

Registration-Based Range-Dependence Compensation for Bistatic STAP Radars

Fabian D. Lapiere

*Department of Electrical Engineering and Computer Science, University of Liege, Building B28,
Sart Tilman, 4000 Liege, Belgium
Email: f.lapiere@fastmail.fm*

Jacques G. Verly

*Department of Electrical Engineering and Computer Science, University of Liege, Building B28,
Sart Tilman, 4000 Liege, Belgium
Email: jacques.verly@ulg.ac.be*

Received 30 December 2003; Revised 18 June 2004

We address the problem of detecting slow-moving targets using space-time adaptive processing (STAP) radar. Determining the optimum weights at each range requires data snapshots at neighboring ranges. However, in virtually all configurations, snapshot statistics are range dependent, meaning that snapshots are nonstationary with respect to range. This results in poor performance. In this paper, we propose a new compensation method based on registration of clutter ridges and designed to work on a single realization of the stochastic snapshot at each range. The method has been successfully tested on simulated, stochastic snapshots. An evaluation of performance is presented.

Keywords and phrases: radar, bistatic, space-time adaptive processing, range-dependence compensation, direction-Doppler curves.

1. INTRODUCTION

Space-time adaptive processing (STAP) is an increasingly popular radar signal processing technique for detecting slow-moving targets in the presence of clutter and jammers [1, 2]. The space dimension arises from the use of an array of N antenna elements and the time dimension from the use of a coherent train of M pulses. The power of STAP comes from the *joint* processing in space and time. STAP radars operate either in monostatic configuration, where the transmitter and receiver are colocated, or in bistatic configuration, where the transmitter and receiver are located on distinct, independently moving platforms.

The data collected by a STAP radar can be viewed as a sequence of $M \times N$ 2D arrays, typically treated as $MN \times 1$ vectors. These arrays or vectors are called “snapshots.” Implementing the optimum STAP processor generally involves inverting the covariance matrix (CM) of the snapshots. This matrix must be estimated using snapshots at neighboring ranges. A major problem for virtually all STAP configurations is that the snapshots’ statistics are not stationary with respect to (w.r.t.) range. One of the most visible manifestations of this is the deformation with range of the 2D clutter power spectrum (PS), where the spectral dimensions

correspond to spatial and Doppler frequencies. Ignoring the lack of stationarity and computing the sample CM by straight averaging of single-sample CMs at neighboring ranges results in a loss of performance. The lack of stationarity of the snapshots w.r.t. range and the series of related issues are referred to in STAP as the “range-dependence (RD) problem.”

Various techniques have been developed to deal with the RD problem. The main ones are Doppler warping [3, 4], angle-Doppler compensation (both deterministic and adaptive) [5, 6], derivative-based updating [4], and registration-based compensation [7, 8, 9]. These methods are briefly reviewed in Section 2.

In [7], we introduced the new concept of registration-based compensation in STAP, but assumed that the (theoretical) CM of the snapshots was known at each range. In contrast, in the present paper, we assume that all we have is a single stochastic realization of the snapshot at each range. This paper shows how to modify the algorithms of [7] so they continue to perform under the new conditions. The work reported here is based exclusively on simulated, stochastic snapshots. Of course, the ultimate test is real data, but this is not considered here.

Figure 1 gives a preview of the level of complexity considered in this paper, as compared to that of [7]. Figure 1a

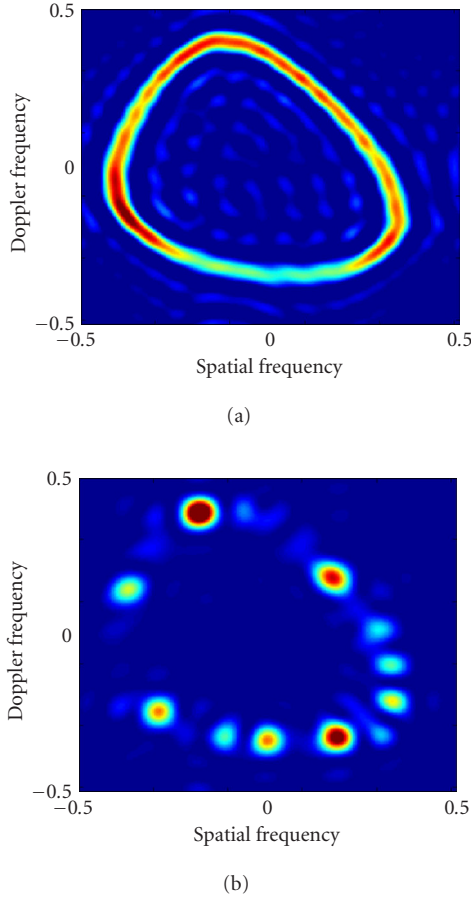


FIGURE 1: Example of (clutter) PS at one range. (a) Expected value of periodogram computed from theoretical CM. (b) Periodogram computed from one realization of a stochastic snapshot.

shows the “interference + noise (I + N)” PS computed at a specific range for a known CM of the I + N snapshots at that range. The “clutter ridge” appears complete and smooth. Its appearance is “deterministic,” that is, fixed for any given configuration and range. Figure 1b shows the corresponding I + N PS estimated from a single realization of the stochastic I + N snapshot at the same range. The clutter ridge has broken down into an “archipelago of small ridges.” Its appearance is “stochastic,” even for a given configuration and range.

Whereas the algorithms in [7] were designed to work on *deterministic* PS such as in Figure 1a, the algorithms discussed here are designed to work on *stochastic* PS such as in Figure 1b. Below, we present our new RD compensation method and discuss its performance.

2. STATE OF THE ART IN RANGE-DEPENDENCE COMPENSATION

Doppler warping was initially developed for monostatic configurations [3] and later applied to bistatic configurations [4]. It applies the appropriate Doppler shift at each range to bring the clutter ridges at all ranges into registration at a specific spatial frequency. Even if this method is simple to imple-

ment, the configuration parameters must be known and the compensation is perfect at only one spatial frequency. (Adaptive) angle-Doppler compensation [5, 6] generalizes Doppler warping by applying an angle-Doppler shift, determined at each range, to bring the spectral center at that range into registration with the spectral center at some reference range. All the above methods apply only to directive sensors.

Derivative-based updating [10] was applied to bistatic configurations in [4]. The idea is to expand the optimum weight as a function of range using a Taylor series expansion limited to 1st order. Even if this method can be used in a wide variety of situations, the number of degrees of freedom is doubled, implying that the number of samples required for estimation is also doubled.

In [7], we proposed registration-based RD compensation methods that compute the sample CM at some reference range gate l by averaging properly transformed single-sample CMs at a series of neighboring range gates k . The single-sample CM at each range k is transformed to bring the corresponding clutter ridge at k into registration with the clutter ridge at l . This registration is guided by analytical formulas describing the “direction-Doppler (DD) curves” [8]. In [7], we distinguished between “true-parameters (TP)” methods, which assume exact knowledge of the configuration parameters, and “estimated-parameters (EP)” methods, which estimate the parameters from the data. The methods in each class rely on a common “registration-based compensation (RBC)” module. The EP methods rely on an additional “configuration-parameters estimation (CPE)” module.

3. BISTATIC RADAR CONFIGURATION

The configuration geometry is shown in Figure 2. The transmitter T is at the center of an (x, y, z) coordinate system. The x -axis points in the same direction as the velocity vector \underline{v}_T of T . The z -axis points vertically and up. The receiver R is located at (x_R, y_R, z_R) . Its velocity vector \underline{v}_R is assumed to be horizontal and to make an angle α_R w.r.t. \underline{v}_T . The linear antenna A is located at R . It is horizontal and makes an angle δ w.r.t. \underline{v}_R . The (bistatic) range R_b to some scatterer S is the distance from T to S to R . The angular positions of S w.r.t. \underline{v}_T , \underline{v}_R , and \underline{s} are denoted by ξ_d^T , ξ_d^R , and ξ_s , respectively. The ground is assumed to be a horizontal plane at $z = -H$. All scatterers corresponding to ground clutter are thus located in this plane. The magnitudes of \underline{v}_T and \underline{v}_R are denoted by v_T and v_R . Any bistatic configuration is fully characterized by the vector of (configuration) parameters

$$\underline{\theta} = (H, x_R, y_R, z_R, v_T, v_R, \alpha_R, \delta). \quad (1)$$

4. DIRECTION-DOPPLER CURVES AND SURFACES

4.1. Important parameters

The three important physical parameters associated with each scatterer S are the range R_b , the angular position ξ_s , and the relative velocity v_r . The related parameters that are more directly measured from the radar returns are the roundtrip

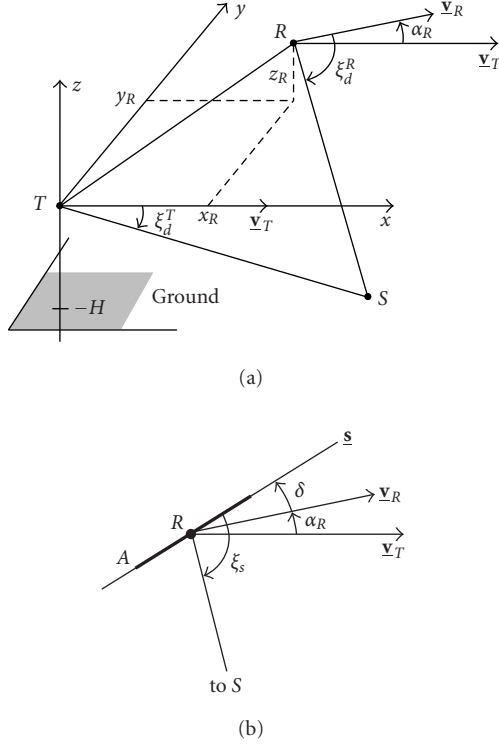


FIGURE 2: Bistatic radar configuration. (a) Transmitter (T)-receiver (R)-scatterer (S) geometry and related parameters. (b) Receiver antenna (A) and related angles.

delay τ_{rt} , the spatial frequency f_s , and the Doppler frequency f_d . For a stationary scatterer, we have $\tau_{rt} = R_b/c$, $f_s = \cos \xi_s/\lambda_c$, and $f_d = \nu_r/\lambda_c$, where λ_c is the carrier wavelength and c is the speed of light. R_b , ξ_s , and ν_r are easily derived from τ_{rt} , f_s , and f_d .

4.2. Isorange curves

All scatterers S characterized by the same range R_b are located on an isorange surface, which is an ellipsoid of revolution with foci at T and R . The intersection of this surface with the ground is an isorange curve, which is an ellipse (parameterized with polar angle ψ).

4.3. 2D direction-Doppler curves

For any given configuration and range, all stationary scatterers S at this range map onto a curve showing the relation between f_s and f_d for any such S . This curve is called a DD curve. DD curves are typically represented in terms of the normalized spatial frequency ν_s , equal to $(\lambda_c/2)f_s$, and the normalized Doppler frequency ν_d , equal to $(\lambda_c/2(\nu_r + \nu_T))f_d$. Figure 3 shows that bistatic DD curves vary significantly with configuration and range. The variation of these curves with range for any particular configuration is one of the most visible manifestations of the RD problem.

To derive the equations of bistatic DD curves, we proceed as follows. First, we express ν_d as a function of ν_s . Since most DD curves have 2 distinct values of ν_d for most ν_s 's, any DD

curve is best described by 2 functions $\nu_d = f_1(\nu_s)$ and $\nu_d = f_2(\nu_s)$, which describe the “bottom” and “top” parts of the curve, respectively.¹ Second, if we express ν_s and ν_d in terms of the angle ψ , which also parameterizes the isorange ellipse, we find a parametric description of the DD curve, that is, $\nu_s = g_1(\psi)$ and $\nu_d = g_2(\psi)$. The derivation of the $f_i(\nu_s)$'s and $g_i(\psi)$'s is lengthy and thus omitted.

4.4. 3D direction-Doppler surfaces

We call the surface obtained by stacking DD curves for successive values of R_b a DD surface. Figure 4 shows the DD surface for the configuration of Figure 3d.

4.5. Recovery of configuration parameters

Consider the DD surface \mathcal{S} corresponding to an arbitrary bistatic configuration $\underline{\theta}_1 = (H, x_R, y_R, z_R, \nu_R, \nu_T, \alpha_R, \delta)$ and to all applicable values of R_b . One can show that the only other set of parameters that produces \mathcal{S} is $\underline{\theta}_2 = (H, x_R, -y_R, z_R, \nu_R, \nu_T, -\alpha_R, -\delta)$. Thus, the inverse problem of recovering $\underline{\theta}$ from \mathcal{S} has 2 related solutions. We can then infer that the inverse problem of recovering $\underline{\theta}$ from a single slice of \mathcal{S} has at least 2 solutions and that at least 2 of them are related.

5. SNAPSHOTS

In each coherent processing interval, a coherent train of M pulses is transmitted from T . The returns are sensed at each of the N elements of the linear antenna array A at R . Finally, the sensed returns are sampled at a number of discrete ranges (called range gates) covering the range interval of interest. Ranges are indexed with $l \in \mathcal{L} = \{0, 1, \dots, L-1\}$. We regard the data as a sequence of $M \times N$ 2D data arrays (snapshots) at successive ranges l . The $M \times N$ snapshot corresponding to a specific l (or even to some arbitrary value of the continuous range) and to a single scatterer S characterized by specific parameters (R_b, ν_s, ν_d) can be written as the $MN \times 1$ vector [2]

$$\underline{\mathbf{y}}(\nu_s, \nu_d) = \beta_r \underline{\mathbf{v}}(\nu_s, \nu_d), \quad (2)$$

where β_r is found from the radar equation and $\underline{\mathbf{v}}(\nu_s, \nu_d)$ is the $MN \times 1$ steering vector

$$\underline{\mathbf{v}}(\nu_s, \nu_d) = \underline{\mathbf{b}}(\nu_d) \otimes \underline{\mathbf{a}}(\nu_s), \quad (3)$$

where \otimes is the Kronecker product, $\underline{\mathbf{a}}(\nu_s)$ is the $N \times 1$ spatial steering vector, and $\underline{\mathbf{b}}(\nu_d)$ is the $M \times 1$ temporal steering vector. For uniform linear arrays, we have

$$\begin{aligned} \underline{\mathbf{a}}(\nu_s) &= (1 \cdots e^{j2\pi\nu_s n} \cdots e^{j2\pi\nu_s(N-1)})^T, \\ \underline{\mathbf{b}}(\nu_d) &= (1 \cdots e^{j2\pi\nu_d m} \cdots e^{j2\pi\nu_d(M-1)})^T. \end{aligned} \quad (4)$$

¹All curves have a bottom and a top, even when they appear flat, as in Figure 3a. The notions of top and bottom continue to be precisely defined for 8-shaped curves, as in Figure 3b. However, the top is not always above the bottom!

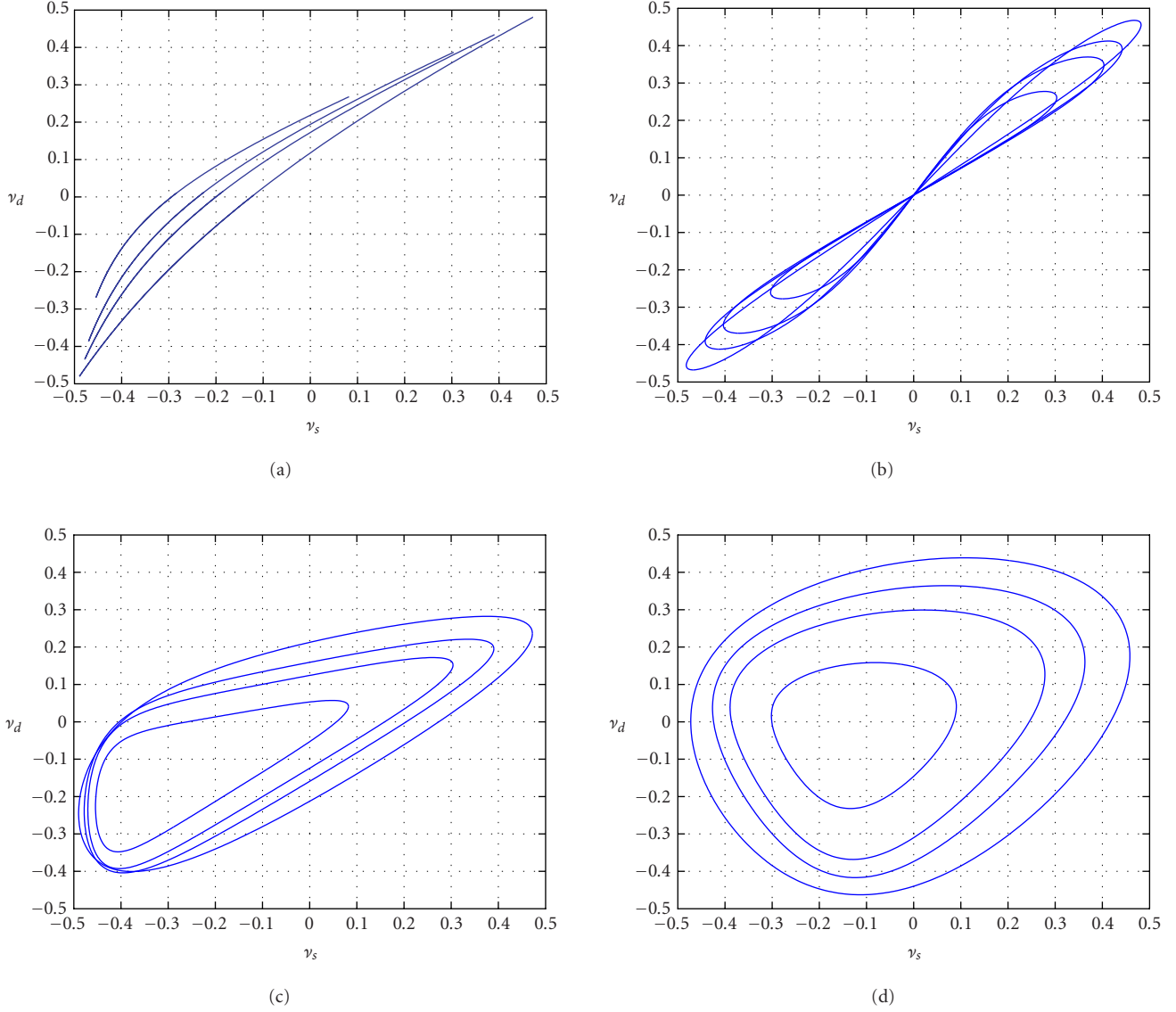


FIGURE 3: Example of DD curves for four bistatic configurations and four ranges R_b (170, 210, 250, and 400 km). Parameters specific to each configuration are (a) $(x_R, y_R, z_R) = (100, 0, 0)$ km, $\alpha_R = 0^\circ$, $\delta = 0^\circ$, (b) $(x_R, y_R, z_R) = (0, 100, 0)$ km, $\alpha_R = 0^\circ$, $\delta = 0^\circ$, (c) $(x_R, y_R, z_R) = (0, 100, 0)$ km, $\alpha_R = 90^\circ$, $\delta = 0^\circ$, and (d) $(x_R, y_R, z_R) = (80, 50, 20)$ km, $\alpha_R = 35^\circ$, $\delta = 60^\circ$. Common configuration parameters are $H = -50$ km and $v_R = v_T = 90$ m/s.

The target snapshot \underline{y}_t is directly given by (2), where we use the appropriate target parameters $(R_b^t, \nu_s^t, \nu_d^t)$. The clutter snapshot \underline{y}_c is found by integrating $\underline{y}(\nu_s, \nu_d)$ over the isorange curve, that is,

$$\underline{y}_c(\nu_s, \nu_d) = \int_0^{2\pi} \beta_c(\psi) \underline{y}(\nu_s(\psi), \nu_d(\psi)) d\psi. \quad (5)$$

Since $\beta_c(\psi)$ is a stochastic process, \underline{y}_c is a stochastic vector. We assume it is wide-sense stationary w.r.t. space and time. It is thus characterized by a constant CM $\underline{\mathbf{R}}_c = E\{\underline{y}_c \underline{y}_c^\dagger\}$. Jammer snapshots are not considered here. The unavoidable noise snapshot \underline{y}_n is assumed to be uncorrelated with \underline{y}_c and to be spatially and temporally white and indepen-

dent of range. It is thus characterized by a constant CM $\underline{\mathbf{R}}_n = E\{\underline{y}_n \underline{y}_n^\dagger\} = P_n \underline{\mathbf{I}}$, where P_n is the noise power. The important quantity in STAP is the I + N snapshot $\underline{y}_q = \underline{y}_c + \underline{y}_n$, characterized by the I + N CM

$$\underline{\mathbf{R}}_q = E\{\underline{y}_q \underline{y}_q^\dagger\}. \quad (6)$$

As alluded to earlier, the present paper differs from [7] primarily through the fact that, here, we use a set of simulated, stochastic I + N snapshots \underline{y}_q generated on the basis of (5), whereas, in [7], we use the theoretical I + N CM $\underline{\mathbf{R}}_q$ defined by (6) and derivable, in major part, from (5).

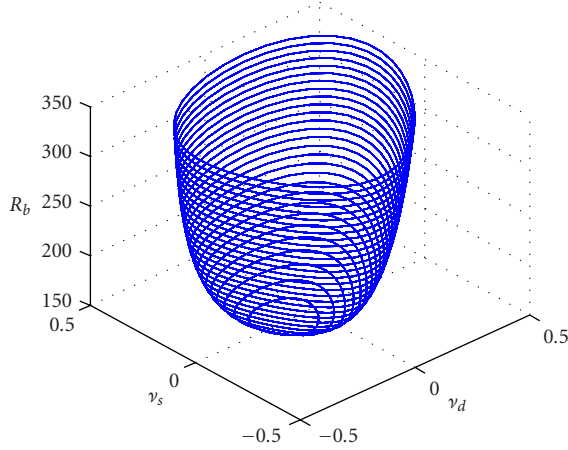


FIGURE 4: Example of DD surface for bistatic configuration of Figure 3d. Range R_b varies from 152 km to 350 km. Note that the minimum of 152 km is the smallest possible value for R_b for this configuration.

6. OPTIMUM PROCESSOR

The structure of the optimum processor (OP) is shown in Figure 5. The primary input is the snapshot $\underline{y} = \underline{y}(l)$ at range gate l (with range R_b). Secondary inputs are specific values of ν_s and ν_d and the theoretical $\underline{\mathbf{R}}_q$ at l . The triplet (R_b, ν_s, ν_d) constitutes the “target hypothesis.” The weight vector that maximizes the output signal-to-interference-plus-noise ratio (SINR) is given by [11]

$$\underline{\mathbf{w}}_o(\nu_s, \nu_d) = \alpha \underline{\mathbf{R}}_q^{-1} \underline{\mathbf{y}}(\nu_s, \nu_d), \quad (7)$$

where α is an arbitrary complex constant. (R_b appears through $\underline{\mathbf{R}}_q$.) The detection statistic is the complex scalar

$$z = \underline{\mathbf{w}}_o^\dagger(\nu_s, \nu_d) \underline{\mathbf{y}}. \quad (8)$$

Its magnitude is compared to a threshold λ to determine whether the target hypothesis is true or false.

$\underline{\mathbf{R}}_q$ in (7) is the theoretical I + N CM. In practice, $\underline{\mathbf{R}}_q$ has to be estimated from the data. This is discussed in the next section.

The performance of a processor using a weight $\underline{\mathbf{w}}$, whether optimal or not, is measured by the SINR loss defined as [2]

$$\text{SINR}_L = \frac{\text{SINR}}{\text{SINR}_0} = \frac{|\underline{\mathbf{w}}^\dagger \underline{\mathbf{y}}|^2}{(\underline{\mathbf{w}}^\dagger \underline{\mathbf{R}}_q \underline{\mathbf{w}})(\underline{\mathbf{y}}^\dagger \underline{\mathbf{y}})}, \quad (9)$$

where SINR_0 is the SINR in the absence of clutter. Optimum performance is achieved for $\underline{\mathbf{w}} = \underline{\mathbf{w}}_o$. Performance is degraded by losses due to the estimation of the I + N CM (even in the absence of RD effects) and to inappropriate handling of the RD problem.

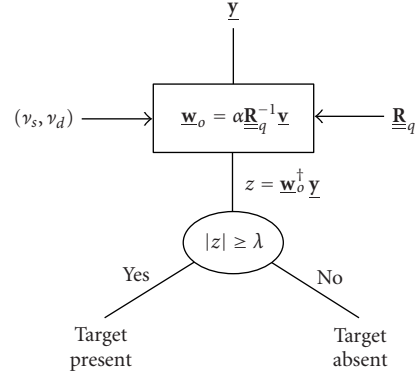


FIGURE 5: Structure of OP determining whether, for the given $\underline{y} = \underline{y}(l)$, the target hypothesis (R_b, ν_s, ν_d) is true or false.

7. ESTIMATION OF I + N COVARIANCE MATRIX $\underline{\mathbf{R}}_q$

At each range l , one might be tempted to estimate $\underline{\mathbf{R}}_q$ via [2]

$$\hat{\underline{\mathbf{R}}}_q(l) = \frac{1}{N_l} \sum_{k \in S_l} \tilde{\underline{\mathbf{R}}}_q(k) \quad \text{with} \quad \tilde{\underline{\mathbf{R}}}_q(k) = \underline{\mathbf{y}}_q(k) \underline{\mathbf{y}}_q^\dagger(k), \quad (10)$$

where N_l is the number of snapshots used for estimation, S_l is the set of snapshot indices k defined by $l - (N_l/2) < k < l + (N_l/2)$, and $\underline{\mathbf{y}}_q(k)$ is the snapshot at range k . (Observe that range l is omitted from the sum in (10).) However, $\hat{\underline{\mathbf{R}}}_q(l)$ is “maximum likelihood” *only if* the $\underline{\mathbf{y}}_q(k)$ ’s are independent and identically distributed (i.i.d.) w.r.t. range and have complex Gaussian probability density functions (identical for all k ’s) [12]. Unfortunately, in virtually all configurations, the $\underline{\mathbf{y}}_q(k)$ ’s are *not* i.i.d. w.r.t. range. One of the clearest manifestations of this lack of stationarity w.r.t. range is the variation with range of the clutter PS and, in particular, of the clutter ridge. All these facts and issues are the essence of the RD problem in STAP.

If no RD compensation is performed, the estimate $\hat{\underline{\mathbf{R}}}_q(l)$ in (10) provides reduced performance. The goal of any RD compensation method should be to produce the best possible estimate for $\underline{\mathbf{R}}_q(l)$ based on the available data and in spite of the nonstationarity w.r.t. range. One such method is described below.

8. PRINCIPLE OF OUR REGISTRATION-BASED RANGE-DEPENDENCE COMPENSATION METHOD

We perform RD compensation by applying a properly designed transformation $T_{kl}^R[\cdot]$ to each single-sample CM $\tilde{\underline{\mathbf{R}}}_q(k) = \underline{\mathbf{y}}_q(k) \underline{\mathbf{y}}_q^\dagger(k)$. Equation (10) then becomes

$$\hat{\underline{\mathbf{R}}}_q(l) = \frac{1}{N_l} \sum_{k \in S_l} \tilde{\underline{\mathbf{R}}}'_q(k) \quad \text{with} \quad \tilde{\underline{\mathbf{R}}}'_q(k) = T_{kl}^R[\tilde{\underline{\mathbf{R}}}_q(k)]. \quad (11)$$

At this point, $T_{kl}^R[\cdot]$ should be regarded as a conceptual transformation. Its precise meaning will become clear later. Specifically, one should not assume it is a linear-transformation matrix.

Since the nonstationarity is most visible in the spectral domain (via the deformation of the clutter ridge), we perform the RD compensation in this domain, using the clutter ridges at the various ranges as guides. The transformation $T_{kl}^R[\cdot]$ is designed to deform the clutter PS at each range $k \in S_l$ to bring its clutter ridge into registration with that of the PS at reference range l . Because of the direct relation between clutter ridges and DD curves, we can also think in terms of DD curves. The registration of DD curves is the key idea behind the method introduced in [7] and its generalization presented here. Note that [5] also applies an elementary form of registration, in that DD curves are brought into registration at their spectral centers, that is, at a single point only.

To implement $T_{kl}^R[\cdot]$ in the spectral domain, we need an estimate $\tilde{P}_k(U, V)$ of the true PS $P_k(U, V)$. Here, we use the periodogram so that [13]

$$\tilde{P}_k(U, V) = \frac{\mathbf{y}^\dagger(U, V) \tilde{\mathbf{R}}_q(k) \mathbf{y}(U, V)}{\mathbf{y}^\dagger(U, V) \mathbf{y}(U, V)}, \quad (12)$$

where $\mathbf{y}(\cdot, \cdot)$ is defined in (3). Appropriately combining (11) and (12), we find

$$\hat{P}_l(U, V) = \frac{1}{N_l} \sum_{k \in S_l} \frac{\mathbf{y}^\dagger(U, V) T_{kl}^R[\tilde{\mathbf{R}}_q(k)] \mathbf{y}(U, V)}{\mathbf{y}^\dagger(U, V) \mathbf{y}(U, V)} \quad (13)$$

or

$$\hat{P}_l(U, V) = \frac{1}{N_l} \sum_{k \in S_l} \tilde{P}_k(U, V), \quad (14)$$

where

$$\tilde{P}_k(U, V) = \frac{\mathbf{y}^\dagger(U, V) T_{kl}^R[\tilde{\mathbf{R}}_q(k)] \mathbf{y}(U, V)}{\mathbf{y}^\dagger(U, V) \mathbf{y}(U, V)}, \quad (15)$$

which we can write as follows:

$$\tilde{P}_k(U, V) = T_{kl}^P[\tilde{P}_k(U, V)]. \quad (16)$$

From (12), (15), and (16), it is clear that it would be extremely difficult to express $T_{kl}^P[\cdot]$ in terms of $T_{kl}^R[\cdot]$. Therefore, $T_{kl}^P[\cdot]$ should only be interpreted as a conceptual transformation, just as $T_{kl}^R[\cdot]$ should be.

$\tilde{P}_k(U, V)$ in (12) is also the 2D DTFT of the estimate $\tilde{\Gamma}_{q,k}[l_m, l_n]$ of the 2D statistical autocorrelation sequence of the snapshot $\mathbf{y}_q(k)$. Denoting by $y_{q,k}(m, n)$ the 2D $M \times N$ array representing the realization at hand of $\mathbf{y}_q(k)$, we have $\tilde{\Gamma}_{q,k} = y_{q,k} \star y_{q,k}^*$, where \star is the 2D correlation operator [13]. Taking the inverse 2D DTFT of (14), we have

$$\hat{\Gamma}_{\underline{q}}(l) = \frac{1}{N_l} \sum_{k \in S_l} \tilde{\Gamma}'_{\underline{q}}(k) \quad \text{with} \quad \tilde{\Gamma}'_{\underline{q}}(k) = T_{kl}^\Gamma[\tilde{\Gamma}_{\underline{q}}(k)], \quad (17)$$

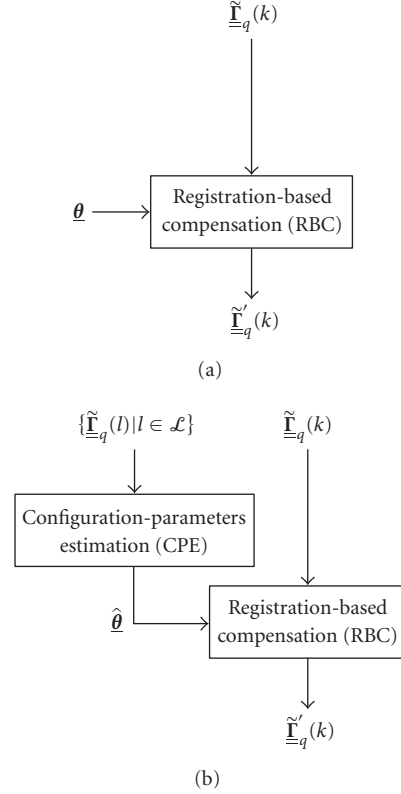


FIGURE 6: The subsystem for properly transforming the reduced-size CM $\tilde{\Gamma}_{\underline{q}}(k)$ into $\tilde{\Gamma}'_{\underline{q}}(k)$ can operate in (a) TP mode (for known θ) or (b) EP mode (for estimated $\hat{\theta}$). In both cases, the transformation is performed by the common RBC module. In the second case, the unknown configuration parameters are provided by the CPE module. Note that the current CPE module uses *all* inputs $\tilde{\Gamma}_{\underline{q}}(l)$, that is, for all $l \in \mathcal{L}$.

where $\tilde{\Gamma}_{\underline{q}}(k)$ is the matrix representing the central $(2M - 1) \times (2N - 1)$ part of $\tilde{\Gamma}_{q,k}[l_m, l_n]$. Here too, $T_{kl}^\Gamma[\cdot]$ is conceptual. Since the $(2M - 1) \times (2N - 1)$ $\tilde{\Gamma}_{\underline{q}}(k)$ carries exactly the same information as the $MN \times MN$ $\tilde{\mathbf{R}}_q(k)$, it is computationally advantageous to perform the compensation using $\tilde{\Gamma}_{\underline{q}}(k)$ rather than $\tilde{\mathbf{R}}_q(k)$.

The transformation $T_{kl}^\Gamma[\cdot]$ is not given in analytical form, but as an algorithm operating on the input $\tilde{\Gamma}_{\underline{q}}(k)$ and producing the output $\tilde{\Gamma}'_{\underline{q}}(k)$. Figure 6 shows two possible implementations of $T_{kl}^\Gamma[\cdot]$, depending upon whether we know the configuration-parameter vector θ or we have to estimate it. These modes of operation are respectively called TP and EP.

Figure 6 highlights the presence of two key processing modules: RBC and CPE. The RBC module is used in both modes of operation and transforms $\tilde{\Gamma}_{\underline{q}}(k)$ using either the true θ or its estimate $\hat{\theta}$. The CPE module is used only in the EP mode of operation to produce the estimate $\hat{\theta}$ from the data. Note that, contrary to the approach discussed in [7], *all* $\tilde{\Gamma}_{\underline{q}}(l)$'s, $l \in \mathcal{L}$, are used to estimate θ .

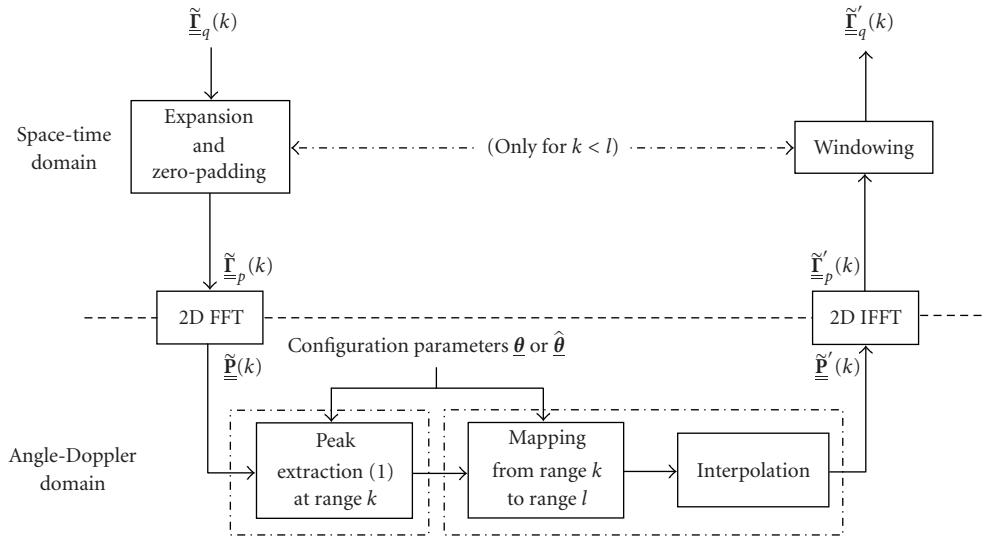


FIGURE 7: Block diagram of processing steps for RBC module.

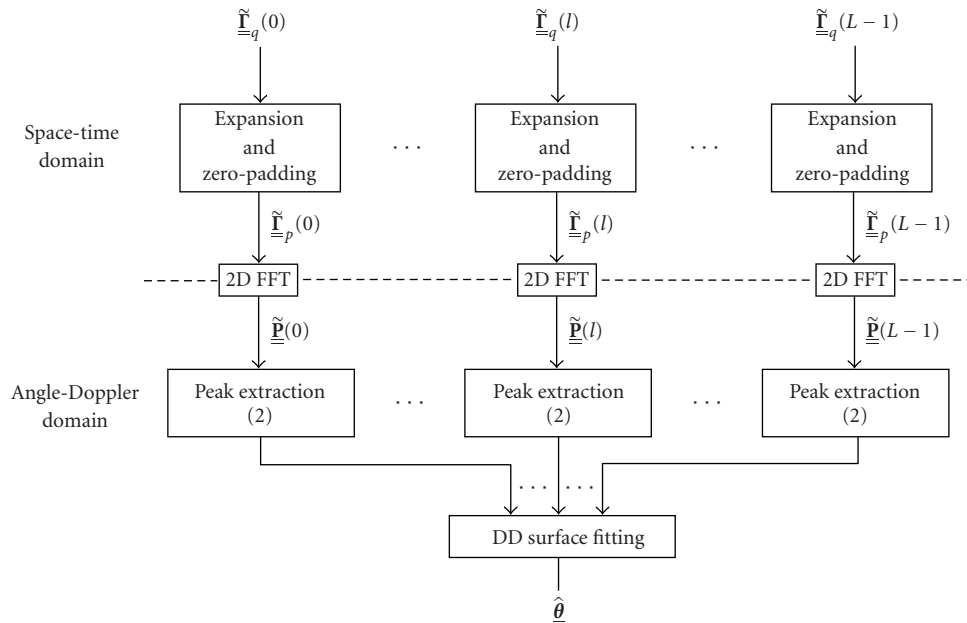


FIGURE 8: Block diagram of processing steps for CPE module.

In spite of the added degree of difficulty in the problem considered here (as compared to that considered in [7]), the RBC module described in [7] is directly applicable here. Its block diagram is shown in Figure 7, but the reader is referred to [7] for additional details.

In contrast, the CPE module of [7] needs a major overhaul to deal with the increased degree of uncertainty considered here and illustrated in Figure 1.

9. CONFIGURATION-PARAMETERS ESTIMATION MODULE

Figure 8 shows the block diagram of the CPE module. This diagram is a generalization of the corresponding diagram in [7]. One main difference is that the PS for all ranges $l \in \mathcal{L}$

are used. The $\tilde{\mathbf{r}}_q(l)$'s are processed individually up to and including peak extraction (described in Section 10). The peaks corresponding to all l 's are then used jointly to find the DD surface that best matches them. This surface yields the estimate $\hat{\underline{\theta}}$ of $\underline{\theta}$.

The significant contributions of this paper are the design and evaluation of the peak extraction (2) and of the DD-surface-fitting processing steps (i.e., algorithms) shown in Figure 8.

Figure 1a shows the PS obtained for a given $\underline{\theta}$ and a given R_b when we use the theoretical $\underline{\mathbf{r}}_q$. The clutter ridge is deterministic and fixed. The peaks are located on, or very close to, the corresponding DD curve, which is fixed. Figure 1b shows the PS obtained for the same $\underline{\theta}$ and R_b when using a single,

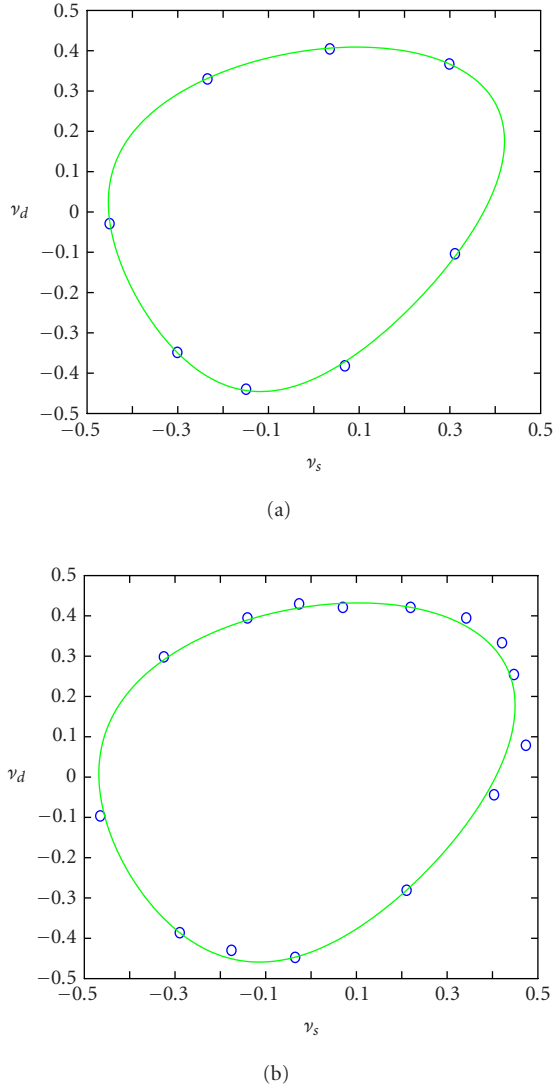


FIGURE 9: Each subfigure shows the peaks extracted at a given range by the peak extraction (2) algorithm. The underlying, theoretical DD curve is also shown. Whereas, in (a), the extracted peaks are very close to the underlying DD curve, in (b), they tend to fall further away, which leads to inaccuracies in the DD-surface-fitting algorithm.

stochastic snapshot \underline{y}_{-q} . From one realization to the next, the PS estimate and, thus, the related clutter ridge will most likely change. However, this clutter ridge will always be located in the vicinity of the underlying DD curve, which is the same as in the 1st case. Comparing both cases, it is clear that the 2nd is more challenging since we may experience difficulties in extracting the peaks and in fitting a DD surface to them.

The reader is referred to [7] for a description of the first two processing steps, that is, expansion and zero-padding and 2D FFT. In fact, these steps are identical to the first two steps of the RBC module (Figure 7). In both cases, the purpose of these steps is to bring us into the spectral domain. (The expansion sizes associated with the 2D FFT may differ in both modules.)

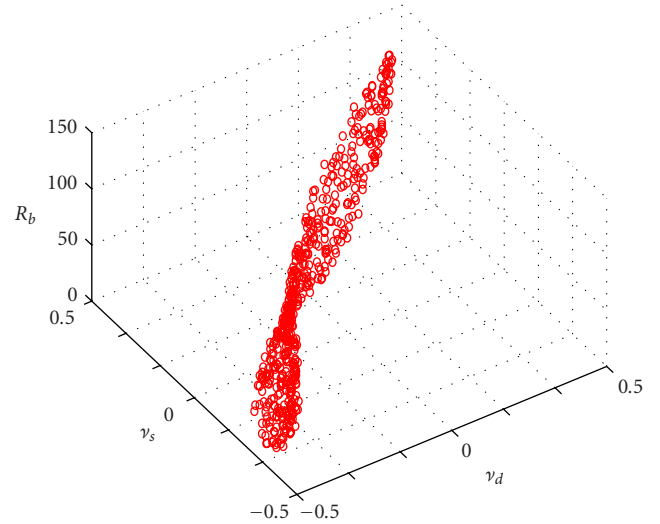


FIGURE 10: 3D scatter plot of extracted peak locations in (ν_s, ν_d, R_b) -axes. The peak locations shown are located in the L horizontal slices corresponding to the L values of $l \in \mathcal{L}$. The locations in each slice are the output of the peak extraction (2) algorithm. The complete collection of peak locations is the input to the DD-surface-fitting algorithm. Ideally, these points should fall on the underlying DD surface. In practice, they should most likely fall in its vicinity.

The other two processing steps, that is, peak extraction (2) and DD surface fitting, are described in the next two sections. Their design, implementation, and evaluation are the major contributions of this paper.

10. PEAK EXTRACTION (2) ALGORITHM

The corresponding algorithm of [7] was significantly modified to deal with the sparsity and stochastic behavior of the peaks in any given realization of a PS. The new algorithm for extracting the desired peaks from the PS at any given range l works as follows: (1) the largest value PS^{\max} in the PS array is found; (2) all *pixels* in the PS array with values less than a given percentage (set here to 30%) of PS^{\max} are set to zero; (3) the remaining nonzero values are grouped into regions by the “connected components” labelling technique of image processing [14] or equivalent; (4) the largest value within each region is found and the location of the corresponding pixel defines the location of the peak for that region.

This peak-extraction algorithm is based on the hope that the largest value in each region will fall on or close to the underlying DD curve. Simulations show that this is generally the case. This is illustrated in Figure 9.

The above algorithm is then applied to each $l \in \mathcal{L}$. This leads to a constellation of peak locations, as illustrated in Figures 10 and 11. This constellation of points is the input to the DD-surface-fitting algorithm.

11. DD-SURFACE-FITTING ALGORITHM

The DD-surface-fitting algorithm we have implemented to recover the configuration parameters is quite complicated.

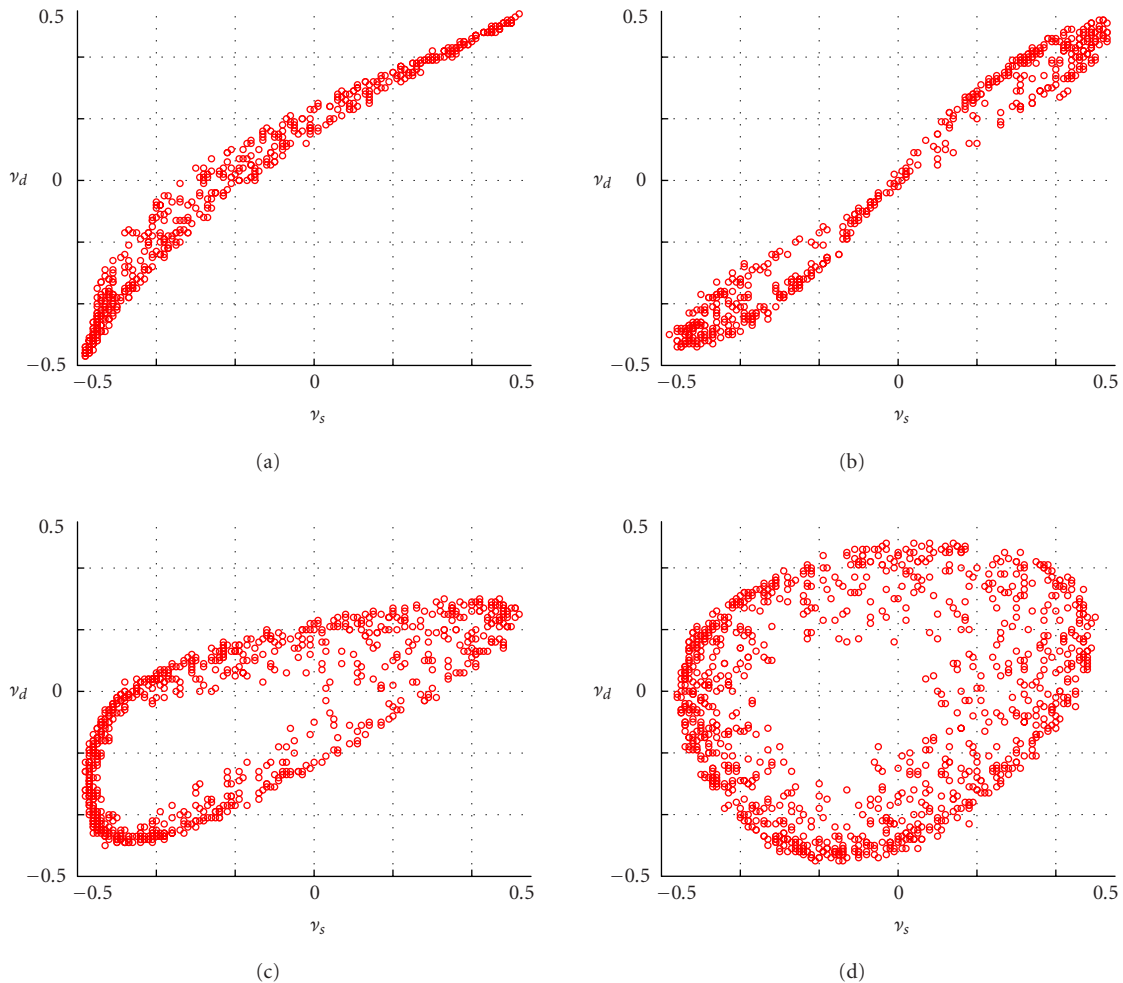


FIGURE 11: Each subfigure is the top view of a 3D scatter plot of peak locations similar to that of Figure 10. The four subfigures labelled (a) through (d) correspond to the configurations of Figures 3a–3d, respectively.

For purpose of conciseness, we limit ourselves here to describing its general principle.

We start from the 3D constellation of points (extracted peak locations) that can be thought of as lying on an *experimental* DD surface that, we hope, approximates well the underlying theoretical DD surface. Our goal is to recover the set of configuration parameters that characterizes this theoretical DD surface (Figure 2). (We know that there may be 2 such sets and that they are related.) Since it is quite logical to assume that we know H and ν_R , we only need to estimate x_R , y_R , z_R , ν_T , α_R , and δ . First, we derive the tightest possible constraints on the possible values of x_R , y_R , and z_R to reduce the search space. Then, we explore each allowed position (x_R, y_R) and estimate the other parameters.

11.1. Determination of constraints on x_R , y_R , and z_R

We apply the same method as in [7]. However, the sparsity of the estimated peak locations prevents us from directly obtaining the estimates $\hat{\gamma}_{s,l}^{\min}$ and $\hat{\gamma}_{s,l}^{\max}$ of the extremities, along the ν_s -axis, of the DD curve at each l . We have thus developed a new method to derive reliable estimates $\hat{\gamma}_{s,l}^{\min}$ and $\hat{\gamma}_{s,l}^{\max}$

at each l by taking into account the set of peaks for all l 's in \mathcal{L} . Results have proven to be more accurate at shorter ranges R_b . Therefore, we focus on a single short range (typically for $l = 4$ rather than for $l = 0$ to improve accuracy) and determine the constraints on x_R , y_R , and z_R exactly as in [7].

11.2. Determination of z_R , ν_T , α_R , and δ for each (x_R, y_R)

As in [7], we apply the following procedure to each allowed pair (x_R, y_R) . First, we compute z_R using a formula we have derived, which relates z_R to x_R , y_R , $\hat{\gamma}_{s,l}^{\min}$, and $\hat{\gamma}_{s,l}^{\max}$. This formula is quite lengthy and is thus omitted. If z_R satisfies the constraints specified in [7], we proceed. Otherwise, we discard the candidate pair. To proceed, we rely on exact analytical expressions for DD surfaces in terms of θ .² For each candidate position (x_R, y_R, z_R) of R , we first insert the values of (x_R, y_R, z_R) in the analytical expressions for the DD surface.

²Deriving these equations is quite complicated. Furthermore, the equations themselves are also quite complicated. Thus, providing the derivation or even the equations is beyond the scope of this paper.

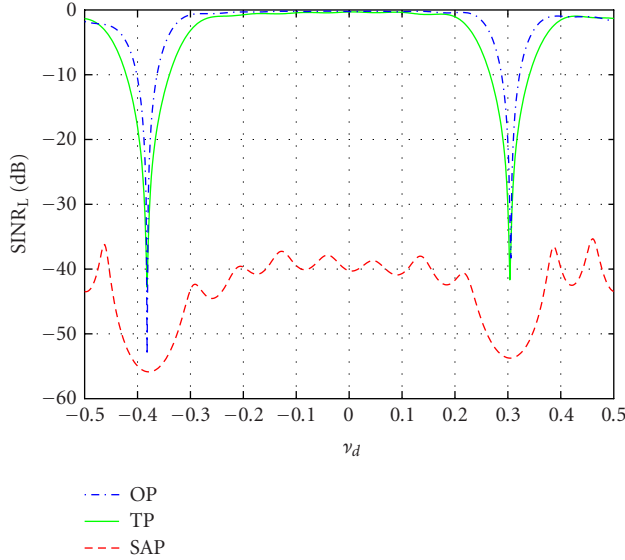


FIGURE 12: The end-to-end performance of a STAP processor using RBC and working in TP mode (known θ) is a direct reflection of the performance of the RBC module (see Figure 6). Performance is shown in terms of cuts of SINR loss at $\nu_s = 0$. The performances of the OP (the best achievable) and of the straight-averaging processor (no RD compensation) are also shown as references.

Then, we estimate the remaining unknown parameters ν_T , α_R , and δ by least-square fit of the resulting parametric surface to the extracted peak locations. We have found that the upper part of the DD surface is better suited for estimating ν_T , α_R , and δ . As a result, we perform the fitting using only the N_{lr} longest ranges. The resulting least-square fit error is also computed. The influence of the choice of value for N_{lr} is discussed in Section 12.2.

11.3. Final parameter selection

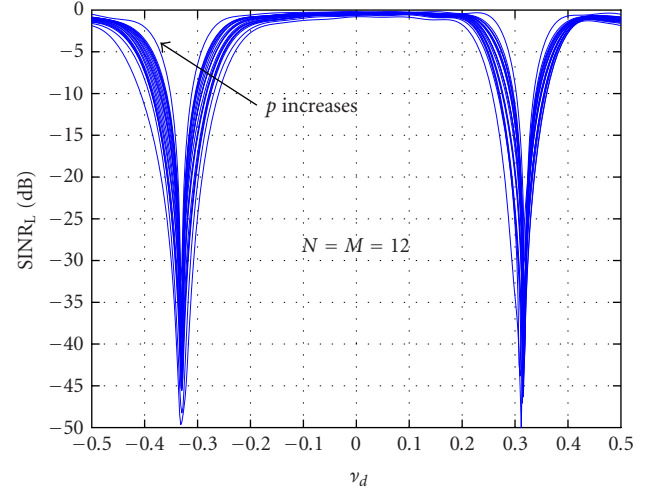
The procedure just described (in Section 11.2) is repeated for all allowed pairs (x_R, y_R) . The set of parameters resulting in the smallest least-square fit error is ultimately chosen. These parameters, as well as the known H and ν_R , make up the final $\hat{\theta}$.

12. PERFORMANCE EVALUATION

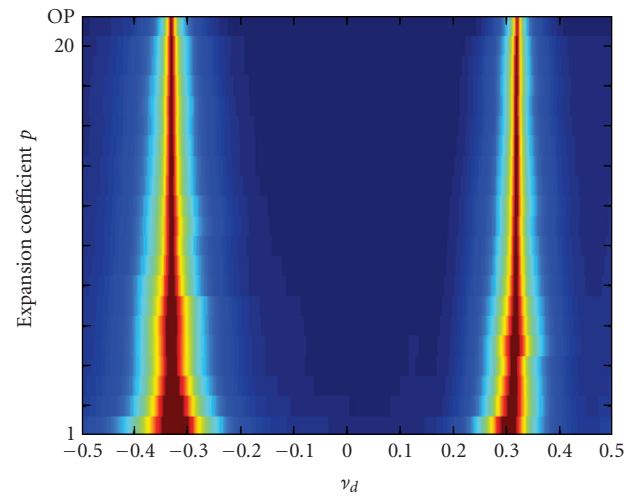
We first discuss the individual performances of the RBC and CPE modules. Then, we compare the end-to-end performance of STAP processors working in TP mode and in EP mode to those of the OP and of the straight-averaging processor (no compensation).

12.1. Performance of RBC module

To evaluate the individual performance of the RBC module, we consider an end-to-end STAP processor using RBC and working in TP mode. Indeed, this mode of operation only involves the RBC module (see Figure 6). The performance of this processor is shown in Figure 12 in terms of SINR loss plots.



(a)



(b)

FIGURE 13: Cuts of SINR loss at $\nu_s = 0$ for various values of expansion coefficient p . OP is shown for reference. (a) Topmost curve corresponds to OP. Other curves, from bottommost to topmost, correspond to p increasing from 1 to 20. (b) Alternate visualization, where grayscale corresponds to SINR loss. Curves are shown in the same order as in (a), with topmost row corresponding to OP. Performance increases with increasing p and gets very close to that of OP for large p .

With reference to Figures 7 and 8, recall that we must typically expand the size of $\hat{\underline{\Gamma}}_q(k)$ when the source range k is smaller than the destination range l . This is done by zero-padding. The relative size of this padding plays a critical role in the performance of the RBC module. To characterize this expansion, we define the expansion coefficients $p = P/(2M - 1)$ and $q = Q/(2N - 1)$, where P , Q , N , and M are related to the sizes $P \times Q$ and $(2M - 1) \times (2N - 1)$ of $\hat{\underline{\Gamma}}_p$ and $\hat{\underline{\Gamma}}_q$, respectively. In our experiments, all arrays are square and, thus, $p = q$.

Figure 13 illustrates the influence of p on the SINR loss. Figure 13a shows the SINR loss as a function of ν_d

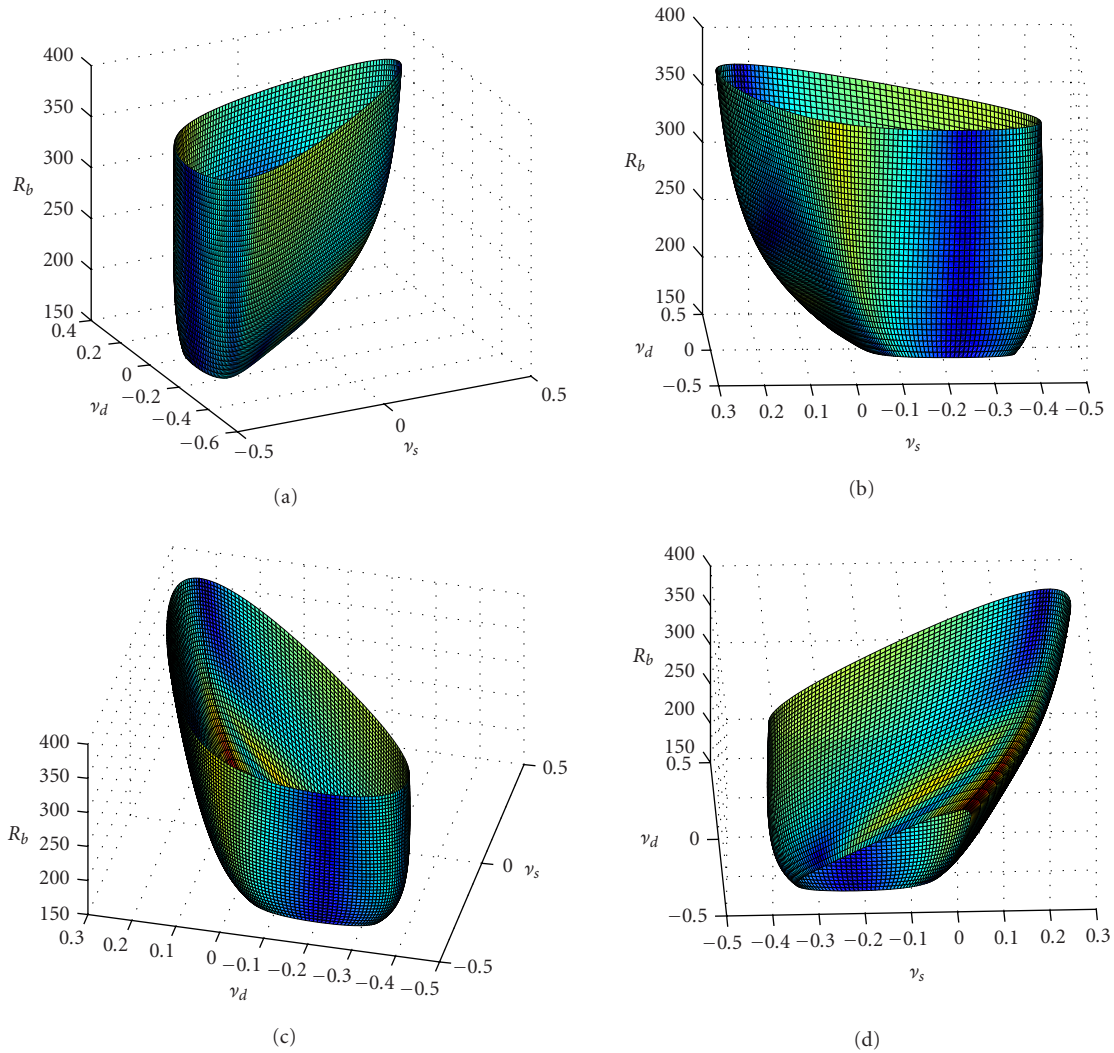


FIGURE 14: Four different views of the true DD surface color coded with the absolute value of the fit error for the configuration of Figure 3c.

for $\nu_s = 0$. The topmost curve corresponds to the OP, while the other curves, from bottommost to topmost, correspond to processors using values of p increasing from 1 to 20. Figure 13b provides an alternate visualization of the same information as in Figure 13a: the vertical axis now corresponds to p and the grayscale to SINR_L . Both plots clearly show that performance increases with p . Furthermore, performance for large p 's gets very close to that of the OP.

Getting close to OP performance with stochastic snapshots implies using a large p , which leads to significant computational requirements. However, this is counterbalanced by a reduction of the sample-support size, N_l . Indeed, the results shown in Figure 13 were obtained with only $N_l = 32$ snapshots, whereas the rule of Reed, Mallett, and Brennan (RMB) [15] requires at least 288 snapshots for $M = N = 12$. Figure 13a shows the improvement over the 3 dB loss associated with the RMB rule.

12.2. Performance of CPE module

To evaluate the individual performance of the CPE module, we pick some parameter $\underline{\theta}$, generate corresponding stochastic snapshots $\underline{y}_{-q}^{(l)}$, $l \in \mathcal{L}$, run the CPE algorithm, and examine the output $\hat{\underline{\theta}}$. We then compute the (absolute value of the) “horizontal” (i.e., in the (ν_s, ν_d) -plane) fit error between the estimated DD surface corresponding to $\hat{\underline{\theta}}$ and the true DD surface corresponding to $\underline{\theta}$. Figure 14 shows the errors as colors on the true DD surface. Observe that the largest errors occur at short ranges, this is because DD curves change more rapidly with range at short ranges than at long ranges. Figure 15 shows the RMS horizontal error between estimated and true DD surfaces for the four configurations of Figure 3. Since the errors are measured in the (ν_s, ν_d) -plane, they are unitless just as ν_s and ν_d are. Since ν_s and ν_d vary from -0.5 to 0.5 , the maximum possible value of the absolute error is 1. The values given in Table 1 show that the estimated and true DD surfaces nearly coincide.

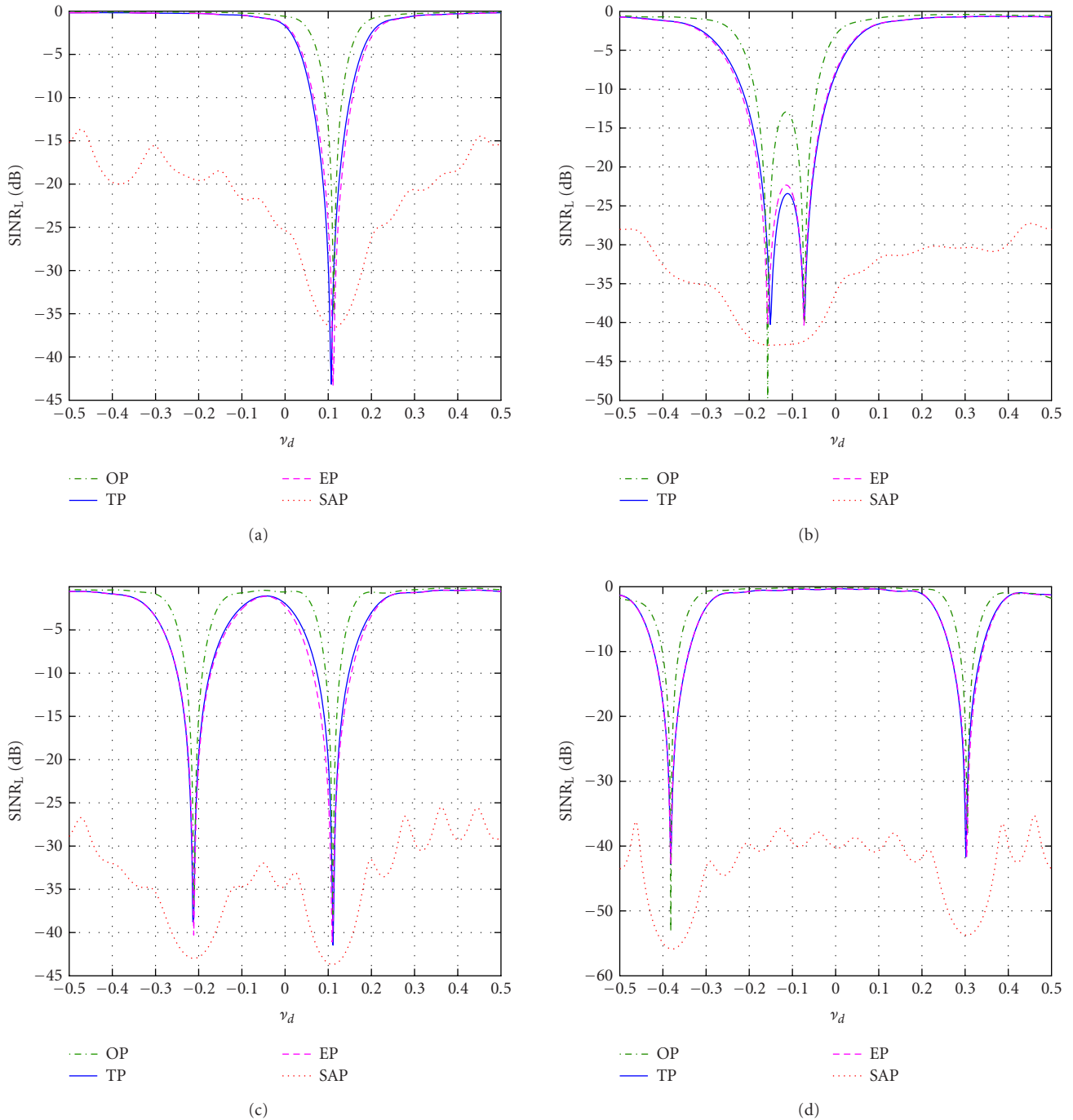


FIGURE 15: Each subfigure quantifies the end-to-end performance of TP, EP, OP, and SAP processors for a given configuration. The four subfigures correspond to the four configurations of Figure 3. Each subfigure shows cuts of SINR loss at $\nu_s = 0$ for the OP (topmost curve), SAP (bottommost curve), and TP and EP (middle curves, with TP above EP). Conclusions drawn from the graphs are given in the text.

Figure 16 shows the RMS horizontal error between the estimated and true DD surfaces as a function of the number of “long ranges,” N_{lr} . We see that an acceptable fit is obtained for relatively small values of N_{lr} . This is significant since a reduction in N_{lr} results in a reduction of the computational load.

12.3. End-to-end performance

We now characterize the performance of a STAP processor using TP mode (known θ) and EP mode (unknown θ) RD compensation. The performance of the OP (known $\underline{\mathbf{R}}_q$; no need for RD compensation) and of the straight-averaging

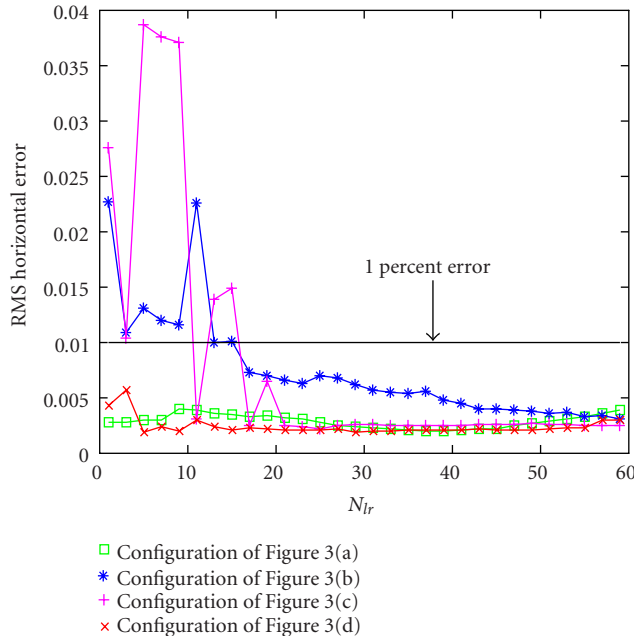


FIGURE 16: RMS horizontal error (between estimated and true DD surfaces) as a function of the number of *long ranges*, N_{lr} , used by the DD-surface-fitting algorithm for the four configurations of Figure 3. In all cases, the error *tends* to decrease with increasing N_{lr} . However, past about $N_{lr} = 15$, there is little improvement.

TABLE 1: RMS horizontal error between estimated and true DD surfaces.

Configuration	RMS horizontal error (unitless)
Figure 3a	0.0024
Figure 3b	0.0062
Figure 3c	0.0026
Figure 3d	0.0019

processor (SAP; no RD compensation) are also shown for reference. Figure 15 shows the SINR loss curves corresponding to the TP, EP, OP, and SAP processors for the four configurations of Figure 3. The plots show that the performance of TP is close to that of OP and much better than that of SAP, thereby demonstrating the soundness of our registration-based approach to RD compensation. The plots also show that EP performance is close to that of TP, thereby demonstrating the soundness of our combined peak extraction and DD-surface-fitting approaches for estimating the configurations parameters.

13. CONCLUSION

The range-dependence (RD) problem in STAP originates from the lack of stationarity of the snapshot statistics w.r.t. range. Its clearest manifestation is the deformation with range of the power spectrum. The usual maximum likeli-

hood estimate of the interference-plus-noise covariance matrix (CM) $\underline{\mathbf{R}}_q$ is no longer optimal and leads to poor target-detection performance. Therefore, it becomes imperative to develop RD compensation methods.

In [7], we introduced registration-based methods for known $\underline{\mathbf{R}}_q$ and developed a general strategy for estimating the CM at each range. This strategy is based on the registration of clutter ridges and direction-Doppler (DD) curves at each range, using known (theoretical) CMs $\underline{\mathbf{R}}_q$ at neighboring ranges.

In this paper, we adapted the strategy of [7] to handle single realizations of stochastic snapshots. The main difficulty lies in the estimation of the configuration parameters. The main contributions of this paper are the modification of the peak-extraction algorithm of [7] and the development of a new 3D DD-surface-fitting algorithm generalizing the 2D DD-curve-fitting algorithm of [7]. The performance of the new algorithms was evaluated in detail.

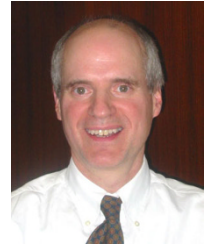
ACKNOWLEDGMENTS

The authors acknowledge useful discussions with Xavier Neyt from the Department of Electrical Engineering, Royal Military Academy, Brussels, Belgium. The authors thank the reviewers for several useful comments. Much of the work of Fabian Lapierre was supported by a fellowship of the Fond National de la Recherche Scientifique (FNRS), Brussels, Belgium.

REFERENCES

- [1] R. Klemm, *Principles of Space-Time Adaptive Processing*, vol. 9 of *Radar, Sonar, Navigation & Avionics*, IEE, London, UK, 2002.
- [2] J. Ward, "Space-time adaptive processing for airborne radar," Tech. Rep. 1015, MIT Lincoln Laboratory, Lexington, Mass, USA, 1994.
- [3] G. K. Borsari, "Mitigating effects on STAP processing caused by an inclined array," in *Proc. IEEE National Radar Conference (RADARCON '98)*, pp. 135–140, Dallas, Tex, USA, May 1998.
- [4] S. M. Kogon and M. A. Zatman, "Bistatic STAP for airborne radar systems," in *Adaptive Sensor Array Processing Workshop*, MIT Lincoln Laboratory, Lexington, Mass, USA, March 2000.
- [5] B. Himed, Y. Zhang, and A. Hajjari, "STAP with angle-Doppler compensation for bistatic airborne radars," in *Proc. IEEE National Radar Conference (RADARCON '02)*, pp. 311–317, Long Beach, Calif, USA, April 2002.
- [6] W. L. Melvin, B. Himed, and M. E. Davis, "Doubly adaptive bistatic clutter filtering," in *Proc. IEEE National Radar Conference (RADARCON '03)*, pp. 171–178, Huntsville, Ala, USA, May 2003.
- [7] F. D. Lapierre, J. G. Verly, and M. Van Droogenbroeck, "New solutions to the problem of range dependence in bistatic STAP radars," in *Proc. IEEE National Radar Conference (RADARCON '03)*, pp. 452–459, Huntsville, Ala, USA, May 2003.
- [8] F. D. Lapierre and J. G. Verly, "Registration-based solutions to the range-dependence problem in STAP radars," in *Adaptive Sensor Array Processing Workshop*, MIT Lincoln Laboratory, Lexington, Mass, USA, March 2003.
- [9] F. D. Lapierre, M. Van Droogenbroeck, and J. G. Verly, "New methods for handling the range dependence of the clutter spectrum in non-sidelooking monostatic STAP radars," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '03)*, vol. 5, pp. 73–76, Hong Kong, April 2003.
- [10] S. D. Hayward, "Adaptive beamforming for rapidly moving arrays," in *Proc. IEEE International Radar Conference*, pp. 480–483, Beijing, China, October 1996.
- [11] L. E. Brennan and I. S. Reed, "Theory of adaptive radar," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 9, no. 2, pp. 237–252, 1973.
- [12] N. R. Goodman, "Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction)," *Annals of Mathematical Statistics*, vol. 34, no. 1, pp. 152–177, 1963.
- [13] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and Adaptive Signal Processing*, McGraw-Hill, New York, NY, USA, 2000.
- [14] D. Ballard and C. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1982.
- [15] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 10, no. 6, pp. 853–863, 1974.

Jacques G. Verly was born in Liège, Belgium. He received the Ingenieur Electronicien (Electronics Engineer) degree from the University of Liège, Belgium, in 1975. Through a sponsorship of the Belgian American Educational Foundation (BAEF), he joined Stanford University, Stanford, Calif, where he received the M.S. and Ph.D. degrees in electrical engineering in 1976 and 1980, respectively. From 1980 to 2000, he was a member of the technical staff at MIT Lincoln Laboratory, Lexington, Mass. Since 2000, he has been a Professor in the Department of Electrical Engineering and Computer Science (also known as the "Institut Montefiore"), the University of Liège, Belgium, where he is the Head of the new signal processing group and in charge of the core courses in signal processing. His current research interests are principally in medical imaging (image-guided surgery), radar signal processing (space-time adaptive processing or STAP), and object tracking in video streams (for videosurveillance and sports analysis). In 2003–2004, he was the holder of one of the two prestigious Francqui Chairs at the Free University of Brussels (ULB). He has about 150 publications, mostly in the area of image processing and computer vision for advanced imaging systems, as well as two US patents. He is a CRB Fellow of the BAEF.



Fabian D. Lapierre was born in Huy, Belgium. He received the Ingenieur Electronicien (Electronics Engineer) degree from the University of Liège, Belgium, in 2000. In the same year, he received a fellowship of the Fonds National de la Recherche Scientifique (FNRS), Brussels, Belgium, to work on a Ph.D. thesis. He is in his final year as a Ph.D. student. His research interest is mainly focussed on space-time adaptive processing (STAP) for bistatic radars.

