFA algorithm, the partition sizes are quite small and the combining of the partitions is done adaptively. Furthermore, the FFA algorithm repartitions the resulting adaptively processed statistics for further processing, a process that is repeated in multiple stages.

An alternative interpretation is that the FFA algorithm is equivalent to a block-diagonal approximation to solving the original fully adaptive AMF matrix problem at each stage. It is worth emphasizing that, as a consequence, the FFA algorithm does not replace traditional fully adaptive processing. Even if adequate training were available, the algorithm would suffer from a performance loss—the FFA algorithm is fully adaptive in the sense that it assigns an *adaptive weight* to each of the available *NM* spatiotemporal DoF. By considering only sub-blocks of the original $NM \times NM$ AMF matrix, the algorithm suffers from a performance loss as compared to the traditional fully adaptive AMF algorithm (if adequate training is available). However, as we will see, in the more practical case of limited training, the FFA algorithm performs extremely well.

The FFA approach described so far is convenient and intuitive but, by forcing the use of disjoint contiguous space-time blocks, is also restrictive. We develop variants of the partitioning scheme that break from the block-diagonal structure and provide a trade-off between performance and computational complexity. The first two are simple extensions to unequal partition sizes and overlapping partitions allowing for implementation with, for example, systems with prime *N* and *M*. This is similar to the smoothing approach with overlapping subarrays. The final and most interesting variant uses a *random* interleaving of the space-time data matrix—this randomization provides an additional level of interference cancellation and also allows for *reuse* of the available DoF.

In this paper, we develop the FFA algorithm and the three variants described. We then analyze the computational complexity of the FFA algorithm and its variants using the lower-complexity JDL algorithm as a benchmark. We test the performance of the algorithms in two different interference scenarios: using simulated homogeneous airborne clutter data and two sets of measured data from a high-frequency surface wave radar (HFSWR) system. As will be shown in the results section, the performance of the FFA method is comparable to algorithms such as JDL processing with homogeneous data. However, the performance of the FFA algorithm proves to be significantly superior to that of the JDL approach in the HFSWR case, where clutter nonhomogeneities play a significant role.

The rest of this paper is organized as follows. Section II presents the system model for the interference scenarios considered in this paper. Section III presents the FFA algorithm and the four spatiotemporal partitioning schemes for this algorithm. One crucial contribution is a *randomized* partitioning scheme. Section IV presents an analysis of the computational complexity of the regular, overlapping, and randomized FFA approaches and compares their complexity to that of the optimal AMF and low-complexity JDL algorithms. Section V presents several simulation results used to evaluate the performance of the FFA in the airborne and HFSWR setups. Section VI concludes this paper and presents suggestions for future work.

II. SYSTEM MODEL AND ADAPTIVE MATCHED FILTERING

We develop the FFA algorithm in the context of an equispaced linear array of point sensors. This choice is largely for ease of exposition and is not essential to the workings of the algorithm. We begin with the system model and a brief review of the fully adaptive AMF scheme. The review will help with the description of the FFA algorithm in the following section.

Consider an equispaced linear radar array of N isotropic, point sensors with elements separated by a distance d. The radar transmits a pulse in a chosen direction ϕ_t, referenced to broadside, searching for potential targets in this direction. This signal reflects off (possibly) a target and other interfering obstacles (e.g., the ground in the airborne radar case). The return signal is match filtered to the pulse shape and sampled L times with each sample corresponding to a range cell.

This process is repeated M times within a CPI. The entire data set can therefore be compiled into an $N \times M \times L$ data cube. For each range bin, the received data can be stored in a length-NM vector $\mathbf{x}$ whose entries, numbered mN to $[(m+1)N-1]$, correspond to the returns at the N elements from pulse number m ($m = 0, 1, \ldots, M-1$). This data vector is a sum of the contributions from external interference sources, thermal

noise, and possibly a target and can be written as

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

where $\mathbf{c}$ represents all interference sources and $\mathbf{n}$ the additive thermal noise and ξ represents the target amplitude and is zero under the null hypothesis (i.e., target absent). Finally, the vector $\mathbf{v}$ in (1) 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],

$$\mathbf{v}(\phi_t, f_t) = \mathbf{b}(f_t) \otimes \mathbf{a}(\phi_t), \tag{2}$$

$$\mathbf{a}(\phi_t) = \begin{bmatrix} 1 & z_s & z_s^2 & \dots & z_s^{(N-1)} \end{bmatrix}^T, \tag{3}$$

$$\mathbf{b}(f_t) = \begin{bmatrix} 1 & z_t & z_t^2 & \dots & z_t^{(M-1)} \end{bmatrix}^T, \tag{4}$$

$$z_s = e^{j2\pi f_s}; \quad z_t = e^{j2\pi f_t / f_R}, \tag{5}$$

where $\otimes$ represents the Kronecker product of two vectors, the superscript 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, that is,

$$y = \mathbf{w}^H \mathbf{x}, \tag{6}$$

$$\Lambda = |y|^2 = \left| \mathbf{w}^H \mathbf{x} \right|^2 \underset{H_0}{\overset{H_1}{\gtrless}} \Lambda_0, \tag{7}$$

where the superscript H denotes the Hermitian of a matrix, Λ_0 is a detection threshold chosen corresponding to a chosen false alarm rate, and H_0 and H_1 represent the target-absent and target-present hypotheses, respectively.

A. Optimal and Adaptive Data Processing

Define $\mathbf{n}_I = \mathbf{c} + \mathbf{n}$ as the total interference and noise received at the radar and $\mathbf{R} = \mathrm{E}[\mathbf{n}_I \mathbf{n}_I^H]$ as the covariance matrix of this interference vector, where $\mathrm{E}[\cdot]$ represents the statistical expectation. If this covariance matrix is known a priori, the optimal weight vector is given by the matched filter under colored noise, $\mathbf{w} = \mathbf{R}^{-1}\ \mathbf{v}$ [1].

Unfortunately, in practice, the interference covariance matrix is unknown a priori 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

$$\widehat{\mathbf{R}} = \frac{1}{K} \sum_{r=1}^{K} \mathbf{x}_k \mathbf{x}_k^H, \tag{8}$$

where $\mathbf{x}_k$ is the space-time data snapshot at the kth secondary range cell. Under zero-mean Gaussian interference, this is the maximum likelihood estimate of $\mathbf{R}$. The AMF weights are then obtained as $\mathbf{w} = \widehat{\mathbf{R}}^{-1}\mathbf{v}$. A formulation with the extremely useful property of constant false alarm rate (CFAR) is the modified sample matrix inversion (MSMI) statistic [14],

$$\Lambda_{MSMI} = \frac{\left| \mathbf{v}^H \widehat{\mathbf{R}}^{-1} \mathbf{x} \right|^2}{\mathbf{v}^H \widehat{\mathbf{R}}^{-1} \mathbf{v}} = \frac{|y|^2}{\mathbf{v}^H \widehat{\mathbf{R}}^{-1} \mathbf{v}}, \tag{9}$$

with y defined in (6). Clearly, this fully adaptive processor assigns an adaptive weight to each of the NM DoF.

However, the fundamental issue that precludes the use of this processor is the need for adequate training. For the matrix estimate in (8) to be accurate, the secondary range cells must be 1) *statistically homogeneous* with the primary range cell, and 2) the training set must have adequate samples. The Reed-Mallet-Brennan rule [3] states that to obtain an output signal-to-interference-plus-noise ratio within 3 dB of optimal, we need $K > 2NM$. Unfortunately, such generous—and homogeneous—sample support is rarely available in practice. For example, in the HFSWR system used in this paper, $N = 16$ and $M = 4096$, while the total number of range cells in the data cube is $L = 270$.

Most solutions to this problem reduce the adaptive DoF, with corresponding reductions in the required sampled support. Popular examples of these schemes are the JDL, $\Sigma\Delta$, and PAMF methods mentioned in the introduction. The JDL method, for example, adaptively processes data within a relatively small localized processing region (LPR) after performing a nonadaptive transformation of the space-time data to the angle-Doppler domain. The LPR, most often, comprises just three angle and three Doppler bins resulting in nine adaptive DoF—and the required sample support dropping to approximately $K = 20$. In the next section, we present a new algorithm that uses all adaptive DoF but still reduces the required sample support to the same level.

III. FFA PROCESSING

We propose a *multistage* adaptive processing scheme that uses a "divide-and-conquer" strategy to overcome the disadvantages associated with the fully adaptive AMF. The FFA scheme requires little training, with the advantage that all available DoF are processed adaptively. The key fact enabling the FFA algorithm is that the AMF approach searches for a signal, buried within the received data, with the space-time structure defined by the steering vector; as long as the steering vector is an accurate representation of the target signal structure, the AMF process, as described in the previous section, is valid. We begin by describing the algorithm followed by the mathematical details.

A. Regular FFA

In this section, we focus on the development of the FFA approach in its most intuitive form. The $N \times M$ space-time matrix is partitioned into $(t_s \times t_t)$ spatiotemporal matrices, each of dimension $N' \times M'$. Here, $N' = N/t_s$ and $M' = M/t_t$ are chosen to satisfy $N' \ll N$ and $M' \ll M$. We then apply the AMF to each of

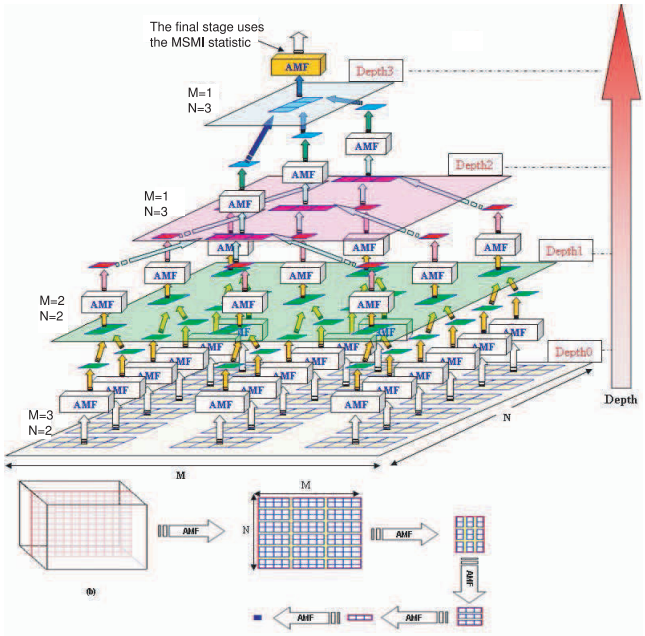

Fig. 1. A tree-like representation of the FFA method for a data cube with $M = 12$ pulses, $N = 12$ elements, spatial-partitioning sequence = [2, 2, 1, 3], and temporal-partitioning sequence = [4, 1, 3, 1].

these partitions, forming a *complex* statistic for each partition. This statistic is of the form in (6), not the magnitude statistic as in (9). The $t_s \times t_t$ statistics are organized in a new *data matrix* whose entries are the complex outputs of the corresponding AMF processes.

Of crucial importance is the impact on the steering vector. In each partition, the weights computed by the AMF are used to track the effect on the steering vector. As a result, the new $t_s \times t_t$ matrix of output statistics, from the first stage of processing, forms a space-time matrix containing the same target (if present in the original data) but with a modified steering vector. One approach is to normalize the weights to retain the steering vector structure with modified interelement and interpulse phase shifts. However, note that as long as the impact of the processing on the steering vector is tracked, the approach is valid.

The resulting $t_s \times t_t$ data matrix is again repartitioned (not necessarily in the same manner as the original space-time snapshot), and each partition is processed by the AMF, yielding the next stage of spatiotemporal outputs. Again, the impact on the steering vector is tracked. This procedure is repeated until a final scalar statistic is obtained whose magnitude, as in (9), is compared to an appropriate threshold to decide on the presence of a target.

Fig. 1 and the block diagram in Fig. 2 illustrate the proposed FFA approach. The first illustration in Fig. 1 shows how the $M \times N$ data matrix is partitioned with each partition processed using the AMF and feeding into the next stage. The second half of the same figure shows how the number of partitions reduces in each stage. Fig. 2 shows that the partitioning, processing, and repartitioning

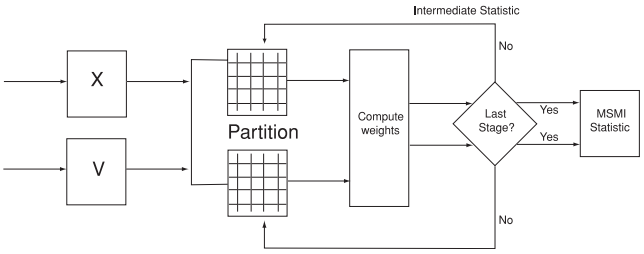

Fig. 2. A block diagram representation of the general FFA framework.

process iterates until a final statistic is available. As seen in the figures, the FFA algorithm comprises three steps:

- *Partitioning*: In the first step, the space-time data and steering matrices are partitioned into several data matrices of smaller dimensions.
- *Processing*: The AMF is used within each submatrix, an adaptive problem of dimension $(N' \times M')$. The weights obtained are applied to both the data and the steering matrices.
- *Updating and Iterating*: The intermediate statistics corresponding to each partition form the new "space-time" data matrix with a corresponding steering vector.

These three steps are then iterated until only a single scalar remains.

Mathematical Formulation: For ease of exposition, we formalize this approach for the case where the partition sizes remain the same at all levels and 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. The mth column of these matrices lists the data and steering vectors obtained for the mth transmitted pulse. Matrices $\mathbf{X}$ and $\mathbf{V}$ are partitioned into $t_s \times t_t$ submatrices each of size $N' \times M'$ (the secondary data are partitioned in a similar manner). Denote the nth spatial and mth temporal partition of the space-time and steering matrices as $\mathbf{X}_{nm}^{(0)}$ and $\mathbf{V}_{nm}^{(0)}$, $n = 0, 1, \ldots, t_s - 1$ and $m = 0, 1, \ldots, t_t - 1$, respectively. The superscript (0) specifies that we are currently processing the starting (zeroth) stage (see Fig. 1).

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

$$\mathbf{V}_{nm}^{(0)} = z_s^{nN'} z_t^{mM'} \mathbf{V}_{11}^{(0)}, \tag{10}$$

where z_s and z_t correspond to the spatial and temporal phase shifts defined in (5).

The AMF is used within each partition. Consider the first partition, with data matrix $\mathbf{X}_{00}^{(0)}$ of size $N' \times M'$. The sample support required to estimate the corresponding interference covariance matrix, $\mathbf{R}_{00}^{(0)}$,is reduced from $2NM$ to approximately $2N'M'$, with corresponding reductions in the computation load in solving the resulting matrix equation. The weight vector for the first partition is given by

$$\mathbf{w}_{00}^{(0)} = (\widehat{\mathbf{R}}_{00}^{(0)})^{-1} \mathbf{v}_{00}^{(0)}, \tag{11}$$

where $\mathbf{x}_{00}^{(0)}$ and $\mathbf{v}_{00}^{(0)}$ are the length-$N'M'$ vectorized forms of $\mathbf{X}_{00}^{(0)}$ and $\mathbf{V}_{00}^{(0)}$, respectively. The intermediate statistic, for the *next* stage, corresponding to this first partition is given by

$$y_{00}^{(1)} = \frac{(\mathbf{w}_{00}^{(0)})^H \mathbf{x}_{00}^{(0)}}{(\mathbf{w}_{00}^{(0)})^H \mathbf{v}_{00}^{(0)}} = \frac{(\mathbf{w}_{00}^{(0)})^H (\mathbf{n}_{I00}^{(0)})}{(\mathbf{w}_{00}^{(0)})^H \mathbf{v}_{00}^{(0)}} + \xi \frac{(\mathbf{w}_{00}^{(0)})^H (\mathbf{v}_{00}^{(0)})}{(\mathbf{w}_{00}^{(0)})^H \mathbf{v}_{00}^{(0)}}, \tag{12}$$

$$= n_{I00}^{(1)} + \xi, \tag{13}$$

where $\mathbf{n}_{I00}^{(0)}$ is the (scalar) colored noise component resulting from the first partition. Due to the normalization in (12), 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 as

$$\mathbf{w}_{nm}^{(0)} = (\widehat{\mathbf{R}}_{nm}^{(0)})^{-1} \mathbf{v}_{nm}^{(0)} \tag{14}$$

$$y_{nm}^{(1)} = \frac{(\mathbf{w}_{nm}^{(0)})^H \mathbf{x}_{nm}^{(0)}}{(\mathbf{w}_{nm}^{(0)})^H \mathbf{v}_{00}^{(0)}} = \frac{(\mathbf{w}_{nm}^{(0)})^H \mathbf{n}_{Inm}^{(0)}}{(\mathbf{w}_{nm}^{(0)})^H \mathbf{v}_{00}^{(0)}} + \xi \frac{(\mathbf{w}_{nm}^{(0)})^H \mathbf{v}_{nm}^{(0)}}{(\mathbf{w}_{nm}^{(0)})^H \mathbf{v}_{00}^{(0)}}, \tag{15}$$

$$= n_{Inm}^{(1)} + \xi z_s^{nN'} z_t^{mM'}, \tag{16}$$

where we made use of the relation in (10) between the (n, m)th steering matrix partition and the first steering matrix partition. Note that, unlike in (12), the statistic in (15) is normalized using the steering vector corresponding to the first partition. This is largely a choice to retain the structure of a space-time steering vector; as long as the normalization is accounted for in the steering vector of the next stage, making a different choice does not change the process.

From (16) in the first stage, the phase shift of the target component between the first and (n, m)th partition is given by $z_s^{nN'} z_t^{mM'}$; that is, using (5), the equivalent phase shifts in the first stage are

$$z_s^{(1)} = z_s^{N'}, \quad z_t^{(1)} = z_t^{M'}. \tag{17}$$

To keep track of the impact of the weights into the next stage, the weight vector is applied to the secondary data as well.

Using (13) and (17), the second stage comprises a $t_s \times t_t$ data matrix containing a target with the same amplitude but with *new spatial and temporal spacings* corresponding to $z_t^{(1)}$ and $z_s^{(1)}$, respectively. The FFA algorithm iterates the partitioning and the processing until a single final decision statistic is obtained. At each stage, the target steering matrix changes as per (17). The final complex statistic is processed to form the CFAR MSMI statistic using (9).

It is worth emphasizing that the process of partitioning, processing with adaptive weights, and iterating is applied to the secondary (training) data as well, exactly as is done for the primary data (denoted by $\mathbf{X}$ above) and steering matrix ($\mathbf{V}$).

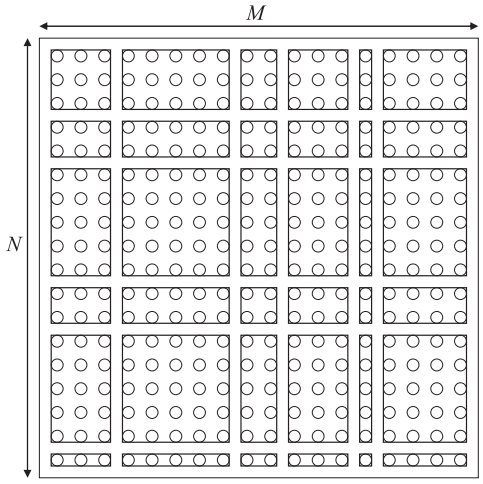

Fig. 3. Illustrating unequal partitions.

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 processed adaptively while significantly reducing both the sample support requirements and the computation load. However, it is important 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, even if adequate sample support were available, some performance degradation would be expected. However, for practical scenarios where sample support is scarce, the fully adaptive AMF is not implementable, and the FFA becomes a strong practical alternative.

With a suitable permutation of the entries in the data and steering vectors, each stage of 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 $NM \times NM$. By focusing on partitions of size of $N' \times M'$, the FFA algorithm processes blocks of size $N'\mathrm{M}' \times N'M'$ individually on the *diagonal* of $\widehat{\mathbf{R}}$. The loss in performance of the FFA, as compared to the fully adaptive AMF, is due to the neglecting of the off-diagonal blocks within $\widehat{\mathbf{R}}$. On the other hand, the use of repeated adaptive stages regains some of this loss in performance.

B. Unequal Partitions

So far, we have assumed that the partitions at each depth are of equal size; that is, all the spatiotemporal rectangles at a certain depth are of equal dimensions. This is largely for ease of exposition and not fundamental to the algorithm. For example, one could use P_s and P_t partitions in the spatial and temporal domains, respectively. Each partition comprises $N_i^0,\ i = 0,\ \ldots,\ P_s - 1$ spatial DoF and $M_i^0,\ i = 0,\ \ldots,\ P_t - 1$ temporal DoF such that $\sum_{i=0}^{P_s-1} N_i^0 = N$ and $\sum_{i=0}^{P_t-1} M_i^0 = M$. Fig. 3 illustrates a possible partition of the original $N \times M$ data matrix.

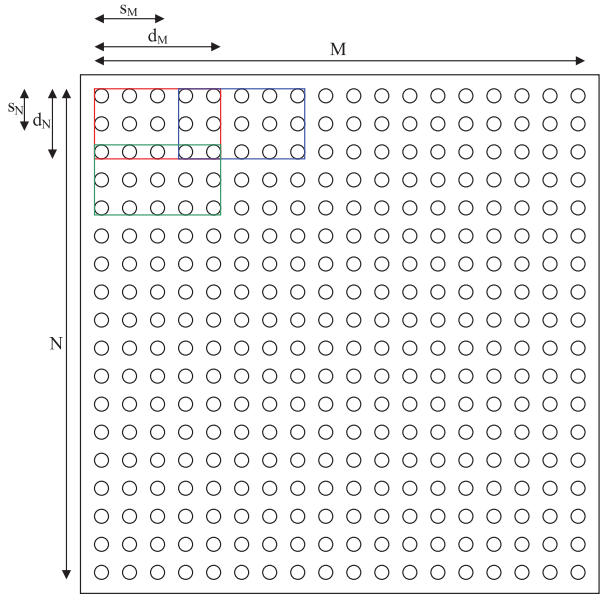

Fig. 4. A pictorial description of the overlapping FFA method.

The impact of the unequal partitions is largely through a change in the steering vector in the next stage. Define

$$\widetilde{\mathbf{v}}_{nm} = z_s^{-S_n^0} z_t^{-S_m^0} \mathbf{v}_{nm}, \tag{18}$$

where $S_n^0 = \sum_{i=0}^{n-1} N_i^0$ and $S_m = \sum_{i=0}^{m=1} M_i^0$. Essentially, (18) creates a steering vector for the (n, m)th partition with the first entry normalized to 1. After AMF processing within the (n, m)th partition, the output statistic, analogous to (15), is

$$\begin{aligned} y_{nm}^{(1)} &= \frac{(\mathbf{w}_{nm}^{(0)})^H \mathbf{x}_{nm}^{(0)}}{(\mathbf{w}_{nm}^{(0)})^H \widetilde{\mathbf{v}}_{nm}^{(0)}} = \frac{(\mathbf{w}_{nm}^{(0)})^H \mathbf{n}_{Inm}^{(0)}}{(\mathbf{w}_{nm}^{(0)})^H \widetilde{\mathbf{v}}_{nm}^{(0)}} + \xi \frac{(\mathbf{w}_{nm}^{(0)})^H \mathbf{v}_{nm}^{(0)}}{(\mathbf{w}_{nm}^{(0)})^H \widetilde{\mathbf{v}}_{nm}^{(0)}}, \\ &= n_{Inm}^{(1)} + \xi z_s^{S_n^0} z_t^{S_m^0}. \end{aligned} \tag{19}$$

The steering vector in the first stage, 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_i^0 and M_i^0, 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. Overlapping FFA

The previous section generalized the FFA scheme to arbitrary values of M and N. However, the data matrix at any stage was partitioned into *disjoint* sets. This is unnecessarily restrictive, and here we generalize the partitioning scheme such that data partitions can overlap. This allows for *reuse* of the available DoF and inherent smoothing across the space-time data matrix. Define d_N and d_M to be the lengths of a spatial and temporal partition, respectively, at a certain depth of the FFA tree. Next, define s_N and s_M to be the separation between two consecutive spatial and temporal partitions, respectively. In the example shown in Fig. 4, $d_N = 3$, $s_N = 2$, $d_M = 5$, and $s_M = 3$. Fig. 4 might represent either the data matrix of space-time samples at a certain depth, the secondary data, or the steering matrix corresponding to the look angle and Doppler (each must be partitioned in an identical manner).

The overlapping FFA scheme then proceeds much like the unequal partitioning scheme described earlier. We therefore do not present the details of this approach. The key difference from the previous section is the reuse of DoF, which allows for improved performance at the expense of added computational load. Note that the choices of interpartition spacing (e.g., s_N) and partition size (e.g., d_N) are *not* independent since the union of all partitions should include all spatial elements. Clearly, the associated benefits are at the cost of increased computation load since each stage now includes more partitions.

D. Randomized FFA

This section presents an alternative partitioning approach but one that is a significant departure from all previous STAP algorithms. All the formulations presented so far are based on *contiguous* partitions. This is, again, restrictive. There are, moreover, other motivations to consider an alternative formulation:

- A significant issue with the FFA method is the lack of a systematic and efficient procedure to determine the optimal partitions that maximize algorithm performance (illustrated later in Fig. 16). A computer search for the "best" sequence becomes impossible for the case of large values of N and M with numerous factors and, in any case, would defeat the purpose of reduced computation complexity. This task becomes even more difficult in the overlapping FFA scenario, where the number of parameters requiring optimization doubles (compared to regular FFA processing).
- From the block-diagonal interpretation presented earlier, all schemes so far have focused at each stage on the diagonal blocks of the space-time covariance matrix. However, a scheme that accounts for the neglected portions of this matrix would clearly improve performance.
- 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.
- An important issue with STAP is that measured data sets are invariably nonhomogeneous. Allowing for random combinations and recombinations of these nonhomogeneities potentially reduces their impact by averaging out the individual contributions.

As seen in the previous sections, 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

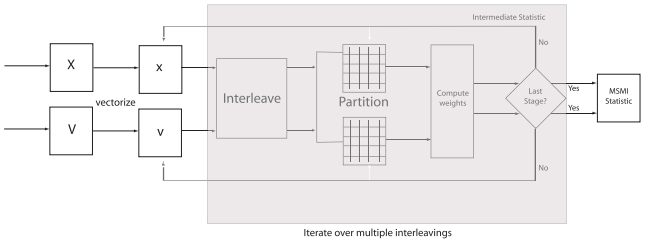

Fig. 5. A block diagram representation of the randomized FFA algorithm.

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.

The steps to the randomized FFA algorithm, illustrated in Fig. 5, are the following:

1) Given the available training data (K), choose as N_{DoF} the maximum number of adaptive DoF that can be processed. Usually, $N_{\text{DoF}} \simeq K/2$.
2) Vectorize the space-time data and steering matrices.
3) *Randomly* interleave (rearrange) the data vector and apply the same interleaver to the steering vector and secondary data vectors.
4) Choose blocks of length N_{DoF} from within the interleaved vectors and process these blocks using the AMF. For example, in the zeroth stage, there would be approximately NM/N_{DoF} blocks, yielding these many output statistics to be combined in the next stage of the FFA process.
5) *Repeat* steps 3 to 4 N_{iter} times, that is, as many times as computationally feasible, to form multiple statistics in each stage of the FFA process. For example, in the zeroth stage, there would be approximately $N_{\text{iter}} \times NM/N_{\text{DoF}}$ statistics to be combined in the next stage.
6) Repeat steps 2 to 5 in each stage of the FFA process until one final statistic is reached. The parameter N_{iter} may be different for each stage of the FFA process.

We emphasize that, at every step, the same interleaving and partitioning is always applied to the data, the corresponding steering vector, and the secondary data.

There are two unusual aspects to the randomized FFA algorithm described above. The first is clearly the use of random subsets of the space-time data vector. The second is that the DoF can be *reused* essentially as many times as necessary by reinterleaving and taking different random subsets. While in theory the total number of subsets is limited by N and M, in practice these numbers are so large that the limiting factor would be the available computation capability.

The randomized FFA approach does not have a structural pattern and loses the FFT-like structure that originally inspired its development. This method becomes increasingly difficult to express mathematically or analyze. However, as the simulations shall show, the randomized approach has the distinct advantage of making the FFA more stable (i.e., less sensitive to the chosen partitioning scheme).

Revisiting the block-diagonal interpretation of the previous FFA variants, at every stage the randomized FFA algorithm accounts for random and noncontiguous $N_{\text{DoF}} \times N_{\text{DoF}}$ submatrices of the original $NM \times NM$ space-time interference matrix. Reinterleaving allows us to process as many of these submatrices as desired.

The randomized FFA has one additional potential benefit—the inherent randomness of the approach suggests that the impact of nonhomogeneities in the training data could be averaged out. As we will see in Section V, this is of particular interest in the HFSWR scenario, wherein the available data are limited and nonhomogeneities are inherent.

IV. COMPLEXITY ANALYSIS

The central aim of the FFA formulations is to address the issue of limited sample support in practical STAP scenarios. The reduction in DoF also leads to an attendant reduction in computation load as compared to fully adaptive processing. In this section, we characterize the computation load of the variants of the FFA algorithm. The computation load is compared with the popular and efficient JDL algorithm—setting the stage for the performance comparison in the next section.

JDL Processing: The JDL algorithm [5] processes space-time data after transforming these data to a localized processing region (LPR) in the angle-Doppler domain. If the LPR comprises η_a angle bins and η_d Doppler bins, the algorithm requires $\eta_a\eta_d$ inner products with the corresponding length-NM steering vectors. The algorithm running time for a single look range can be approximated as $T_{\text{JDL}} = \eta_a\eta_d\, T_1 + T_2$, where T_1 is the time needed for a single-length-NM inner product, which is on order of $O(NM)$, and T_2 is the time needed to perform inversion of the covariance matrix and is in the order of $O((\eta_a\eta_d)^3)$. In this regard, the JDL algorithm is one of the most computationally efficient STAP algorithms, requiring a single transformation to angle-Doppler space followed by the solution of a small matrix equation (usually 9×9) for each look angle-Doppler.

Regular FFA: The complexity associated with the processing of a spatiotemporal partition of size $N' \times M'$ is that of the AMF and is defined primarily by the complexity associated with the inversion of an $N'M' \times N'M'$ matrix; that is, $O((N'M')^3)$. There are $(t_s^{(i)}t_t^{(i)})$ partitions at every depth, i, each of which must undergo AMF processing. For the regular FFA method, $t_s^{(i)}$ must be a factor of $N(i)$, and $t_t^{(i)}$ must be a factor of $M(i)$ at every depth i. Consider a given spatial partitioning sequence $S_N = \{n_1,\ n_2, \ldots, n_{\max(k_N,k_M)}\}$ and a temporal partitioning sequence $S_M = \{m_1,\ m_2, \ldots, m_{\max(k_N,k_M)}\}$, where m_i and n_i are, respectively, the temporal and spatial

dimensions of the partitions at depth i, and k_N and k_M are the largest non-unit spatial and temporal factor indices, respectively. In order to make both spatial and temporal sequences the same length, we augment the shorter sequence with ones until both sequences are the same length. Thus, the maximum depth of the FFT tree is given by $\max(k_N, k_M)$. As a result, the complexity of the FFA per range can be written as

$$T_{reg\_FFA} = \sum_{i=1}^{\max(k_N,k_M)} \left\{ \frac{NM}{\prod_{j=1}^{i} n_j m_j} O\left((n_i m_i)^3\right) \right\} = \sum_{i=1}^{\max(k_N,k_M)} \left\{ \left(\prod_{j=i+1}^{\max(k_N,k_M)} n_j m_j \right) O\left((n_i m_i)^3\right) \right\}. \tag{20}$$

This equation shows that by requiring solutions to small AMF problems, the FFA algorithm is computationally efficient. In addition, the individual FFA problems could be executed in parallel if such a capability were available. However, the repeated combinations and multiple stages make it far more computationally complex in comparison with the JDL algorithm.

The complexity of the unequally partitioned FFA scheme can be derived in a similar manner and is not presented here.

Overlapping FFA: The overlapping FFA method's complexity is somewhat more difficult to characterize since it involves twice the number of parameters as the regular FFA method. At a certain depth i, assuming that the spatial and temporal parameters are $d_N(i)$, $s_N(i)$, $d_M(i)$, and $s_M(i)$, we must process $(M'_i\, N'_i)$ spatiotemporal partitions using the AMF method, where

$$M'_i = \left\lfloor \frac{M(i) - d_M(i)}{s_M(i)} + 1 \right\rfloor, \qquad N'_i = \left\lfloor \frac{N(i) - d_N(i)}{s_N(i)} + 1 \right\rfloor, \tag{21}$$

are the sizes of the spatial and temporal dimensions of the next depth (i.e., $M'_i = M_{i+1}$, and $N'_i = N_{i+1}$). Each AMF solution requires the inversion of a $d_N(i)d_M(i)d_N(i)d_M(i)$ matrix, which has a computation load on order of $O((d_N(i)d_M(i))^3)$. Thus, the complexity of the overlapping FFA method can be shown to be

$$T_{\text{int}\_FFA} = \sum_{i=1}^{\max(k_N,k_M)} \left\{ N'_i M'_i O((d_N(i)d_M(i))^3) \right\} = \sum_{i=1}^{\max(k_N,k_M)} \left\{ \left\lfloor \frac{N_i - d_N(i)}{s_N(i)} + 1 \right\rfloor \left\lfloor \frac{M_i - d_M(i)}{s_M(i)} + 1 \right\rfloor \times O((d_N(i)d_M(i))^3) \right\}, \tag{22}$$

TABLE I
Complexity Expressions for the AMF, JDL, Regular FFA, Overlapping FFA, and Randomized FFA Methods

Algorithm	Complexity per Look Range
AMF	$O((NM)^3)$
JDL	$O((\eta_a \eta_d)^3)$
Regular FFA	$\sum_{i=1}^{\max(k_N,k_M)} \left\{ \left(\prod_{j=i+1}^{\max(k_N,k_M)} n_j m_j \right) O((n_i m_i)^3) \right\}$
Overlapping FFA	$\sum_{i=1}^{\max(k_N,k_M)} \left\{ \left\lfloor \frac{N(i)-d_N(i)}{s_N(i)} + 1 \right\rfloor \left\lfloor \frac{M(i)-d_M(i)}{s_M(i)} + 1 \right\rfloor O((d_N(i)d_M(i))^3) \right\}$
Randomized FFA	$\sum_{i=1}^{\max\_\text{depth}} N_{\text{iter}} \times [g(i) \times O\{(N_{\text{DOF}})^3\} + O\{(a(i))^3\}]$

where

$$M_i = \left\lfloor \frac{M_0 - \sum_{k=0}^{i-1} \left(\prod_{l=0}^{k-1} s_M(l) \left[d_M(k) - s_M(k) \right] \right)}{\prod_{k=0}^{i-1} s_M(k)} + 1 \right\rfloor, \tag{23}$$

where we define

$$d_M(-1) = s_M(-1) = 1, \tag{24}$$

and N_i is given by a similar expression similar to (23). Note that (22) reduces to (20) if we set $s_M(i) = d_M(i)$ and $s_N(i) = d_N(i)$ at every depth i.

Randomized FFA: In the randomized FFA scheme, at a given depth i, if we denote by $L(i)$ the size of the data vector at that depth, we use the AMF to process approximately $g(i) = \lfloor L(i)/N_{\text{DoF}} \rfloor$ partitions each of length N_{DoF} (the approximation arises because $L(i)$ may not be a multiple of N_{DoF}. This process is repeated N_{iter} times using N_{iter} different interleaved versions of the data vector at the current depth. As a result, the computation complexity associated with the randomized FFA per range can be written as

$$T_{rand\_FFA} = \sum_{i=1}^{\max\_\text{depth}} N_{\text{iter}} \times \left[g(i) \times O\left\{ (N_{\text{DoF}})^3 \right\} + O\left\{ (a(i))^3 \right\} \right], \tag{25}$$

where $a(i) = L(i) - g(i) \times N\text{DoF}$, $\quad g(i) = [\frac{L(i)}{N\text{DoF}}]$. The overall complexity for r look ranges is therefore just rT_{rand_FFA}.

Table I summarizes the computational complexities of the five schemes: the three FFA schemes, JDL, and the original AMF approach. While clearly the various parameters of each algorithm can be manipulated to favor any algorithm, in general, in terms of computation load, JDL has the distinct advantage in that it requires the solution of a single-size $\eta_a \eta_d$ matrix equation, while the FFA approach (in all of its variants) requires several AMF solutions. The results in the next section provide some sample numbers for the various parameters involved. Note that the regular FFA algorithm uses all NM adaptive DoF, while the JDL is restricted to just $\eta_a \eta_d$ adaptive DoF. The

TABLE II
Airborne Radar Simulation Parameters

Parameter	Value
Elements N	18
Element spacing	0.5 λ
Array transmit pattern	uniform
Mainbeam transmit azimuth	0°
Backlobe attenuation	30 dB
Target Doppler	25 Hz
Pulses M	18
PRF	300 Hz
Clutter slope β	1
Thermal noise power	0dB
Target azimuth	0°
False alarm rate (P_{FA})	0.001

overlapping and randomized FFA methods reuse the DoF at each depth in the computation of intermediate statistics and as a result outperform the regular FFA method. Unfortunately, therefore, this improvement in performance comes at the price of an increase in computational complexity.

V. NUMERICAL EVALUATION

In this section, we present results of simulations used to test the efficacy of the FFA approaches developed. The two sets of results are based on simulated airborne clutter and measured HFSWR data. We compare the performance of the FFA schemes against the nonadaptive filter matched to the space-time steering vector, AMF (when possible), multistage Wiener filter (MWF) [15], and JDL algorithms. The JDL algorithm is chosen for its extremely low computation complexity and sample support requirements and as representative of the class of reduced-rank algorithms. The MWF is chosen as an alternate multistage approach. As we will see, in the airborne radar scenario with homogeneous data, the JDL algorithm outperforms the regular and overlapping FFA schemes with performance comparable to the randomized FFA scheme. However, the FFA schemes vastly outperform JDL when tested with measured HFSWR data.

The simulated airborne radar data are generated using the clutter model of [1], wherein the clutter is simulated as a superposition of point clutter sources. Unless otherwise specified, the simulations use the model parameters listed in Table II. The HFSWR data were measured using a system developed by Raytheon Canada. The radar, operated by Defense Research and Development Canada at Cape Race on the Canadian East Coast [16], 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 62.75 km with each range cell covering 1.5 km. The 4096 pulses use a PRF of 15.625 Hz. The interelement distance of the uniform linear array is $d = 33.33$ m. The examples here use the data set measured on March 25, 2002, at 03.02.57 a.m. Of the 270 range cells, the last 93 include ionospheric clutter.

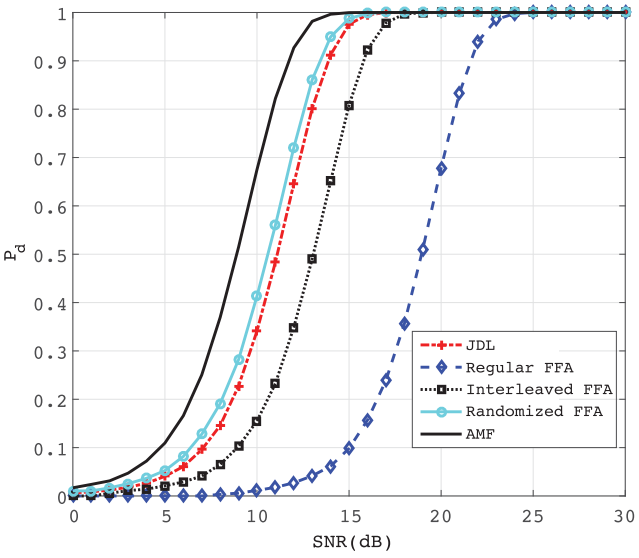

Fig. 6. Probability of detection versus SNR for a $P_{FA} = 0.001$ in the airborne scenario using 3240 secondary samples.

A. Probability of Detection

In this section, we investigate the probability of detection (P_D) as a function of target signal-to-noise ratio (SNR) given a fixed false alarm rate for the simulated airborne radar. We emphasize that the simulation results in statistically homogeneous clutter.

Example 1: The algorithm parameters used for this simulation are the following:

- JDL: three angle bins, three Doppler bins, $1/(N\sqrt{2})$normalized angle spacing, and $1/(M\sqrt{2})$ normalized Doppler spacing. This is the smallest possible and most popular choice of LPR.
- Regular FFA: spatial-partitioning sequence = [3, 2, 3], and temporal-partitioning sequence = [3, 2, 3].
- Overlapping FFA: $d_M = \mathrm{d}_N = [3, 3, 3, 3, 3, 3, 3, 3, 2]$, and $s_M = \mathrm{s}_N = [1, 1, 1, 1, 1, 1, 1, 1, 1]$.
- Randomized FFA: $N_{\mathrm{DoF}} = 9$ and $N_{\mathrm{iter}} = 9$ in the first two stages and $N_{\mathrm{iter}} = 1$ thereafter.

For the JDL algorithm, the angle and Doppler spacings were chosen *via exhaustive search* to maximize performance, whereas for the FFA case, no such search is performed. The comparisons, therefore, are to the best-case scenario for the JDL algorithm. We note that both the JDL and the FFA schemes use the same maximum number of DoF. In practice, this would be dictated by the available training; 10^4 independent interference and noise realizations were used to obtain the detection threshold corresponding to the chosen false alarm rate. All examples are based on the MSMI statistic of (9).

Fig. 6 plots the P_D as a function of SNR when a large training set is available. This figure uses $K = 10\,NM = 3240$ homogeneous range cells to estimate the interference covariance matrix. With a large amount of training, the AMF performs best as expected (a full-rank MWF would perform the same as the AMF given the large available

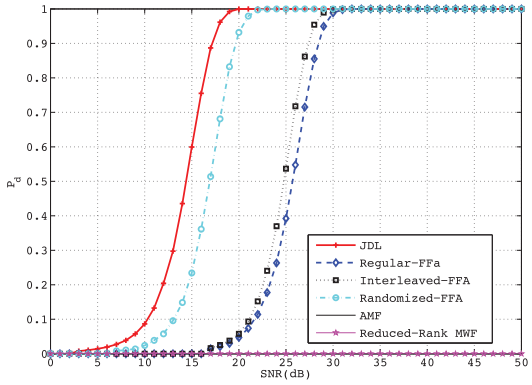

Fig. 7. Probability of detection versus SNR for a $P_{FA} = 0.001$ in the airborne scenario using 20 secondary samples.

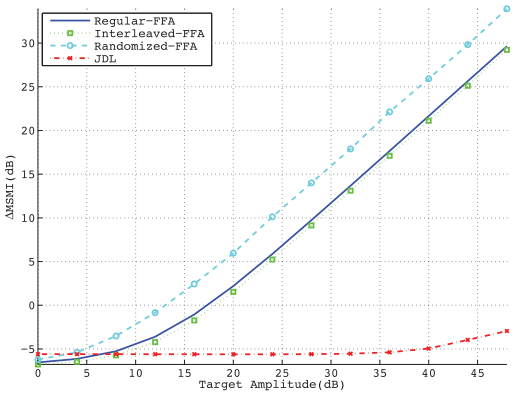

Fig. 8. ΔMSMI versus target amplitude for the JDL and FFA algorithms using $K = 93$ secondary samples.

training). Using $P_D = 0.5$ as the point of comparison, the JDL scheme outperforms the regular and overlapping FFA methods by approximately 7.5 dB and 2 dB, respectively. This is because of the optimization of the angle and Doppler spacings and the significant gain achieved by the transformation from the space-time to angle-Doppler domain. The performance of the JDL approach is also helped by the fact that the secondary data here are statistically homogeneous. On the other hand, the *randomized* FFA scheme outperforms JDL by approximately 0.5 dB and is only 1.5 dB from optimal performance.

Example 2: The second example uses a more realistic number for sample support. The algorithm parameters are unchanged from the first example, but only $K = 20$ range cells are used to estimate the interference covariance matrix. This value is about the lowest that can be used with the 9 DoF used in the JDL and FFA schemes. Fig. 7 plots the probability of detection versus SNR for each of the methods with reduced sample support. Again, the MWF is implemented using 200 iterations to ensure that we obtain its best performance given the available training.

As this plot reveals, as expected, the performance of both the AMF and the MWF with reduced rank is severely degraded. The AMF uses all 324 DoF and is implemented using a matrix pseudoinverse, while the rank of the MWF is reduced from $NM = 324$ to $K = 20$. The MWF is implemented using the conjugate gradient method with 200 iterations. The 200 iterations ensure that the reduced-rank MWF performance is as good as possible. Both schemes essentially fail for all reasonable SNR levels. On the other hand, both the JDL and the FFA methods are affected to a lesser extent. The performance of the JDL algorithm maintains an edge of about 11 dB over the regular FFA method and an edge of 10 dB over the overlapping FFA for a probability of detection of 0.5. The JDL also outperforms the randomized FFA by approximately 2.5 dB. Although the regular FFA does not perform quite as well as the JDL method in the airborne radar setup, it does demonstrate a greater immunity to reduced sample support as compared to its "parent" AMF algorithm. It is worth emphasizing that the parameters of the JDL algorithm are optimized using an exhaustive search, but those of the FFA schemes are not.

Example 3: The previous two examples showed that, with homogeneous training data, the JDL algorithm outperforms the FFA approach. Given the fact the JDL algorithm is also computationally efficient, these examples raise the question as to why the FFA algorithm would be useful. To motivate use of the FFA algorithm, the following examples use measured HFSWR data.

This example develops a test for the measured ionospheric HFSWR clutter data similar to the detection probability plot used in the first two examples. Using a measured data set does not allow for an adequate number of independent realizations to form a true 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. Small or negative values of ΔMSMI indicate potential false alarms. This process is repeated for all 93 range cells with ionospheric clutter, and the results are averaged. The parameters used are the following:

- JDL: three angle bins, three 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: $N_{\text{DoF}} = 16$, $N_{\text{iter}} = 1$; that is, iterations over random interleavings are not used.

Fig. 8 plots the results for the JDL and FFA methods. We emphasize that the abscissa here is the amplitude of the injected target in dB and not an SNR. This is because the noise level in the data is unknown, as are scale factors introduced in the measurement process. 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. Again, the randomized FFA

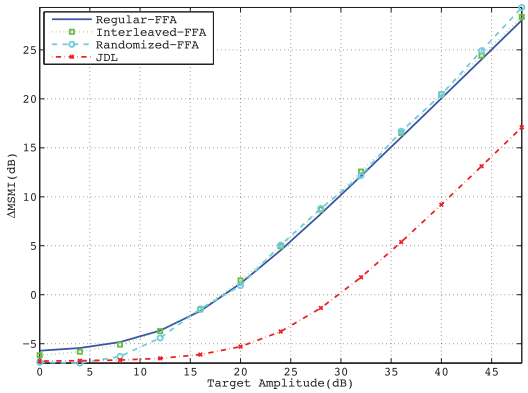

Fig. 9. ΔMSMI versus target amplitude for the JDL and FFA algorithms reduced sample support scenarios ($K = 20$).

scheme shows the best performance with about a 4.5-dB gap from the regular and overlapping FFA schemes, which show similar performance.

Example 4: 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 20 range bins are chosen symmetrically about the range cell of interest after allowing for a single-range bin guard zone on either side.

The algorithm parameters utilized in this simulation are the following:

- JDL: three angle bins, three 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], and 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: $N_{\text{DoF}} = 10$ in the first two stages followed by $N_{\text{DoF}} = 9$ in the following stages. $N_{\text{iter}} = 1$; that is, DoF are not reused.

The results of this simulation are shown in Fig. 9. 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 counterintuitive, we attribute the behavior of the JDL algorithm to the nonhomogeneity of the ionospheric clutter across wider range spans. Over shorter spans, such as 20 range cells, reduced nonhomogeneity leads to a more accurate estimate of the interference covariance for the JDL algorithm and, in turn, to an improvement in performance.

Interestingly, the FFA methods seem to behave in the opposite manner; that is, they seem to be less sensitive to the nonhomogeneity 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 15 dB. The randomized FFA enters its linear region at approximately 7 dB, while JDL enters the linear region at about 23 dB. In the high-SNR region, all three FFA schemes show similar performance for the reduced sample support scenario and outperform JDL by approximately 12.5 dB. Note that in this case, the randomized FFA scheme does not reuse adaptive DoF.

Figs. 8 and 9 allow for some speculation as to the source of the stability of the FFA algorithms in nonhomogeneous clutter scenarios. While each individual "small" AMF implementation within the larger FFA scheme is severely impacted by clutter nonhomogeneities, the figures suggest that the several combinations and recombinations average out this impact, making the final result robust.

B. MSMI Statistic Versus Range

In previous publications using measured data (e.g., [5]), a popular approach to algorithm testing is plots of the detection statistic versus range. In this section, we inject range-spread 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. As in Examples 3 and 4 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.

In this example, we used $K = 30$ secondary data cells with a target magnitude chosen such that nonadaptive processing cannot detect the inject target. The algorithm parameters used are the following:

- JDL: three angle bins, three 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], and 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: $N_{\text{DoF}} = 10$ in the first two stages followed by $N_{\text{DoF}} = 9$ in the following stages. $N_{\text{iter}} = 1$; that is, data are not reused.

The results are shown in Figs. 10 and 11. Fig. 10 confirms that nonadaptive processing does not detect the target; note that the JDL approach is unable to clearly identify the target (though the plot does show that the corresponding MSMI statistic is largest at the correct location). On the other hand, the three FFA approaches provide significant discrimination between the target and surrounding clutter. In particular, the regular FFA approach yields a ΔMSMI of almost 12 dB, while randomized FFA provides discrimination of about 16 dB.

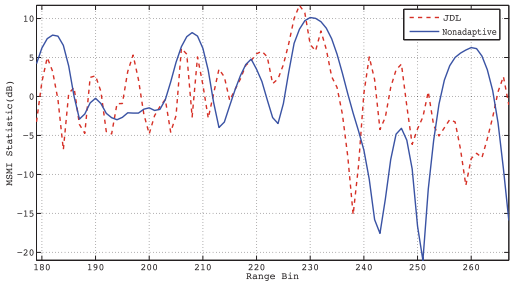

Fig. 10. MSMI versus range plots for using the nonadaptive and JDL methods to detect a 45-dB real target in ionospheric clutter region.

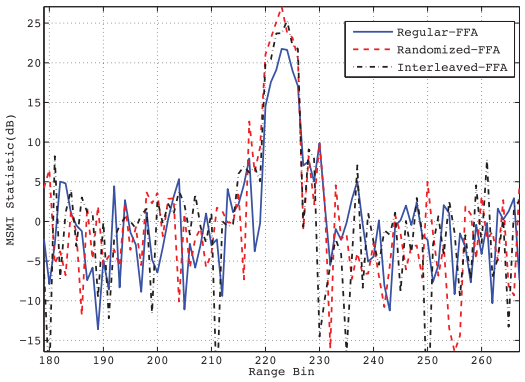

Fig. 11. MSMI versus range plots for using the three FFA methods to detect a 45-dB real target in ionospheric clutter region.

C. Detection With Measured Data Exclusively

In the previous example, we showed that the FFA approaches provide better discrimination against injected targets. In this example, we remove this final artificiality and work with measured data exclusively. Using HFSWR data provided, our results compare the performance of the nonadaptive and regular FFA approaches. Here, in contrast to the previous example, we focus on the ranges dominated by sea clutter. This is because we have access to target detections within this region using the nonadaptive processor (coupled with a tracker) [17].[1] For target detection in sea clutter, we focus on spatial adaptive processing only.

Example 1: The data provided comprise 1024 pulses and cover 150 measured data cubes from 0:07 a.m. to 6:05 a.m. on March 31, 2010. Here we examine 10 consecutive data cubes measured from 5:25 a.m. to 6:00 a.m. since these data cubes seemed to include the most targets. Based on the tracking data provided, there are 36 targets in these data cubes. Table III lists the targets in the first data cube that are detected *using nonadaptive processing* between look angles of 0° and 10°. All these are fairly strong targets. This table provides the information in terms of the range cell number, Doppler frequency, and azimuth angle of the targets with respect to the antenna array. False positives (false alarms) are reduced by declaring a target present only if it is detected in 10 consecutive data cubes. Details of how to pick the secondary data are given in [17].

Figs. 12 and 13 plot the MSMI statistics, as a function of range and Doppler, after nonadaptive processing and regular FFA processing, respectively. As is clear, the FFA scheme is better at suppressing interference and false alarms. The adaptive scheme finds significantly fewer

[1] The authors would like to gratefully acknowledge the contribution of these data sets by Raytheon Canada.

TABLE III
List of Targets in the First Data Cube That Are Detected at the Look Angles Between 0° and 10°

Target ID	Track Range Number	Doppler Frequency (Hz)	Azimuth Angle (°)
1	83	−0.0610	8.7
2	55	−0.0839	−1.6
3	7	0.0916	5.0
4	15	0.0343	8.9
5	69	−0.0572	−5.4
6	74	−0.0648	−0.1
7	14	−0.0267	−2.2

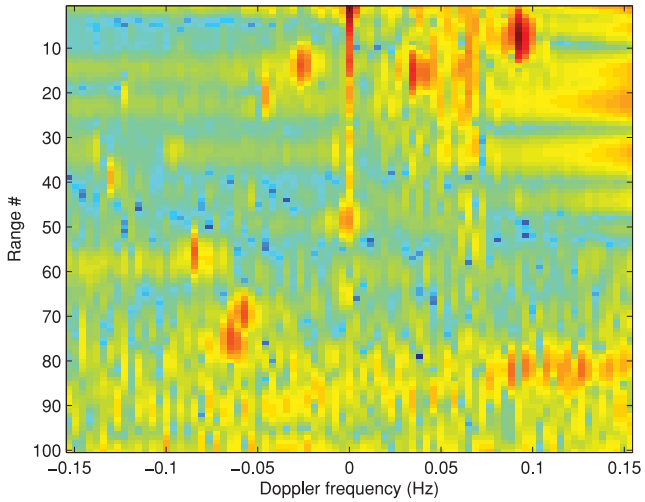

Fig. 12. Nonadaptive statistic versus range and Doppler frequency.

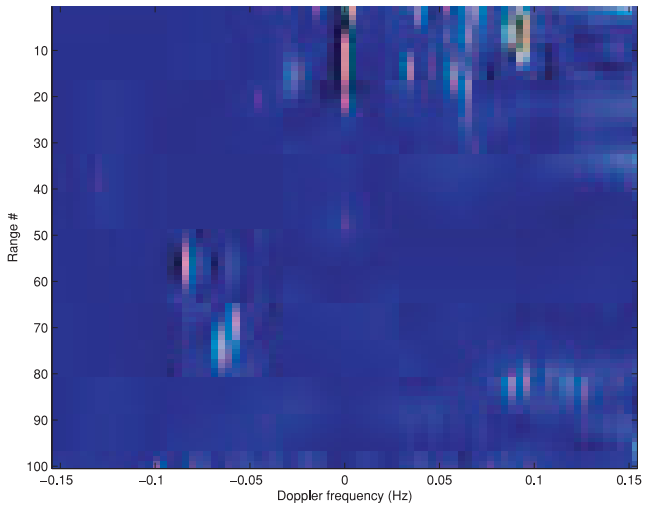

Fig. 13. MSMI statistic versus range and Doppler using modified approach.

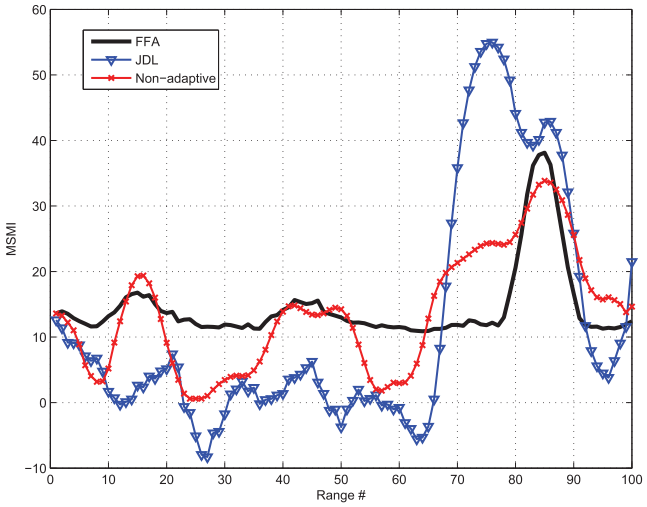

Fig. 14. MSMI statistic versus range at angle and Doppler frequency of the first target listed in Table III.

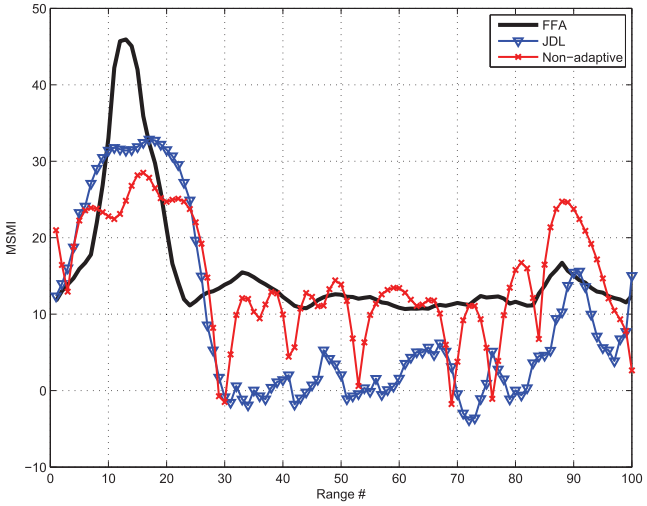

Fig. 15. MSMI statistic versus range at angles and Doppler frequency of a target not detected by nonadaptive processing

targets. To underline this advantage gained, Fig. 14 plots the MSMI statistic versus range at Doppler frequencies of the first target in Table III. For comparison with another STAP algorithm, the plot also shows the results of using the JDL algorithm. As is clear from the figure, the nonadaptive and JDL approaches lead to false alarms and a broad target spread in range, while the FFA method shows just one peak at the target range cell. Essentially, the FFA scheme provides far better discrimination against the sea clutter. These figures, therefore, again illustrate the performance improvements associated with the FFA algorithm when using measured data.

Example 2: While the previous example focused on targets detected by nonadaptive processing, this example illustrates the ability of the FFA approach to detect and isolate missed targets. Fig. 15 plots the MSMI statistic as a function of range for the look angle and Doppler of a target not identified by nonadaptive processing. We note both the improved discrimination between target and surrounding clutter statistics but also the sharper peak in range. This is consistent with Fig. 14 as well.

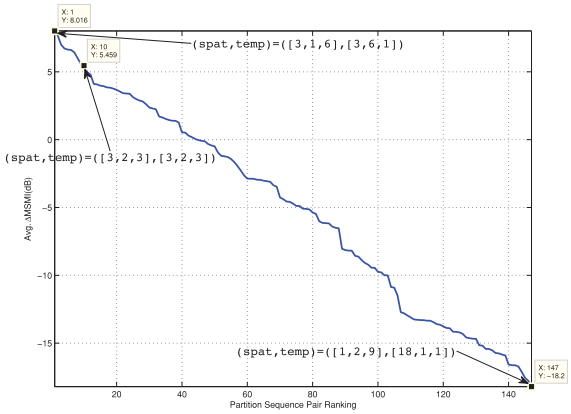

Fig. 16. ΔMSMI versus sequence ranking in the airborne scenario for a 30 dB ideal target using the regular FFA method for a sample support size of 20 secondary samples.

D. FFA Parameter Selection

The performance of the regular FFA method is characterized by three parameters: the spatial-partitioning sequence, the temporal-partitioning sequence, and the available sample support. The choice of spatial and temporal sequences (as well as their ordering) can significantly impact performance of the FFA method.

To illustrate effect of the partition sequences, we performed an exhaustive search across all possible spatiotemporal partitioning sequences of an airborne data cube with $N = 18$ elements and $M = 18$ pulse. For each partition sequence, we injected a 30 dB ideal target into the data cube at a specific range and computed the ΔMSMI associated with this partitioning sequence. We averaged the results of each partitioning sequence tested over 20 independent data cube instances. Since we used only 20 secondary samples to estimate the interference covariance for each look range, we will consider only the sequences whose required sample support requirement is less than or equal to 20 samples, that is, $2N'M' \leq 20$.

Fig. 16 plots the average ΔMSMI versus the sequence ranking for all possible spatiotemporal sequences with a secondary sample support of $K \leq 20$. The sequences are ranked in decreasing order of performance. Note that such a plot is possible only for scenarios where both M and N are relatively small and have only a few factors. The figure clearly indicates that the performance of the regular FFA method is sensitive to the choice of partitioning sequences selected, with the difference between the average ΔMSMI of the best- and worst-performing sequences as high as 26 dB.

Importantly, most of the poor sequences were found to have multiple ones in either the spatial or the temporal partition sequence; that is, several poor sequences are adaptive in either the spatial or the temporal direction only. Interestingly, the intuitive sequence chosen in the examples of Section V-A provides close-to-optimal

performance. Essentially, like with the JDL algorithm, wherein the implementation requires the choice of multiple algorithm parameters, the FFA algorithm performance can suffer if a poor parameter choice is made. However, as with the JDL algorithm (wherein a 3×3 LPR has shown to provide good performance in a variety of settings), intuitive choices of parameters in the FFA algorithm provide good performance as well.

In Section III-D, we proposed a randomized partitioning scheme as a more generalized form of the regular FFA approach. Simulation results in Section V indicated that the randomized FFA outperforms the regular and overlapping FFA schemes under all the simulation scenarios tested. We also mentioned in Section III-D that the randomized partitioning scheme could also yield an FFA algorithm more stable than that using the regular partitioning scheme. In this section, we will attempt to verify this claim through simulation.

To illustrate the randomized FFA's greater immunity to the choice of the spatiotemporal partitioning scheme, we used $N_{\text{DoF}} = [9,9,4]$ and $N_{\text{iter}} = 1$, that is, no DoF recycling. The injected target has an absolute amplitude of 30 dB but otherwise has the same characteristics as in Section V-A. The results were obtained for 4000 independent simulations, each using a different random permutation of the indices of the data matrix at every depth. Thus, although a fixed partition length sequence is used for all the iterations, each iteration combined the available adaptive DoF at every depth differently. The obtained standard deviation of the ΔMSMI statistics computed over all the iterations is as little as 2.5 dB. This result indicates that the randomized FFA is relatively insensitive to the manner in which the adaptive DoF are grouped into the various partitions at each depth for a fixed partition sequence.

E. Discussion

As seen in the examples presented in Sections V-A and V-B, the performance of the algorithms available in the literature is as expected with statistically homogeneous data available for training. The AMF and MWF schemes perform extremely well when adequate training is available, and their performance degrades severely with very limited training. On the other hand, schemes such as the JDL algorithm perform very well in both scenarios. The FFA schemes introduced in this paper follow a pattern similar to the JDL algorithm. While performance suffers when the available training is reduced, this loss is not as severe as for the AMF scheme. In our testing, the JDL algorithm performs on par with the regular and randomized FFA schemes when its parameters are optimized and homogeneous training is available.

The key advantage of the divide-and-conquer approach underlying the FFA schemes arises with the measured data with sea and ionospheric clutter, interference well accepted as being inherently nonhomogeneous. This is confirmed by the fact that the performance of the JDL algorithm can actually improve with reduced sample support; the data are homogeneous over short ranges but not over longer ranges.

In detecting targets buried in ionospheric clutter, the FFA algorithms significantly outperform the JDL approach. In the examples presented, the randomized FFA scheme *without recycling adaptive DoF* provides the best performance. This suggests that the use of multiple "smaller" AMF processes within the FFA multistage structure and combining their outputs average out the impact of nonhomogeneities. This is made possible by the fact that, at each stage, the impact of the processing on the original space-time steering vector is accounted for. This allows for the intermediate outputs to be combined coherently such that the target components add up but the interference components average out.

VI. CONCLUSION

In this paper, we introduced the FFA STAP algorithm that uses a divide-and-conquer strategy to significantly reduce 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.

The performance of the FFA schemes was tested using simulated airborne radar data and measured HFSWR data. The performance of the FFA scheme was comparable to the previously available JDL scheme in the case of the simulated data. This is because training used in the case of simulated data is statistically homogeneous. In the case of nonhomogeneous 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 nonhomogeneities. In using purely measured data, with sea clutter, the FFA approach significantly outperforms both the nonadaptive processor and the JDL approach.

The complexity analysis presented in this paper suggests that 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 problem.

ACKNOWLEDGMENT

The work of Saleh, Ravan, and Adve was supported by funding from Defence Research and Development Canada.

REFERENCES

[1] Ward, J.
Space-time adaptive processing for airborne radar.

MIT Lincoln Laboratory, Cambridge, MA, Tech. Rep. F19628-95-C-0002, Dec. 1994.

[2] Wicks, M., Rangaswamy, M., Adve, R., and Hale, T.
Space-time adaptive processing: a knowledge-based perspective.
IEEE Signal Processing Magazine, **23**, 1 (Jan. 2006), 51–65.

[3] Reed, I. S., Mallett, J., and Brennan, L.
Rapid convergence rate in adaptive arrays.
IEEE Transactions on Aerospace and Electronic Systems, **10**, 6 (Nov. 1974), pp. 853–863.

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

[5] Adve, R. S., Hale, T. B., and Wicks, M. C.
Joint domain localized adaptive processing in homogeneous and non-homogeneous environments. part I: homogeneous environments.
IEE Proceedings on Radar Sonar and Navigation, **147**, 2 (Apr. 2000), 57–65.

[6] Roman, J., Rangaswamy, M., Davis, D., Zhang, Q., Himed, B., and Michels, J.
Parametric adaptive matched filter for airborne radar applications.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (2000), 677–692.

[7] Wang, P., Li, H., and Himed, B.
Knowledge-aided parametric adaptive matched filter with automatic combining for covariance estimation.
IEEE Transactions on Signal Processing, **62**, 9 (Sep. 2014), 4713–4722.

[8] Brown, R., Schneible, R., Wicks, M., Wang, H., and Zhang, Y.
STAP for clutter suppression with sum and difference beams.
IEEE Transactions on Aerospace and Electronic Systems, **36**, 2 (Feb. 2000), 634–646.

[9] Yang, E., Chun, J., and Adve, R.
A hybrid D3-Sigma Delta STAP algorithm in non-homogeneous clutter.
In *Proceedings of the IEEE International Radar Conference*, Edinburgh, UK, 2007, pp. 1–5.

[10] Wang, X., Aboutanios, E., and Amin, M.
Reduced-rank STAP for slow-moving target detection by antenna-pulse selection.
IEEE Signal Processing Letters, **22**, 8 (Aug. 2015), 1156–1160.

[11] Degurse, J., Savy, L., and Marcos, S.
Reduced-rank STAP for target detection in heterogeneous environments.
IEEE Transactions on Aerospace and Electronic Systems, **50**, 2 (Feb. 2014), 1153–1162.

[12] Pillai, S. U., and Kwon, B. H.
Forward/backward spatial smoothing for coherent signal identification.
IEEE Transactions on Acoustics, Speech and Signal Processing, **37**, 1 (Jan. 1989), 8–15.

[13] Pillai, S. U., Kim, Y. L., and Guerci, J. R.
Generalized forward/backward subaperture smoothing techniques for sample starved STAP.
IEEE Transactions on Signal Processing, **48**, 12 (Dec. 2000), 3569–3574.

[14] Kelly, E.
An adaptive detection algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **22** (Jul. 1986), 115–127.

[15] Goldstein, J., Scharf, L., and Reed, I.
A multistage representation of the Wiener filter based on orthogonal projections.
IEEE Transactions on Information Theory, **44**, 7 (Jul. 1998), 2943–2959.

[16] Chan, H. C.
Characterization of ionospheric clutter in HF surface-wave radar.
Defense Research and Development Canada, Ottawa, Tech. Rep. DRDC Ottawa TR 2003-114, Sep. 2003.

[17] Ravan, M., and Adve, R.
Robust STAP for HFSWR in dense target scenarios with nonhomogeneous clutter.
In *Proceedings of the 2012 IEEE Radar Conference*, Atlanta, GA, Apr. 2012.

Oliver Saleh has close to a decade of experience in the big data industry working for leaders in their domain, such as Microsoft, Blackberry, and Ooyala, as well as experience in the start-up scene. He is an expert in big data technologies, distributed systems, signal processing, and machine learning techniques. He holds an M.A.Sc. degree in electrical engineering from the University of Toronto with specialization in radar signal processing.

Maryam Ravan (M'10–SM'12) received her Ph.D. degree from Amirkabir University of Technology, Tehran, Iran. She was a postdoctoral fellow with the Departments of Electrical and Computer Engineering at University of Toronto, McMaster University, and Ryerson University, Ontario, Canada, from May 2007 to April 2013, where she was involved in solving forward modeling and inverse problems and the related signal/image processing techniques for biomedical, radar systems, and nondestructive testing (NDT) applications. In the field of biomedical engineering, her focus was on development of brain source localization and machine learning approaches for investigating the effectiveness of medications in patients with depression and schizophrenia and also the state of brain developments in infants. In the field of radar systems, her research was on providing a reliable space-time adaptive processing to detect targets in nonhomogeneous ionospheric and ocean clutters, and in the field of NDT techniques, her focus was on developing inversion techniques to estimate the shape of three-dimensional arbitrary defects. She was also a lecturer in the School of Computational Engineering and Science, McMaster University, Hamilton, Ontario, Canada, from 2009 to 2012. She is currently a senior research scientist at LivaNova PLC (formerly Cyberonics), Houston, Texas, where her work focuses on efficacy of closed-loop vagus nerve stimulation therapy for epilepsy. Her research interests include signal and image processing, beamforming, machine learning, microwave holography, MIMO radar systems and space-time adaptive processing, and nondestructive testing. Dr. Ravan has authored/coauthored over 50 journal and conference papers and a book chapter.

Ryan Riddolls received his B.Sc. (1997), M.Eng. (1999), and Ph.D. (2003) degrees, all in electrical engineering, from the Massachusetts Institute of Technology in Cambridge, Massachusetts, USA. Since 2004, he has been a defense scientist at Defence Research and Development Canada in Ottawa, Ontario, Canada, which is an agency of the Canadian Department of National Defence. He is currently the principal investigator for the Canadian over-the-horizon radar program. His interests are in signal processing, electronics, and plasma physics.

Raviraj S. Adve (S'88–M'97–SM'06) was born in Bombay, India. He received his B.Tech. in electrical engineering from IIT, Bombay, in 1990 and his Ph.D. from Syracuse University in 1996. His thesis received the Syracuse University Outstanding Dissertation Award. Between 1997 and August 2000, he worked for Research Associates for Defense Conversion Inc. on contract with the Air Force Research Laboratory at Rome, New York. He joined the faculty at the University of Toronto in August 2000, where he is currently a professor. Dr. Adve's research interests include analysis and design techniques for heterogeneous networks, energy harvesting networks, and, in signal processing, techniques for radar and sonar systems. He received the 2009 Fred Nathanson Young Radar Engineer of the Year Award.

DOCUMENT CONTROL DATA

(Security markings for the title, abstract and indexing annotation must be entered when the document is Classified or Designated)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g., Centre sponsoring a contractor's report, or tasking agency, are entered in Section 8.)

IEEE Transactions on Aerospace and Electronic Systems

2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)

CAN UNCLASSIFIED

2b. CONTROLLED GOODS

NON-CONTROLLED GOODS
DMC A

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Fast Fully Adaptive Processing: A Multistage STAP Approach

4. AUTHORS (last name, followed by initials – ranks, titles, etc., not to be used)

Saleh, Oliver; Ravan, Maryam; Riddolls, Ryan; Adve, Raviraj;

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)	6b. NO. OF REFS (Total cited in document.)
December 2017	16	17

7. DESCRIPTIVE NOTES (The category of the document, e.g., technical report, technical note or memorandum. If appropriate, enter the type of report, e.g., interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

External Literature (P)

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

DRDC – Ottawa Research Centre
Defence Research and Development Canada
3701 Carling Avenue
Ottawa, Ontario K1A 0Z4
Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC-RDDC-2017-P121	

11a. FUTURE DISTRIBUTION (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Public release

11b. FUTURE DISTRIBUTION OUTSIDE CANADA (Any limitations on further dissemination of the document, other than those imposed by security classification.)

12. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

13. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g., Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Space-time adaptive processing, reduced rank STAP, fast fully adaptive processing, multistage STAP, high frequency surface wave radar