LEARNING-BASED SPACE-TIME ADAPTIVE PROCESSING

by

Alaa El Khatib

A Thesis Presented to the Faculty of the
American University of Sharjah
College of Engineering
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Electrical Engineering

Sharjah, United Arab Emirates

June 2013

Approval Signatures

We, the undersigned, approve the Master's Thesis of Alaa El Khatib.

Thesis Title: Learning-Based Space-Time Adaptive Processing

Signature **Date of Signature**
(dd/mm/yyyy)

Dr. Hasan S.Mir
Assistant Professor, Department of Electrical Engineering
Thesis Advisor

Dr. Khaled Assaleh
Professor, Department of Electrical Engineering
Thesis Co-Advisor

Dr. Mohammed El-Tarhuni
Associate Professor, Department of Electrical Engineering
Thesis Committee Member

Dr. Naoufel Werghi
Associate Professor, Department of Electrical Engineering
Khalifa University (Sharjah Campus)
Thesis Committee Member

Dr. Mohammed El-Tarhuni
Head
Department of Electrical Engineering

Dr. Hany El-Kadi
Associate Dean
College of Engineering

Dr. Leland Blank
Interim Dean
College of Engineering

Dr. Khaled Assaleh
Director of Graduate Studies

Acknowledgments

After giving due praise to our Lord, the Most Merciful, for all His blessings, I would like to thank my thesis advisors, Dr. Hasan Mir and Dr. Khaled Assaleh, for their support and guidance through the completion of this thesis. I also would like to thank the department of Electrical Engineering at the American University of Sharjah for providing me with the opportunity to be part of their graduate program. Last but not least, I would like to express my gratitude to my parents, family and friends for the undeserved support and compassion with which they overwhelm me all the time.

To those unwilling

to give up on their dreams…

Abstract

The probability of target detection in airborne-radar missions depends on the target signal-to-interference-plus-noise ratio. In order to maximize the probability of detection, it is necessary to maximize the target signal-to-interference-plus-noise ratio by suppressing the interference to an acceptable level. The type of interference encountered by airborne radars is of a distinctive nature; it spreads in both the spatial and the temporal dimensions, exhibiting a relationship between the amount of Doppler shift in the temporal dimension and the spatial direction of the echo source. In practical situations, the characteristics of the interference present are not known a priori; thus, they have to be estimated in real-time. The two-dimensional nature of the unknown interference dictates the use of two-dimensional adaptive filters to suppress it. Such filters are called space-time adaptive filters. In practical situations, the amount of secondary training data needed to accurately compute the space-time adaptive filter weights is not available. Thus, it is necessary to develop algorithms that are able to suppress the unknown interference with limited amounts of training data. Many such algorithms have been developed over the past few decades, each with its own advantages and drawbacks. In this report, a new algorithm called "learning-based space-time adaptive processing" is proposed. The proposed algorithm transforms the filtering problem into a pattern classification problem, where the secondary data is used to train a classifier, instead of estimating the interference characteristics. The results show that the proposed algorithm achieves a higher target signal-to-interference-plus-noise ratio than space-time adaptive processing when the amount of secondary data is limited and the target power is not extremely low compared to interference power. The proposed system is able to overcome two more problems faced by space-time adaptive processing: target-cancellation and clutter variation. Finally, a cascaded system of space-time adaptive processing followed by learning-based space-time adaptive processing is proposed. The cascaded system offers a performance gain compared to the individual systems.

Abbreviations

ADC	-	Analogue-to-Digital Converter
CPI	-	Coherent Processing Interval
FDA	-	Fisher Discriminant Analysis
LBSTAP	-	Learning-Based Space-Time Adaptive Processing
LSMI	-	Loaded Sample Matrix Inversion
ML	-	Maximum Likelihood
MTI	-	Moving Target Indication
PCA	-	Principle Component Analysis
PRF	-	Pulse Repetition Frequency
PRI	-	Pulse Repetition Interval
RCS	-	Radar Cross-Section
SINR	-	Signal-to-Interference-plus-Noise Ratio
STAP	-	Space-Time Adaptive Processing
ULA	-	Uniform Linear Array

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Moving Target Indication (MTI) Overview

Moving target indication refers to the utilization of radars in the detection of moving targets [1]. The probability of detection, P_D, mainly depends on two factors: the probability of false alarm, P_F, and the signal-to-interference-plus-noise ratio (SINR). Therefore, to maximize the probability of detection, it is necessary to maximize the SINR at the radar receiver output.

The main focus of this report is airborne radar systems. For such systems, the interference that needs to be suppressed in order to maximize the SINR consists of two components: clutter and jamming.

Jamming signals are correlated in the spatial dimension but uncorrelated in the temporal dimension. Therefore, they appear centered at a certain azimuth angle but spread over all frequencies. Jamming signals are easier to suppress than clutter signals; they can be filtered by placing a null in the radar array pattern in the direction of the jammer.

Clutter refers to unwanted returns from the ground or sea that can be mistaken as a target. For airborne radars, these returns are Doppler-shifted as a consequence of the relative motion of the radar platform with respect to ground. This means that the clutter main lobe will no longer be centered at zero Doppler frequency; hence, if the clutter power is high enough, it will appear as a moving target.

The geometry of the platform motion with respect to earth creates a dependency of the Doppler frequency of clutter returns on the azimuth angle of the clutter source (patch). The characteristics of clutter and the other signals at hand will be discussed in later sections; however, it is enough for now to note the 2D spatial-temporal nature of the clutter signal. This 2D nature brings about the need for 2D filters.

Space-time adaptive processing (STAP) refers to the technique employed in interference environments to achieve the required increase in SINR for target detection. As the name implies, STAP encompasses the use of 2D spatiotemporal digital filters, whose weights are adaptively computed in real-time in accordance with estimated parameters of an unknown changing interference [2]. In general, STAP algorithms estimate the covariance matrix of interference at a certain range gate (which corresponds to distance on the ground) from secondary data, and use the estimated matrix in the computation of the filter weights. The secondary data is obtained from adjacent range gates, assuming they have the same characteristics as the range gate under test. The output of the filter is compared to a threshold value and accordingly a decision is made on the presence of a target.

1.2 Limitations of STAP

The secondary data used by STAP in estimating the interference covariance matrix should satisfy the following conditions:

a. They should be target-free (to avoid what is known as target-cancellation).
b. They should have the same statistical properties as the range gate under test.

The violation of these conditions will result in either a poor estimate of the interference covariance matrix, and accordingly a poor filter output, or the loss of the target as a consequence of target-cancellation.

Moreover, according to the rule derived by Reed, Mallett and Brennan in their 1974 paper [3], known as the RMB rule, the output SINR of STAP depends on the amount of secondary data used in the estimation of the interference covariance matrix. In many practical situations, the amount of secondary data needed to achieve an acceptable performance level is not available.

1.3 Contributions of this Thesis

Many variations to the original STAP algorithms have been developed to overcome the aforementioned limitations. These algorithms mainly aim to obtain a more accurate estimate of the interference, and to reduce the amount of secondary data needed as well. Some of these algorithms will be discussed along with their own limitations as well in the coming sections.

This thesis contributes the following:

a. Developing a new algorithm for target detection based on pattern classification techniques. The proposed technique is given the name learning-based space-time adaptive processing (LBSTAP). It is shown in this report that the proposed method has the following advantages:

   i. It outperforms the original STAP algorithm in terms of SINR in cases where the amount of secondary data is limited.

   ii. It is able to overcome the problem of target-cancellation.

   iii. It is more robust than STAP in changing clutter environments.

   These advantages are evident when the target power is not drastically lower than interference power (interference power should not be more than 20 dB higher than the target power; i.e. SIR ≥ -20 dB).

b. Developing a cascaded system, where a variation of the proposed algorithm is used after STAP. The cascaded system is able to outperform both individual systems. The cascaded system is able to overcome the problem faced by LBSTAP in the detection of low-power targets.

1.4 Thesis Arrangement

The rest of the thesis report is organized as follows. Chapter 2 provides a background on the original STAP algorithm, other variations of STAP and their limitations. Chapter 3 is an overview of pattern classification theory. Chapter 4 introduces the proposed technique and the selected classifier. Chapter 5 presents simulation results comparing LBSTAP with the original STAP algorithm and shows the performance of the cascaded system as well. Finally, Chapter 6 concludes the thesis report.

Chapter 2: Space-Time Adaptive Processing

In this chapter[*], a basic STAP system will be presented. The system is mainly made of an array receiver, a down-convertor, a matched filter and an analogue-to-digital converter (ADC) for sampling. Models of the transmitted signal, clutter, jammers, noise and potential targets will be presented.

After describing the basic STAP system, other STAP algorithms will be presented, such as reduced-rank STAP, knowledge-aided STAP and deterministic STAP.

2.1 Problem Geometry

A pulsed Doppler radar is mounted on a moving platform with velocity $\boldsymbol{v_p}$ and at an altitude h. A uniform linear array (ULA) is used as a receiver and is made of *N* channels uniformly separated by a distance *d*. Figure 1 shows the geometry of the moving platform with respect to ground. θ represents the elevation angle, and φ represents the azimuth angle. The radar array illuminates a patch on the ground defined by the direction vector $\boldsymbol{a}$, given by

$$\boldsymbol{a} = \cos\theta \sin\varphi \;\boldsymbol{i} + \cos\theta \cos\varphi\, \boldsymbol{j} + \sin\theta \;\boldsymbol{k}, \tag{1}$$

where $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ are unit vectors in the directions of the positive x-, y- and z-axes, respectively.

[*] This chapter benefited from the technical report on STAP by Ward [2].

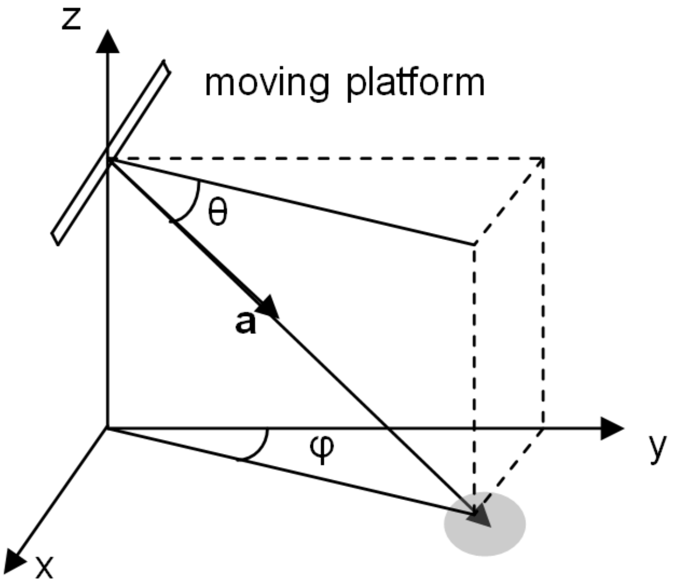

Figure 1 Radar problem geometry

The array elements are assumed to have their reference point at the first channel, as shown in Figure 2. Accordingly, the *N* elements are labeled *n = 0, 1, 2,…, N−1.* Then, the distance of each element from the reference point is *nd.* Moreover, the array elements have the same radiation pattern.

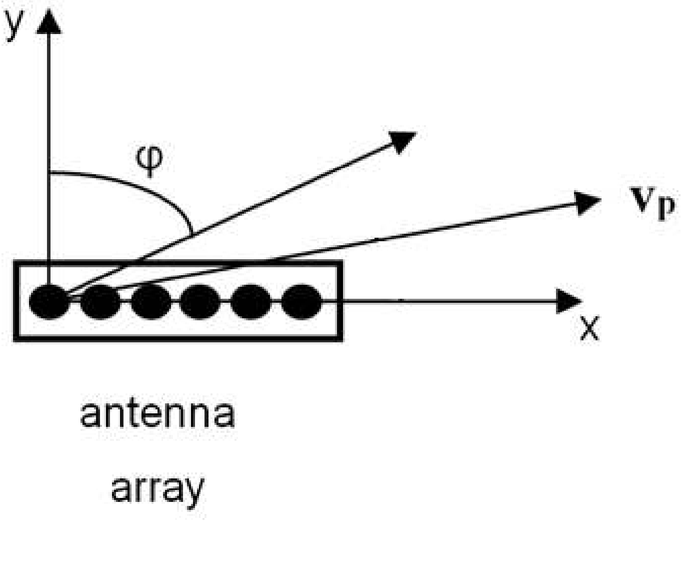

Figure 2 Array elements

The N receive channels provide N degrees of freedom for filtering in the spatial dimension (i.e. N degrees of freedom for the array pattern).

As previously mentioned, a jammer signal appears at the radar receiver centered at a certain azimuth angle. Hence, if jamming was the only interference present, it can be filtered out using the N receive channels alone (given that the total number of independent jammers is less than or equal to $N-1$). If these conditions – jamming being the only interference present, and the number of spatial channels is greater than the number of independent jammers by at least one – hold, then maximizing the SINR at the receiver output is a matter of placing nulls in the array pattern at the azimuth/elevation angles of the jammers. This process is called spatial beamforming, and it will be explained in further details in the coming sections.

Unfortunately, in MTI radars, jamming is never present without clutter. Hence, the use of 1D spatial filters is not sufficient. Still, understanding the 1D spatial filtering process is important for understanding the 2D filtering process – STAP.

2.2 Signal Models

The radar transmits a pulsed waveform at a fixed pulse repetition frequency (PRF). M pulses are sent per coherent processing interval (CPI). The total duration of each CPI is equal to MT_r, where T_r refers to the pulse repetition interval (PRI). This concept is illustrated in Figure 3.

The M pulses provide the filter with M temporal degrees of freedom. Therefore the total number of degrees of freedom in a fully-adaptive STAP system is equal to the total number of spatial receive channels, N, multiplied by the total number of pulses per CPI, M – that is, a total of NM degrees of freedom.

The transmitter transmits at a carrier frequency $f_o = c/\lambda_o$, where c is the speed of light and λ_o is the carrier wavelength. Each pulse has a bandwidth B. (Note that it is assumed that the radar's instantaneous bandwidth, B, is less than f_o, and at the same time larger than the pulse repetition frequency, PRF. This assumption is important, as it affects the way the signals are correlated in both the spatial and temporal dimensions).

The transmitted signal per CPI can be represented as follows:

$$s(t) = a_t p(t) e^{j(\omega_o t+\beta)} \tag{2}$$

$$p(t) = \sum_{m=0}^{M-1} b(t - mT_r), \tag{3}$$

where a_t is the signal amplitude, b is the pulse waveform, β is an arbitrary phase shift, ω_o is the carrier frequency in radians per second and t represents time.

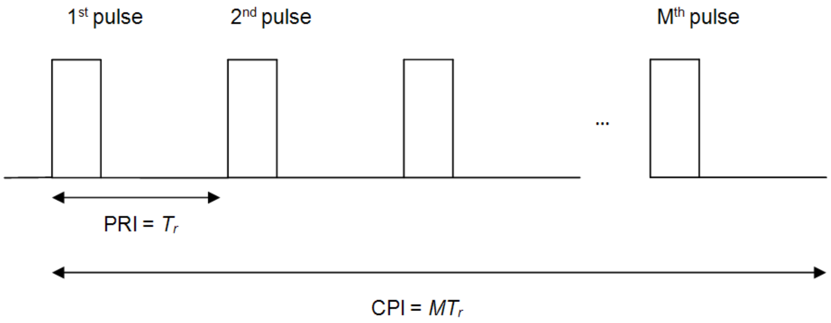

Figure 3 CPI (rectangular pulses are used only for illustration)

Given the geometry described earlier, the echo received at the n^{th} channel will be delayed with respect to the echo received by the reference channel by a time delay expressed by:

$$\tau_n = n\frac{d}{c}\cos\theta \sin\varphi. \tag{4}$$

Define the variable γ, called the spatial frequency, as:

$$\gamma = \frac{d}{\lambda_o}\cos\theta \sin\varphi. \tag{5}$$

Then, combining equations (4) and (5), the time delay at the *n*th channel can be written as:

$$\tau_n = \frac{n\gamma}{f_o}. \tag{6}$$

From the time delay and frequency, the phase delay at the *n*th channel, ρ_n, can be computed as:

$$\rho_n = \omega_o \tau_n = 2\pi n\gamma. \tag{7}$$

Moreover, the echo will have another delay caused by the path to the reflector (target or ground) and back. This delay is given by:

$$\tau_o = \frac{2R}{c}, \tag{8}$$

where R is the distance between the radar and the reflector.

The received echo is Doppler-shifted according to the relative velocity of the reflecting surface (target or ground) with respect to the radar. The Doppler shift is given by:

$$f_D = 2\frac{v}{\lambda_o}, \tag{9}$$

where v is the surface's relative velocity. The Doppler frequency is usually presented normalized. The normalized Doppler frequency is given by:

$$\omega_D = \frac{f_D}{f_r}, \tag{10}$$

where f_r is the pulse repetition frequency (PRF).

At each channel, the received signal goes through three stages. First, it is down-converted to baseband. Next, the output from the down-conversion step is passed through a matched filter. Finally, the signal is sampled by an ADC at an appropriate sampling frequency.

Given the aforementioned formulations for the time delays and Doppler shift, the received signal at the output of each of the three stages can be defined as follows. The signal at the down-convertor output is given by:

$$s_{dc}(t) = \zeta p(t-\tau_o)e^{j2\pi f_d t}e^{j2\pi n\gamma}, \tag{11}$$

where ζ is a complex amplitude that includes the echo amplitude and all constant phase terms.

Next, the signal passes through a filter matched to the original pulse waveform *p(t)*. The output of this matched filter is given by:

$$s_m(t) = \zeta e^{j2\pi n\gamma}\sum_{m=0}^{M-1} e^{j2\pi m\omega_D}\chi(t-\tau_o-mT_r,\omega_D), \tag{12}$$

where χ is the waveform ambiguity function, given by:

$$\chi(\tau,f) = \int_{-\infty}^{\infty} p(\alpha)p^*(\alpha-\tau)\, e^{j2\pi f\alpha}d\alpha, \tag{13}$$

where * represents the conjugate operator.

Finally, after the sampling stage, and looking at the reflector's range gate, the signal at the n^{th} channel and m^{th} PRI interval simplifies to:

$$x_{nm} = \zeta e^{j2\pi n\gamma}e^{j2\pi m\omega_D}. \tag{14}$$

Looking at the samples from one PRI interval, the returns from the *N* spatial channels can be written as:

$$\boldsymbol{x}_m = [x_{0m}, x_{1m}, x_{2m}, \dots, x_{N-1m}]^T, \tag{15}$$

where ^{T} is the matrix transpose.

Equation (15) can be written as:

$$\boldsymbol{x}_m = \zeta e^{j2\pi m\omega_D}\boldsymbol{t}_s, \tag{16}$$

where $\boldsymbol{t}_s$ is an $N \times 1$ vector called the spatial steering vector. For a given array size, $\boldsymbol{t}_s$ depends on the spatial frequency, γ, which in turn depends on the azimuth and elevation angles of the look direction. The spatial steering vector is given by:

$$\boldsymbol{t}_s(\gamma) = \begin{bmatrix} 1 \\ e^{j2\pi\gamma} \\ e^{j4\pi\gamma} \\ \vdots \\ e^{j(N-1)2\pi\gamma} \end{bmatrix}. \tag{17}$$

Every range gate, l, is described by NM space-time samples. These samples are arranged in an $NM \times 1$ vector, called the space-time snapshot vector, which is given by:

$$\boldsymbol{x}_l(\gamma, \omega_D) = \zeta \begin{bmatrix} \{\boldsymbol{t}_s(\gamma)\} \\ \{e^{j2\pi\omega_D}\boldsymbol{t}_s(\gamma)\} \\ \vdots \\ \{e^{j2\pi(M-1)\omega_D}\boldsymbol{t}_s(\gamma)\} \end{bmatrix}. \tag{18}$$

Similar to the spatial steering vector, define an $M \times 1$ temporal steering vector, as:

$$\boldsymbol{t}_t(\omega_D) = \begin{bmatrix} 1 \\ e^{j2\pi\omega_D} \\ e^{j4\pi\omega_D} \\ \vdots \\ e^{j(M-1)2\pi\omega_D} \end{bmatrix}. \tag{19}$$

Using equations (18) and (19), the space-time snapshot vector representing range gate l can be written as:

$$\boldsymbol{x}_l(\gamma, \omega_D) = \zeta \boldsymbol{t}_t(\omega_D) \otimes \ \boldsymbol{t}_s(\gamma), \tag{20}$$

where $\otimes$ represents the Kronecker product operator. The space-time snapshot vector, then, can be divided into parts: a complex amplitude, ζ, and an $NM \times 1$ space-time steering vector, given by:

$$\boldsymbol{s}(\gamma, \omega_D) = \boldsymbol{t_t}(\omega_D) \otimes \boldsymbol{t_s}(\gamma). \tag{21}$$

The ADC acquires a total of L samples per PRI. Each sample corresponds to a certain range gate (which corresponds to distance). Overall, there are NMLsamples acquired per CPI. These samples are usually arranged in an $N \times M \times L$ cube, called the CPI data cube. Figure 4 illustrates this concept.

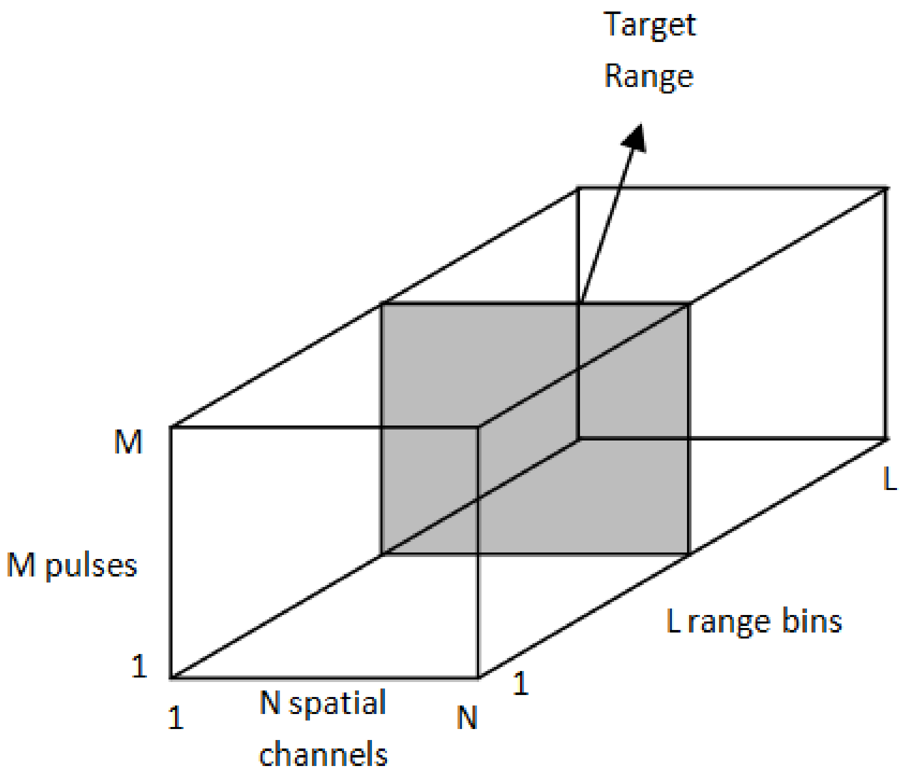

Figure 4 CPI data cube

The CPI data cube and how it is used in the filtering process will be discussed in the coming sections. The characteristics of the different types of signals (target, noise, jammer, clutter) are now discussed.

2.2.1 Target. A target is characterized by five main variables: its azimuth and elevation angles, relative velocity with respect to the radar platform, range (distance to the radar) and target power. The azimuth and elevation angles make up the target's spatial steering vector. Moreover, the relative velocity of the target determines its Doppler frequency, which in turn defines its temporal steering vector. The range of the target determines the time delay of the signal echo, which determines the position of the target in the range-gate dimension of the CPI data cube. Finally, the received target power depends on many variables defined in radar theory. These variables include the radar cross-section (RCS) of the target, the range of the target, the transmitted power, the array power gain, the radiation pattern, the radar's operating frequency and the receiver noise.

For a target with parameters shown in Table 1, the component of the space-time snapshot vector (derived from the received signal) at the target range gate is given by:

$$\boldsymbol{x}_t(\gamma_t, \omega_t) = \zeta_t \boldsymbol{t_t}(\omega_t) \otimes \ \boldsymbol{t_s}(\gamma_t), \tag{22}$$

where $\gamma_t = \frac{d}{\lambda_o} \cos\theta_t \sin\varphi_t$, $\omega_t = \frac{2\frac{v_t}{\lambda_o}}{f_r}$, and $\boldsymbol{t_t}$ and $\boldsymbol{t_s}$ are the temporal and spatial steering vectors respectively (d, λ_o and f_r are the array spacing, the operating wavelength and the pulse repetition frequency, respectively). Simply put, a target vector (the target component of the space-time snapshot vector) has two components: target amplitude and target steering vector.

Table 1 Target parameters

Target power *(assuming constant amplitude, i.e. not random)*	ζ_t^2
Target azimuth angle	φ_t
Target elevation angle	θ_t
Target velocity	v_t

2.2.2 Jamming. Jamming signals are received from other airborne or ground transmitters. They aim to increase the background noise power level at the radar receiver to prevent target detection. For that reason, jamming signals usually have very high power. There are different types of jamming; each type has its own tactical advantages and disadvantages. In this report, only barrage jamming is considered.

Barrage jamming cover all of the receiver's frequency range [4]. This means that the jammer power will be divided over a range of frequencies, and hence will decrease at each single frequency. Nonetheless, it can still prevent the detection of targets, depending on the power difference between the target and the jamming signals.

It is assumed that *1/B* is large compared to the signal's propagation time across the array (as mentioned earlier, $B < f_o$); there is no decorrelation across the radar array. On the other hand, the instantaneous bandwidth is assumed to be larger than the pulse repetition frequency [2]; the signal decorrelates from pulse to pulse. In other words, jamming signals appear to be concentrated at one angle (spatially-correlated), while they appear like thermal noise in the frequency domain, covering all frequencies (temporally-uncorrelated).

Given a single jammer with the parameters shown in Table 2, its component in the space-time snapshot vector is formulated as follows. Since the jammer signal is spatially-correlated, its spatial form will be defined by the spatial steering vector, given by:

$$\boldsymbol{t}_s(\gamma_u) = \begin{bmatrix} 1 \\ e^{j2\pi\gamma_u} \\ e^{j4\pi\gamma_u} \\ \vdots \\ e^{j(N-1)2\pi\gamma_u} \end{bmatrix}, \tag{23}$$

where γ_u is the jammer's spatial frequency, given by:

$$\gamma_u = \frac{d}{\lambda_o} \cos\theta_u \sin\varphi_u. \tag{24}$$

Since the jammer signal is uncorrelated in the temporal domain, its contribution to the snapshot vector from PRI, m, will have a random amplitude ζ_{um}. Therefore, the total contribution of a single jammer to the space-time snapshot vector has the form:

$$\boldsymbol{x}_u = \boldsymbol{\zeta}_u \otimes \ \boldsymbol{t}_{\boldsymbol{s}}(\gamma_u), \tag{25}$$

where $\boldsymbol{\zeta}_u$ is the jammer amplitude vector, given by:

$$\boldsymbol{\zeta}_u = \begin{bmatrix} \zeta_{u0} \\ \zeta_{u1} \\ \vdots \\ \zeta_{uM-1} \end{bmatrix}. \tag{26}$$

Table 2 Jammer parameters

Jammer power spectral density (W/Hz)	J_o
Jammer azimuth angle	φ_u
Jammer elevation angle	θ_u

Since a jammer is concentrated at one angle, it can be filtered out relatively easily by 1D filters (without the use of STAP). This can be done by placing a null in the array pattern in the estimated direction of the jammer.

The total number of jammers that can be filtered is determined by the total number of spatial degrees of freedom (which is equal to the total number of spatial channels in fully-adaptive STAP systems). In theory, the total number of degrees of freedom should be at least one plus the total number of independent jammers.

In the case of having N_j jammers, their total contribution to the space-time snapshot vector is given by:

$$\boldsymbol{x}_u = \sum_{n_j=1}^{N_j} [\boldsymbol{\zeta}_{u,n_j} \otimes \ \boldsymbol{t}_{\boldsymbol{s}}\left(\gamma_{u,n_j}\right)], \tag{27}$$

where $\boldsymbol{\zeta}_{u,n_j}$ and γ_{u,n_j} are the amplitude vector and spatial frequency, respectively, for the n_jth jammer.

2.2.3 Clutter. Clutter refers to signal returns (i.e. reflections) that are not considered of importance to the search/detection task of the radar. Accordingly, in an ideal situation, clutter signals should not appear at the radar output, due to their insignificance to the detection process. Therefore, the task of the processing stage at the radar receiver is to distinguish between actual targets and clutter returns, and consequently, filter the latter out.

Clutter returns, which come from reflections off the ground or sea, are Doppler-shifted due to the relative motion of the radar platform with respect to earth. Moreover, clutter power levels can sometimes be significantly higher than that of an actual target. This means that they can appear, and accordingly be mistakenly classified, as a moving target. Due to this, it is important to understand the nature of clutter signals, and be able to estimate their characteristics in real situations, in order to separate them from target signals. In this report, only ground clutter will be considered.

The Doppler shift of clutter returns depends on the azimuth angle of the clutter source. This dependency results in the spreading of clutter returns in both spatial and temporal dimensions. The 2D nature of clutter signals makes them relatively difficult to filter out compared to jamming signals.

Clutter signals are modeled as returns from discrete sources, called clutter patches; each of which has the same model as a target. For example, the clutter from a ring of radius r is modeled as N_p discrete clutter patches evenly distributed in azimuth. Each patch will have its own azimuth angle and random amplitude. The total clutter returns from the ring, then, will be the sum of the clutter returns from the N_p patches.

The radar's unambiguous range is given by:

$$R_u = \frac{c}{2f_r}, \tag{28}$$

and it refers to the range after which the returns from one pulse can be confused with the returns of the previous pulse, due to the fact that the time it takes for the two-way journey of a pulse will be more than the pulse repetition interval.

Moreover, the radar horizon range is approximated by [2]:

$$R_h \approx \sqrt{2r_e h}, \tag{29}$$

where r_e refers to the effective radius of earth (which can be assumed to be 4/3 the actual radius), and h is the radar altitude. The horizon range refers to the range at which the line between the radar and earth (clutter patch) will be tangent to the surface of earth. Radar signals after this range will not be reflected back to the receiver.

For a clutter ring at a range R_c (within the unambiguous range of the radar), one of two scenarios can take place:

a) $R_u > R_h$

If this condition holds, the number of clutter rings that can contribute to the clutter signal is either zero (in the case that R_c is larger than R_h) or one only. In this case the clutter is said to be unambiguous in range.

b) $R_u < R_h$

If this condition holds, N_r rings contribute to the total clutter signal, where N_r can be greater than one. In this case the clutter is said to be ambiguous in range.

Note that the number of discrete rings is also governed by the range resolution, which is given by:

$$\Delta R = \frac{c}{2B}, \tag{30}$$

and that the number of discrete patches per ring is determined by the choice of N_p in the chosen model.

In general, the contribution of clutter from range R_c to the total space-time snapshot vector of the range gate corresponding to R_c is derived as follows. The total number of rings is N_r, each of which is made of N_p discrete patches. Each patch is defined by an azimuth angle, φ_{rp}, elevation angle, θ_{rp}, and an amplitude, ζ_{rp}.

$$\boldsymbol{x}_c = \sum_{r=1}^{N_r} \sum_{p=1}^{N_p} [\zeta_{rp} \boldsymbol{t_t}(\omega_{rp}) \otimes \ \boldsymbol{t_s}(\gamma_{rp})], \tag{31}$$

where ω_{rp} is the normalized Doppler frequency of the rp^{th} patch given by equation (9) (the relative velocity will be determined by the platform alone, since the earth is assumed to be stationary), and $\gamma_{rp}$is the spatial frequency of the rp^{th} patch given by equation (5).

It can be shown from the geometry of the moving platform with respect to earth that the normalized Doppler frequency of the rp^{th} clutter patch is given by:

$$\omega_{rp} = \frac{2v_p}{\lambda_o} T_r \cos\theta_{rp} \sin\varphi_{rp}. \tag{32}$$

Substituting equation (5) into (32), the normalized Doppler frequency can be written in terms of the spatial frequency as:

$$\omega_{rp} = \frac{2v_p T_r}{d} \gamma_{rp}. \tag{33}$$

The dependency of the normalized Doppler frequency on the spatial of frequency of clutter results in a distinct shape for its spread over the 2D Doppler-azimuth space. If the clutter signal is plotted over the normalized Doppler and spatial frequency space, it will form what is known as clutter ridges. The clutter ridges have a slope given by:

$$slope = \frac{2v_p T_r}{d}. \tag{34}$$

Depending on its slope, the clutter ridge can cover part or all of the Doppler frequency range (which extends from –PRF/2 to PRF/2). If the slope of the clutter ridge increases beyond the point where it covers the whole frequency range exactly one time, higher frequencies start to appear; however, they fold over into the PRF frequency range. Beyond this point, the clutter is said to be ambiguous in Doppler. Figures 5 and 6 illustrate the concept of ambiguity of clutter in Doppler.

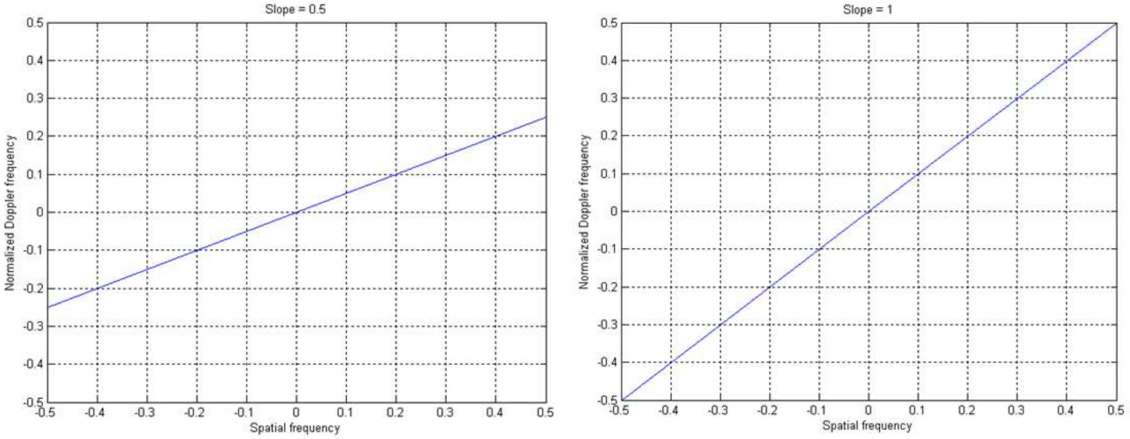

Figure 5 Unambiguous-in-Doppler clutter

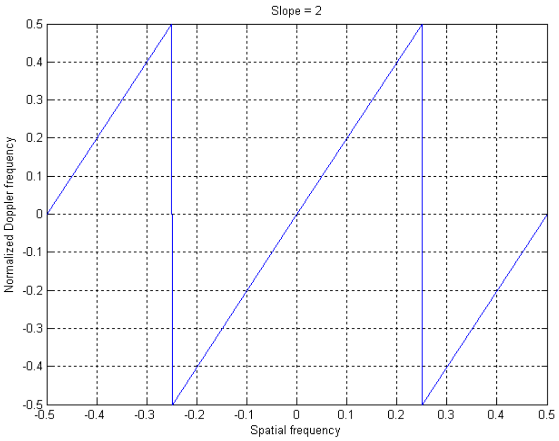

Figure 6 Ambiguous-in-Doppler clutter

2.2.4 Noise. Noise in this report refers to thermal noise inside the receiver elements only. Since each receiver channel processes the received signal separately at the first stages (down-converter, matched filter and sampling), the noise signal is assumed to be uncorrelated from one channel to another. Moreover, the noise samples are temporally uncorrelated (i.e. from pulse to pulse). This means that the noise will have a covariance matrix equal to an $NM \times NM$ identity matrix multiplied by the noise power. The noise power, η, is determined from the noise power spectral density, N_o, and the receiver bandwidth B as:

$$\eta = N_o B. \tag{35}$$

2.3 Fully-Adaptive STAP

In this section, the fully-adaptive STAP algorithm and architecture are described. Fully-adaptive STAP refers to STAP algorithms where all the degrees of freedom provided by the system are used in the filtering process.

So far, the need for 2D spatiotemporal filters has been justified by the nature of clutter returns. Next, the need for an adaptive system is discussed.

A radar array acquires $N \times M$ space-time samples corresponding to a certain range gate. If the properties of interference present are known a priori, the filtering process can be done as follows. Suppose that the only interference present is caused by N_j jammers. No clutter is present. Moreover, suppose that the space-time snapshot vectors describing the spatial-temporal position of each of the N_j jammers are known. Then, the 2D filter weights can be calculated by solving a set of independent equations for the values of the *NM* weights.

Define the filter weights as:

$$\boldsymbol{w} = [w_{11} \quad w_{21} \quad w_{31} \quad \cdots \quad w_{N1} \quad w_{12} \quad w_{22} \quad \cdots \quad w_{N2} \quad \cdots \quad w_{1M} \quad \cdots \quad w_{NM}]^T. \quad (36)$$

Moreover, let the space-time steering vector for jammer J_n be given by:

$$\boldsymbol{t}_n(\gamma_n, \omega_n) = \begin{bmatrix} \{\boldsymbol{t}_s(\gamma_n)\} \\ \{e^{j2\pi\omega_n}\boldsymbol{t}_s(\gamma_n)\} \\ \vdots \\ \{e^{j2\pi(M-1)\omega_n}\boldsymbol{t}_s(\gamma_n)\} \end{bmatrix}, \quad (37)$$

where γ_n and ω_n refer to the spatial and normalized Doppler frequencies of jammer J_n respectively.

Then, the filtering process reduces to placing nulls in the array pattern in the known directions of the N_j jammers. This can be done by setting the filter weights to the values found by solving the following set of equations:

$$\boldsymbol{w}'\boldsymbol{t}_1 = 0, \quad (38)$$

$$\boldsymbol{w}'\boldsymbol{t}_2 = 0, \tag{39}$$

$$\vdots$$

$$\boldsymbol{w}'\boldsymbol{t}_{N_j} = 0, \tag{40}$$

$$\boldsymbol{w}'\boldsymbol{t}_x = 1, \tag{41}$$

where $'$ denotes conjugate transpose, and $\boldsymbol{t}_x$ is the space-time steering vector describing the position of the target in the spatial-temporal 2D space. Thus, the described set of equations is solved for the weight values that maximize the output of the receiver array in the direction of the target, while, simultaneously, placing nulls in the directions of the target. In fact, the filter in this case does not make use of the M temporal degrees of freedom. Therefore, the maximum number of independent jammers that can be filtered is $N - 1$.

The same concept can be extended to include clutter-contaminated data. However, in practical situations, the clutter and jamming properties are not known a priori, and thus need to be estimated adaptively from the surrounding environment; here comes the need for space-time adaptive processing.

A STAP algorithm uses the data provided by a CPI data cube to filter the space-time snapshot vector representing a certain range gate. Let the target be present at range gate l_t. The target range gate is represented by one slice of the CPI data cube containing $N \times M$ space-time samples. Likewise, each of the adjacent slices of the data cube corresponds to a certain range on the ground away from the target range.

The snapshot vector representing the target range gate can be written as:

$$\boldsymbol{x}_{l_t} = \boldsymbol{x}_t + \boldsymbol{x}_n + \boldsymbol{x}_c + \boldsymbol{x}_j, \tag{42}$$

where $\boldsymbol{x}_n$, $\boldsymbol{x}_c$ and $\boldsymbol{x}_j$ refer to the snapshot vectors representing noise, clutter and jamming respectively. As mentioned earlier, the task of the filter is to maximize the output target SINR, and thus maximize the probability of detection. The filter output SINR can be written as:

$$SINR = \frac{|w'x_t|^2}{E\{|w'x_i|^2\}}\,, \tag{43}$$

where $\boldsymbol{x}_i$ refers to the sum of clutter, jamming and noise signals. Noise and interference signals are random variables; therefore, the expectation of the output noise and interference power is used.

The weight vector $\boldsymbol{w}$ is not a random variable. Thus, equation (43) can be written as:

$$SINR = \frac{|w'x_t|^2}{\boldsymbol{w}'E\{x_i x_i{}'\}\boldsymbol{w}}\,. \tag{44}$$

Furthermore, the expectation given by:

$$E\{\boldsymbol{x}_i \boldsymbol{x}_i{}'\} \tag{45}$$

is the noise-plus-interference covariance matrix, $\boldsymbol{R}$.

Therefore, the task of maximizing the filter output SINR reduces to the solving the following optimization problem:

$$\max_{\{w\}} \frac{|w'x_t|^2}{\boldsymbol{w}'\boldsymbol{R}\boldsymbol{w}}\,. \tag{46}$$

This maximization problem given by (46) is solved for the optimum filter weights, $\boldsymbol{w}$, and the solution is given by:

$$\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{s}, \tag{47}$$

where $\boldsymbol{s}$ refers to the target space-time steering vector.

The filter weights are optimum for an exact covariance matrix. However, as mentioned earlier, the exact covariance matrix is not known, and an estimate of it has to be used. Thus, the weight vectors in this case will be referred to as suboptimum weights.

The covariance matrix is estimated using the rest of the range gates in the CPI data cube, excluding a few guard cells on either side of the target range gate (see Figure 7), since exact target range is not known in advance, and only an approximate location is available. Moreover, the target returns show range side lobes. Therefore, the guard cells are left out of the covariance matrix estimation process to avoid target-cancellation. The covariance matrix is estimated by:

$$\boldsymbol{R}_e = \frac{1}{N_e}\sum_{n_e=1}^{N_e} \boldsymbol{x}_{n_e}\boldsymbol{x}_{n_e}{}', \tag{48}$$

where $\boldsymbol{x}_{n_e}$ refers to the secondary (training) data obtained from the rest of the CPI data cube, and N_e refers to the total number of training range gates used. The accuracy of the covariance matrix estimate (the degree with which it resembles the actual noise and interference present at the target range gate) depends on the number of training range gates used in the estimation process.

Reed, Mallett and Brennan (RMB) derived a quantitative description of the effect of the estimation accuracy of the covariance matrix on the output SINR [3]. The ratio between the expected SINR, using an estimated covariance matrix, and the optimum SINR, using the exact covariance matrix, is given by:

$$\rho = \frac{N_e - N_{DoF} + 2}{N_e + 1}, \tag{49}$$

where N_{DoF} refers to the number of degrees of freedom, and N_e refers to the number of range gates used in the estimation. Equation (49) holds if the following conditions are satisfied:

a) The training range gates are target-free.

b) The training range gates are independent and identically distributed (i.i.d) Gaussian random variables, having the same statistical characteristics as the range gate under test.

Equation (49), also known as the RMB rule, predicts that for a 3dB SINR loss (compared to the optimum SINR) approximately $2NM$ training range gates satisfying the above conditions are needed for the covariance matrix estimation. It can be seen that as the number of training range gates approaches infinity, the ration, ρ, approaches 1 (optimum SINR).

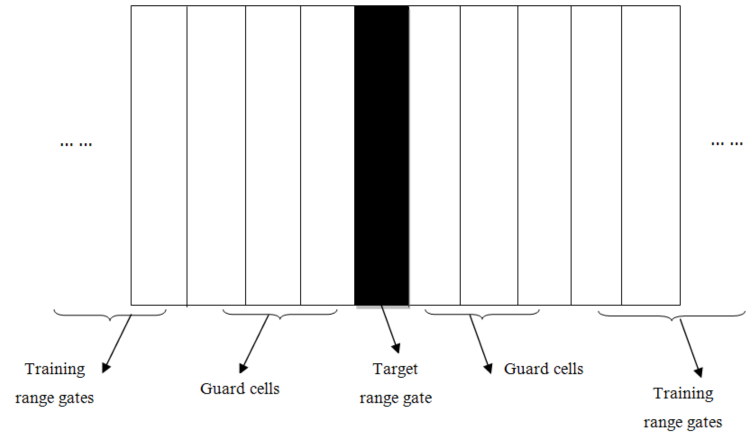

Figure 7 Training and guard range gates

After estimating the covariance matrix, the suboptimum weight vector corresponding to the direction defined by the space-time steering vector ***s*** is computed using equation (47). Then, the output of the filter is computed as:

$$y = \boldsymbol{w}'\boldsymbol{x}_{l_t}. \tag{50}$$

The magnitude of the output, $|y|$, is compared to a threshold value, T_h, to decide on the presence of a target. In other words, the system has to choose one of the following two hypotheses:

a) H_0: Target is present ($\boldsymbol{x}_{l_t} = \boldsymbol{x}_t + \boldsymbol{x}_n + \boldsymbol{x}_c + \boldsymbol{x}_j$).

b) H_1: Target is not present ($\boldsymbol{x}_{l_t} = \boldsymbol{x}_n + \boldsymbol{x}_c + \boldsymbol{x}_j$).

Note that the weight computed according to equation (47), and consequently the output value and the decision made, all correspond to one point on the 2D azimuth-Doppler space and one range gate. In reality, a radar array scans a range of azimuth angles. The target velocity (and consequently its Doppler frequency) is not known in advance. Therefore, at each azimuth angle, the system has to compute a suboptimum weight vector that maximizes the output SINR for each of the possible Doppler frequencies. The range of Doppler frequencies extends from –PRF/2 to PRF/2, and the number of Doppler bins is usually set to *M* (the number of degrees of freedom in the temporal dimension). After that, an output value that corresponds to each of the possible Doppler bins is computed using equation (50). Comparing the *M* outputs to the threshold value, the decision can be made on the Doppler frequency (and velocity) of the target (if present). This process is repeated at every azimuth angle scanned, and at every range gate searched.

Figure 8 shows a summary of a fully-adaptive STAP algorithm. From the CPI data cube, the target space-time snapshot vector and the secondary (training) data are extracted. The training data are used to estimate the covariance matrix at the target range gate. Next, the estimated covariance matrix is fed to the weight computation block. Note that the diagram illustrates the process for one weight computation only. In reality, this block carries many computations to find the suboptimum weights for all the Doppler bins. Accordingly, the next block computes the output for all the Doppler bins. All outputs are compared to a threshold value to decide on the velocity of the target (if present at the azimuth angle represented by the CPI data cube).

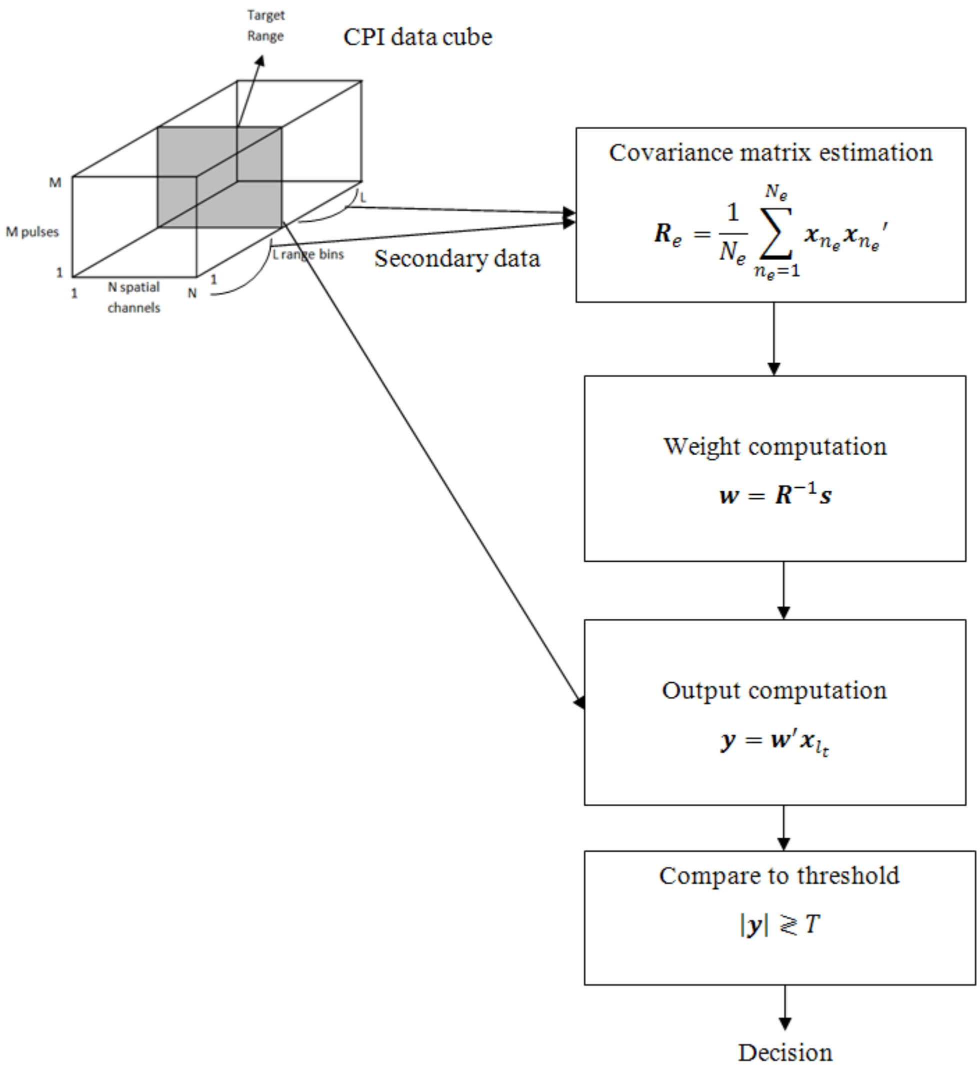

Figure 8 STAP summary

A plot of the output power (in dB) is usually plotted over the Doppler-range space as shown in Figure 9. The plot shows a target at a range of around 138 km and with a Doppler frequency of −80 Hz.

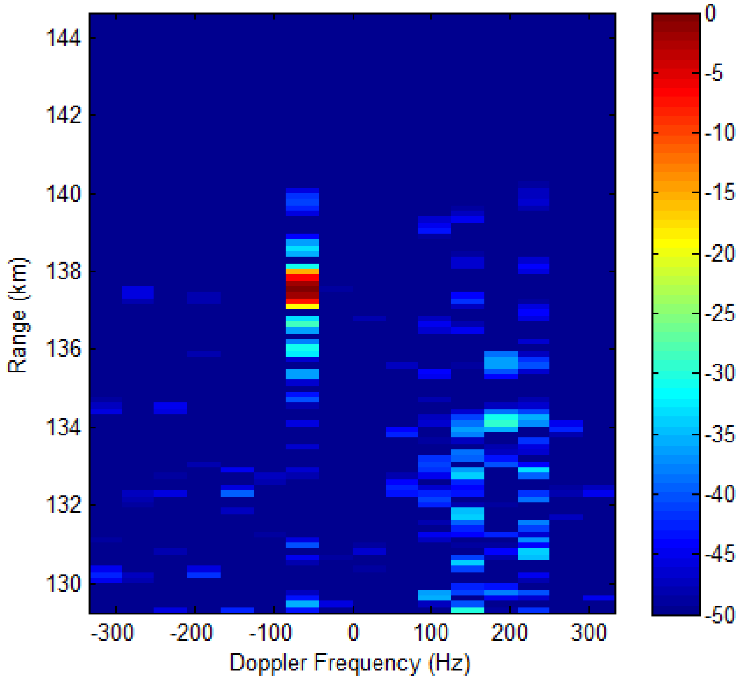

Figure 9 Normalized output power plot

2.4 Literature Review

Since the early work done on STAP by Brennan and Reed in the 1970s [5], much research has been conducted to overcome three major difficulties. The three difficulties, which have been touched upon earlier in the report, are:

a. The amount of secondary data needed for an acceptable estimate of the interference covariance matrix is usually unavailable. As mentioned earlier, fully-adaptive STAP requires *2NM* secondary training range gates for an output SINR within 3 dB of the optimum SINR (which is achieved using the exact covariance matrix). For example, a 224-dimensional snapshot vector requires the use of 448 secondary range gates. Depending on the operating frequency and the sampling rate, this number of range gates translates to tens of kilometers on the ground – and this leads to the second difficulty.

b. The secondary data has to be homogenous (having the same statistical properties as the range gate under test) and target-free. This requirement is usually not completely satisfied, due to different sources that result in clutter heterogeneity. These sources include internal clutter motion (such as sea echoes), weather conditions, discrete clutter sources, moving clutter sources, and other sources [6]. Clutter heterogeneity results in inaccurate interference covariance matrix estimation from secondary data.

c. The third difficulty is the computational load involved in fully-adaptive STAP. The computational complexity of fully-adaptive STAP is of the order $O(N^3M^3)$ [7].

2.4.1 Reduced-Rank STAP. Many reduced-rank STAP algorithms have been proposed over the past years to alleviate the effects of these difficulties. The general idea involved is the transformation of the data snapshot vectors and all the subsequent processing to a lower-dimensional space using a transformation matrix $\boldsymbol{T}$ [2]. The design of the transformation matrix, $\boldsymbol{T}$, has been the main subject of reduced-rank STAP research. For a given snapshot vector, x, the reduced-rank snapshot vector, $\tilde{x}$, is given by:

$$\tilde{x} = \boldsymbol{T}x. \tag{51}$$

The transformation matrix is a $P \times NM$ matrix, where P is desired to be significantly less than NM. Accordingly, a reduced-rank, $P \times P$ covariance matrix, $\widetilde{\boldsymbol{R}}$, is estimated from the transformed secondary snapshot vectors using the same procedure applied in full-rank STAP. The rest of the steps then will follow in the same way: compute the reduced-dimensional weight vectors and apply them to the reduced dimension test snapshot vector, and finally, compare the output to a threshold value.

The effectiveness of reduced-rank techniques is brought about by the fact that the interference covariance matrix is generally low-rank [2]. This low-rank nature of the covariance matrix is sometimes reduced by Eigen spectrum spreading, which can be caused by internal clutter motion, receiver instabilities and diffused multipath [8]. However, in general, studies show that reduced-rank STAP can outperform full-rank STAP when the amount of secondary data is limited [9], though this can come at the expense of some loss in adaptivity [2] and/or interference cancellation performance [7].

The amount of secondary data needed for reduced-rank algorithms depends on the interference rank and not the snapshot vector dimension. Thus, assuming low-rank interference, the amount of secondary data needed is reduced. This reduction also reduces the computational load on the processor. Moreover, it can be seen that the effect of heterogeneity in clutter increases as the amount of secondary data increases (since the range covered on ground would increase). Thus, for dimensionality reduction techniques, the three difficulties appear to be correlated; they are overcome together.

The design of the transformation matrix, $\boldsymbol{T}$, can be divided into two types: data-dependent and data-independent. Data-dependent techniques achieve better results than the latter; however, this comes at the expense of computational complexity [9]. One of the first data-dependent techniques used was using principle component (PC) analysis. In PC STAP, the data vectors are projected onto a lower dimensional subspace based on the Eigen decomposition of the interference-plus-noise covariance matrix [7]. The dominant Eigen vectors define the interference subspace [10]. Adaptivity can take place in the interference subspace [11] [12], or in the complementary subspace [13]. Other data-dependent algorithms include the parametric adaptive matched filter [14] [15] and the multistage Wiener filter [16].

Data-independent reduced-rank algorithms include joint domain localization (JDL) algorithms [7]. In general, JDL algorithms work to transform the received data to the angle-Doppler space using only a few angular and Doppler bins centered around the desired angle-Doppler direction [17]. In this way, the data vectors are reduced from NM-dimensional to N_aM_d-dimensional vectors, where N_a and M_d are the number of bins in the angular and Doppler dimensions, respectively. Accordingly, the amount of secondary data needed is reduced.

2.4.2 Diagonal Loading. Loaded sample matrix inversion (LSMI) or diagonal loading refers to the addition of a weighted NM-dimensional identity matrix to the full-rank interference covariance matrix. LSMI has a close behavior to reduced-rank STAP [9]; specifically, it works similar to PC algorithms [18]. However, diagonal loading results in less-deep interference nulls compared to PC methods [18].

2.4.3 Knowledge-Aided STAP. It is important to note that there is no one “best” technique to use out of the abovementioned techniques; each technique has its own pros and cons [17]. Knowledge-aided STAP (KA-STAP) refers to the utilization of a priori knowledge of the interference environment and data from other sensors and information sources to enhance the filtering process [19] [20]. The filtering process can be enhanced by selecting the most suitable algorithm based on such knowledge. Moreover, a priori knowledge and sensor data can be used to adjust the probability of false alarm, PRF and other parameters, depending on the environment [17].

One problem faced by KA-STAP algorithms is the negative effect of inaccurate a priori knowledge. Assuming a priori knowledge is given in terms of an interference covariance matrix, it is necessary to search for the optimum combination of the a priori covariance matrix and the estimated covariance matrix obtained using secondary data [21] [22]. The combination is done to alleviate the effects on the filtering process in the case of having inaccurate a priori knowledge. In other words, it is important to maximize the gain obtained from a priori knowledge and, at the same time, minimize the risk arising from the knowledge being potentially-inaccurate.

2.4.4 Direct Data Domain (D3). Direct data domain methods were developed to overcome the problem of nonhomogeneous interference environments [17] [23]. D^3 methods are non-statistical methods that work on data from the range gate of interest only, and thus, they overcome the problem of limited secondary data.

D^3 algorithms assume exact knowledge of the target Doppler frequency and angle [24]. Given this knowledge, the phase shift from one spatial channel to another, p_s, and the phase shift from one pulse to another, p_t, can be found. Assuming that the total signal at the m^{th} pulse and the n^{th} channel is x_{nm}, the signals given by:

$$x_{nm} - {p_s}^{-1} x_{n+1,m} \tag{52}$$

$$x_{nm} - {p_t}^{-1} x_{n,m+1} \tag{53}$$

should be interference-plus-noise signals. Accordingly, D^3 algorithms work to minimize the power in these signals and, at the same time, maximize the power in the target direction [17].

D^3 algorithms are effective in suppressing interference from discrete sources and quickly-varying clutter. However, given their non-statistical nature, they are not as effective in suppressing homogenous interference [17]. Moreover, D^3 algorithms significantly deteriorate in cases of the inaccurate target information [24].

Chapter 3: Pattern Classification

In this chapter, an overview of pattern classification theory is provided. The different stages involved in the design of a pattern classification system are presented. After that, linear classifiers are introduced. The chapter is concluded by a description of two dimensionality-reduction techniques: principle component analysis (PCA) and Fisher discriminant analysis (FDA).

3.1 Overview of Pattern Classification Systems

In general, pattern classification refers to the task of recognizing discriminating patterns in objects and, accordingly, classifying these objects into separate groups (called classes) based on their defining properties (called features). The applications of pattern classification techniques extend into numerous fields. They are used in biomedical applications, military applications, industrial applications, security checks, speech recognition and many other applications.

The design of a general pattern classification system consists of the following stages:

a. Data collection.
b. Pre-processing.
c. Feature extraction.
d. Classifier selection.
e. Training.
f. Testing.

The data collection stage is the first step. In this stage, the raw data that will be used in the training and testing stages is collected. This set of data has to be representative for each class. Subsequently, in the pre-processing stage, the useful data is separated from the rest of the data. After that, in the feature extraction stage, the features that will be used in designing the classifier are extracted. The features have to be discriminating attributes. That is, they must have the capability to distinguish the different classes from each other. Moreover, they have to be as few as possible, to reduce the computational complexity of the system. Figure 10 illustrates the distribution of two classes (represented by blue and pink dots) using two features. As seen in the plot on the left, Feature 1 is not a suitable feature for discriminating between the two classes shown. On the other hand, Feature 2 offers a clear discrimination capability for the two classes.

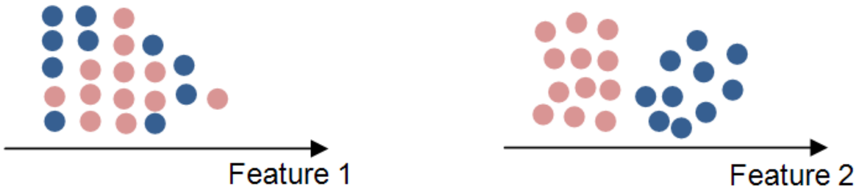

Figure 10 Class distributions

For each data sample, the features are arranged in one vector called the feature vector. Hence, all processing after the feature extraction stage will not deal with the actual physical objects, but with feature vectors describing the data samples.

After choosing the appropriate classifier, the classifier is trained using the training data and then tested used a different set of data. A different set of data is used in the testing stage to ensure that the classifier can also attain a high classification rate for data not used in the training stage.

The following example clarifies these stages further. Suppose that a system is required to recognize 20 different persons using their facial information. The data collection stage then is a set of pictures of faces for each person in varying environments (light, background, angle, etc…). In the pre-processing stage, the faces are separated from the background (segmented). The discriminating features can be chosen by an expert (if they are easy to recognize, such as length, weight, etc…) or using dimensionality-reduction techniques, which select the dimensions of the feature space that exhibit the highest discrimination capacity. The data set is then split into two sets: training set and testing set. Finally, the two sets are used as shown in Figure 11, to train and test the chosen classifier.

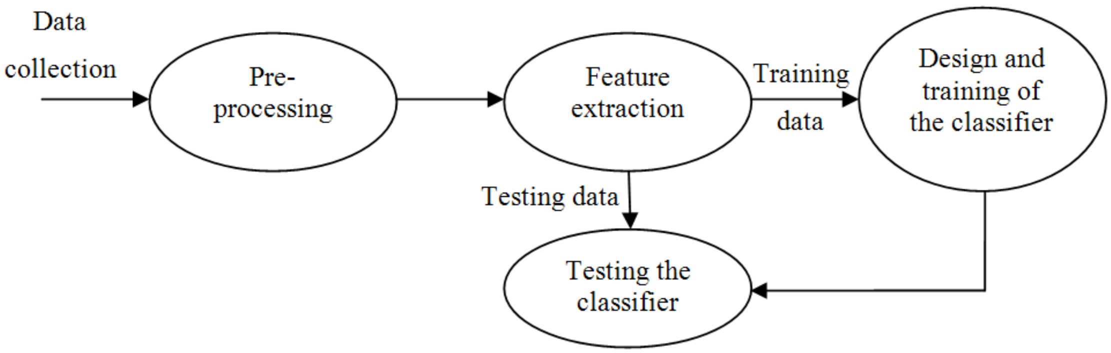

Figure 11 Classifier design

Classification methods fall into different categories: parametric and non-parametric methods. In parametric methods, the shape of the distribution of the different classes in the feature space is known a priori. In such a case, the training data is used to estimate the parameters of the distributions of the classes using parameter estimation techniques such as maximum likelihood (ML). In other cases, the parameters of the distributions are also given a priori. If so, then the training phase can be eliminated and an optimum classifier can be designed using Bayesian theory.

On the other hand, many applications (such as the subject of this report) fall into the second category and require non-parametric methods for designing an appropriate classifier. In such cases, no information is given about the distributions of the classes a priori; the only given is the data samples themselves. The data samples can be sometimes given without any information about their corresponding classes. These cases fall under what is known as unsupervised learning [25], where clustering techniques have to be used to estimate the number of classes from the given data and assign the data samples to the different classes.

In non-parametric methods, the data samples can be used to estimate the class distribution [26], and based on the estimated distributions design the classifier using parametric methods. If $\boldsymbol{x}$ is the pattern that is to be classified, then a discriminant function can be defined as "a function of the pattern $\boldsymbol{x}$ that leads to a classification rule" [27]. Each discriminant function represents a class. The shape of the discriminant functions are either known a priori or selected by assumption. Accordingly, training a classifier in non-parametric techniques reduces to finding the parameters of the discriminant functions. Estimating the parameters of discriminant functions is less complex than estimating the shape and parameters of class distributions. And thus, in many applications, using discriminant functions is favored over using distribution functions.

The simplest discriminant function is the linear discriminant function. Therefore, the first step in designing a classifier based on discriminant functions is to attempt to use a linear machine (a classifier that uses linear discriminant functions [27]). A linear classifier is optimum for Gaussian-distributed classes with equal covariance matrices. However, it can be used for other distributions as long as the classification rate is acceptable. If it is not, then nonlinear discriminant functions can be used to increase the classification rate.

3.2 Linear Classifier

The application of classification techniques to MTI radars falls under non-parametric classification. In this section, the theory for a linear classifier is provided. Nonlinear classifiers will be introduced in the next chapter.

Linear classifiers assume that the classes are linearly-separable. In a linear classifier, each class is represented by a linear discriminant function of the form:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_o, \tag{54}$$

where $\boldsymbol{x}$ represents a feature vector, $\boldsymbol{w}$ is a weight vector and w_o is a constant bias value. Equating the discriminant functions representing two classes, the decision boundary separating them is found.

Figure 12 illustrates the use of a linear classifier in a two-class problem, where the feature space is two-dimensional.

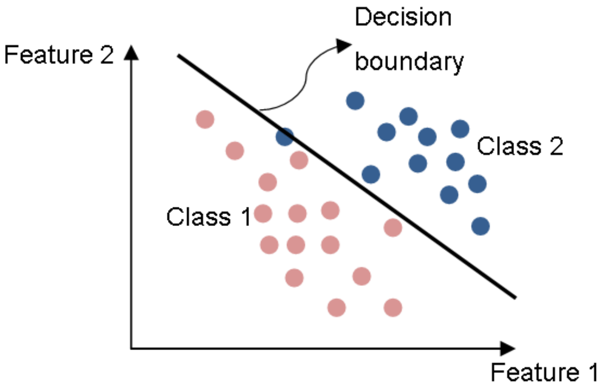

Figure 12 Linear classifier

The decision boundary in this example is a line. For a one-dimensional, two-class problem, the decision boundary would be a point. For a three-dimensional feature space, it would be a plane. For higher-dimensional spaces, the decision boundary would be a hyperplane.

To train a linear classifier, the decision boundaries separating the classes have to be determined. This can be done by different optimization methods. One method is to use a gradient descent algorithm, to find the weight vector that minimizes the error between a target output vector and the actual output vector of the classifier. For example, the target output vector for a class-1 data point can be assigned an output vector $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$, and $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ for a class-2 data point. The error that has to be minimized is the difference between the target output vector and the actual output vector constructed from the values of the discriminant functions of the two classes for the current weight vector and data point. Another method for designing a linear classifier is to minimize the mean-squared error (MSE) using the pseudoinverse of the matrix constructed from the feature vectors [26]. This method is presented in Section 4.2.

3.3 Dimensionality Reduction

Dimensionality reduction refers to the transformations applied on a vector space to reduce its dimension. Dimensionality reduction can be done for representation purposes or for the goal of extracting discriminating feature subspace from the full feature space.

3.3.1 Principle Component Analysis (PCA). Principle component analysis transforms the feature space into a lower-dimensional subspace that provides the most accurate representation of the data samples, given the new dimension. PCA does not incorporate class information in finding the new subspace. Thus, it is not always the best dimensionality reduction technique to use for feature extraction.

PCA is done through Eigen decomposition of the data covariance matrix. A PCA algorithm projects the sample data points (irrespective of their class) onto a lower-dimensional space defined the by the vectors pointing in the direction of greatest variance in the data. In other words, PCA projects the data points onto the subspace defined by the Eigen vectors corresponding to the largest Eigen values.

3.3.2 Fisher Discriminant Analysis (FDA). Fisher discriminant analysis is a dimensionality reduction technique that is designed for classification purposes. Unlike PCA, FDA incorporates class information in the computation of the lower-dimensional space. The class information included is the mean and variance of the each class. Thus, FDA ensures that the new subspace maximizes the distances between class means and, at the same time, minimizes the variance of each class. This ensures that the classes are centered at distant points and, at the same time, concentrated about their centers.

The main drawback of FDA is that it limits the dimension of the new subspace by one less than the number of classes. Therefore, even though PCA is not designed for pattern classification, it can outperform FDA in cases where the data is so complex that it cannot be represented by the subspace dimension allowed by FDA.

Chapter 4: Learning-Based Space-Time Adaptive Processing

In this chapter, the proposed learning-based space-time adaptive processing (LBSTAP) system architecture will be described. First, a description of how the problem is transformed into a classification problem is provided. Next, a comparison between different classifiers and dimensionality-reduction techniques is provided. Finally, a description of the proposed cascaded system (STAP followed by LBSTAP) is provided.

4.1 From STAP to LBSTAP

The main task of the processing stage, be it STAP or LBSTAP, is to maximize the SINR, in order to increase the probability of detection. In STAP, this translates into a filtering process, where secondary data is used to estimate the present noise and interference, and accordingly suppress them. This is not the case in LBSTAP.

LBSTAP transforms the filtering process into a classification problem. This is done as follows. Testing one range gate for the presence of a target and identifying its Doppler frequency (if present) translates, using pattern classification terminology, into classifying the range gate under test into one class out of all the possible classes. As the radar receiver scans over a range of azimuth angles, at each angle, the number of classes is equal to the number of Doppler frequency bins covering the range from –PRF/2 to PRF/2. The number of frequency bins is set equal to the number of pulses per coherent processing interval (CPI), M. This value is chosen because it is equal to the number of temporal degrees of freedom. And thus, having a higher number of classes will not contribute to having better differentiation ability; it will only increase the smoothness of the output. In other words, it is the highest Doppler resolution possible, given a fixed PRF.

Each class represents one Doppler frequency bin out of the possible bins. An extra class is added to represent the case where no target is present. Thus, the first hypothesis, H_0, symbolizing the presence of a target, is represented by M classes covering the

Doppler frequency range. The second hypothesis, H_1, symbolizing the absence of a target is represented by one class.

The training phase entails the modeling of these classes. Each class (except the target-free class) is made of two parts: a target and interference-plus-noise. High-power targets are artificially-generated, using a point-target model, with different frequencies corresponding to their designated classes. The number of training vectors used per class is limited by the amount of secondary data available in the CPI data cube. The secondary data is divided among the different classes to represent the interference-plus-noise part. That is, each training vector representing any class (with the exception of the target-free class) will be the sum of an artificially-generated target, with a Doppler frequency falling within the range of frequencies covered by that class, and a space-time snapshot vector coming from the secondary range gates to represent interference-plus-noise.

These training vectors are stacked together, each as a row vector, in one data matrix $\boldsymbol{X}_{tr}$. A target matrix, $\boldsymbol{Y}_{tar}$, is constructed from binary row vectors, with each vector corresponding to one class. For example, the target vector representing class 5 out of 8 available classes will have the form:

$$[0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]. \tag{55}$$

Having defined the data matrix and the target matrix, the training can be completed, using an appropriate classifier, to determine the outcome of the training phase: the weight matrix $\boldsymbol{W}$.

After the training phase is completed, the space-time snapshot vector corresponding to the range gate under test is passed through the classifier. The output of the classifier is a value representing the degree with which the test vector fits into each of the classes modeled in the training phase. Similar to the final step in STAP, the outputs are compared to a threshold value to decide on the presence of a target and, if present, its Doppler frequency.

4.2 LBSTAP Training and Testing

In this section, the training and testing stages of LBSTAP are described, assuming a linear classifier is used. The application of nonlinear classifiers will be discussed in the next section of this chapter.

Let the secondary data in a CPI data cube be represented by:

$$\boldsymbol{X}_{sd} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \boldsymbol{x}_3^T \\ \vdots \\ \boldsymbol{x}_{N_e}^T \end{bmatrix}, \tag{56}$$

where $\boldsymbol{x}_{n_e}(1 \leq n_e \leq N_e)$ are $NM \times 1$ column vectors, each representing one range gate of the secondary data. Thus, $\boldsymbol{X}_{sd}$ is an $N_e \times NM$ matrix, representing the interference-plus-noise part of the training data.

Using the target model, given by:

$$\boldsymbol{x}_t(\gamma_t, \omega_t) = \zeta_t \boldsymbol{t_t}(\omega_t) \otimes\ \boldsymbol{t_s}(\gamma_t), \tag{57}$$

N_t target space-time snapshot vectors are generated. The target amplitudes, ζ_t, are chosen such that the targets have high power relative to the interference-plus-noise present in the environment. The spatial frequency, $\gamma_{t,}$ is defined by the current look-direction (elevation and azimuth angles) of the radar array. The normalized Doppler frequency, ω_t, changes from one class to another, covering the frequency range between –PRF/2 and PRF/2. Supposing that the frequency range is divided into M Doppler bins, each class will be represented by approximately $\frac{N_e}{M+1}$ training vectors, where the extra one in the denominator is added to represent the target-free class. If the target-free class is represented by N_f training vectors, then,

$$N_e = N_t + N_f, \tag{58}$$

and

$$N_t = MN_c, \tag{59}$$

where N_c is the number of training vectors per target-containing class. Accordingly, the target-free class will be represented by $N_e - MN_c$ training vectors.

The complete data matrix used for training can now be written as:

$$\boldsymbol{X}_{tr} = \boldsymbol{X}_{sd} + \begin{bmatrix} \boldsymbol{X}_{tar} \\ \boldsymbol{0}_{N_f \times NM} \end{bmatrix}, \tag{60}$$

where $\boldsymbol{X}_{tar}$ is an $N_t \times NM$ matrix containing the artificially-generated targets as row vectors, and is given by:

$$\boldsymbol{X}_{tar} = \begin{bmatrix} \boldsymbol{x}_{t,1,1}^T \\ \boldsymbol{x}_{t,1,2}^T \\ \vdots \\ \boldsymbol{x}_{t,1,N_c}^T \\ \boldsymbol{x}_{t,2,1}^T \\ \vdots \\ \boldsymbol{x}_{t,2,N_c}^T \\ \vdots \\ \boldsymbol{x}_{t,M,N_c}^T \end{bmatrix}, \tag{61}$$

where $\boldsymbol{x}_{t,m,n_c}$ is one of the n_c artificially-generated $NM \times 1$ target space-time snapshot vectors representing class *m*.

To train a linear classifier, a target matrix, $\boldsymbol{Y}_{ta,}$ has to be constructed. $\boldsymbol{Y}_{ta}$ will be defined as an $N_e \times (M+1)$ binary matrix. The n_eth row in $\boldsymbol{Y}_{ta}$ represents the target outputs for each of the classes, corresponding to the training vector in the n_eth row of the training data matrix, $\boldsymbol{X}_{tr}$.

For example, let N_c, M, N_e, N_t and N_f be defined by Table 3.

Table 3 LBSTAP parameter values

Parameter	Value
N_c	2
M	4
N_e	10
N_t	8
N_f	2

Then, the first row vector in $\boldsymbol{Y}_{tar}$, given by:

$$\boldsymbol{y}_1 = [1 \quad 0 \quad 0 \quad 0 \quad 0], \tag{62}$$

represents the target vector for a class-1 space-time snapshot vector. Thus, it is constructed of a "1" for the first element and "0"s for the rest. Defining the rest of the rows of $\boldsymbol{Y}_{tar}$ using equation (62), the complete target matrix for the example defined by Table 3 becomes:

$$\boldsymbol{Y}_{tar} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{63}$$

The training data matrix and the target matrix are used to compute the weight matrix, $\boldsymbol{W}$. For a linear classifier, this is done by solving:

$$\boldsymbol{Y}_{tar} = \boldsymbol{X}_{tre}\boldsymbol{W} \tag{64}$$

for $\boldsymbol{W}$, where $\boldsymbol{X}_{tre}$ is defined as:

$$\boldsymbol{X}_{tre} = [\boldsymbol{X}_{tr} \quad \mathbf{1}_{N_{e\times 1}}]. \tag{65}$$

The vector of ones is added to the original data matrix to allow for the computation of constant bias values in the weight matrix. In general, $\boldsymbol{X}_{tre}$ is not a square matrix. Thus, $\boldsymbol{W}$ is computed by:

$$\boldsymbol{W} = \boldsymbol{X}_{tre}{}^{t}\boldsymbol{Y}_{tar}, \tag{66}$$

where $\boldsymbol{X}_{tre}{}^{t}$ denotes the pseudo-inverse of $\boldsymbol{X}_{tre}$, and is given by:

$$\boldsymbol{X}_{tre}{}^{t} = (\boldsymbol{X}_{tre}{}^{T}\boldsymbol{X}_{tre})^{-1}\boldsymbol{X}_{tre}{}^{T}. \tag{67}$$

After computing the weight matrix, the next step in LBSTAP is to compute the outputs corresponding to the range gate test under test. The output for a test range gate, $\boldsymbol{x}_{lt}$, is given by:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{x}_{lt} \\ 1 \end{bmatrix}^{T} \boldsymbol{W}. \tag{68}$$

The output vector, $\boldsymbol{y}$, contains values corresponding to the degree with which the test vector fits each class. The outputs are compared to a threshold value, T, to decide on the class of the test vector.

As done in STAP, a plot of the output power (in dB) is generated to visually represent the position of a target (if present) on the Doppler-range space (see Figure 9).

The formulations in this section were done for a linear classifier. Next, the concept will be extended to a type of nonlinear classifiers called polynomial classifiers.

4.3 Polynomial Classifier

According to the Weierstrass approximation theorem, "every real-valued continuous function on a finite closed interval [a, b] can be uniformly approximated by polynomials with real coefficients" and "every complex-valued continuous function on a finite closed interval [a, b] can be uniformly approximated by polynomials with complex coefficients" [28]. Based on this theorem, polynomial classifiers can be thought of as "universal approximators to the optimal Bayes classifier" [29].

The training process using the polynomial classifier is illustrated in Figure 13.

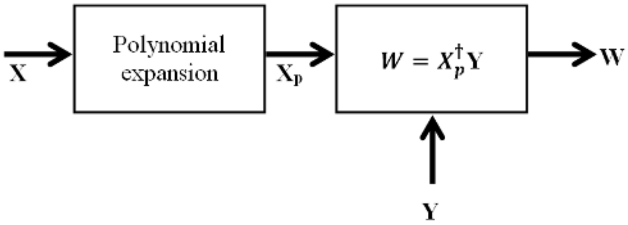

Figure 13 Polynomial classifier training

where $\boldsymbol{X_p}$ is the polynomial expansion of $\boldsymbol{X}$. To demonstrate what is meant by polynomial expansion, suppose that $\boldsymbol{X}$ is composed of three, 2-dimensional vectors, as shown in equation (69).

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{33} & x_{32} \end{bmatrix} \tag{69}$$

Then, $\mathbf{X_p}$ is given by:

$$\boldsymbol{X_p} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{11}x_{12} & x_{11}^2 & x_{12}^2 \\ 1 & x_{21} & x_{22} & x_{21}x_{22} & x_{21}^2 & x_{22}^2 \\ 1 & x_{33} & x_{32} & x_{31}x_{32} & x_{31}^2 & x_{32}^2 \end{bmatrix}. \tag{70}$$

The polynomial expansion defined by equation (70) is the second order (quadratic) expansion. After determining the weight matrix $\boldsymbol{W}$, the snapshot vector of the range gate under test is expanded in the same way to its polynomial expansion, $\boldsymbol{x}_{tp}$. Finally, the output vector is determined by:

$$\boldsymbol{y} = \boldsymbol{x}_{tp}{}^{T}\boldsymbol{W}. \tag{71}$$

4.4 Classifier Selection

The classifier selected for use in LBSTAP is the polynomial classifier. The linear classifier was tried first, but showed unsatisfactory results. The use of multilayer neural networks was attempted as well. However, given that the number of classes and feature-vector dimensionalities are typically high, the computational complexity of multilayer neural networks becomes infeasible for MTI applications. Polynomial classifiers were found to achieve the best results among the studied classifiers, at an acceptable computational cost.

In this report, the space-time snapshot vectors are 224-dimensional. Hence, their polynomial expansion results in very-high-dimensional vectors (more than 25000-dimensional). Therefore, the polynomial expansion stage is followed by a dimensionality reduction step, where, using principle component analysis (PCA), the snapshot vectors are reduced to dimensions between 30 and 400, depending on the size of the CPI data cube. The variability in the dimension of the reduced vectors is dictated by computational limitations of the PCA algorithm used; the maximum dimension that can be achieved cannot exceed the number of training range gates.

Although PCA is not designed for pattern classification, it was found to achieve better results than FDA. This is because the number of classes throughout this report is set to 17. Thus, FDA reduces the dimensionality of the feature vectors to a maximum of 16. The results show that using these low-dimensional feature vectors, the separability of the classes is significantly diminished.

4.5 Cascaded System

The use of a cascaded system (STAP followed by LBSTAP) can outperform STAP and LBSTAP when used separately. The cascaded system has the following architecture in the training phase:

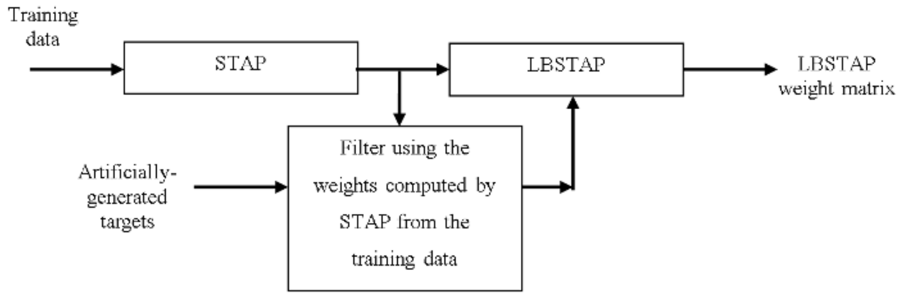
Training data
STAP
LBSTAP
LBSTAP weight matrix
Artificially-generated targets
Filter using the weights computed by STAP from the training data

Figure 14 Cascaded system

After the training phase is complete, the space-time snapshot vector of the range gate under test is passed through STAP and then the output is passed through LBSTAP, to obtain an output vector. The output vector is compared to a threshold value to decide which Doppler bin contains a target (if present).

4.5.1 Note on Computational Complexity. It must be said that no system can be superior in all situations and from all aspects. As such, it should be noted that although the cascaded system outperforms both STAP and LBSTAP, it suffers from high computational complexity compared to the individual systems. However, this high computational complexity is balanced by the high performance it provides (see Chapter 5 for the results).

Chapter 5: Results

5.1 Note on Thresholds and Decision-Making

Although the description given in this report defines the decision-making process by a comparison between an output value and a threshold value, it should be noted that this is not actually done in the simulations presented in this report. The outputs presented here are normalized and the position of the maximum output is compared to the known target position, to decide if the processing algorithms were able to correctly locate the target. Moreover, the difference between the maximum and the rest of the outputs in power is used as a measure of effectiveness, alongside the output SINR at the target range gate.

Also note that in LBSTAP, the target-free class outputs are discarded. That is, the output plots shown in this chapter show only the classes falling within the Doppler range (target-present classes). Since the position of the actual target is already known in the simulations presented, the decision and performance evaluation can be made using the technique described above. However, in real situations, an appropriate threshold has to be calculated and used to decide on the presence of a target in each class. Still, the target-free class output would be discarded, as the failing of all target-present class outputs to exceed the threshold would imply the absence of a target, without the need to check the target-free class output.

5.2 STAP vs. LBSTAP

In this section, the performance of STAP will be compared to that of polynomial LBSTAP in scenarios where the number of training range bins is limited. Table 4 shows the parameters used throughout the results section. Target and jammer powers are given relative to clutter power.

Table 4 Simulation parameters

Parameter	Value
Number of spatial channels (N)	14
Number of PRIs per CPI (M)	16
Jammer azimuth angle	-10°
Jammer power	10 dB
Target azimuth	30°
Target Doppler frequency	-80 Hz
PRI	1.6 ms

The target power in the first scenario is -20 dB, and the number of training range bins available is limited to 96. Figure 15 shows the unprocessed data.

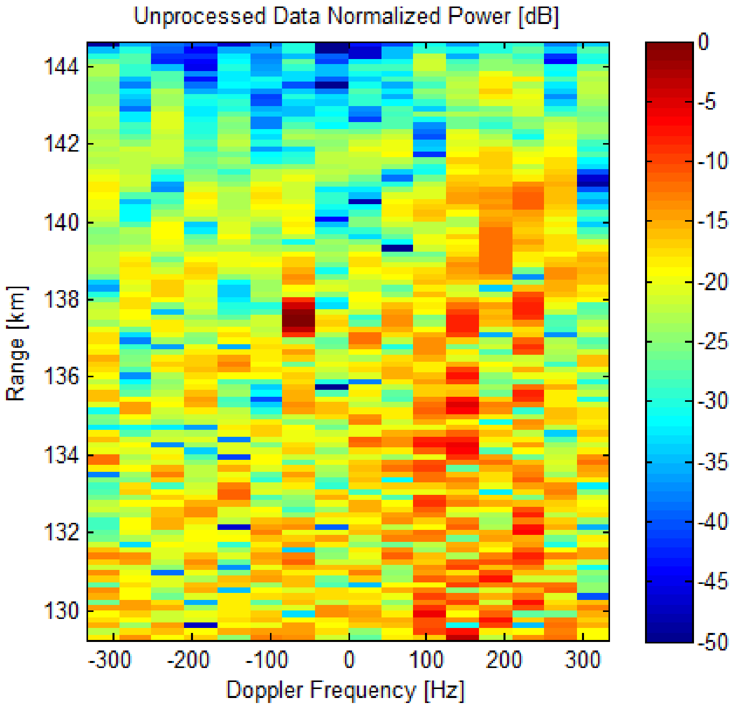

Figure 15 Unprocessed data

The clutter is mostly present with Doppler frequencies between 100 Hz and 250 Hz, as can be seen from the unprocessed data. The target is present around a range of 137 km. Figure 16 shows the output of STAP.

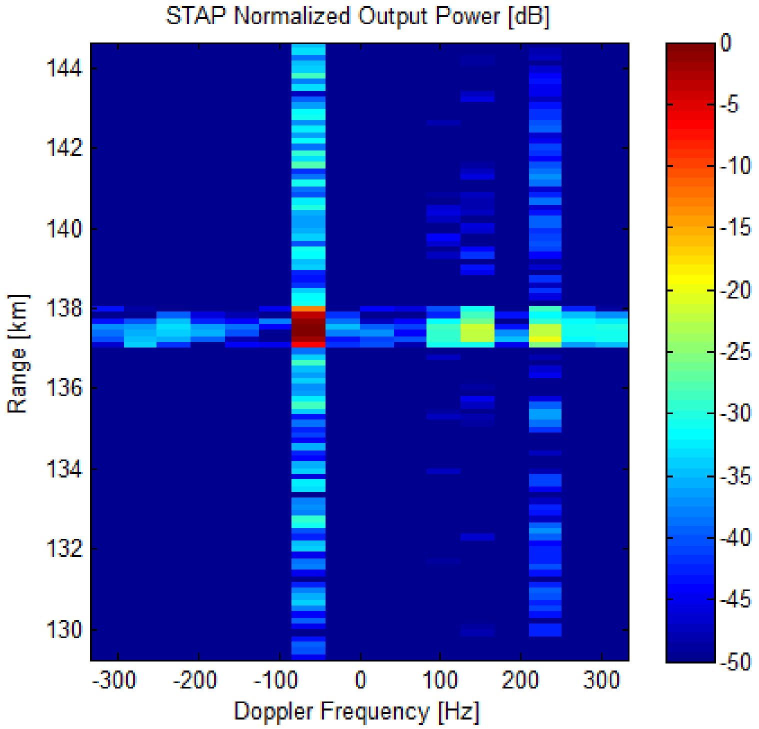

Figure 16 STAP output

It is seen from Figure 16 that STAP was able to maximize the output power at the target position. However, it can also be seen that the clutter was not completely suppressed, especially at the range of the target. This can be explained by the limited number of training range bins used, leading to a poor estimate of the interference covariance matrix. This resulted in a distortion of the null at the clutter position. The residual clutter returns can be seen at all ranges; however, they are highest at the range bins close to the target. This is due to the fact that these range bins were left out of the training process as guard cells.

Moreover, the STAP output shows persistent target range side lobes. These side lobes are clearly seen in the range profile shown in Figure 17.

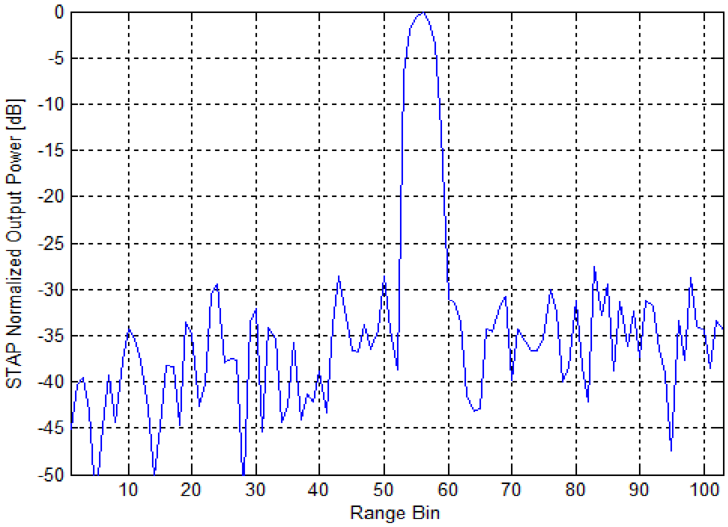

Figure 17 STAP range profile

The output of polynomial LBSTAP for the same scenario is shown in Figure 18. The range vectors used in LBSTAP are the quadratic expansion of the original 224-dimensional vectors. The expanded vectors are then reduced in dimension, using principle component analysis (PCA), to become 30-dimensional.

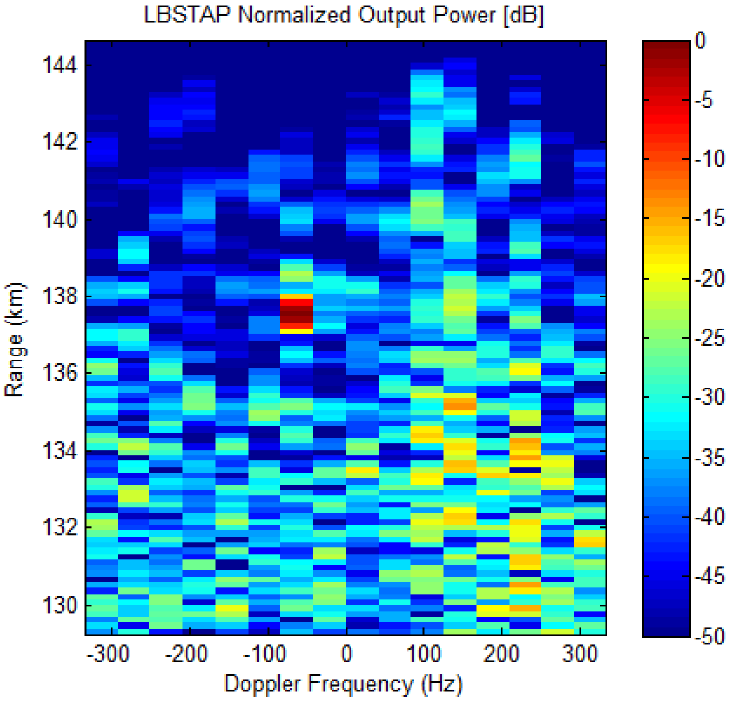

Figure 18 LBSTAP output

The range profile of LBSTAP is shown in Figure 19.

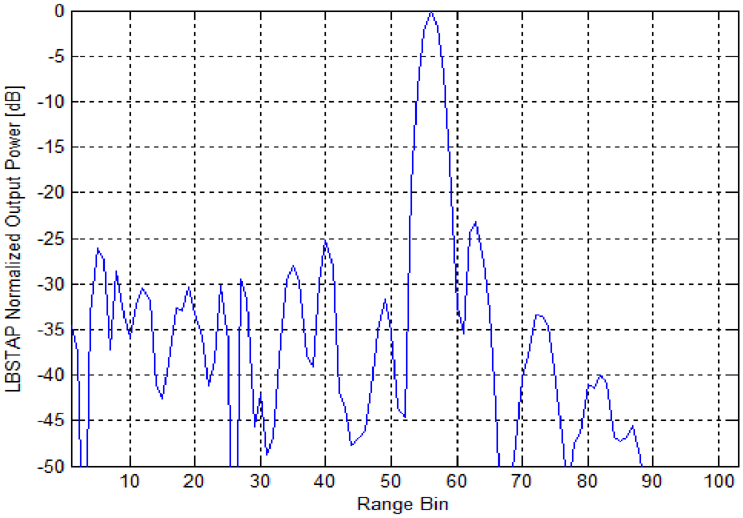

Figure 19 LBSTAP range profile

It can be seen from Figure 19 that the level of range side lobes of the output of LBSTAP is approximately the same as that of the output of STAP.

5.2.1 Effect of the Size of Training Data on SINR. The performance of STAP in terms of SINR depends on the number of training range bins used in estimating the interference covariance matrix. According to the RMB SINR equation, the minimum number of training range bins needed for an SINR of 3 dB less than the optimum SINR (using the exact covariance matrix) is twice the total number of degrees of freedom (twice 224 for the simulations in this paper). For different practical reasons, STAP is sometimes required to work with a limited number of training range bins, well below the RMB requirement. Figure 20 shows the output SINR for STAP (with and without 20 dB diagonal loading) and polynomial LBSTAP versus the total number of training range gates used for each processing technique. (Note that PCA in this case reduces the polynomial expansion of vectors in LBSTAP to a dimension equal to the number of training range bins used).

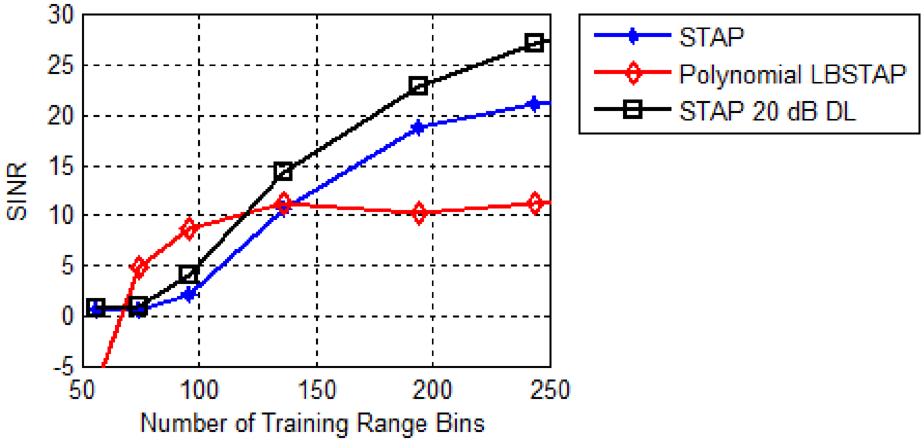

Figure 20 Output SINR comparison

It can be seen from Figure 20 that when the amount of training data available is high enough, STAP will have a higher output SINR. However, when the amount of training range bins is limited between 70 and 120, polynomial LBSTAP shows an advantage over STAP (with and without diagonal loading) in terms of output SINR. This is a significant advantage, given that the amount of training data available is usually limited. It will be shown next that the range where polynomial LBSTAP outperforms STAP and the size of the advantage of using the former depend greatly on the target power.

5.2.2 Effect of Target Power. Target power does not have any significant effect on the interference covariance matrix estimation in STAP. Therefore, having a target of higher power or lower power will not affect the degree with which the interference will be rejected. (It will not affect the depth and precision of the clutter and jammer nulls). On the other hand, LBSTAP, being a classification technique, depends greatly on the relative target to interference power. Therefore, an increase in target power relative to interference should result in greater advantage for LBSTAP over STAP.

Figure 21 and Figure 21 show the outputs of STAP and polynomial LBSTAP, respectively, for the same simulation environment and parameters used in the earlier simulation; however, the target power is increased from −20 dB to −10 dB.

The two figures show an advantage of using polynomial LBSTAP in terms of target range side lobes and clutter suppression over STAP in cases where the target power is relatively high. The target range side lobes reduction can be seen clearly in the range profile plots shown in Figure 23 for STAP and Figure 24 for LBSTAP.

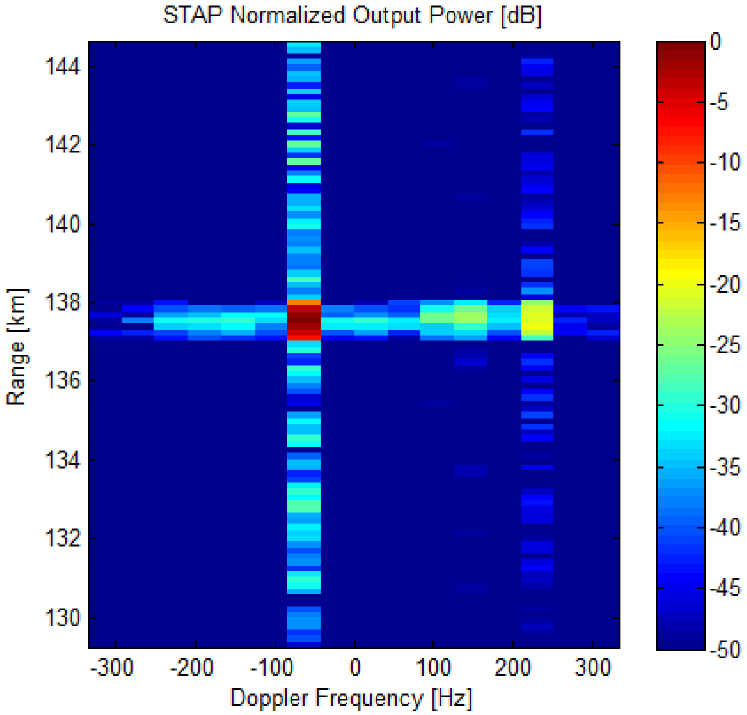

Figure 21 STAP output (higher target power)

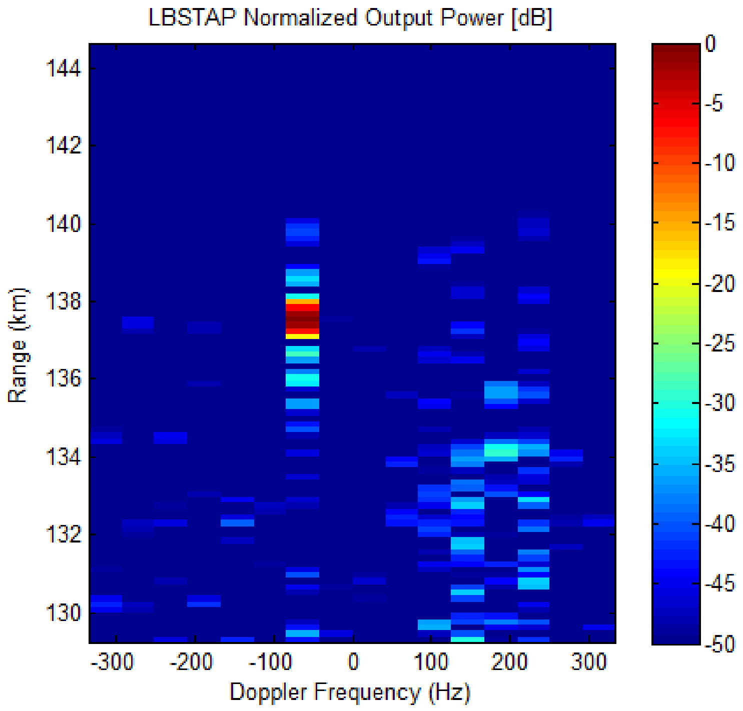

Figure 22 LBSTAP output (higher target power)

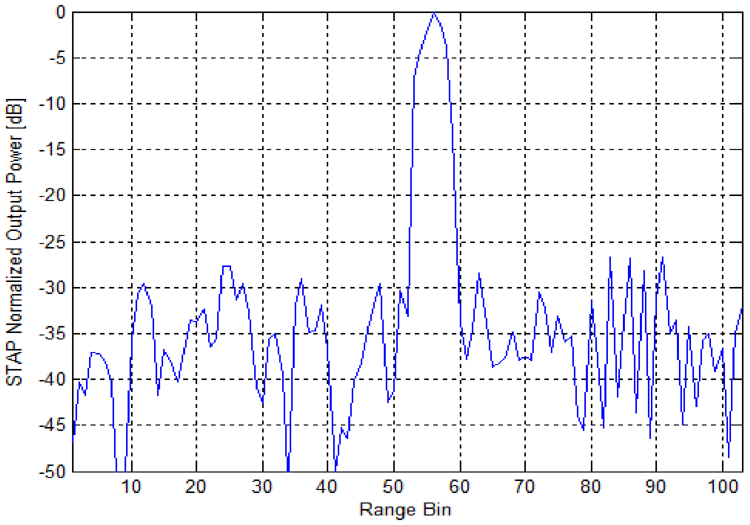

Figure 23 STAP range profile (higher target power)

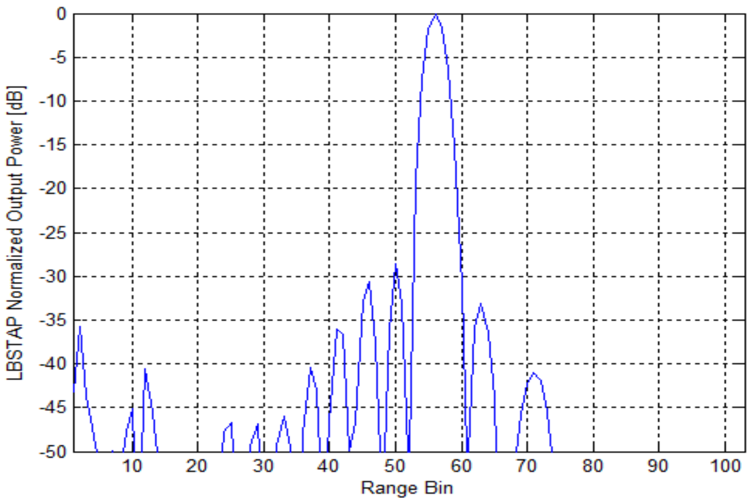

Figure 24 LBSTAP output (higher target power)

Figure 25 shows the effect of the number of training range bins on the output SINR curves for STAP and polynomial LBSTAP for the higher target power case.

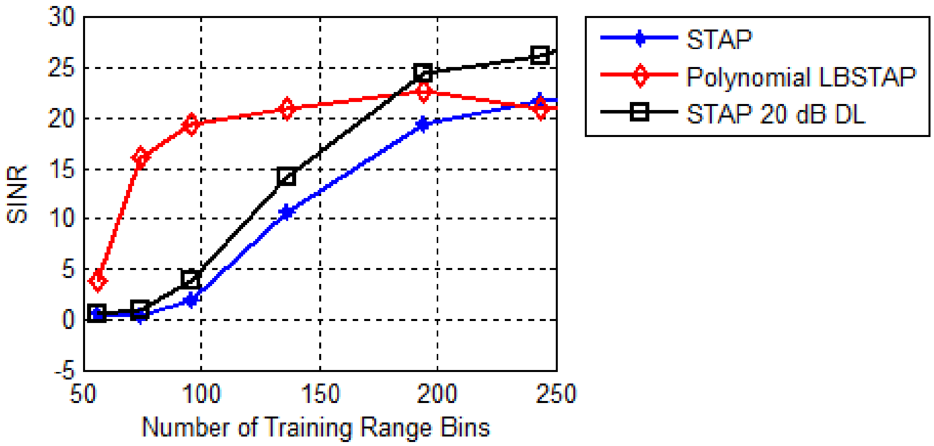

Figure 25 Output SINR comparison (higher target power)

Comparing Figure 25 with Figure 20, it is clear that the gain in SINR as a result of having a higher target power is significantly higher for polynomial LBSTAP than for STAP. It can be seen that the range where polynomial LBSTAP outperforms STAP is extended to cover the range between 50 to 230 training range bins (50 to 170 for STAP with 20 dB diagonal loading).

It should be also said that reducing the target power well below the clutter power will cause the performance of LBSTAP to degrade significantly, while, again, STAP will not be affected as much by the variation in target power.

5.2.3 Target-Cancellation. The theory of STAP requires that the training range bins used to estimate the interference covariance matrix must be target-free, to avoid what is known as target cancellation. This means that, to use STAP, it is required to have some prior estimate of the location of the target. For example, if the target was mistakenly assumed to be 10 range gates away from its actual position, STAP will fail to detect the correct location of the target.

This condition does not apply to LBSTAP. This is due to the fact that the training process of LBSTAP uses part of the training range gates to model the interference present at each class. The effect of the range bin containing the target will limited to one class, and at the same time will be marginalized by the rest of the training range bins used to represent the interference at that class.

Figure 26 shows the output of STAP for a scenario where the target was assumed to be 10 range bins away from its actual position. It is clear from Figure 26 that STAP fails to detect the target in such a scenario.

On the other hand, Figure 27 shows that LBSTAP was able to perform almost the same as it did when the range bin containing the target was left out during the training process.

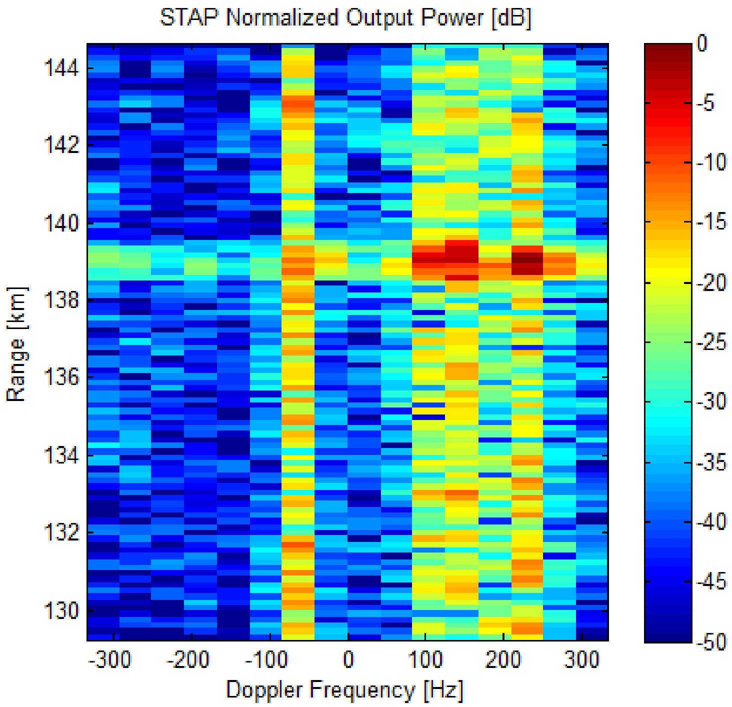

Figure 26 STAP output (target-cancellation)

This means that the steps taken to have an initial estimate of the target position can be skipped, and the whole CPI data cube (including target and guard range gates) can be used in the training process.

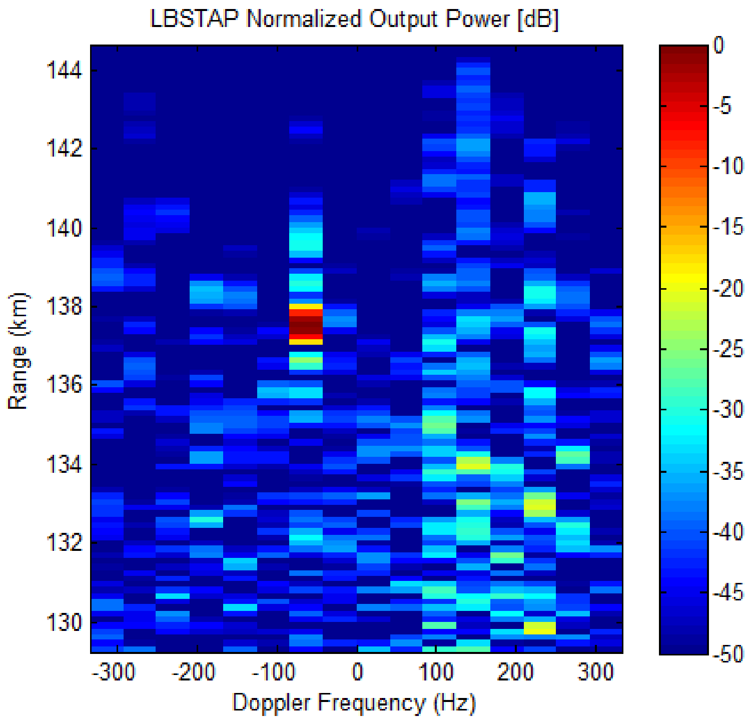

Figure 27 LBSTAP output (target-cancellation)

5.2.4 Clutter-Variation. In this section, the robustness of both STAP and LBSTAP in changing clutter environments is put under test. The clutter change of environment is simulated by a 10% increase in platform velocity, resulting in an increase in the slope of the clutter ridge. In other words, the training will be done in one clutter environment, while the testing will be done in a slightly different environment.

Figure 28 and Figure 29 show the outputs for STAP and polynomial LBSTAP respectively.

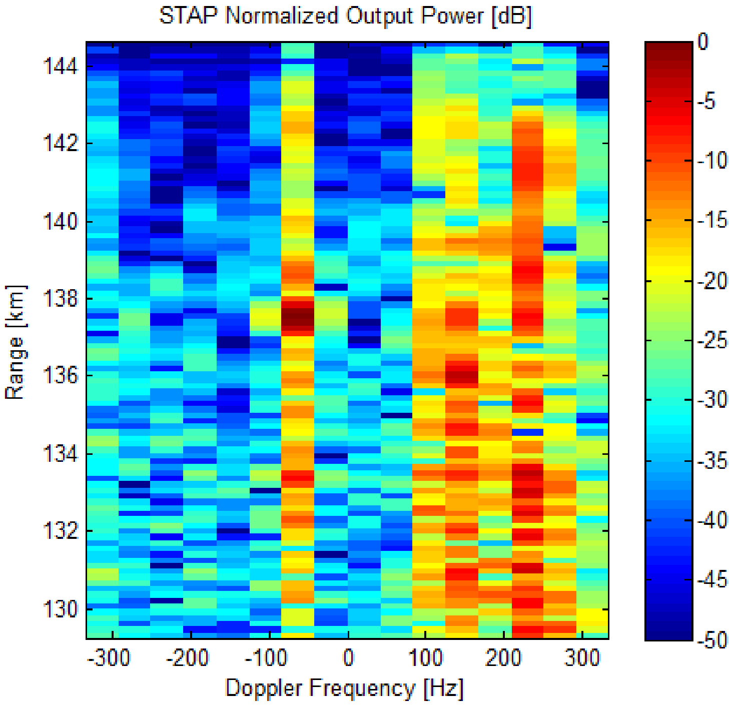

Figure 28 STAP output (clutter-variation)

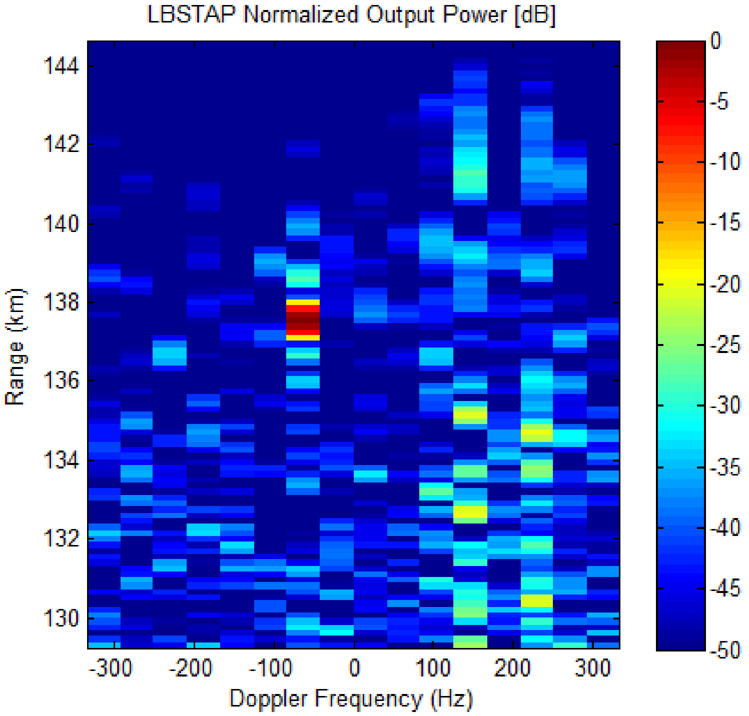

Figure 29 LBSTAP output (clutter-variation)

Figure 28 shows that STAP was unable to suppress the clutter, which can be explained by the fact that the null position estimated in the training process is different than the position of the clutter in the test environment.

On the other hand, polynomial LBSTAP was able to correctly detect the target while suppressing the clutter; albeit there is a slight degradation in performance when compared to the original scenario, where the training and testing were done in the same environment (shown in Figure 22).

The robustness of LBSTAP to clutter variation can be explained the following. Unlike STAP, LBSTAP is not trying to place a null in the position of clutter. LBSTAP works to classify the range gate under test into one of the possible classes. Hence, a variation in clutter will, roughly speaking, equally reduce the degree with which the range under test matches the models of all classes produced in the training phase. Hence, the effect of clutter variation will be limited to a reduction in the power output at all range and Doppler bins.

5.3 Cascaded System

To conclude the simulation results section, it must be said that both STAP and LBSTAP have their own points of weaknesses and strengths. LBSTAP will completely fail in cases where the target power is significantly lower than the clutter power. In such cases, STAP will be the better technique to use. On the other hand, the performance of STAP degrades significantly when the number of training range bins is very limited. Therefore, cascaded system of STAP followed by LBSTAP can be used to take advantage of the strengths of both systems while overcoming their weaknesses. The STAP stage can reduce the clutter power relative to the target power to a level low enough for LBSTAP to perform effectively. This means that a cascaded system can be used in severe cases were both STAP and LBSTAP fail to work independently.

To test the cascaded systems, the clutter power is increased so that the target power is now at -50 dB. In such a scenario LBSTAP is expected to fail, as shown in Figure 30. In addition, the number of training range gates is limited to 96.

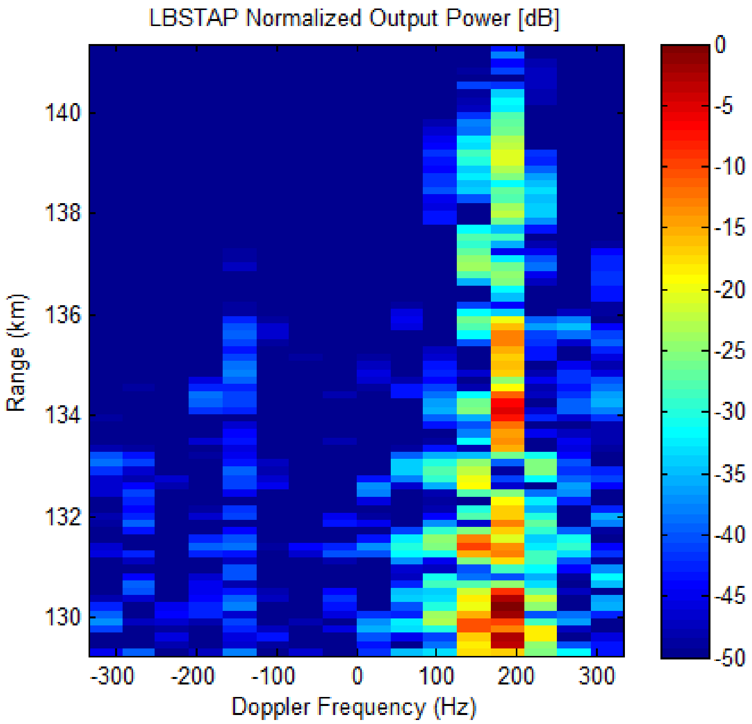

Figure 30 LBSTAP output (severe environment)

Finally, another jammer signal with a power of −30 dB is added very close to the azimuth angle of the target (target is at 30°; jammer is at 33°). The limited number of training range bins, in addition to the close proximity of the jammer null to the target maximum, is expected to degrade the performance of STAP, as shown in Figure 31.

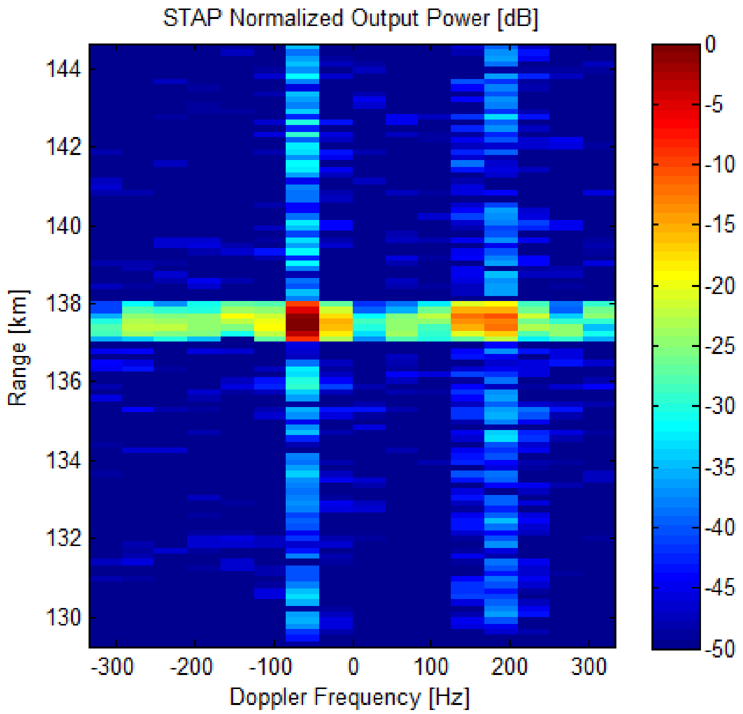

Figure 31 STAP output (severe environment)

Figure 32 and Figure 33 show the output of the cascaded system and the corresponding range profile, respectively.

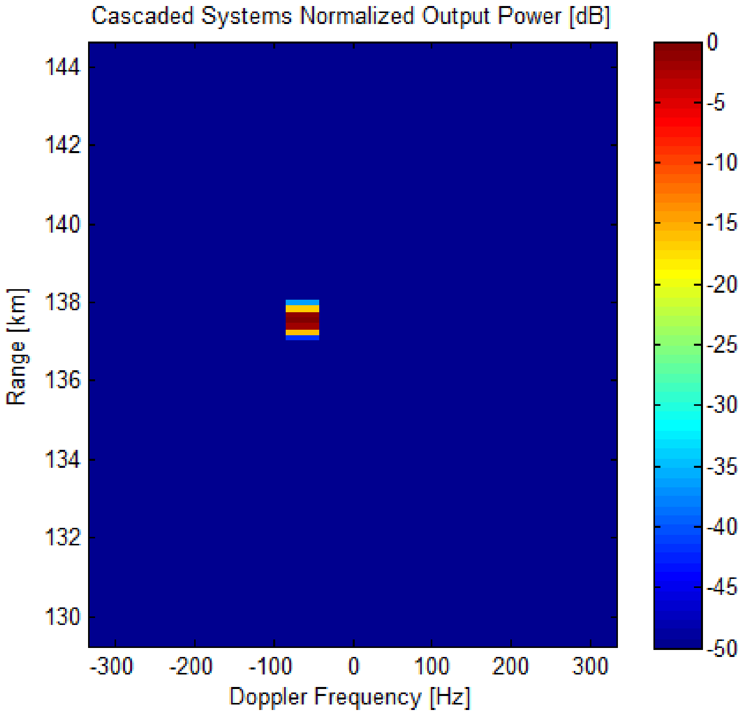

Figure 32 Cascaded system output (severe environment)

Note that the polynomial classifier used in this case is of degree four. The LBSTAP is applied on range vectors of dimension 16 (the number of PRIs per CPI), which are extracted from the output obtained from STAP. The expanded vectors are then reduced in dimension, using PCA, to become 30-dimensional.

The cascaded systems were able to outperform both systems when used separately. The target was detected, while all the interference is suppressed to levels below −50 dB. The output plot shown in Figure 32 shows approximately no range side lobes or Doppler side lobes.

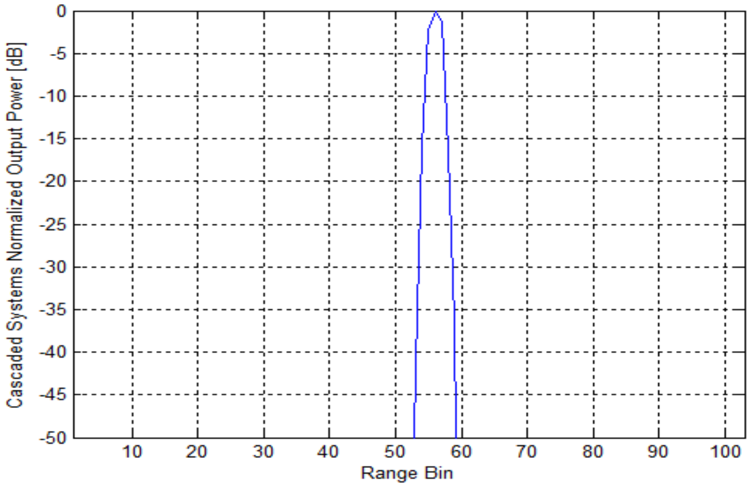

Figure 33 Cascaded system range profile (severe environment)

5.4 Mountaintop Data

The introduced methods (LBSTAP and cascaded system) were also tested with real radar data obtained from the mountaintop data set. The results are compared to STAP output. Figure 34 shows the unprocessed data.

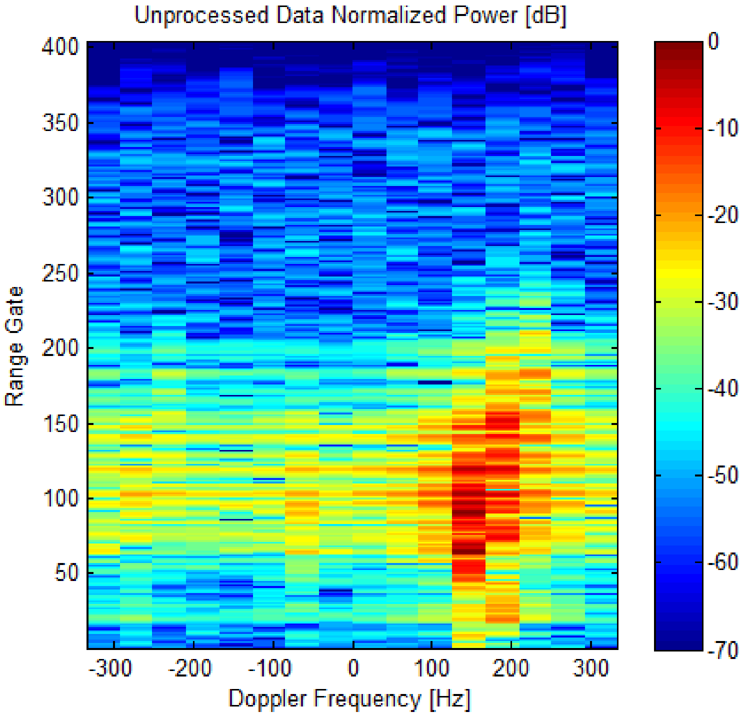

Figure 34 Unprocessed data (mountaintop)

A target is present at range gate around 150, with a Doppler frequency of 150 Hz. Clutter returns are present at the same Doppler frequency and range gate; however, at a different azimuth angle.

The CPI data cube is made of 403 range gates; however, the training processes in all methods will be limited to 83 training range gates. This is done to see the performance of each method in circumstances involving limited secondary data.

The output of STAP is shown in Figure 35.

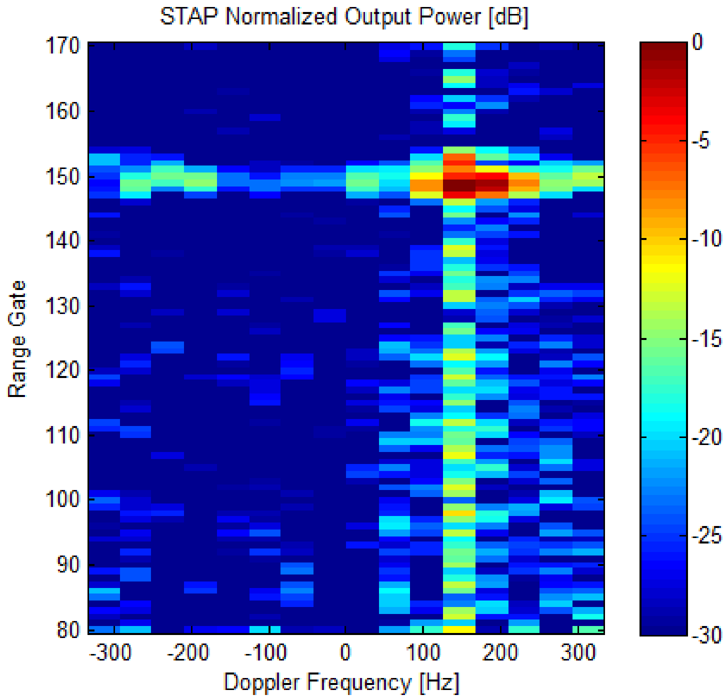

Figure 35 STAP output (mountaintop)

The target was detected using STAP; however, the Doppler side lobes and range side lobes are relatively high, even though Doppler tapering has been used, in addition to diagonal loading, in order to achieve as good an output as possible.

Figure 36 shows the output of polynomial LBSTAP. It is clear from the output that LBSTAP fails in detecting the target. This is due to the fact that the target power is considerably lower than the clutter power; and as mentioned earlier, LBSTAP degrades in performance as the target power falls with respect to clutter power.

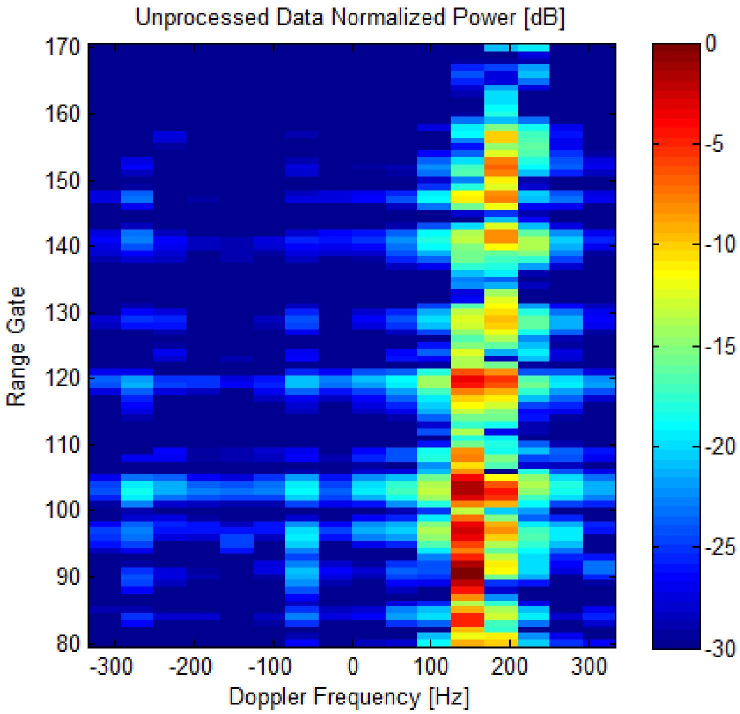

Figure 36 LBSTAP output (mountaintop)

Figure 35 shows that the performance of STAP, when the number of training range gates is limited, is relatively poor and suffers from high range and Doppler side lobes. Also, Figure 36 shows that LBSTAP fails at detecting the target when the clutter power is considerably higher than that of the target.

Figure 37 shows the output of the cascaded system (STAP followed by polynomial LBSTAP). As was the result for the simulated data, the cascaded-system method was able to suppress the range and Doppler side lobes to very low levels.

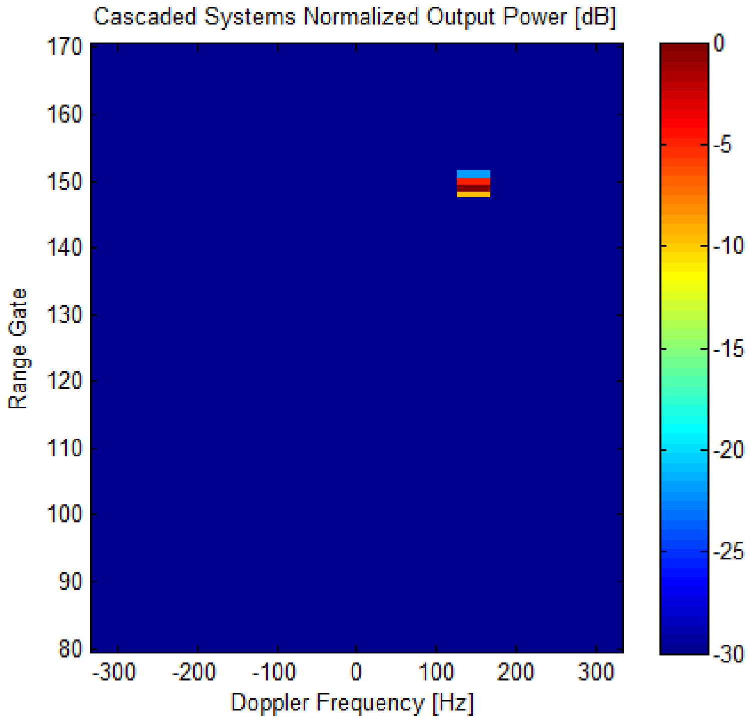

Figure 37 Cascaded system output (mountaintop)

Figure 38 and Figure 39 show range profiles for STAP and the cascaded systems respectively.

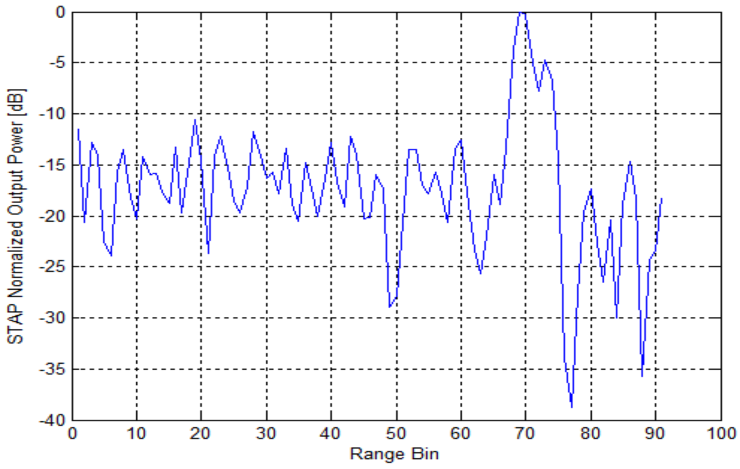

Figure 38 STAP range profile (mountaintop)

It is clear from the range profiles that the cascaded system is able to suppress the range side lobes from around −15 dB (after STAP) to around −150 dB.

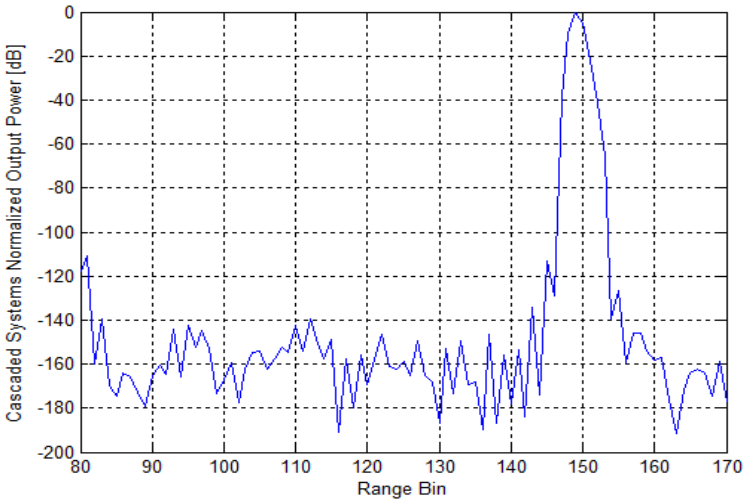

Figure 39 Cascaded system range profile (mountaintop)

Chapter 6: Conclusions

The practical constraints limiting the availability of secondary training data needed for the computation of STAP filter weights make the development of algorithms capable of achieving acceptable performance with limited secondary data a necessity. While there have been many improvements on the original fully-adaptive STAP algorithm, the necessity still holds, for each of these algorithms has its own limitations.

Reduced-rank STAP algorithms suffer from the need for a priori knowledge of the low-rank interference subspace. Given the fact that a priori knowledge is generally not available, reduced-rank STAP algorithms often have to sacrifice part or all of their adaptivity in one dimension, depending on the algorithm of dimensionality-reduction applied.

It was shown in this report that the proposed learning-based space-time adaptive processing technique is able to achieve higher target SINR compared to STAP for limited secondary data scenarios. This outcome holds as long as the target power does not fall to an extreme level below the clutter power. The range of values for the amount of secondary data used by STAP/LBSTAP over which LBSTAP outperforms STAP was shown to improve with the increase in target power relative to clutter power. For example, it was shown that for targets having a power 20 dB below the clutter power this range is limited between 70 and 130 secondary training bins. However, when the target power was increased by 10 dB, this range extended to cover low values (around 50 training bins) and continued until reached values of around 200 training bins. These results were obtained for 224-dimensional snapshot vectors.

Moreover, it was shown that LBSTAP overcomes two of the problems faced by STAP algorithms; namely, target-cancellation and clutter variation. It was shown in the results section that when the target range gate was mistakenly included in the training data, the output of STAP was unsatisfactory; the target was not detected in the correct position. On the other hand, LBSTAP continued to perform almost unaffected by the inclusion of the target range gate in the training data.

To test the effect of clutter variations on the performance of both systems, simulations were run in which the training phase of both systems was done in a clutter environment slightly different than that of the testing phase. It was shown that LBSTAP continued to perform almost unaffected while STAP failed to detect the target in the correct position.

Although these results show a significant advantage for LBSTAP compared to STAP, the fact that LBSTAP fails to detect low-power targets can be a significant downside under certain circumstances. The cascaded system was proposed to overcome this issue. It was shown that the cascaded system is able to outperform both individual systems under severe conditions. STAP was used first to lower the gap between the target power and the clutter power. This facilitated the application of LBSTAP to detect the target and suppress interference to very low levels. STAP (when used alone) was not able to achieve acceptable results, due to the limited secondary data and the very-close proximity of the target and the jammer signals in the spatial dimension. On the other hand, LBSTAP (when used alone) failed, due to the low target power level.

The proposed cascaded system was tested also using real radar data obtained from the mountaintop data set. It was shown to outperform both systems when used separately with limited secondary data.

Finally, although it is apparent that all developments come with their own downsides and limitations, it is necessary to state that the proposed algorithms (LBSTAP and the cascaded system) cannot outperform STAP and other upgrades of STAP in ALL conditions. However, they do stand as alternatives that, in some possible conditions, can offer a performance advantage.

References

[1] William L Melvin, "A STAP Overview," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 19-35, Jan. 2004.

[2] J. Ward, "Space-Time Adaptive Processing for Airborne Radar," Lincoln Laboratory, MIT, Massachusetts, Technical Report 1015, 1994.

[3] I.S. Reed, J.D. Mallett, and L.E. Brennan, "Rapid Convergence Rate in Adaptive Arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 10, no. 6, pp. 853-863 , Nov. 1974.

[4] George W. Stimson, *Introduction to Airborne Radar*. Mendham, N.J. , USA: SciTech Pub, 1998.

[5] L.E. Brennan and I.S. Reed, "Theory of Adaptive Radar ," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 9, no. 2, pp. 237-252 , Mar. 1973.

[6] William L. Melvin, "Space-Time Adaptive Radar Performance in Heterogeneous Clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 631-633, Apr 2000.

[7] Rui Fa and Rodrigo C. De Lamare, "Reduced-Rank STAP Algorithms using Joint Iterative Optimization of Filters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 1668-1682, Jul. 2011.

[8] J. R. Guerci, J. S. Goldstein, and I. S. Reed, "Optimal and Adaptive Reduced-Rank STAP," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 647-660, Apr 2000.

[9] C. D. Peckham, A. M. Haimovich, T. F. Ayoub, J. S. Goldstein, and I. S. Reed, "Reduced-Rank STAP Performance Analysis," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 664-674, Apr. 2000.

[10] A. Haimovich, "The Eigencanceler: Adaptive Radar by Eigenanalysis

Methods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 2, pp. 532-542, Apr. 1996.

[11] L. Chang and C.-C. Yeh, "Performance of DMI and Eigenspace-Based Beamformers ," *IEEE Trans. Antennas Propag.*, vol. 40, no. 11, pp. 1336-1347 , Nov. 1992.

[12] R. Kumaresan and D.W. Tufts, "Data-Adaptive Principal Component Signal Processing," in *Proceedings of the 19th IEEE Conference on Decision and Control*, Albuquerrque, NM, 1980, pp. 949-954.

[13] I.P. Kirsteins and D.W. Tufts, "Adaptive Detection using Low Rank Approximation to a Data Matrix ," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 1, pp. 55-67, Jan. 1994.

[14] Chaoshu Jiang, Hongbin Li Li, and M. Rangaswamy, "Conjugate Gradient Parametric Adaptive Matched Filter," in *Proceedings of the IEEE Radar Conference*, Washington, DC, 2010, pp. 740-745.

[15] J.R. Roman et al., "Parametric Adaptive Matched Filter for Airborne Radar Applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 677-692, Apr. 2000.

[16] J.Scott Goldstein, Irving S. Reed, and P.A. Zulch, "Multistage Partially Adaptive STAP CFAR Detection Algorithm ," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 2, pp. 645-661 , Apr. 1999.

[17] Michael C. Wicks, Muralidhar Rangaswamy, Raviraj Adve, and Todd B. Hale, "Space-Time Adaptive Processing," *IEEE Signal Process. Mag.*, pp. 51-65, Jan. 2006.

[18] J. R. Guerci, *Space-Time Adaptive Processing for Radar*. Norwood, MA, USA: Artech House, 2003.

[19] Fulvio gini and Muralidhar Rangaswamy, Eds., *Knowledge Based Radar Detection, Tracking, and Classification*. Hoboken, NJ, USA: Wiley, 2008.

[20] Stephanie Bidon, Olivier Besson, and Jean-Yves Tourneret, "Knowledge-Aided STAP in Heterogeneous Clutter using a Hierarchical Bayesian Algorithm,"

IEEE Trans. Aerosp. Electron. Syst., vol. 47, no. 3, pp. 1863-1879, Jul. 2011.

[21] Xumin Zhu, Jian Li, and Petre Stoica, "Knowledge-Aided Space-Time Adaptive Processing ," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1325-1336 , Apr. 2011.

[22] Rui Fa, R. C. De Lamare, and V. H. Nascimento, "Knowledge-Aided STAP Algorithm using Convex Combination of Inverse Covariance Matrices for Heterogenous Clutter ," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Dallas, TX , 2010, pp. 2742-2745.

[23] Tapan Kumar Sarkar et al., "A Deterministic Least-Squares Approach to Space-Time Adaptive Processing (STAP)," *IEEE Trans. Antennas Propag.*, vol. 49, no. 1, pp. 91-103, Jan. 2001.

[24] Diego Cristallini and Wolfram Burger, "A Robust Direct Data Domain Approach for STAP," *IEEE Trans. Signal Process.*, vol. 60, no. 3, pp. 1283-1294, Mar. 2011.

[25] Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition*. Boston, USA: Academic Press, 1990.

[26] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. New York, USA: Wiley, 2001.

[27] Andrew R. Webb, *Statistical Pattern Recognition*. New Jersey, USA: Wiley, 2002.

[28] Borwein Peter and Tamas Erdelyi, *Polynomials and Polynomial Inequalities*. New York, USA: Springer, 1995.

[29] W.M. Campbell, K.T. Assaleh, and C.C. Broun, "Speaker Recognition with Polynomial Classifiers," *IEEE Trans. Speech and Audio Process.*, vol. 10, no. 4, pp. 205-212, May 2002.

Vita

Alaa El Khatib received his B.S. (cum laude) in Electrical Engineering from the American University of Sharjah (AUS) in 2011. After that, he joined the graduate program at the department of Electrical Engineering at AUS as a teaching assistant and completed his M.S. in Electrical in Engineering degree requirements in 2013. During this time, he participated in the First International Conference on Communications, Signal Processing, and their Applications (ICCSPA'13), with a paper on his research findings.