

## Two-Step Low-Complexity Space-Time Adaptive Processing (STAP)

Pun, M-O.; Sahinoglu, Z.; Shah, S.; Hara, Y.; Wang, P.

TR2010-119 December 06, 2010

### Abstract

This work proposes a low-complexity space-time adaptive processing (STAP) algorithm for sensing applications built on a moving platform in the presence of strong clutters. The proposed algorithm achieves low-complexity computation via two steps. First, it utilizes improved fast approximated power iteration methods to compress the data into a much smaller subspace. To further reduce the computational complexity, a progressive singular value decomposition (SVD) approach is employed to update the inverse of the covariance matrix of the compressed data. As a result, the proposed low complexity STAP algorithm can achieve order-of-magnitude computational complexity reduction as compared to conventional STAP algorithms. Simulation results are shown to confirm the validity of the proposed algorithm.

*IEEE Global Telecommunications Conference (GLOBECOM)*

© 2010 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



## TWO-STEP LOW-COMPLEXITY SPACE-TIME ADAPTIVE PROCESSING (STAP)

Man-On Pun\*, Zafer Sahinoglu\*, Sagar Shah†, Yoshihisa Hara‡ and Pu Wang§

\*Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA

†Dept. of Electrical and Computer Eng., Drexel University, Philadelphia, PA 19104, USA

‡Mitsubishi Electric Corporation, 325 Kamimachiya, Kamakura Kanagawa 247-8520, Japan

§ Dept. of Electrical and Computer Eng., Stevens Institute of Tech., Hoboken, NJ 07030, USA.

### ABSTRACT

This work proposes a low-complexity space-time adaptive processing (STAP) algorithm for sensing applications built on a moving platform in the presence of strong clutters. The proposed algorithm achieves low-complexity computation via two steps. First, it utilizes improved fast approximated power iteration methods to compress the data into a much smaller subspace. To further reduce the computational complexity, a progressive singular value decomposition (SVD) approach is employed to update the inverse of the covariance matrix of the compressed data. As a result, the proposed low-complexity STAP algorithm can achieve order-of-magnitude computational complexity reduction as compared to conventional STAP algorithms. Simulation results are shown to confirm the validity of the proposed algorithm.

### 1. INTRODUCTION

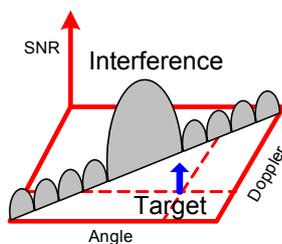


Fig. 1. Illustration of clutter interference spectrum.

In sensing applications built on a moving platform, returned signals are commonly contaminated by clutter interference of different incoming angles and Doppler frequencies as shown in Fig. 1. To accurately detect moving targets, effective clutter suppression techniques have become indispensable. Among many techniques developed for clutter suppression in the literature, space-time adaptive processing (STAP) has emerged as one of the most promising techniques [1]. In STAP, returned signals are filtered *simultaneously* over both

space and time domains. As a result, clutter interference can be effectively removed regardless of its incoming angle and Doppler frequency. However, the conventional STAP suffers from prohibitively expensive computational complexity. Denote by  $M$  and  $N$  the number of pulses and antennas, respectively. The conventional STAP requires intensive matrix inversion of dimension  $MN \times MN$ . For practical systems with  $MN$  on the order of hundreds, such a matrix inversion requirement renders impractical the implementation of STAP.

To circumvent this obstacle, considerable research efforts have been devoted to developing low-complexity STAP. According to Brennan's rule, the rank of the clutter interference covariance matrix is known to be much smaller than  $MN$ . Thus, one way to achieve complexity reduction is to compress the return signal into its signal subspace of rank  $r \ll MN$  without incurring any information loss. In particular, [2] has recently developed a low-complexity STAP scheme by exploiting a subspace tracking algorithm called fast approximated power iteration (FAPI) [3]. It was demonstrated in [3] that FAPI can be employed to effectively compress the return signal into a much smaller signal subspace, which enables low-complexity STAP operating on the compressed data.

This work proposes a two-step low-complexity STAP scheme in which the first step develops *modified* FAPI schemes with improved convergence speed before applying the modified schemes to compress the return data. Next, taking advantage of progressive singular vector decomposition (PSVD) [4], we propose a PSVD-based low-complexity technique to compute the inverse of covariance matrix of the compressed data recursively. The resulting low-complexity STAP reduces the computational complexity from  $\mathcal{O}((MN)^3)$  to  $\mathcal{O}((MN)r)$ . Finally, simulation results are shown to demonstrate that the proposed STAP achieves near-optimal performance as compared to the conventional STAP using full matrix inversion.

**Notation:** Vectors and matrices are denoted by boldface letters.  $\|\cdot\|$  represents the Euclidean norm of the enclosed vector and  $|\cdot|$  denotes the amplitude of the enclosed complex-valued quantity.  $\mathbf{I}_N$  is the  $N \times N$  identity matrix. Furthermore, we use  $(\cdot)^H$  for Hermitian transposition. Finally,

$[A]_{i,j}$  denotes the  $i$ -th row and  $j$ -th column entry of matrix  $A$  whereas  $A(j, :)$  the  $j$ -th column of  $A$ .

## 2. SIGNAL MODEL

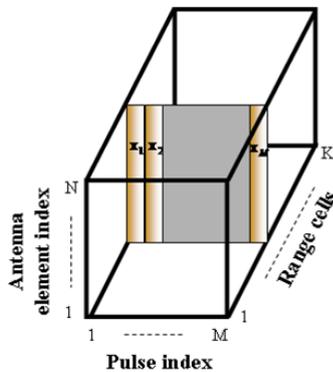


Fig. 2. Illustration of return signal structure over range cells.

Received signals are arranged in range cells as shown in Fig. 2. STAP first constructs data vectors  $\mathbf{x}(k)$  of length  $MN$  by stacking up the samples collected over  $M$  pulses from each antenna in the  $k$ -th range cell, where  $k = 1, 2, \dots, K$ . To examine if a target is present in the  $k$ -th range cell, a clutter-plus-noise covariance matrix  $C(k)$  of dimension  $MN \times MN$  is computed from its neighboring range cells assuming that these neighboring range cells are impaired by the same clutters and yet target-free. Denote by  $\Omega_k$  the index set of range cells used to compute  $C(k)$ . Thus,  $C(k)$  can be expressed as

$$C(k) = \frac{1}{|\Omega(k)|} \sum_{\ell \in \Omega(k)} \mathbf{x}(\ell)\mathbf{x}(\ell)^H, \quad (1)$$

where  $|\cdot|$  stands for the cardinality of the enclosed quantity.

It has been shown that clutter can be optimally suppressed by [1]:

$$\mathbf{z}(k) = C(k)^{-1}\mathbf{x}(k). \quad (2)$$

Upon obtaining  $\mathbf{z}(k)$ , target detection algorithms can be performed. Despite the good performance of (2), the matrix inversion  $C(k)^{-1}$  incurs prohibitively expensive computation. To cope with this problem, different subspace-tracking algorithms have been proposed to first reduce the dimension of  $\mathbf{x}(k)$  before performing the matrix inversion. Denote by  $\mathbf{W}$  of dimension  $MN \times r$  the subspace concentration matrix, where  $\text{rank}(C(k)) < r \ll MN$ . The compressed signal after subspace concentration process can be mathematically expressed as:

$$\mathbf{y}(k) = \mathbf{W}^H \mathbf{x}(k), \quad (3)$$

with  $\mathbf{W}$  is given by the following optimization function

$$\mathbf{W} = \arg \min_{\tilde{\mathbf{W}}} \left\| \mathbf{y}(k) - \tilde{\mathbf{W}} \tilde{\mathbf{W}}^H \mathbf{x}(k) \right\|^2. \quad (4)$$

The optimization problem in (4) can be numerically solved with subspace-tracking algorithms such as FAPI [2].

Next, target detection can be similarly performed after the compressed signal is filtered with

$$\mathbf{r}(k) = \mathbf{R}(k)^{-1} \mathbf{y}(k). \quad (5)$$

where  $\mathbf{R}(k)$  is the corresponding *compressed* clutter-plus-noise covariance matrix and it reads

$$\mathbf{R}(k) = \frac{1}{|\Omega(k)|} \sum_{\ell \in \Omega(k)} \mathbf{y}(\ell)\mathbf{y}(\ell)^H. \quad (6)$$

It is worth noting that  $\mathbf{R}(k)$  is of dimension  $r \times r$ , which is significantly smaller than  $C(k)$ .

In the following section, we first develop a modified FAPI with expedited convergence behavior by taking into account noise impairment. Furthermore, a progressive SVD approach is proposed to compute  $\mathbf{R}(k)^{-1}$  from  $\mathbf{R}_{k-1}^{-1}$  in a low-complexity recursive fashion by exploiting the correlation between  $\mathbf{R}(k)^{-1}$  and  $\mathbf{R}_{k-1}^{-1}$  for  $k > 1$ .

## 3. PROPOSED SCHEME

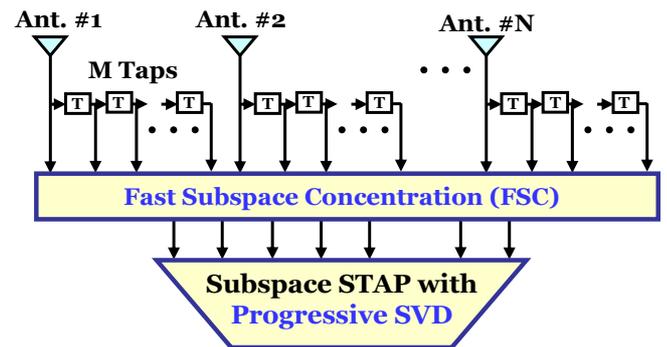


Fig. 3. Illustration of the proposed low-complexity two-step STAP.

Figure 3 depicts the proposed two-step STAP comprised of the fast subspace concentration step and the progressive SVD step. In the following, we will discuss two steps in details. Since the proposed scheme is an iterative algorithm, we use  $\mathbf{W}(k)$  to denote  $\mathbf{W}$  updated with the  $(k-1)$ -th input vector  $\mathbf{x}(k)$  for  $k = 1, 2, \dots, K$ .

### 3.1. Modified fast approximated power iteration (MFAPI)

The fast approximated power iteration (FAPI) algorithm developed in [3] is an efficient approximation of the conventional projection approximation subspace tracking (PAST) algorithm proposed in [5]. By exploiting the approximation of  $\mathbf{W}(k) \approx \mathbf{W}(k-1)$ , FAPI can reduce the computational complexity of PAST from  $\mathcal{O}(NM^2r)$  to  $\mathcal{O}(3NM^2r + 5r^2)$ .

However, the derivation of FAPI does not *explicitly* take into account the impact of additive noise. As a result, its performance degrades as signal-to-noise ratio (SNR) decreases. More specifically, FAPI is derived from the approximated power iteration (API) algorithm [6]. In API, the auxiliary matrix  $\mathbf{Z}$  is updated as

$$\mathbf{Z}(k) = \frac{1}{\beta} \Theta(k)^H [\mathbf{I}_r - \mathbf{g}(k)\mathbf{y}(k)^H] \mathbf{Z}(k-1)\Theta(k)^{-H}, \quad (7)$$

where

$$\Theta(k) = \mathbf{W}(k-1)^H \mathbf{W}(k), \quad (8)$$

and  $\mathbf{g}(k)$  of length  $r$  has the same definition as that in [3].

It is important to observe that the last term in (7),  $\Theta(k)^{-H}$ , not only incurs  $\mathcal{O}(r^3)$  operation but also may enhance noise if  $\Theta(k)$  is noisy. Motivated by this observation, we propose the following two modifications of (7). Recalling that  $\Theta$  is nearly orthonormal [3], it is reasonable to approximate  $\Theta(k)^{-H}$  as

$$\Theta(k)^{-H} \approx \Theta(k). \quad (9)$$

As a result, (7) becomes

$$\mathbf{Z}(k) = \frac{1}{\beta} \Theta(k)^H [\mathbf{I}_r - \mathbf{g}(k)\mathbf{y}(k)^H] \mathbf{Z}(k-1)\Theta(k). \quad (10)$$

Note that (10) has the same computational complexity as (7). Further computation reduction can be achieved by observing that  $\mathbf{W}$  comprises of orthonormal column vectors. Hence, we can approximate  $\Theta(k)^{-H} \approx \mathbf{I}_r$  in (7) and  $\mathbf{Z}(k)$  takes the following form:

$$\mathbf{Z}(k) = \frac{1}{\beta} \Theta(k)^H [\mathbf{I}_r - \mathbf{g}(k)\mathbf{y}(k)^H] \mathbf{Z}(k-1). \quad (11)$$

It should be pointed out that (11) has  $\mathcal{O}(r^3)$  less operation as compared to (7) and (10).

Following the same procedures proposed in [3], we can derive the modified FAPI (MFAPAI) by incorporating (10) and (11). After some straightforward algebraic manipulations, we can find the update functions for  $\mathbf{Z}(k)$  using (10) and (11) are given as follows, respectively.

$$\mathbf{Z}(k) = \frac{1}{\beta} (\mathbf{Z}(k-1) - \mathbf{g}(k)\mathbf{h}'(k) - \epsilon(k)\mathbf{g}(k)^H), \quad (12)$$

and

$$\mathbf{Z}(k) = \frac{1}{\beta} (\mathbf{Z}(k-1) - \mathbf{g}(k)\mathbf{h}'(k)), \quad (13)$$

where the definitions of  $\mathbf{h}'(k)$  and  $\epsilon(k)$  are the same as those in [3] and given in Algorithm 1.

In the sequel, the MFAPAI algorithms employing (12) and (13) are referred to as the Noise-Robust MFAPAI (NR-MFAPAI) and Low Complexity MFAPAI (LC-MFAPAI). The total computational complexity of NR-MFAPAI and LC-MFAPAI is  $\mathcal{O}(3NMr + 5r^2)$  and  $\mathcal{O}(3NMr + 3r^2)$ , respectively.

The proposed NR/LC-MFAPAI algorithms are summarized in Algorithm 1.

---

### Algorithm 1 Modified fast approximated power iteration (NR/LC-MFAPAI)

---

**Initialization:**  $\mathbf{W}(0) = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{0}_{(n-r) \times r} \end{bmatrix}$ ,  $\mathbf{Z}(0) = \mathbf{I}_r$ .

**Procedure:**

- 1: **for**  $k = 1, 2, \dots, K$  **do**
  - 2:    $\mathbf{y}(k) = \mathbf{W}(k-1)^H \mathbf{x}(k)$
  - 3:    $\mathbf{h}(k) = \mathbf{Z}(k-1)\mathbf{y}(k)$
  - 4:    $\mathbf{g}(k) = \frac{\mathbf{h}(k)}{\beta + \mathbf{y}(k)^H \mathbf{h}(k)}$
  - 5:    $\epsilon^2(k) = \|\mathbf{x}(k)\|^2 - \|\mathbf{y}(k)\|^2$
  - 6:    $\tau(k) = \frac{\epsilon^2(k)}{1 + \epsilon^2(k)\|\mathbf{g}(k)\|^2 + \sqrt{1 + \epsilon^2(k)\|\mathbf{g}(k)\|^2}}$
  - 7:    $\eta(k) = 1 - \tau(k)\|\mathbf{g}(k)\|^2$
  - 8:    $\mathbf{y}'(k) = \eta(k)\mathbf{y}(k) + \tau(k)\mathbf{g}(k)$
  - 9:    $\mathbf{h}'(k) = \mathbf{Z}(k-1)^H \mathbf{y}'(k)$
  - 10:   NR-MFAPAI:  
      $\epsilon(k) = \frac{\tau(k)}{\eta(k)} (\mathbf{Z}(k-1)\mathbf{g}(k) - (\mathbf{h}'(k)^H \mathbf{g}(k)) \mathbf{g}(k))$   
     and (12) or
  - LC-MFAPAI: (13)
  - 11:    $\mathbf{e}'(k) = \eta(k)\mathbf{x}(k) - \mathbf{W}(k-1)\mathbf{y}'(k)$
  - 12:    $\mathbf{W}(k) = \mathbf{W}(k-1) + \mathbf{e}'(k)\mathbf{g}(k)^H$
  - 13: **end for**
- 

### 3.2. Progressive SVD (PSVD)

Despite the fact that the output of the subspace concentration,  $\mathbf{y}(k)$ , has a much smaller dimension as compared to  $\mathbf{x}(k)$ , computation of  $\mathbf{R}(k)^{-1}$  in (5) for  $k = 1, 2, \dots, K$  can remain computationally expensive. To cope with this problem, it is important to observe that  $\mathbf{R}(k)$  and  $\mathbf{R}(k-1)$  are correlated. This is because they are derived from some common compressed data vectors and clutter variation between consecutive pulse intervals is correlated. Thus motivated, we propose a progressive SVD (PSVD) approach [4] to compute  $\mathbf{R}(k)^{-1}$  in terms of  $\mathbf{R}(k-1)^{-1}$  and  $\Delta\mathbf{R}(k) = \mathbf{R}(k) - \mathbf{R}(k-1)$ , assuming  $\mathbf{R}(k-1)^{-1}$  is given and  $\text{rank}(\Delta\mathbf{R}(k)) \ll r$ . Upon obtaining  $\mathbf{R}(k)^{-1}$ , the same procedures can be repeated to derive  $\Delta\mathbf{R}(k+1)$  recursively.

The low-rank assumption of  $\Delta\mathbf{R}(k)$  allows us to decompose it into the following form:  $\Delta\mathbf{R}(k) = \sum_{d=1}^{D_k} \alpha_d \mathbf{q}_d \mathbf{q}_d^H$ ,

where  $D_k \stackrel{\text{def}}{=} \text{rank}(\Delta\mathbf{R}(k))$ . Furthermore,  $\mathbf{q}_d$  and  $\alpha_d$  are the eigenvectors and the associated eigenvalues, respectively, with  $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_{D_k}$ . To achieve low-complexity computation, we propose to employ the following rank-one approximation to decompose  $\Delta\mathbf{R}(k)$ :

$$\Delta\mathbf{R}(k) \approx \mathbf{a}\mathbf{a}^H. \quad (14)$$

where  $\mathbf{a} = \sqrt{\alpha_1} \mathbf{q}_1$ .

Finally, assuming that  $\mathbf{R}(1)^{-1}$  is given and  $\mathbf{R}(1)$  can be decompose as

$$\mathbf{R}(1) = \mathbf{U}\mathbf{S}\mathbf{U}^H, \quad (15)$$

the rank-one PSVD algorithm is given in Algorithm 2.

---

**Algorithm 2** Rank-one Progressive SVD (PSVD)

---

**Initialization:**  $\mathbf{R}(1) = \mathbf{U}\mathbf{S}\mathbf{U}^H$

**Procedure:**

- 1: **for**  $k = 2, 3, \dots, K$  **do**
- 2: Obtain  $\mathbf{R}(k)$  according to (6);
- 3: Compute  $\Delta\mathbf{R}(k) = \mathbf{R}(k) - \mathbf{R}(k-1)$ ;
- 4: Decompose  $\Delta\mathbf{R}(k) \approx \mathbf{a}\mathbf{a}^H$ ;
- 5: Obtain  $\mathbf{p}$  and  $\gamma$  such that

$$\gamma\mathbf{p} = (\mathbf{I} - \mathbf{U}\mathbf{U}^H)\mathbf{a};$$

- 6: Construct an  $(r+1) \times (r+1)$  diagonal matrix  $\mathbf{K}$  whose  $n$ -th diagonal element is given by

$$[\mathbf{K}]_{n,n} = \left[ \begin{array}{cc} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array} \right] + \left[ \begin{array}{c} \mathbf{U}^H \mathbf{a} \\ \gamma \end{array} \right] \left[ \begin{array}{c} \mathbf{U}^H \mathbf{a} \\ \gamma \end{array} \right]^H \Bigg]_{n,n},$$

for  $n = 1, 2, \dots, r+1$ .

- 7: Denote by  $\mathcal{I}_k = \{i_j\}$  the index set satisfying  $[\mathbf{K}]_{i_n, i_n} \leq [\mathbf{K}]_{i_m, i_m}$  for  $1 \leq n < m \leq r+1$ .
- 8: Define two auxiliary matrices:  $\tilde{\mathbf{U}} \stackrel{\text{def}}{=} [\mathbf{U} \ \mathbf{p}]$  and a  $m \times m$  diagonal matrix  $\mathbf{\Lambda}$  with  $[\mathbf{\Lambda}]_{n,n} = [\mathbf{K}]_{i_n, i_n}$ ;
- 9: Update

$$\mathbf{U} \stackrel{\text{def}}{=} \tilde{\mathbf{U}}(i_{1:m}, :),$$

where  $m$  is a design parameter with  $1 \leq m \leq r+1$ ;

- 10: Finally, compute

$$\mathbf{R}(k)^{-1} = \mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^H.$$

- 11: **end for**
- 

Note that the computation complexity of Algorithm 2 is  $\mathcal{O}((3+2m)r^2)$ , as compared to  $\mathcal{O}(r^3)$  for direct matrix inversion of  $\mathbf{R}(k)^{-1}$ . It should be emphasized that, rather than (14), higher-rank approximation of  $\Delta\mathbf{R}(k)$  may lead to better approximation accuracy at the price of higher computational complexity. As shown in Sec. 4, simulation results suggest that rank-one approximation in (14) is usually sufficient to result in satisfactory performance.

### 3.3. Computational Complexity

The total computational complexity of the proposed two-step STAP algorithm is thus given by  $\mathcal{O}(3NM r + (8+2m)r^2)$  and  $\mathcal{O}(3NM r + (5+2m)r^2)$  for PSVD in conjunction with NR-MFAPI and LC-MFAPI, respectively. Clearly, this stands for a substantial computational reduction as compared to the full matrix inversion  $\mathbf{C}(k)^{-1}$  required in (2), particularly for practical values of  $N$  and  $M$ .

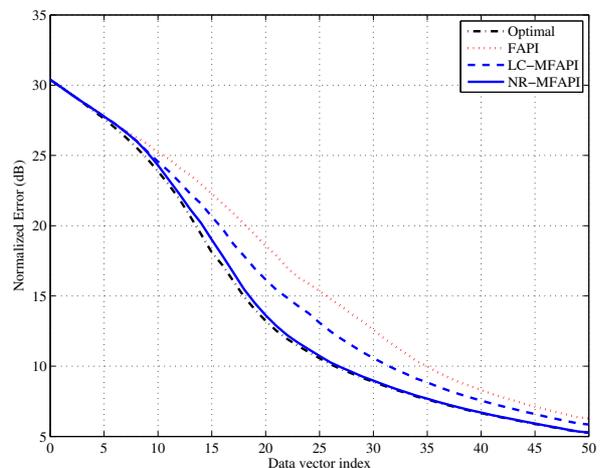
## 4. SIMULATION RESULTS

In this section, computer simulation is performed to verify the performance of the proposed algorithm. Unless otherwise specified, we use  $N = 8$ ,  $M = 8$  and  $K = 50$  in our following simulation. From the Brennan's rule, the rank of the clutter covariance matrix is about 15. Thus, it is reasonable to argue that setting  $r = 35$  is sufficient to retain most of the signal information contained in the original  $NM = 64$ -dimensional space. Furthermore, we set parameters such as pulse repetition frequency (PRF) and so on according to Table 2.1 in [1] and fix the look angle and platform mobile speed at  $\varphi_L = \frac{\pi}{3}$  and 90 m/s, respectively. Finally, we consider target-free clutter signal corrupted by additive white Gaussian noise (AWGN) with clutter-to-noise power ratio (CNR) given by  $\rho$ .

We first investigate the performance of the proposed MFAPI in terms of the normalized error defined as:

$$J(\mathbf{W}) = \frac{1}{K} \sum_{k=1}^K \|\mathbf{y}(k) - \mathbf{W}\mathbf{W}^H \mathbf{x}(k)\|^2. \quad (16)$$

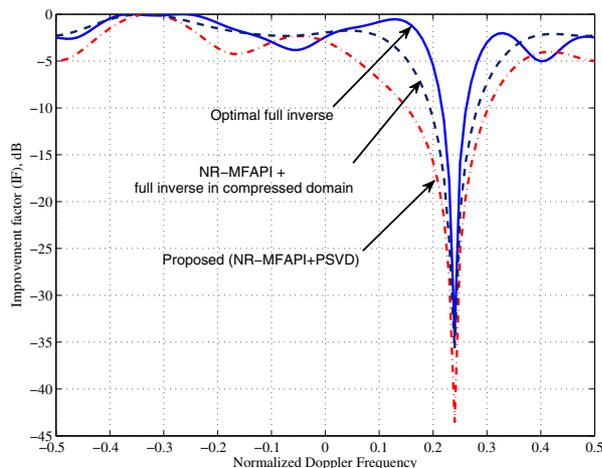
Fig. 4 shows the convergence behavior of different subspace-tracking algorithms as a function data vector index at  $\rho = 20$  dB. Inspection of Fig. 4 suggests that NR-MFAPI approaches the optimal curve derived from direct orthonormalization whereas LC-MFAPI outperforms FAPI in terms of convergence rate. After updating with 50 data vectors, all algorithms converge to approximately the same signal subspace.



**Fig. 4.** Convergence behavior as a function data vector index at CNR of 20 dB.

Next, we provide a holistic performance comparison using MFAPI in conjunction with PSVD. We first employ the improvement factor (IF) as the performance metric. IF is designed to measure the ratio of signal-to-noise ratio at the output and input [1]. Fig. 5 depicts the IF as a function of the normalized Doppler frequency normalized with respect to PRF. For the given look angle and platform mobile

speed, the clutter has a normalized Doppler frequency of about 0.2. Fig. 5 shows that the proposed two-step low-complexity algorithm using NR-MFAPI and PSVD has comparable performance as compared to the optimal algorithm using full inversion of  $C(k)$  (i.e. without compression). Furthermore, comparison between the performance curves corresponding to the proposed algorithm and the algorithm using NR-MFAPI to compressed the data but full inversion of  $R(k)$  indicates that the PSVD incurs marginal performance degradation. However, it should be borne in mind that the proposed algorithm achieves the comparable performance with an order-of-magnitude complexity reduction as compared to the other two algorithms.

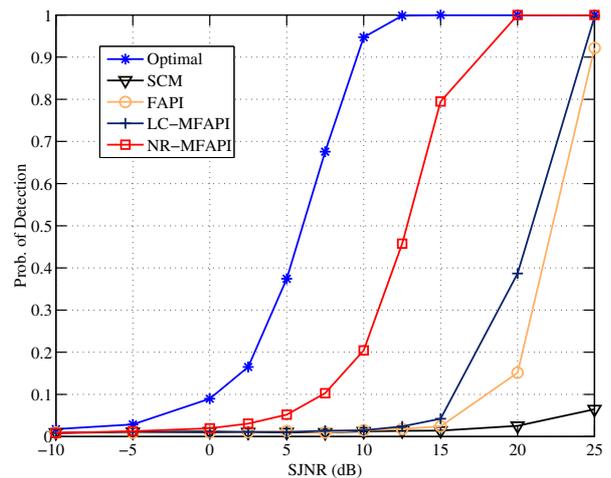


**Fig. 5.** Performance comparison in terms of improvement factor (IF).

Finally, we compare the performance of different STAP algorithms in terms of probability of detection as a function of signal-to-jamming-noise ratio (SJNR) [1]. In this simulation, we set  $M = 4$ ,  $N = 16$ ,  $K = 64$  and  $r = 49$ . Furthermore, two jamming signals are simulated at angles of  $-40^\circ$  and  $25^\circ$ , respectively. Figure 6 shows that the conventional sample covariance matrix (SCM)-based method suffers from poor performance due to insufficient data as well as jamming. In contrast, the proposed NR-MFAPI exhibits good performance with about 5 dB degradation with respect to (w.r.t.) the optimal detector using the true clutter-and-jamming covariance matrix whereas the LC-MFAPI and FAPI have similar performance of 15 dB degradation w.r.t. the optimal.

## 5. CONCLUSION

In this work, we proposed a low-complexity space-time adaptive processing (STAP) algorithm for sensing applications built on a moving platform in the presence of strong clutters. The proposed algorithm achieves low-complexity computation via two steps. It first utilizes improved fast approximated



**Fig. 6.** Probability of detection obtained with different STAP algorithms.

power iteration methods to compress the data into a much smaller subspace followed by updating the inverse of the covariance matrix of the compressed data in a progressive singular value decomposition approach. As a result, the proposed low-complexity STAP algorithms can achieve order-of-magnitude computational complexity reduction as compared to conventional STAP algorithms. Simulation results have confirmed the validity of the proposed STAP algorithms.

## 6. REFERENCES

- [1] R. Klemm, *Principles of space-time adaptive processing*, 3rd ed. The Institution of Engineering and Technology, London, 2006.
- [2] H. Belkacemi and S. Marcos, "Fast iterative subspace algorithms for airborne STAP radar," *EURASIP Journal on applied signal processing*, pp. 1–8, Jul. 2006.
- [3] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Trans. Signal Proc.*, vol. 53, pp. 2931 – 2941, Aug. 2005.
- [4] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra and its Applications*, vol. 415, pp. 20–30, May 2006.
- [5] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Proc.*, vol. 43, pp. 95 – 107, Jan. 1995.
- [6] R. Badeau, G. Richard, and B. David, "Approximated power iterations for fast subspace tracking," in *Proc. 7th International Symp. Signal Processing and Applications*, Paris, France, Jul. 2003, pp. 583–586.