Article

A Track-Before-Detect Strategy Based on Sparse Data Processing for Air Surveillance Radar Applications

Nicomino Fiscante [1,*], **Pia Addabbo** [2], **Carmine Clemente** [3], **Filippo Biondi** [4], **Gaetano Giunta** [1] **and Danilo Orlando** [5]

1. Department of Engineering, University of Roma TRE, 00154 Rome, Italy; gaetano.giunta@uniroma3.it
2. Department of Telecommunication Engineering , University of Study "Giustino Fortunato", 82100 Benevento, Italy; p.addabbo@unifortunato.eu
3. Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G11XQ, UK; carmine.clemente@strath.ac.uk
4. Joint Satellite Remote Sensing Center, Italian Ministry of Defence, 00187 Rome, Italy; biopippoo@yahoo.it
5. Department of Engineering, University of Study "Niccolò Cusano", 00166 Rome, Italy; danilor78@gmail.com
* Correspondence: nicomino.fiscante@uniroma3.it

Abstract: In this paper we consider the tracking problem of a moving target competing against noise and clutter in a surveillance radar scenario. For a single array-antenna multiple-target tracking system and according to the Track-Before-Detect paradigm, we present a novel approach based on a three-stage processing chain that involves the Sparse Learning via Iterative Minimization algorithm, the *k-means* clustering method and the ad hoc detector by exploiting the sparse nature of the operating scenario. Under the latter assumption, the detection strategy declares the presence of targets subsequently to the retrieval of their corresponding tracks performed by jointly processing the received echoes of multiple consecutive radar scans. Simulation results show that the proposed approach is able to provide good tracking and detection capabilities for different multiple target trajectories with low Signal-to-Interference-plus-Noise ratio and results in providing advantages when compared to a number of other reference Track-Before-Detect strategies based on sparse data processing techniques.

Keywords: air surveillance radar; clustering; machine learning; multiple-target tracking; radar; sparse data recovery; Track-Before-Detect

1. Introduction

In a multiple target tracking problem, the objective is to estimate an unknown and time varying number of targets as well as their trajectories from sensor data [1]. Traditional radar tracking algorithms are designed assuming that the sensor provides a set of point measurements at each scan, obtained by thresholding the output of a matched filter fed by a baseband version of the collected echoes [2–4]. This approach is effective in the limited case of Signal-to-Interference-plus-Noise ratio (SINR), while it experiences some limitations when the signal amplitude is weak compared to the background interference, and in general, in the detection of dim targets. Specifically, for dim targets the threshold must be low enough to allow for sufficient probability of target detection with the drawback that a low threshold leads to high volumes of false alarms.

An alternative approach, referred to as Track-Before-Detect (TBD), consists in supplying the processor with either thresholded or unthresholded data from several consecutive scans [5]. Unlike conventional techniques that declare the presence of a target at each scan of the radar, TBD-based procedures jointly process more consecutive scans and, exploiting the physically admissible target transitions, jointly declare the presence of a target and its track. Track-Before-Detect improves track accuracy and allows the tracker to effectively deal with low-observable targets, i.e., objects with a low SINR. The main difficulty in TBD

techniques is that the measurements depend on the target state in a highly nonlinear way. Typically, the target state describes the kinematic evolution of the target and may also include its amplitude. A possible means to solve the nonlinear estimation problem is to resort to particle filtering [6]. An alternative is to discretize the target state space, since when the state is discrete, the linearity is irrelevant [7–11]. In this case, the problem with using a discrete state space is that it leads to high computation and memory resource requirements. Summarizing, if on one hand the TBD paradigm allows for a detection performance improvement, on the other hand, it leads high computational costs due to the huge amount of data collected at each scan. We refer the readers to [12] for a comparison of several TBD strategies to detect low amplitude targets.

The idea behind TBD has been developed in the past 20 years and its importance in low SINR radar application was first investigated in [13,14]. Since then, a number of TBD techniques and applications have been studied [15–19]. In recent years, security applications employing radar sensors for surveillance objectives are becoming increasingly important, so it is convenient to apply TBD techniques because targets with a low SINR appear [20–24]. In [21] the authors present a family of TBD procedures for early detection of moving targets from airborne radars exploiting a sectorization of the coverage area, the received echoes are jointly processed in the fast and slow time domains. In [22], TBD strategies for space-time adaptive processing radars are derived. The authors introduce the target and noise models in discrete-time form and a generalized likelihood ratio test strategy for target detection into two different scenarios. Results show that the proposed procedures might be a viable means to implement early detection and track initiation of weak moving targets. In [23], a number of new algorithms are proposed for adaptive detection and tracking based on spatial-time data. The possible spillover of target energy to adjacent range cells is taken into account at the design stage. The research proposed in [24] gives application of TBD processing against airborne radar surveillance data. The sensitivity of TBD processing to the choice of clutter model is examined through processing of real and simulated data containing non-coherent radar returns of a small maritime targets in sea clutter. The results confirm the good detection and tracking performance.

Leaving aside the TBD approach for the moment, we recall here that a radar system illuminates a given area and attempts to detect prospective targets present in its field of view by also estimating their locations and reflectivity. This task is accomplished by partitioning the region of interest into range, azimuth, and elevation bins. Remarkably, the prospective targets occupy only few bins as shown in Figure 1 and hence, if we neglect the interference, the 3D data matrix representing the observed scenario is sparse as most of its entries are approximately zero. Remarkably, it is important to highlight that considering multiple radar scans over time, the data sparsity increases even more. In [25], the authors exploit such features to devise radar processors based on compressed sensing techniques [26–28]. As a matter of fact, the exploitation of sparsity, originally proposed in statistics, signal processing, and machine learning communities, can improve the performance of radar systems under certain conditions and is therefore of considerable practical value.

Compressed sensing based methods have been applied to radar systems [29–39] to recover the target scene from a fewer number of measurements than traditional methods. In [29], it is demonstrated that compressed sensing can eliminate the need for matched filter at the receiver and has the potential to reduce the required sampling rate. Specifically, the authors present an adaptive clutter suppression method for airborne random pulse repetition interval radar by using prior knowledge of clutter boundary in Doppler spectrum. The research developed in [30] focuses on monostatic chaotic multiple-input-multiple-output radar systems and investigates the performance of sparsity-exploiting algorithms for the parameter estimation of targets with low-SINR. Remarkably, in the context of TBD, compressed sensing provides an effective means to address the case of multiple targets allowing to solve the nonlinear problem encountered in the traditional TBD algorithms. In [32] the authors propose an innovative, computational efficient TBD algorithm which exploits the sparse nature of measured data. They propose a two-step approach:

the first step discards unreliable measurements, i.e., those with likelihood ratio below a preassigned thresholds; the second step, instead, uses the correlation among observations taken at different time instants to makes the final decision. In [33] a novel compressed sensing based TBD algorithm is devised. The latter reconstructs the whole radar scenario (direction of arrival (DOA)-Doppler plane) for each range gate at consecutive scans using an improved stagewise orthogonal matching pursuit (StOMP) algorithm, resulting in a three-dimensional range-DOA-Doppler space. The improved StOMP algorithm together with the temporal tracking, can effectively distinguish true targets from false targets and clutter based on information from multiple illuminations. In the context of synthetic aperture radar (SAR), refs. [34–39] present compressed sensing based data acquisition and imaging algorithms.
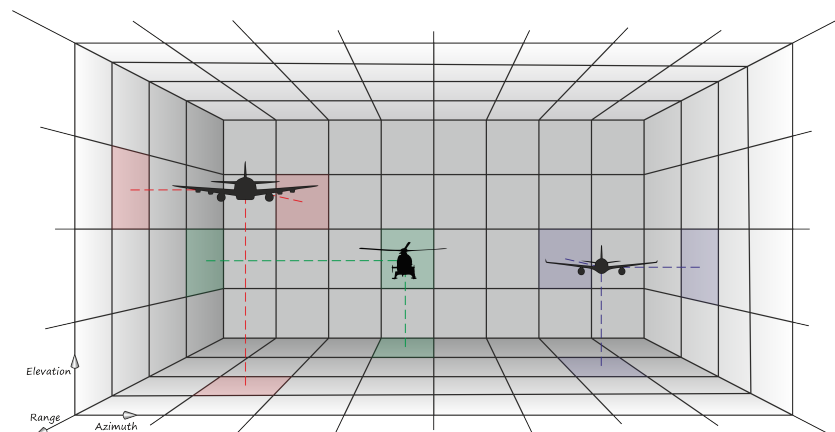
Figure 1. A particular 3D view of the radar datacube to highlight its data sparsity. It is important to note that the three targets occupy only three bins of the entire definition domain.

With the above remarks in mind, in this paper we propose a TBD strategy in the context of air surveillance radar that exploits the sparsity of measured data. The main technical contribution is represented by the development of a multiple target detection architecture comprising three processing stages using a monostatic array-antenna. More precisely, the first processing stage relies on the Sparse Learning via Iterative Minimization (SLIM) [40–42]. This choice is dictated by the fact that SLIM exhibits a good trade off between low computational cost and reconstruction performance, to estimate targets informations, i.e., position and speed, in the scene under consideration. More importantly, SLIM allows to avoid the nonlinear problem in the search space retrieval. The next step in the receiving chain is the formation of target tracks. To this end, the second stage is aimed at clustering the estimates provided by SLIM with each cluster representing a target trajectory. This stage connects the target information over time to create a trajectory. In the third and last stage, an ad hoc detector is derived resorting to a likelihood ratio test (LRT) that relies on the estimates provided by the clustering stage. The performance of the entire processing chain is assessed through simulated data accounting for different scenarios of practical interest. In addition, we compare the proposed technique with two compressed-sensing-based TBD algorithms exploiting the Orthogonal Matching Pursuit (OMP) [43] and the Compressive Sampling Matching Pursuit (CoSaMP) [44] algorithms, which represent the main benchmarks for compressive sensing recovery algorithms. Finally, the numerical analyses show the superiority of the proposed approach over the considered competitors.

The remainder of the paper is organized as follows. In Section 2, we formulate the TBD highlighting its sparse nature, then we provide a full description of the receiver architecture and the data processing framework. In Section 3 we assess the performance of the proposed architecture by means of numerical examples, while in Section 4 we analyse in detail the obtained results. Conclusions and future developments are provided in Section 5.

Notations

Vectors and matrices are denoted by boldface lower-case and upper-case letters, respectively. The (k,l)th entry of a generic matrix $\boldsymbol{A}$ is denoted using $\boldsymbol{A}(k,l)$ and, similarly, the lth entry of a generic vector $\boldsymbol{a}$ with $\boldsymbol{a}(l)$. Symbol $(\cdot)^T$ denotes transpose operation, $(\cdot)^\dagger$ denotes complex conjugate transpose operation, $|\cdot|$ denotes cardinality for a set while modulus for complex number, $\otimes$ denotes the Kronecker product, $\mathrm{vec}(\cdot)$ refers to the vectorization operation (e.g., stacking the columns of a matrix on top of each others), $\mathrm{diag}(\cdot)$ denotes the diagonalization operation, $\det(\cdot)$ and $\mathrm{tr}(\cdot)$ denote the determinant and the trace of a square matrix, respectively. $\boldsymbol{I}$ and $\boldsymbol{0}$ represent the identity matrix and the null vector or matrix of suitable dimensions, respectively. As to the numerical sets, $\mathbb{R}$ is the set of real numbers, $\mathbb{R}^{N \times M}$ is the set of $(N \times M)$-dimensional real matrices (or vectors if $M = 1$), $\mathbb{C}$ is the set of complex numbers, and $\mathbb{C}^{N \times M}$ is the set of $(N \times M)$-dimensional complex matrices (or vectors if $M = 1$). Finally, the imaginary unit is denoted by j.

2. Material and Methods

2.1. Problem Formulation

The considered scenario involves a transmitter and a receiver, at the same location, equipped with an array of sensors, one or more point-like targets at a certain distance from the array in its far zone, and a narrowband signal that travels the round-trip between the radar and the target [45]. Specifically, we assume an uniform linear array of N_s identical sensors with inter-element spacing h. The array illuminates the surveillance area by transmitting M coherent pulse trains, each consisting of N_p rectangular pulses, before deciding whether or not a target is present. We denote by PRI the Pulse Repetition Interval, $\mathrm{PRF} = 1/\mathrm{PRI}$ the Pulse Repetition Frequency and SRT the Scan Repetition Time. The signal collected by the ith sensor is amplified, filtered, and down converted. Then, as customary, a discrete form for the signal received at the ith sensor is obtained by properly sampling the output of a filter matched to the transmitted pulse. For a fixed angle of view, the radar collects $N = N_s \times N_p$ samples from the rth range cell at mth scan; we denote by $\boldsymbol{z}_{m,r}$ the received N-dimensional complex vector of noisy returns obtained by stacking up the N_p vectors collected by the N_s antenna sensors.

Let us assume that an unknown number L of point-like targets are present in the considered scenario at each scan with $L \in \{1, \cdots, N_T\}$. Moreover, the number of range bins under test is $N_R \gg L$. In order to formulate the detection problem, we need to define (r_{l_m}, d_{l_m}) for $l = 1, \ldots, L$ and $m = 1, \ldots, M$, that is the position of the lth target within the range-Doppler grid at the mth scan. The problem of detecting L multiple targets can be formulated in terms of the following multiple hypothesis test (1):

$$
\begin{cases}
H_0 : \boldsymbol{z}_{m,r} = \boldsymbol{n}_{m,r}, & r \in \Omega_R, m = 1, \ldots, M, \\[2mm]
H_1 : \begin{cases} \boldsymbol{z}_{m,r} = \alpha_{m,r,d}\,(\boldsymbol{s}_d \otimes \boldsymbol{a}) + \boldsymbol{n}_{m,r,d}, & (r,d) = (r_{1_m}, d_{1_m}) \in \Omega_R \times \Omega_D, m = 1, \ldots, M, \\ \boldsymbol{z}_{m,r} = \boldsymbol{n}_{m,r}, & (r,d) \in \Omega_R \times \Omega_D \setminus \{(r_{1_m}, d_{1_m})\}, m = 1, \ldots, M, \end{cases} \\[4mm]
H_2 : \begin{cases} \boldsymbol{z}_{m,r} = \alpha_{m,r,d}\,(\boldsymbol{s}_d \otimes \boldsymbol{a}) + \boldsymbol{n}_{m,r,d}, & (r,d) \in \{(r_{l_m}, d_{l_m}), l = 1, 2\} \in \Omega_R \times \Omega_D, m = 1, \ldots, M, \\ \boldsymbol{z}_{m,r} = \boldsymbol{n}_{m,r}, & (r,d) \in (\Omega_R \times \Omega_D) \setminus \{(r_{l_m}, d_{l_m}), l = 1, 2\}, m = 1, \ldots, M, \end{cases} \\[4mm]
\vdots \\[2mm]
H_L : \begin{cases} \boldsymbol{z}_{m,r} = \alpha_{m,r,d}\,(\boldsymbol{s}_d \otimes \boldsymbol{a}) + \boldsymbol{n}_{m,r,d}, & (r,d) \in \{(r_{l_m}, d_{l_m}), l = 1, \ldots, L\} \in \Omega_R \times \Omega_D, m = 1, \ldots, M, \\ \boldsymbol{z}_{m,r} = \boldsymbol{n}_{m,r}, & (r,d) \in \Omega_R \times \Omega_D \setminus \{(r_{l_m}, d_{l_m}), l = 1, \ldots, L\}, m = 1, \ldots, M, \end{cases}
\end{cases}
\tag{1}
$$

where:

- $\Omega_R = \{1, \ldots, N_R\}$ is the set of the range bins, and $\Omega_D = \{1, \ldots, N_D\}$ indexes a discrete set of Doppler values covering the unambiguous Doppler interval;
- $\alpha_{m,r,d} \in \mathbb{C}$ is a factor representative of the target response and channel effects;
- $\boldsymbol{n}_{m,r} \in \mathbb{C}^{N \times 1}$ for $m = 1, \ldots, M$ and $r = 1, \ldots, N_R$, denote the additive interference components in the rth range bin at the mth scan, and are independent and identically

distributed (iid) complex normal random vectors with zero mean and unknown positive definite Interference Covariance Matrix (ICM) $\boldsymbol{M}_m \in \mathbb{C}^{N \times N}$;

- $\boldsymbol{s}_d$ is the temporal steering vector defined as:

$$\boldsymbol{s}_d = \begin{bmatrix} 1 & e^{j2\pi f_d PRI} & \cdots & e^{j2\pi f_d PRI(N_p-1)} \end{bmatrix}^T \in \mathbb{C}^{N_p \times 1}, \tag{2}$$

with $f_d = 2v_d/\lambda$ and v_d the prospective target radial velocity, positive if the target is approaching the radar;

- $\boldsymbol{a}$ is the spatial steering vector defined as:

$$\boldsymbol{a} = \begin{bmatrix} 1 & e^{j2\pi \frac{h}{\lambda} cos\theta_0} & \cdots & e^{j2\pi \frac{h}{\lambda} cos\theta_0(N_a-1)} \end{bmatrix}^T \in \mathbb{C}^{N_a \times 1}, \tag{3}$$

with θ_0 the fixed polar angle of target respect to the away direction. Figure 2 shows the radar geometry problem.
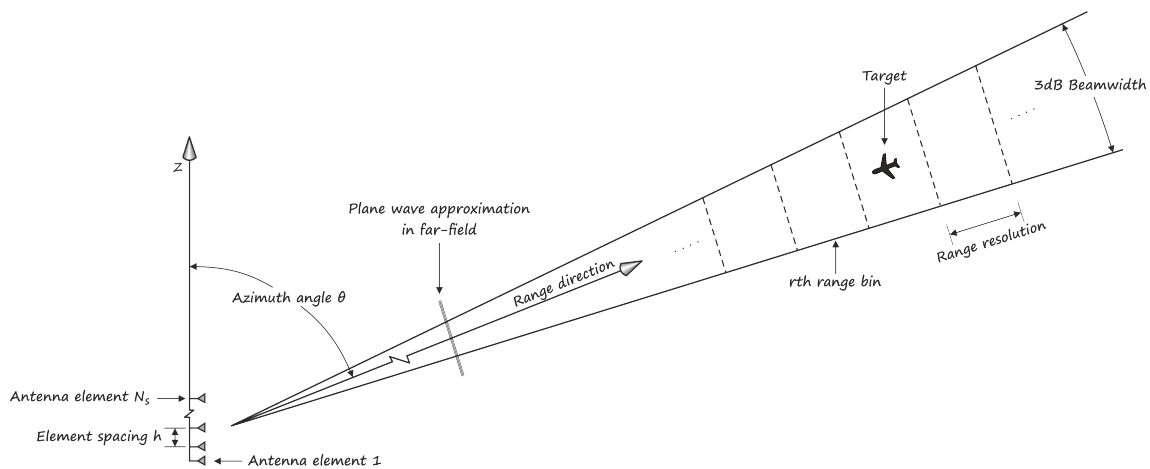

Figure 2. Illustration of the radar geometry problem. The target is assumed to be far enough away that the far-field approximation applies.

Thus, for a specific *m*th radar scan, the received signal matrix $\boldsymbol{Z}_m \in \mathbb{C}^{N \times N_r}$ for the N_R range bins and N_D Doppler points can be written as:

$$\boldsymbol{Z}_m = \left(\sum_{r=1}^{N_R} \sum_{d=1}^{N_D} \alpha_{m,r,d} (\boldsymbol{s}_d \otimes \boldsymbol{a}) \boldsymbol{q}_r^T \right) + \boldsymbol{N}_m, \tag{4}$$

where $\alpha_{m,r,d} \neq 0$ if a target exists in the *r*th range bin and at the *d*th Doppler point, $\boldsymbol{N}_m = [\boldsymbol{n}_1, \cdots, \boldsymbol{n}_{N_R}]$ and $\boldsymbol{q}_r$ represents the shift vector used to describe reflected waveforms from different range bins, and it is defined by:

$$\boldsymbol{q}_r = \begin{bmatrix} \underbrace{0 & 0 & \cdots & 0 & 1}_{r\text{th position}} & 0 & \cdots & 0 \end{bmatrix}^T \in \mathbb{C}^{N_R \times 1}. \tag{5}$$

In order to recast (4) in vector form, we define:

$$\boldsymbol{z}_m = \text{vec}(\boldsymbol{Z}_m) = \boldsymbol{A}\boldsymbol{\alpha}_m + \boldsymbol{n}_m, \tag{6}$$

where $\boldsymbol{A}$ is the space-time steering matrix defined as:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{v}_{1,1} & \boldsymbol{v}_{1,2} & \cdots & \boldsymbol{v}_{N_R,N_D} \end{bmatrix} \in \mathbb{C}^{NN_R \times N_R N_D}, \tag{7}$$

with:

$$\boldsymbol{v}_{r,d} = \text{vec}\left((\boldsymbol{s}_d \otimes \boldsymbol{a}) \boldsymbol{q}_r^T \right) \in \mathbb{C}^{NN_R \times 1}, \tag{8}$$

$\boldsymbol{\alpha}_m$ is signal vector:

$$\boldsymbol{\alpha}_m = \begin{bmatrix} \alpha_{m,1,1} & \alpha_{m,1,2} & \cdots & \alpha_{m,N_R,N_D} \end{bmatrix}^T \in \mathbb{C}^{N_R N_D \times 1}, \tag{9}$$

and $\boldsymbol{n}_m = \text{vec}(\boldsymbol{N}_m)$ is a $NN_R \times 1$ complex normal random vector with zero mean and covariance matrix $\boldsymbol{I} \otimes \boldsymbol{M}_m$.

As $L \ll N_R N_D$, it follows that vector $\boldsymbol{\alpha}_m$ in the linear model in (6) is sparse, i.e., most of its elements are zeros. Moreover, it is clear that with a limited number of measurement, sparsity constraints is required since, without it, the linear system has infinite solutions.

Furthermore, following the compact vector notation defined in (6), the problem (1) can be reformulated as follows:

$$\begin{cases} H_0 : \boldsymbol{z}_m \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{I} \otimes \boldsymbol{M}_m), & m = 1, \dots, M, \\ H_1 : \boldsymbol{z}_m \sim \mathcal{CN}(\boldsymbol{A}\boldsymbol{\alpha}_m, \boldsymbol{I} \otimes \boldsymbol{M}_m), & m = 1, \dots, M, \end{cases} \tag{10}$$

where $\boldsymbol{\alpha}_m$ accounts for the unknown number of targets.

2.2. Proposed TBD Framework

The detection architecture is composed of three stages and its schematic overview is shown in Figure 3. Specifically, the first stage is aimed at estimating the sparse vector $\boldsymbol{\alpha}_m$ exploiting the SLIM algorithm, the second stage is responsible for the track formation and is based on the *k-means* clustering method, and, finally, the third stage implements an ad hoc detector, obtained through the LRT approach.
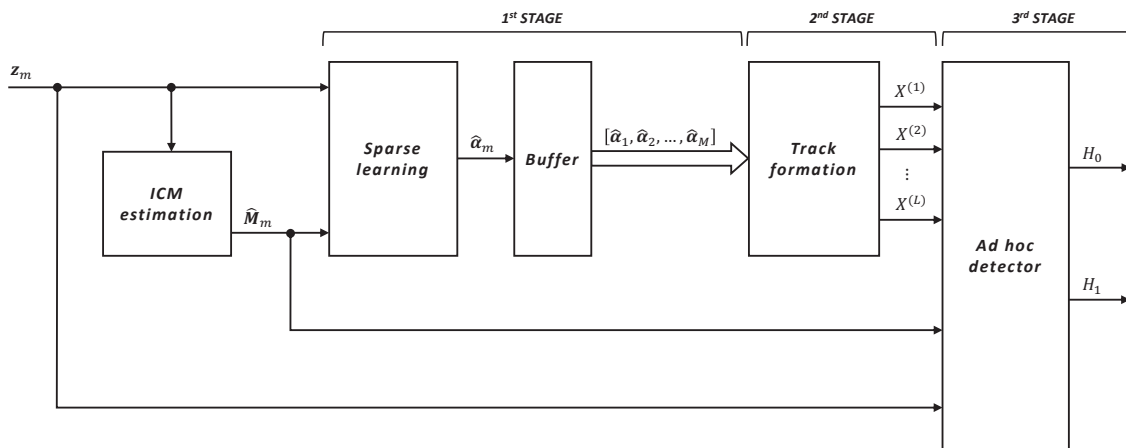

Figure 3. The three-stage detection architecture.

2.2.1. Sparse Learning

As stated before, the main objective of the first stage is to estimate the sparse vector $\boldsymbol{\alpha}_m$ defined in the linear model (6). To this end, we apply the SLIM framework [40,42] assuming that $\boldsymbol{\alpha}_m$ is a random complex vector ruled by a sparsity promoting prior [46,47]. It follows that:

$$\boldsymbol{z}_m | \boldsymbol{\alpha}_m \sim \mathcal{CN}(\boldsymbol{A}\boldsymbol{\alpha}_m, \boldsymbol{I} \otimes \boldsymbol{M}_m), \tag{11}$$

whereas the prior of $\boldsymbol{\alpha}_m$ is:

$$f(\boldsymbol{\alpha}_m; q) \propto \prod_{i=1}^{N_R N_D} e^{-\frac{2}{q}(|\alpha_{m,i}|^q - 1)}, \tag{12}$$

where $q \in (0, 1]$ is a parameter controlling the sparsity level. In fact, when $q = 1$, the prior becomes Laplacian which has a finite peak at 0 while when $q \to 0$ it has an infinite peak at 0. Note, also, $\boldsymbol{M}_m$ is not known and, hence, must be estimated from data. Specifically,

we assume that a set of training data collected in the spatial proximity of the range under test is available and use them to obtain the following ICM estimate:

$$\hat{M}_m = \frac{1}{K} \sum_{k=1}^{K} z_{m,k} z_{m,k}^{\dagger}, \tag{13}$$

where $z_{m,1}, \cdots, z_{m,K}$ are the training data at the mth scan. Thus, we can estimate $\boldsymbol{\alpha}_m$ as follows:

$$\max_{\boldsymbol{\alpha}_m} f(z_m; I \otimes \hat{M}_m | \boldsymbol{\alpha}_m) f(\boldsymbol{\alpha}_m; q), \tag{14}$$

where:

$$f(z_m; I \otimes \hat{M}_m | \boldsymbol{\alpha}_m) = \frac{1}{\pi^{NN_R} \det(I \otimes \hat{M}_m)} e^{[-(z_m - A\boldsymbol{\alpha}_m)^{\dagger}(I \otimes \hat{M}_m)^{-1}(z_m - A\boldsymbol{\alpha}_m)]}, \tag{15}$$

is the conditional probability density function (pdf) of z_m given $\boldsymbol{\alpha}_m$. By taking the negative logarithm it is equivalent to

$$\min_{\boldsymbol{\alpha}_m} \underbrace{\left\{ \|y_m - V\boldsymbol{\alpha}_m\|_2^2 + \sum_{i=1}^{N_R N_D} \frac{2}{q}(|\alpha_{m,i}|^q - 1) \right\}}_{g_q(\boldsymbol{\alpha}_m)}, \tag{16}$$

where $y_m = (I \otimes \hat{M}_m)^{-1/2} z_m$ and $V = (I \otimes \hat{M}_m)^{-1/2} A$. Note that the the first addendum of $g_q(\boldsymbol{\alpha}_m)$ corresponds to a fitting term, whereas the second term promotes the sparsity.

Setting to zero the first derivative of $g_q(\boldsymbol{\alpha}_m)$ with respect to $\boldsymbol{\alpha}_m$ leads to:

$$\frac{d}{dx}[g_q(\boldsymbol{\alpha}_m)] = V^{\dagger} V \boldsymbol{\alpha}_m - V^{\dagger} y_m + P_q^{-1} \boldsymbol{\alpha}_m = 0, \tag{17}$$

where $P_q = \text{diag}(p_q)$, with $p_q = [|\alpha_{m,1}|^{2-q}, |\alpha_{m,2}|^{2-q}, \cdots, |\alpha_{m,N_r N_d}|^{2-q}]^T$. Supposing that an initial estimate of $\boldsymbol{\alpha}_m$ is available, it is possible to apply a cyclic optimization procedure as in [40,42], and the step at the nth iteration can be expressed as:

$$\boldsymbol{\alpha}_{m,q}^{(n)} = P_q^{(n-1)} V^{\dagger} \left(V P_q^{(n-1)} V^{\dagger} + I \right)^{-1} y_m, \tag{18}$$

where $P_q^{(n-1)} = \text{diag}(p_q^{n-1})$ comes from the $(n-1)$th iteration. The optimization procedure can terminate after a fixed number of iterations or when the following convergence criterion is satisfied:

$$\frac{\left\| \boldsymbol{\alpha}_{m,q}^{(n)} - \boldsymbol{\alpha}_{m,q}^{(n-1)} \right\|_2}{\left\| \boldsymbol{\alpha}_{m,q}^{(n)} \right\|_2} < \Delta_{SLIM}, \tag{19}$$

with Δ_{SLIM} a user defined suitable small positive number to obtain the desired estimation accuracy. Specifically, as described in [40] SLIM algorithms converges quickly and in most cases it shows no improvements after 15 iterations. For the SLIM initialization, we apply the same condition based on matched filter as defined in [40,42]:

$$\boldsymbol{\alpha}_{m,i}^{(0)} = \frac{v_i^{\dagger} \left(I \otimes \hat{M}_m \right)^{-1} y_m}{v_i^{\dagger} \left(I \otimes \hat{M}_m \right)^{-1} v_i}, \tag{20}$$

for $i = 1, 2, \ldots, N_r N_d$ and v_i is the ith column of A. To automate the estimation of q, so that SLIM can essentially become user parameter free, the Bayesian Information Criterion (BIC) [48], a model selection tool, is applied [40]:

$$BIC_q = 2\|y_m - V\boldsymbol{\alpha}_m\|_2 + 4h(q) \times \ln(NN_R), \tag{21}$$

where $h(q)$ represents the number of peaks in the SLIM result for a particular value of q. The first term in (21) represents a fitting term for the estimate $\hat{\boldsymbol{\alpha}}_m$, the second term denotes a penalty term, which increases as the number of peaks in the estimate increases and hence is larger for values of q approaching 1. The coefficient 4 in (21) represents the number of unknowns target parameters: complex amplitude, range bin and Doppler shift. It is important to observe that from a practical point of view, after this first stage a data buffer is necessary to temporarily store all the estimated signal vector during the M scans so that it collects $\boldsymbol{\Gamma} = [\hat{\boldsymbol{\alpha}}_1, \cdots, \hat{\boldsymbol{\alpha}}_M] \in \mathbb{C}^{NN_R \times M}$.

2.2.2. Track Formation

The second processing stage is devoted to the extraction of target trajectories from the estimated matrix $\boldsymbol{\Gamma}$. This task is accomplished by means of two consecutive steps. The first one concerns the estimation of the target number (i.e., L). This value is automatically computed supposing that, for a specific radar scan, a target cannot be located in more than one range-Doppler grid. Therefore, L is computed by searching column-wise the maximum number of non-zero element in $\boldsymbol{\Gamma}$. More precisely, let Y_m the set of non-zero element for the estimated vector $\hat{\boldsymbol{\alpha}}_m$:

$$Y_m = \{\hat{\alpha}_{m,r,d} \in \hat{\boldsymbol{\alpha}}_m \ : \ \hat{\alpha}_{m,r,d} \neq 0, \ m = 1, \ldots, M\}, \tag{22}$$

therefore, the estimation of maximum target number is computed as:

$$L = \max_{m=1,\ldots,M} \left(|Y_m| \right) \tag{23}$$

The target to track association is carried out by the application of a clustering algorithm to a dataset containing the information about target positions during radar scanning. Specifically, the *k-means* clustering algorithm, which can be considered one of the simplest and popular unsupervised machine learning algorithms, is applied [49–51]. This algorithm works iteratively, starting from an initial dataset and given the number of clusters, to produce dataset partitions representing the tracks. The iterative refinement clustering is based on a well-defined objective function, also called error-function. In this context, the *k-means* algorithm converges at least to local optimum (i.e., not necessarily the best possible outcome) [51].

For the case at hand, the number of clusters corresponds to the number of targets L, and the dataset, namely X, to be clustered is formed starting from $\boldsymbol{\Gamma}$ matrix, as follows:

$$X = \left\{ \bigcup_{m,r,d} x_{m,r,d} \right\}, \tag{24}$$

where $x_{m,r,d} = [m, r, d, \hat{\alpha}_{m,r,d}]^T$, for all $\hat{\alpha}_{m,r,d} \in Y_m$, with $m = 1, \ldots, M$. The *k-means* algorithm starts with initial estimates for the centroids, which in general can either be randomly generated or randomly selected from the dataset, then it iterates between two steps:

1. data assignment step: each centroid defines one cluster, therefore each vector data is assigned to its nearest centroid. This association in based on the Euclidean metric which is defined as objective function to be locally minimized:

$$\arg\min_{\boldsymbol{\mu}^{(k)}} \sum_{l=1}^{|X|} \sum_{k=1}^{L} \left\| \boldsymbol{W} x_l - \boldsymbol{\mu}^{(k)} \right\|_2^2 \Rightarrow x_l = x_l^{(k)} \ : \ X^{(k)} = \left\{ \bigcup_k x^{(k)} \right\}, \tag{25}$$

where $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{w})$ is a weight matrix, $\boldsymbol{\mu}^{(k)}$ represents the kth cluster centroid; $x_l^{(k)}$ stands for the assignment of the generic vector x_l to k-cluster, and $X^{(k)}$ indicates the kth cluster;

2. centroid update step: the centroids are recomputed by taking the mean of all vector data assigned to that cluster centroid:

$$\mu_k = \frac{1}{|X^{(k)}|} \sum_{x^{(k)} \in X^{(k)}} x^{(k)}. \tag{26}$$

The above steps are repeated until a stopping criteria is met (i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached). The output of interest of the *k-means* algorithm, that is also the output for this processing stage, is the final *L*-partitioning of the starting dataset: $X = \{X^{(1)}, \cdots, X^{(L)}\}$. Each partition contains the range-Doppler point coordinates, that combined in time through the radar scanning information they make the target track.

2.2.3. Ad Hoc Detector

The last stage performs the validation of the tracks formed in the previous stages by comparing a suitable statistics exploiting such tracks to a threshold that allows to control the probability of false track detection. To this end, we resort to the LRT where the unknown parameters are replaced by the previously obtained estimated. Specifically, such decision scheme has the following expression:

$$\frac{f_1(\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_M; \boldsymbol{\Gamma}, \hat{\Omega}_1^{(L)}, \cdots, \hat{\Omega}_M^{(L)}, \hat{\boldsymbol{M}}_1, \cdots, \hat{\boldsymbol{M}}_M)}{f_0(\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_M; \hat{\Omega}_1^{(L)}, \cdots, \hat{\Omega}_M^{(L)}, \hat{\boldsymbol{M}}_1, \cdots, \hat{\boldsymbol{M}}_M)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta, \tag{27}$$

where:

$$f_1(\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_M; \boldsymbol{\Gamma}, \hat{\Omega}_1^{(L)}, \cdots, \hat{\Omega}_M^{(L)}, \hat{\boldsymbol{M}}_1, \cdots, \hat{\boldsymbol{M}}_M) = \prod_{m=1}^{M} \prod_{(r,d) \in \hat{\Omega}_m^{(L)}} f(z_{m,r}; \hat{\alpha}_{m,r,d}(\boldsymbol{s}_d \otimes \boldsymbol{a}), \hat{\boldsymbol{M}}_m),$$

$$f_0(\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_M; \hat{\Omega}_1^{(L)}, \cdots, \hat{\Omega}_M^{(L)}, \hat{\boldsymbol{M}}_1, \cdots, \hat{\boldsymbol{M}}_M) = \prod_{m=1}^{M} \prod_{(r,d) \in \hat{\Omega}_m^{(L)}} f(z_{m,r}; \boldsymbol{0}, \hat{\boldsymbol{M}}_m),$$

with $\hat{\Omega}_m^{(L)}$ the set of integers indexing the range bins and Doppler points corresponding to the estimated target position at the *m*th scan, and

$$f(z_m; \hat{\alpha}_{m,r,d}(\boldsymbol{s}_d \otimes \boldsymbol{a}), \hat{\boldsymbol{M}}_m) = \frac{1}{\pi^N \det(\hat{\boldsymbol{M}}_m)} e^{\{-[z_{m,r} - \hat{\alpha}_{m,r,d}(\boldsymbol{s}_d \otimes \boldsymbol{a})]^\dagger \hat{\boldsymbol{M}}_m^{-1} [z_{m,r} - \hat{\alpha}_{m,r,d}(\boldsymbol{s}_d \otimes \boldsymbol{a})]\}},$$

$$f(z_m; \boldsymbol{0}, \hat{\boldsymbol{M}}_m) = \frac{1}{\pi^N \det(\hat{\boldsymbol{M}}_m)} e^{\{-z_{m,r}^\dagger \hat{\boldsymbol{M}}_m^{-1} z_{m,r}\}}.$$

It is interesting to note that the LRT under the noise only assumption is readily obtained from the above formulations as well. Once the likelihoods of the states are evaluated, a state estimate can be obtained according to the ratio test. In the event that this state is the null state, then the detector reports that there are no targets, otherwise a detection is reported.

3. Results

Numerical simulation are performed following the specifications of the Lockheed Martin FPS-117 surveillance radar [52]. FPS-117 is a 3-D solid-state radar designed to provide long-range accurate aircraft identification and position data for air defence, navigational assistance and tactical control for both close air support and counter air operations [53]. All simulations refer to a radar system with $N_s = 4$, $N_p = 4$, carrier frequency $f_c = 1.388$ GHz, PRF equal to 917 Hz and SRT $= 0.68$ s calculated considering a mechanical rotation scanning with an horizontally array-antenna with a 3 dB beamwidth equal to $2.3°$.

We consider three specific scenarios, with two moving targets, to analyse how the sparse learning and the track formation stages work. These scenarios involves $N_R = 15$ range bins located in the far field of the radar with spatial resolution $R_{min} = 30$ m. The Doppler interval considered, according to the unambiguous frequency Doppler condition $| f_d | = \text{PRF}/2$, is taken from -458.5 Hz to $+458.5$ Hz, with 18.34 Hz separation between Doppler points (thus $N_d = 51$). The SINR is defined after pulse integration considering the actual steering vector at the mth scan:

$$\text{SINR} = | \alpha_{m,r,d} |^2 \, v_{r,d}^\dagger M_m^{-1} v_{r,d}. \tag{28}$$

In each considered scenario, targets move according to a constant velocity dynamic model, that is positive if target is approaching the radar. Neglecting the edge effects between adjacent range cells, Q accounts for the maximum number of range bins transitions of the target between two successive scans:

$$Q = \left\lceil \frac{| v_l | \, SRT}{R_{min}} \right\rceil, \tag{29}$$

where v_l for $l = 1, \dots, L$ is the target velocity. Thus, according to the frequency Doppler definition, the maximum target velocity is $| v_{l,max} | = 49.5$ m/s so that it results $Q_{max} = 2$.

Both SLIM and clustering are iterative algorithms, so it is essential to set some convergence criteria. For both algorithms, a convergence threshold is set equal to $\Delta_{SLIM} = \Delta_{k\text{-}means} = 10^{-3}$, that means the estimation performed can be accepted because it doesn't vary significantly between two consecutive iterations. Moreover, the SLIM maximum number of iterations is 25.

Since closed-form expression of probability of detection and tracking (P_{dt}), defined as the probability to detects the targets and to estimate their tracks correctly in range-Doppler domain, and the desired false track detection probability (P_{ft}) are not available, we resort to Monte Carlo counting technique. More precisely, in order to evaluate the threshold necessary to ensure a preassigned value P_{ft}, we resort to $100/P_{ft}$ and 10^4 independent trials, respectively. Finally, we compute also the Root Mean Square Error (RMSE) defined as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{m=1}^{M} \sum_{(r,d) \in \hat{\Omega}_m^{(L)}} (\Delta r_r^2 + \Delta d_d^2)}, \tag{30}$$

where n is the number of the Monte Carlo trials, Δr_r and Δd_d are the range and Doppler error respectively between the true and the estimated range-Doppler target coordinates.

As natural competitors for the proposed TBD algorithm, we consider two compressed-sensing-based TBD algorithms obaitined by exploiting OMP and CoSaMP which are referred to in the following as OMP-based and CoSaMP-based TBD algorithms, respectively. This choice is dictated by the fact OMP and CoSaMP are considered the main benchmark in the context of compressed sensing [54].

For the sake of clarity, Table 1 lists the simulation parameters used in the three test cases and performance analysis, and a detailed discussion can be found in Sections 3 and 4 respectively.

Table 1. Main simulation parameters. In bold italics are indicated values adopted in Monte Carlo simulation for performance analysis as described in Section 4.

Parameter	Value
Number of antenna sensors (N_s)	4
Antennas inter-element spacing (h)	$\lambda/2$
Number of pulses (N_p)	4
Carrier frequency (f_c)	1.388 GHz
Pulse Repetition Frequency (PRF)	917 Hz
Number of range bins (N_R)	15, *12*
Range spatial resolution (R_{min})	30 m
Doppler interval	$\lvert f_d \rvert \le \text{PRF}/2$
Number of Doppler points (N_D)	51, *25*
Azimuth view angle (θ_0)	$\pi/2$
Number of targets (L)	2
Maximum radial target velocity	49.5 m/s
SLIM convergence threshold (Δ_{SLIM})	10^{-3}
Maximum SLIM iterations	25
k-means convergence threshold ($\Delta_{k\text{-}means}$)	10^{-3}
Number of radar scans (M)	5, *4*
Scan Repetition Time (SRT)	0.68 s
Indipendent Monte Carlo trials (n)	*1000*

3.1. Case Study 1

In the first test case, we consider the two separate targets approaching the radar with different and constant velocities. At the beginning of the radar scanning, the first target is located in the range bin $r_{1_1} = 6$ (starting from a relative position of 15 m) with velocity $v_1 = +25.76$ m/s while the second target in the range bin $r_{2_1} = 12$ (starting from a relative position of 15 m) with velocity $v_2 = +41.62$ m/s. Considering the carrier frequency value and according to the velocity-frequency Doppler relationship, the velocities of the targets correspond to the Doppler frequency $f_{d_1} = +238.42$ Hz (Doppler point $d_{1_1} = 39$) and $f_{d_2} = +385.14$ Hz (Doppler point $d_{2_1} = 47$) for the first and the second target respectively. Figure 4 shows the outputs of interest obtained for a single Monte Carlo trial at a fixed SINR = 9 dB according to the definition given in (28). In detail, Figure 4a shows the output of the matched filter at the receiver, that is the input of the overall processing chain. In Figure 4b the output of the SLIM algorithm is presented. By this figure, it is evident that the estimated signal vector (i.e., $\hat{\boldsymbol{\alpha}}_m$) during the scanning is sparse, in fact most of its value are zeros. However, non-zero values are representative of the range-Doppler target coordinate and its amplitude. This concept is better highlighted in Figure 4c where the Doppler points are stacked over the correspondent range bin. Thanks to this particular representation, is evident in an intuitive way the target paths during the radar scanning, but it is important to remark that this kind of representation is useful for a visual understanding and it is not used in the next stage of data processing, the track formation, because it leads to a loss of informations in the Doppler domain. Figure 4d,e,f illustrate in different ways the output obtained by the track retrieval stage. In particular, Figure 4d,e show the range bins and the Doppler point estimated for the detected targets during the radar scanning compared to true target positions. In this specific simulation no errors occur in range bins estimation (Figure 4d), while from Figure 4e is possible to identify some errors in Doppler point estimations attributable to the SLIM algorithm. Finally, Figure 4f summarizes the range-Doppler bin estimation or each targets, highlighting the spatial targets' movements during the radar scanning.
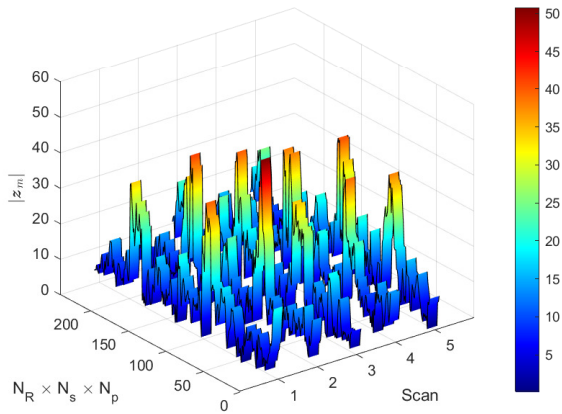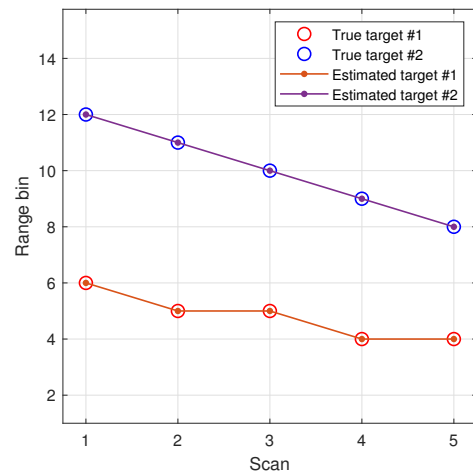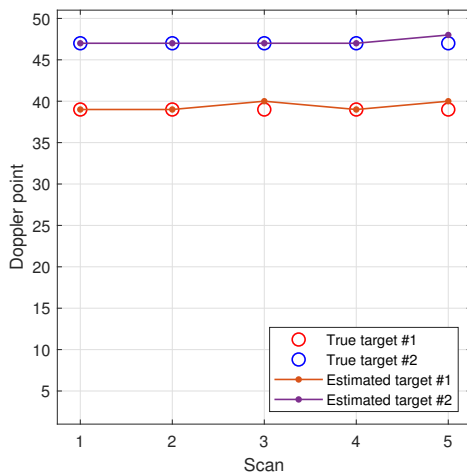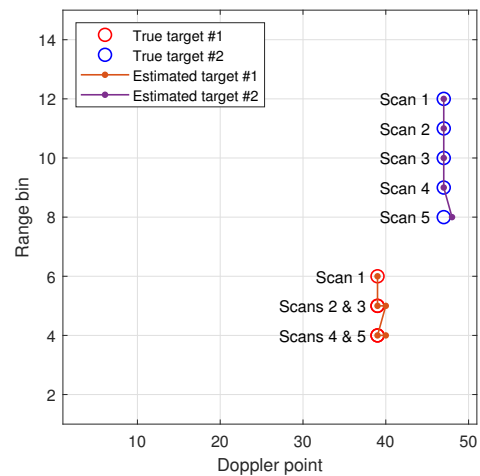
(**a**) *Measured data*
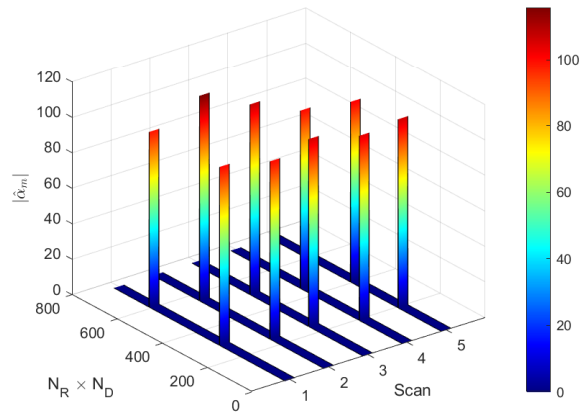
(**b**) *SLIM output*
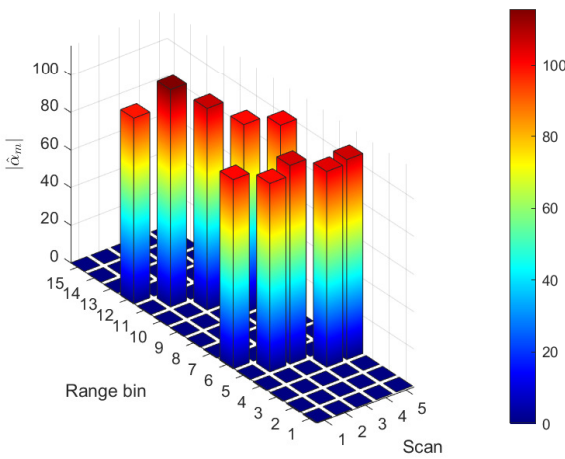
(**c**) *Stacked Doppler points*

(**d**) *Range track* vs. *scan*

(**e**) *Doppler point* vs. *scan*
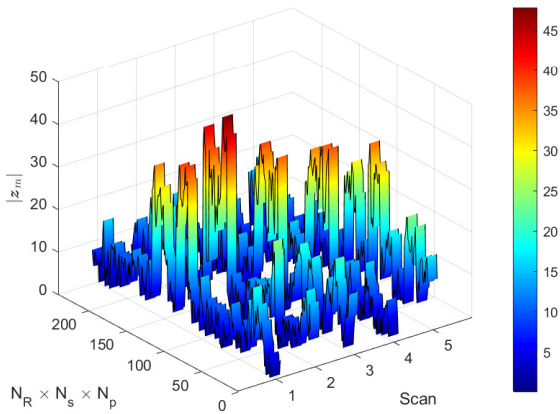

(**f**) *Range track* vs. *Doppler*

Figure 4. Sparse Learning via Iterative Minimization (SLIM)-based Track-Before-Detect (TBD) for the first test case. The radar scenario provides two distant targets in range and Doppler approaching the radar. Simulation is performed for a single Monte Carlo trial at Signal-to-Interference-plus-Noise ratio (SINR) = 9 dB.

3.2. Case Study 2

In the second test case we consider again the two targets approaching the radar with an intersection between their trajectories. Specifically, at the beginning of the radar scanning, the first target is located in the range bin $r_{1_1} = 10$ (starting from a relative position of 25 m) and the second one in the range bin $r_{2_1} = 8$ (starting from a relative position of 15 m). First target velocity is $v_1 = +47.56$ m/s (corresponding to frequency Doppler $f_{d_1} = +440.16$ Hz and Doppler point $d_{1_1} = 50$), while for the second target it is $v_2 = +7.92$ m/s (corresponding to the frequency Doppler $f_{d_2} = +73.36$ Hz and Doppler point $d_{2_1} = 30$) so that during the M radar scans the first target overtakes the second one. This test case is quite interesting because, being the targets spatially close, it is possible to understand some limitations of the proposed processing chain. As in the previous test case, Figure 5 shows the specific results obtained for a fixed SINR = 9 dB. In particular, in Figure 5d is shown the correct range bin trajectory for each detected target while from Figure 5e it is possible to notice some errors, due to the SLIM algorithm, in determining the correct Doppler position. Regarding range bins track estimation, it is also interesting to note that the first target has a double range bin transition between the second and the third scans. This test case is equivalent to which one where targets are spatially distant but with comparable velocities.

3.3. Case Study 3

For the third test case, we consider the two targets moving in opposite directions that means the first target moves approaching the radar while the second one moves away from it. In detail, at the beginning of the radar scanning, the first target is located in the range bin $r_{1_1} = 10$ (starting from a relative position of 15 m) with velocity $v_1 = +39.63$ m/s (corresponding to the frequency Doppler $f_{d_1} = +366.80$ Hz and Doppler point $d_{1_1} = 46$) while the second target is in the range bin $r_{2_1} = 6$ (starting from a relative position of 15 m) and moves with velocity $v_2 = -25.76$ m/s (corresponding to the frequency Doppler $f_{d_2} = -238.42$ Hz and Doppler point $d_{2_1} = 13$). According this configuration targets trajectories cross during the radar scans. Figure 6 shows the specific results for SINR = 9 dB. In Figure 6d is shown the correct Range bins retrieval for both targets during radar scans while Figure 6e reveals the Doppler points estimation during scanning. It is possible to note a single error occurring in Doppler point retrieval only for the first target in the fist scan.
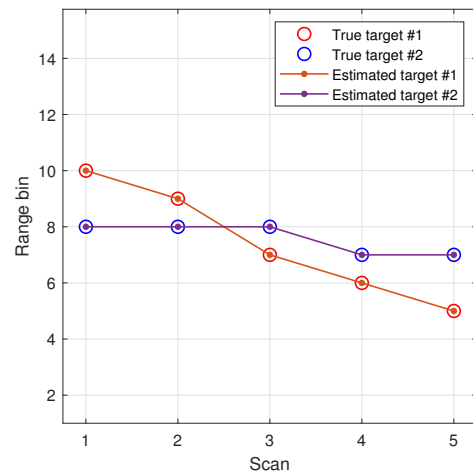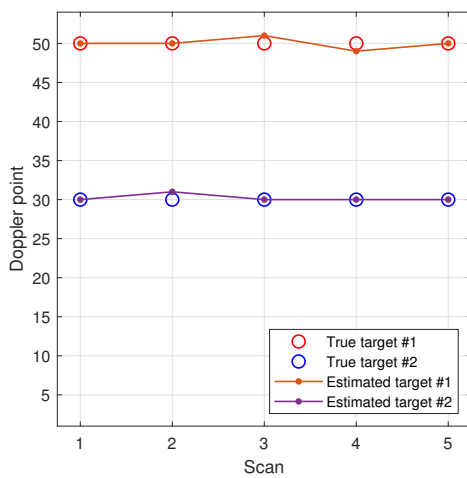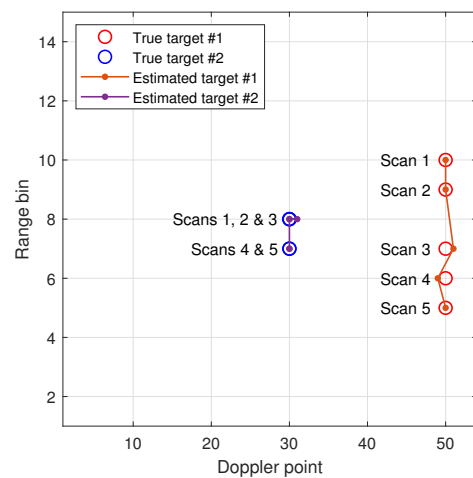
(**a**) *Measured data*
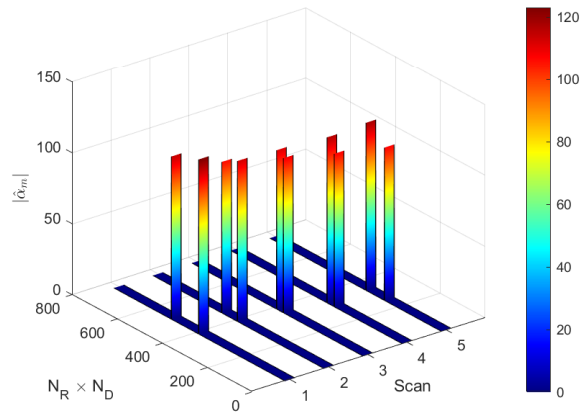
(**b**) *SLIM output*
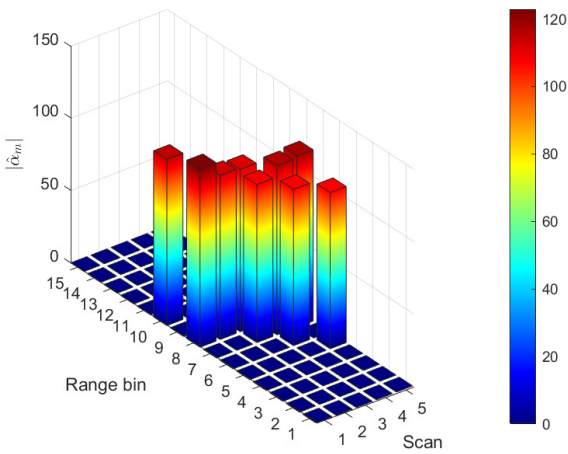
(**c**) *Stacked Doppler points*

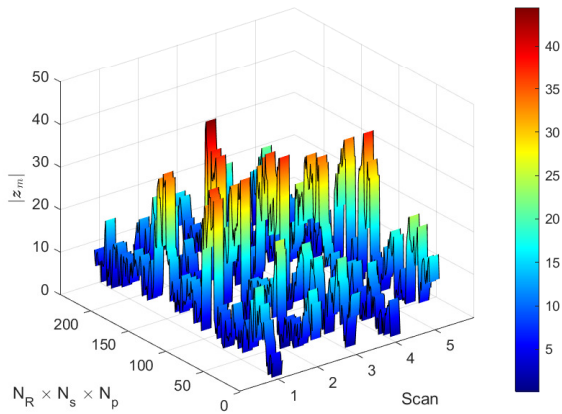(**d**) *Range track* vs. *scan*
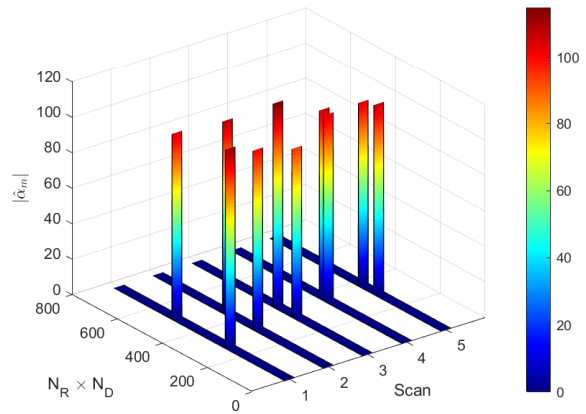

(**e**) *Doppler point* vs. *scan*

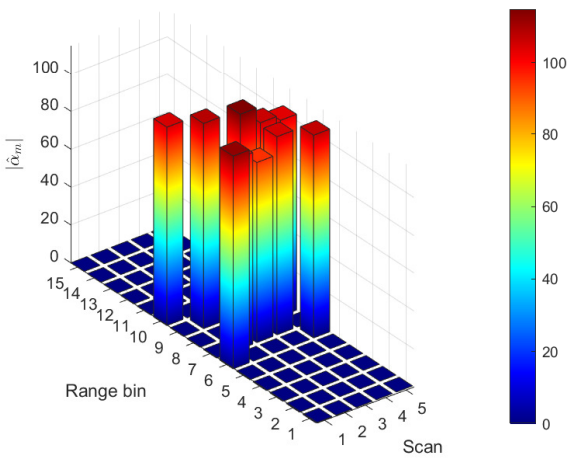(**f**) *Range track* vs. *Doppler*

Figure 5. SLIM-based TBD for the second test case. The radar scenario provides two targets approaching the radar with overlapping trajectories. Simulation is performed for a single Monte Carlo trial at SINR = 9 dB.
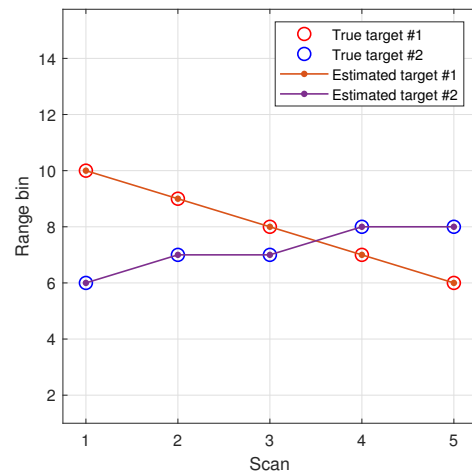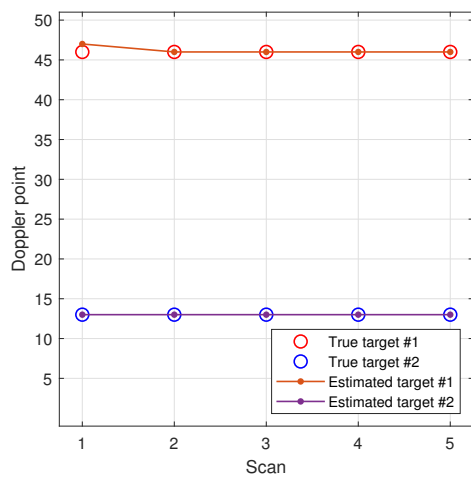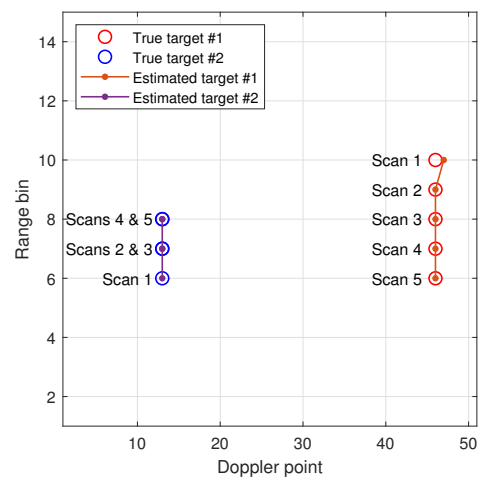
(**a**) *Measured data*

(**b**) *SLIM output*

(**c**) *Stacked Doppler points*

(**d**) *Range track* vs. *scan*

(**e**) *Doppler point* vs. *scan*

(**f**) *Range track* vs. *Doppler*

Figure 6. SLIM-based TBD for the third test case. The radar scenario provides two targets that move in different directions overlapping trajectories. Simulation is performed for a single Monte Carlo trial at SINR = 9 dB.

4. Discussion

In the previous Section, three different scenarios are presented to show the tracking performance of the proposed SLIM-based TBD method. Here, we focus on the performance of the overall proposed processing chain compared with other TBD strategies based on alternative compressive sensing algorithms. More precisely, we resort to the OMP and CoSaMP [43,44,54] the main benchmarks in the context of the compressed sensing. In what follows, we refer to these competitors as OMP-based TBD and CoSaMP-based TBD algorithm. The performance evaluation of the proposed processing chain is carried out resorting to Monte Carlo simulation with 1000 independent trials for each algorithm.

We take into account the scenario as defined in the first test case and, to avoid long simulation times, some parameter intervals are suitably reduced without losing in generality. Specifically, in this numerical simulations we consider $N_R = 12$, $N_D = 25$ and $M = 4$.

Simulation results are shown in Figure 7. More precisely, in Figure 7a we plot the P_{dt} versus SINR, while in Figure 7b we plot RMSE versus the SINR, for the three algorithm respectively. By inspecting Figure 7a, it can be seen that the proposed SLIM-based TBD algorithm obtains unity probability of track detection at the lowest SINR (15 dB) compared with the OMP-based TBD and CoSaMP-based TBD algorithms which is almost the same (21 dB). This means that TBD strategy based on SLIM guarantees a significant of gain of 6 dB respect to the matching pursuit approaches. This also shows that the proposed SLIM-based TBD algorithm is resilient to the measurement noise. It can be seen from Figure 7b that the proposed algorithm obtains the lowest RMSE compared with the other two algorithms. The proposed TBD strategy can perform perfect detection of targets with measurement noise when the SINR is above 15 dB.

Finally, we describe the software and the hardware to do used for the numerical results and performance analysis. Specifically, we use a workstation equipped with an Intel® Xenon® CPU E-5-1620 v2 @ 3.70 GHz and 16 GB RAM. MATLAB® scripts are implemented and developed to simulate the overall processing chain.

(**a**)　　　　(**b**)
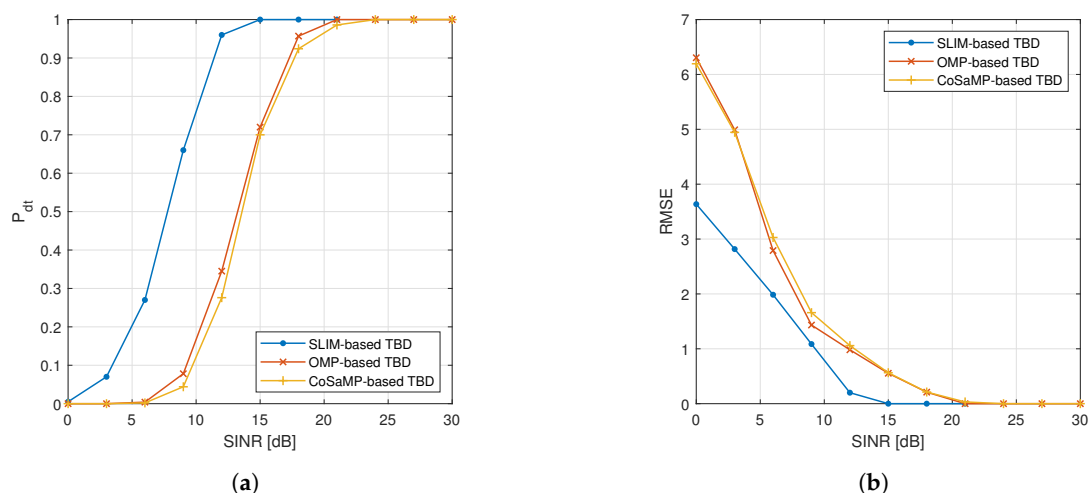
Figure 7. Probability of detection and correct track estimation versus SINR in subplot (**a**), and RMSE versus SINR in subplot (**b**) computed resorting to Monte Carlo simulation.

5. Conclusions

In this paper we have proposed a novel approach to multiple targets TBD for air surveillance radar applications based on sparse data processing. In the first part of the paper we have presented the problem formulation, then we have proposed the relative processing chain. This is based on three stages: the sparse learning, the track formation, and the ad hoc detector. To test it, we have analysed initially three operating scenarios which differ for targets configuration in terms of relative positions and velocities. The adopted

radar parameters refer to the Lockheed Martin FPS-117, a military radar system currently in use. The performance assessment has highlighted that the proposed detection architecture exhibits satisfactory performances and, more importantly, the SLIM-based detector guarantees significant gain in terms of detection probability and track estimation, compared with the natural competitors OMP-based and CoSaMP-based TBD algorithms. Moreover, simulation results show that the proposed processing chain can track and detect multiple targets successfully.

Regarding future developments, considerations could be, firstly, the extension of the present work to the range-azimuth-Doppler domain. Furthermore, particular configurations could be taken into consideration, such as targets with different scattering properties or that change their trajectory. Improvements in the processing chain mainly concern the clustering algorithm for the track formation. In fact, specific constraints could be adopted in the *k-means* algorithm application to achieve better clustering capabilities, or, in addition, it could be useful to investigate other different clustering or path finding algorithms, such as Viterbi.

Abbreviations

The following abbreviations are frequently used in this manuscript:

CoSaMP	Compressive Sensing Matching Pursuit
ICM	Interference Covariance Matrix
iid	independent and identically distribuired
LRT	Likelihood Ratio Test
OMP	Orthogonal Matching Pursuit
SINR	Signal to Interference plus Noise ratio
SLIM	Sparse Learning via Iterative Minimization
PRF	Pulse Repetition Frequency
PRI	Pulse Repetition Interval
RMSE	Root Mean Square Error
SRT	Scan Repetition Time
TBD	Track-Before-Detect

References

1. Mahler, R.P.S. *Statistical Multisource-Multitarget Information Fusion*; Artech House, Inc.: Norwood, MA, USA, 2007.
2. Bar-Shalom, Y. *Tracking and Data Association*; Academic Press Professional, Inc.: Cambridge, MA, USA, 1987.
3. Skolnik, M.I. *Introduction to Radar Systems*, 2nd ed.; McGraw Hill Book Co.: New York, NY, USA, 1980.
4. Blackman, S.S. *Multiple-Target Tracking with Radar Applications*; Dedham: Andover, MA, USA, 1986.
5. Kramer, J.D.R.; Reid, W.S. Track-before-detect processing for an airborne type radar. In Proceedings of the IEEE International Conference on Radar, Arlington, VA, USA, 7–10 May 1990; pp. 422–427, doi:10.1109/RADAR.1990.201202.
6. Ristic, B.; Arulampalm, S.; Gordon, N. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*; Artech House: Norwood, MA, USA, 2004.
7. Barniv, Y.; Bar-Shalom, Y. Dynamic programming algorithm for detecting dim moving targets. In *Multitarget-Multisensor Tracking: Advanced Applications*; Bar-Shalom, Y., Ed.; Artech House: Norwood, MA, USA, 1990; Chapter 4.
8. Hyoungjun, I.; Taejeong, K. Optimization of multiframe target detection schemes. *IEEE Trans. Aerosp. Electron. Syst.* **1999**, *35*, 176–187.
9. Johnston, L.A.; Krishnamurthy, V. Performance analysis of a dynamic programming track before detect algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **2002**, *38*, 228–242.
10. Pohlig, S.C. An algorithm for detection of moving optical targets. *IEEE Trans. Aerosp. Electron. Syst.* **1989**, *25*, 56–63.
11. Tonissen, S.M.; Bar-Shalom, Y. Maximum likelihood track-before-detect with fluctuating target amplitude. *IEEE Trans. Aerosp. Electron. Syst.* **1998**, *34*, 796–809.
12. Davey, S.J.; Rutten, M.G.; Cheung, B. A comparison of detection performance for several Track-Before-Detect algorithms. In Proceedings of the 2008 11th International Conference on Information Fusion, Cologne, Germany, 30 June–3 July 2008; pp. 1–8.
13. Salmond, D.J.; Birch, H. A particle filter for track-before-detect. In Proceedings of the 2001 American Control Conference (Cat. No.01CH37148), Arlington, VA, USA, 25–27 June 2001; Volume 5, pp. 3755–3760.
14. Boers, Y.; Driessen, J.N. Particle filter based detection for tracking. In Proceedings of the 2001 American Control Conference (Cat. No.01CH37148), Arlington, VA, USA, 25–27 June 2001; Volume 6, pp. 4393–4397.
15. Fallon, M.F.; Godsill, S. Acoustic Source Localization and Tracking Using Track Before Detect. *Trans. Audio Speech Lang. Proc.* **2010**, *18*, 1228–1242, doi:10.1109/TASL.2009.2031781.
16. Boers, Y.; Driessen, H. Particle filter track-before-detect application using inequality constraints. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1483–1489.
17. Saha, S.; Boers, Y.; Driessen, H.; Mandal, P.K.; Bagchi, A. Particle based MAP state estimation: A comparison. In Proceedings of the 2009 12th International Conference on Information Fusion, Seattle, WA, USA, 6–9 July 2009; pp. 278–283.
18. Vo, B.; Vo, B.; Pham, N.; Suter, D. Joint Detection and Estimation of Multiple Objects From Image Observations. *IEEE Trans. Signal Process.* **2010**, *58*, 5129–5141.
19. Wong, J.; Vo, B.T.; Vo, B.N.; Hoseinnezhad, R. Multi-Bernoulli based track-before-detect with road constraints. In Proceedings of the 2012 15th International Conference on Information Fusion, Singapore, 9–12 July 2012; pp. 840–846.
20. Jevtić, M.; Marković, K.; Mikluc, D.; Mišić, B.; Pajić, T. Real-time implementation of target tracking system for air surveillance radar applications. In Proceedings of the 2013 11th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS), Niš, Serbia, 16–19 October 2013; Volume 2, pp. 557–560.
21. Buzzi, S.; Lops, M.; Venturino, L. Track-before-detect procedures for early detection of moving target from airborne radars. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 937–954.
22. Orlando, D.; Venturino, L.; Lops, M.; Ricci, G. Track-Before-Detect Strategies for STAP Radars. *IEEE Trans. Signal Process.* **2010**, *58*, 933–938.
23. Orlando, D.; Ricci, G.; Bar-Shalom, Y. Track-Before-Detect Algorithms for Targets with Kinematic Constraints. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 1837–1849.
24. McDonald, M.; Balaji, B. Impact of Measurement Model Mismatch on Nonlinear Track-Before-Detect Performance for Maritime RADAR Surveillance. *IEEE J. Ocean. Eng.* **2011**, *36*, 602–614.
25. Herman, M.A.; Strohmer, T. High-Resolution Radar via Compressed Sensing. *Trans. Sig. Proc.* **2009**, *57*, 2275–2284, doi:10.1109/TSP.2009.2014277.
26. Candes, E.J.; Tao, T. Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? *IEEE Trans. Inf. Theory* **2006**, *52*, 5406–5425.
27. Candes, E.; Wakin, M. An Introduction To Compressive Sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30, doi:10.1109/MSP.2007.914731.
28. Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306.
29. Baraniuk, R.; Steeghs, P. Compressive Radar Imaging. In Proceedings of the 2007 IEEE Radar Conference, Waltham, MA, USA, 17–20 April 2007; pp. 128–133.
30. Liu, Z.; Wei, X.Z.; Li, X. Adaptive clutter suppression for airborne random pulse repetition interval radar based on compressed sensing. *Prog. Electromagn. Res.* **2012**, *128*, 291–311.
31. Yang, M.; Zhang, G. Compressive Sensing Based Parameter Estimation for Monostatic MIMO Noise Radar. *Prog. Electromagn. Res. Lett.* **2012**, *30*, 133–143, doi:10.2528/PIERL12010702.

32. Grossi, E.; Lops, M.; Venturino, L. A track-before-detect procedure for sparse data. In Proceedings of the 2012 IEEE Statistical Signal Processing Workshop (SSP), Arbor, MI, USA, 4–7 August 2012; pp. 772–775.

33. Liu, J.; ChongZhao, H.; XiangHua, Y.; Feng, L. Compressed Sensing Based Track before Detect Algorithm for Airborne Radars. *Prog. Electromagn. Res.* **2013**, *138*, 433–451, doi:10.2528/PIER13022006.

34. Liu, J.; Li, X.; Xu, S.; Zhuang, Z. ISAR Imaging of Non-Uniform Rotation Targets with Limited Pulses via Compressed Sensing. *Prog. Electromagn. Res. B* **2012**, *41*, 285–305.

35. Wei, S.J.; Zhang, X.L.; Jun, S.; Liao, K.F. Sparse array microwave 3-D imaging: Compressed sensing recovery and experimental study. *Prog. Electromagn. Res.* **2013**, *135*, 161–181, doi:10.2528/PIER12082305.

36. Li, J.; Zhang, S.; Chang, J. Applications of compressed sensing for multiple transmitters multiple azimuth beams SAR imaging. *Prog. Electromagn. Res.* **2012**, *127*, 259–275, doi:10.2528/PIER12021307.

37. Chen, J.; Gao, J.; Zhu, Y.; Yang, W.; Wang, P. A novel image formation algorithm for high-resolution wide-swath spaceborne SAR using compressed sensing on azimuth displacement phase center antenna. *Prog. Electromagn. Res.* **2012**, *125*, 527–543, doi:10.2528/PIER11121101.

38. Wei, S.J.; Zhang, X.L.; Jun, S. Linear array SAR imaging via compressed sensing. *Prog. Electromagn. Res.* **2011**, *117*, 299–319, doi:10.2528/PIER11033105.

39. Wei, S.J.; Zhang, X.L.; Jun, S.; Xiang, G. Sparse reconstruction for SAR imaging based on compressed sensing. *Prog. Electromagn. Res.* **2010**, *109*, 63–81, doi:10.2528/PIER10080805.

40. Tan, X.; Roberts, W.; Li, J.; Stoica, P. Sparse Learning via Iterative Minimization With Application to MIMO Radar Imaging. *IEEE Trans. Signal Process.* **2011**, *59*, 1088–1101.

41. Addabbo, P.; Aubry, A.; De Maio, A.; Pallotta, L.; Ullo, S.L. HRR profile estimation using SLIM. *IET Radar Sonar Navig.* **2019**, *13*, 512–521.

42. Yan, L.; Addabbo, P.; Hao, C.; Orlando, D.; Farina, A. New ECCM Techniques Against Noiselike and/or Coherent Interferers. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 1172–1188.

43. Tropp, J.A.; Gilbert, A.C. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. *IEEE Trans. Inf. Theory* **2007**, *53*, 4655–4666, doi:10.1109/TIT.2007.909108.

44. Needell, D.; Tropp, J. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal.* **2009**, *26*, 301–321, doi:10.1016/j.acha.2008.07.002.

45. Galati, G.; Mazzenga, F.; Naldi, M. *Elementi di Sistemi Radar*; Aracne: Rome, Italy, 1996.

46. Gelman, A.; Carlin, J.B.; Stern, H.S.; Rubin, D.B. *Bayesian Data Analysis*, 2nd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2004.

47. Robert, C.P.; Casella, G. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*; Springer: Berlin/Heidelberg, Germany, 2005.

48. Stoica, P.; Selen, Y. Model-order selection: A review of information criterion rules. *IEEE Signal Process. Mag.* **2004**, *21*, 36–47.

49. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137.

50. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*; Le Cam, L.M., Neyman, J., Eds.; University of California Press: Berkeley, CA, USA, 1967; Volume 1, pp. 281–297.

51. Ball, G.H.; Hall, D. A clustering technique for summarizing multivariate data. *Behav. Sci.* **1967**, *12*, 153–155.

52. Lockheed Martin Corporation. *AN/FPS-117 Long-Range Air Surveillance Radars*; Technical Report; Lockheed Martin Corporation: Washington, DC, USA, 2013.

53. Bartolic, J. Near Field Measurements of the FPS-117 Solid-State Phased Array Antenna (Part I)-Self-Oscillating Antennas and Arrays (Part II). Available online: https://www.researchgate.net/publication/305969338 (accessed on 11 January 2021).

54. Karakuş, C.; Gürbüz, A.C. Comparison of iterative sparse recovery algorithms. In Proceedings of the 2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey, 20–22 April 2011; pp. 857–860, doi:10.1109/SIU.2011.5929786.