2016

Applications of Optimal Subspace Estimation

Christopher Grayson
University of Rhode Island, grayson221@my.uri.edu

Follow this and additional works at: <https://digitalcommons.uri.edu/theses>

Recommended Citation

Grayson, Christopher, "Applications of Optimal Subspace Estimation" (2016). *Open Access Master's Theses*. Paper 959.
<https://digitalcommons.uri.edu/theses/959>

This Thesis is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Master's Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu.

APPLICATIONS OF OPTIMAL SUBSPACE ESTIMATION

BY

CHRISTOPHER GRAYSON

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2016

MASTER OF SCIENCE THESIS
OF
CHRISTOPHER GRAYSON

APPROVED:

Thesis Committee:

Major Professor Richard Vaccaro

Steven Kay

Tom Bella

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2016

ABSTRACT

Optimal Subspace Estimation (OSE) is a technique for estimating the signal subspace of a noisy data matrix or covariance matrix using a specific structure known as shift-invariance to generate an accurate estimate. For the applications in this thesis, the datasets are generated using a line array of sensors and multiple snapshots to produce data which should have the desired structure. The OSE algorithm exploits this structure in the data matrix in order to generate an estimate of the underlying noise-free signal subspace. Two applications have been chosen where the covariance matrices of the signal data should exhibit this desired structure. The applications are common signal processing subjects known as Adaptive Beamforming and Space-Time Adaptive Processing (STAP).

OSE is an ideal estimation method for the application of Adaptive Beamforming. It is capable of achieving performance better than a commonly used algorithm known as Dominant Mode Rejection (DMR) while using orders of magnitude fewer snapshots. The OSE algorithm stands out in its ability to quickly approach the Cramer-Rao bound with many fewer snapshots of data. This is especially true when the line array of sensors is able to be carefully calibrated. Results are obtained by multiple simulations that outline the overall performance of theoretical data and test the robustness of the chosen algorithms.

The application of Space-Time Adaptive Beamforming is chosen because it is essentially a two-dimensional adaptation of the Adaptive Beamforming example. The data matrix now contains angle and timing information which becomes increasingly difficult to generate a good estimate as real world physics are applied to the model. Processing this data quickly becomes computationally inefficient and requires a modified OSE algorithm known as Subspace Averaging (SSA) was needed. With theoretical data a lower bound is quickly approached but like the al-

gorithms used for comparison, practical results are not easily achieved with simple processing.

ACKNOWLEDGMENTS

I would like to take the opportunity to thank Dr. Richard Vaccaro for his expert advisement during my thesis research and overall studies at the University of Rhode Island. I owe the completion of this thesis as well as a passion for control theory to him (although, this thesis has nothing to do with controls). I am very thankful for the large amount of time Dr. Vaccaro devoted to deriving a new OSE-like algorithm for the more complex space-time adaptive processing application.

I would also like to thank the additional members of the thesis committee. I took multiple signal processing courses with Dr. Steven Kay during my graduate studies which greatly prepared me for this thesis topic. With the foundation Dr. Kay provided I was able to focus on the intricacies of the simulations rather than just the basic execution. Dr. Tom Bella sparked a strong respect for Mathematics early in my Undergraduate studies. I had the pleasure of taking multiple courses with him, all of which I found challenging and enriching. Kind regards to Dr. Vasilije Perović for serving as committee chair, I wish you great success during your tenure at URI.

Finally, thank you to Dr. Peter Swaszek for allowing me the opportunity to work with him for multiple semesters, finding me my current job, and providing a "gentle" nudge every time he saw me to get this thing done.

DEDICATION

This thesis is dedicated to all of the wonderful people in my life before I began this journey and had the pleasure of meeting along the way. I owe the biggest debt of gratitude to my mother for giving me the motivation and the ability to drop everything and dedicate my time to earning a degree. Thank you to all my family and friends for your support, as well as entertaining me every time I said my thesis would be done soon...

Lastly, thank you to my beautiful fiancée Ashley Boulais for standing by my side while I made it a point to dedicate my focus to everything but her. We started and ended this together and its completion has come at the perfect time. I cannot wait for our amazing wedding this coming summer and all that our new "no school" life will bring us. I love you!

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS	iv
TABLE OF CONTENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
1 Introduction	1
1.1 Applications	1
List of References	6
2 Beamforming	7
2.1 Adaptive Beamformer	7
2.2 Dominant Mode Rejection Beamformer	8
2.3 Optimal Subspace Estimation Beamformer	9
2.4 Simulation Results	10
List of References	24
3 Space-Time Adaptive Processing	25
3.1 STAP Overview	25
3.2 Introduction to Subspace Averaging	26
3.3 Structured Covariance Matrix Simulation	28
3.4 Structured Covariance Matrix Results	30

	Page
3.5 KASSPER Simulation	32
3.6 KASSPER Simulation Results	35
List of References	42
4 Future Work	43
 APPENDIX	
A Derivation of X_1 for OSE Beamforming	45
List of References	46
B One-dimensional OSE MATLAB script	47
C Two-dimensional OSE MATLAB script	49
D Sub-Space Averaging MATLAB script	51
BIBLIOGRAPHY	53

LIST OF TABLES

Table		Page
1	KASSPER Data Set 1 Parameters	42

LIST OF FIGURES

Figure		Page
1	Notch depths of ideal DMR and OSE beamformers with INR=10 dB.	11
2	Notch depths of ideal DMR and OSE beamformers with INR=20 dB	12
3	Notch depths of ideal DMR and OSE beamformers with INR=30 dB	13
4	Illustration of sensor array perturbation for $\sigma^2 = 0.03$	14
5	Sensor position perturbation notch depths, INR=10 dB	15
6	Sensor position perturbation notch depths, INR=20 dB	16
7	Sensor position perturbation notch depths, INR=30 dB	17
8	Gain perturbation notch depths, INR=10 dB	18
9	Gain perturbation notch depths, INR=20 dB	19
10	Gain perturbation notch depths, INR=30 dB	20
11	Phase perturbation notch depths, INR=10 dB	21
12	Phase perturbation notch depths, INR=20 dB	22
13	Phase perturbation notch depths, INR=30 dB	23
14	Example of Subspace selection for Shift=3	27
15	Signal to Interference Noise Ratio for narrowband interference, INR=30 dB	30
16	Signal to Interference Noise Ratio for wideband interference, INR=30 dB	31
17	Diagonally Loaded SMI: Normalized SINR vs. Angle and Doppler, $c = 0.5$, 75 snapshots	37

Figure		Page
18	Fast Maximum Likelihood: Normalized SINR vs. Angle and Doppler, 75 snapshots	38
19	Rank-constrained Maximum Likelihood: Normalized SINR vs. Angle and Doppler, 75 snapshots	39
20	Subspace Averaging: Normalized SINR vs. Angle and Doppler, 75 snapshots	40

CHAPTER 1

Introduction

1.1 Applications

The application of Optimal Subspace Estimation (OSE) seeks to prove the merit of capitalizing upon the underlying subspace structure of certain matrices. When regular arrays (uniform linear spacing in one or two dimensions) are employed, a property called shift-invariance exists. OSE produces a subspace estimate that is constrained to have this shift-invariant structure. When this particular structure exists in the data, impressive results are achieved, which may give results that are orders of magnitude better than those obtained with standard covariance estimation techniques.

The first application considered in this thesis is adaptive beamforming. In this application, a single output signal is formed by taking a linear combination of the outputs of each sensor in an array. The weights used to form this linear combination are calculated so that a source signal from a specified direction is seen in the output signal, while source signals from all other directions are greatly attenuated. In order to calculate the weights, one must first estimate the covariance matrix from noisy data, and then extract the “signal” and “noise” subspaces. Many of the current methods use the eigenvectors of the sample covariance matrix to estimate the subspaces. This results in less than optimal results, especially with low sample support (few snapshots). OSE uses the shift-invariant subspace structure of the true (noise-free) subspace to construct a new subspace estimate that is much more accurate than the eigenvector estimate in the low sample support realm. If the array is carefully constructed/calibrated the OSE algorithm quickly approaches the lower bound for the best achievable subspace accuracy. [1]

The second application considered in this thesis is Space-Time Adaptive Pro-

cessing (STAP). STAP is a method commonly used in radar applications where clutter and jamming are of concern, such as airborne moving target indication (MTI) radar. This space-time clutter is most often colored noise which makes it very difficult to model with the constantly changing ground clutter. The motion of the airborne array causes a Doppler spreading effect that is not present in stationary radar systems which further complicates the problem[2]. Despite the colored noise the application to OSE seemed possible due to the mention of enforcing Toeplitz structures on the radar data in recent papers [3, 4].

It is easiest to demonstrate the shift-invariant property used by the OSE algorithm with an example appropriate for the beamforming application. Consider a matrix $\mathbf{Y}$, which is the product of a tall matrix $\mathbf{A}(\theta)$ of rank r and a wide iid random matrix $\mathbf{S}$, which represents the signal impinging the array [5]

$$\mathbf{Y}_{m \times n} = \mathbf{A}(\theta)_{m \times r} \mathbf{S}_{r \times n} \quad . \quad (1)$$

In this case the matrix dimensions may be thought of where m is the number of array sensors, n is the number of data snapshots, and r is the rank or number of interferers. Most importantly, the matrix $\mathbf{A}(\theta)$ is shift-invariant, which is defined in the following discussion. For the case of a one-dimensional array, a steering vector $\mathbf{a}(\omega)$ is constructed corresponding to $N = m$ array elements where ω is the spatial frequency $2\pi \cos \theta$, and θ is the direction-of-arrival of a source signal:

$$\mathbf{a}_{N \times 1}(\omega) = \begin{bmatrix} 1 \\ \mathbf{e}^{j\omega} \\ \mathbf{e}^{j2\omega} \\ \vdots \\ \mathbf{e}^{j(N-1)\omega} \end{bmatrix} . \quad (2)$$

The vector $\mathbf{a}(\omega)$ is said to have Vandermonde structure because each element in the vector is equal to the element above it times a fixed number, $\mathbf{e}^{j\omega}$. This structure is present because the source signal is narrow-band with a certain wavelength and the

uniform spacing between the sensors in the line array equals half of the wavelength. For this example, we choose the array to have five sensors, which gives the steering vector the following form,

$$\mathbf{a}_{5 \times 1}(\omega) = \begin{bmatrix} 1 \\ \mathbf{e}^{j\omega} \\ \mathbf{e}^{j2\omega} \\ \mathbf{e}^{j3\omega} \\ \mathbf{e}^{j4\omega} \end{bmatrix}. \quad (3)$$

We choose the phase reference of the array to be the center to align with the beamforming example given in chapter 2. This is simply performed by multiplying the array by $\mathbf{e}^{-(N-1)\frac{\omega}{2}}$, where N is the number of sensors:

$$\mathbf{a}(\omega) = \mathbf{e}^{-2j\omega} \begin{bmatrix} 1 \\ \mathbf{e}^{j\omega} \\ \mathbf{e}^{j2\omega} \\ \mathbf{e}^{j3\omega} \\ \mathbf{e}^{j4\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{e}^{-j2\omega} \\ \mathbf{e}^{-j\omega} \\ 1 \\ \mathbf{e}^{j\omega} \\ \mathbf{e}^{j2\omega} \end{bmatrix}. \quad (4)$$

To define the concept of shift invariance, partition $\mathbf{a}(\omega)$ into its upper and lower portions, denoted $\bar{\mathbf{a}}(\omega)$ and $\underline{\mathbf{a}}(\omega)$:

$$\bar{\mathbf{a}}(\omega) = \begin{bmatrix} \mathbf{e}^{-j2\omega} \\ \mathbf{e}^{-j\omega} \\ 1 \\ \mathbf{e}^{j\omega} \end{bmatrix}, \quad \underline{\mathbf{a}}(\omega) = \begin{bmatrix} \mathbf{e}^{-j\omega} \\ 1 \\ \mathbf{e}^{j\omega} \\ \mathbf{e}^{j2\omega} \end{bmatrix}, \quad (5)$$

and notice that

$$\bar{\mathbf{a}}(\omega)\mathbf{e}^{j\omega} = \underline{\mathbf{a}}(\omega). \quad (6)$$

That is, the upper and lower subvectors are related by a scale factor. This example may be extended to two sources. We now have,

$$\underline{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \quad \omega_k = \pi \cos(\theta_k), \quad k = 1, 2. \quad (7)$$

In this case the, $\mathbf{A}(\underline{\theta})$ has two columns, one for each source:

$$\mathbf{A}(\underline{\theta}) = \begin{bmatrix} \mathbf{e}^{-j2\omega_1} & \mathbf{e}^{-j2\omega_2} \\ \mathbf{e}^{-j\omega_1} & \mathbf{e}^{-j\omega_2} \\ 1 & 1 \\ \mathbf{e}^{j\omega_1} & \mathbf{e}^{j\omega_2} \\ \mathbf{e}^{j2\omega_1} & \mathbf{e}^{j2\omega_2} \end{bmatrix}. \quad (8)$$

In the same fashion as the one-dimensional case we split $\mathbf{A}$ into its upper and lower portions:

$$\overline{\mathbf{A}} = \begin{bmatrix} \mathbf{e}^{-j2\omega_1} & \mathbf{e}^{-j2\omega_2} \\ \mathbf{e}^{-j\omega_1} & \mathbf{e}^{-j\omega_2} \\ 1 & 1 \\ \mathbf{e}^{j\omega_1} & \mathbf{e}^{j\omega_2} \end{bmatrix}, \quad \underline{\mathbf{A}} = \begin{bmatrix} \mathbf{e}^{-j\omega_1} & \mathbf{e}^{-j\omega_2} \\ 1 & 1 \\ \mathbf{e}^{j\omega_1} & \mathbf{e}^{j\omega_2} \\ \mathbf{e}^{j2\omega_1} & \mathbf{e}^{j2\omega_2} \end{bmatrix}. \quad (9)$$

Then,

$$\overline{\mathbf{A}} \begin{bmatrix} \mathbf{e}^{j\omega_1} & 0 \\ 0 & \mathbf{e}^{j\omega_2} \end{bmatrix} = \underline{\mathbf{A}} \quad (10)$$

Notice that because $\overline{\mathbf{A}}$ is multiplied by a square, nonsingular matrix, the matrix $\underline{\mathbf{A}}$ is simply a change of basis for the columns of $\overline{\mathbf{A}}$, and therefore $\overline{\mathbf{A}}$ and $\underline{\mathbf{A}}$ share the same column space [6],

$$\text{col}(\overline{\mathbf{A}}) = \text{col}(\underline{\mathbf{A}}). \quad (11)$$

A matrix $\mathbf{A}$ satisfying equation 11 is said to be shift invariant.

In a similar fashion it is possible to show that principal left singular vectors of $\mathbf{Y}$ are also shift-invariant. Looking at the noise-free case and taking its singular value decomposition we have

$$\mathbf{Y} = \mathbf{A} \times \mathbf{S} = \mathbf{U} \Sigma \mathbf{V}^H, \quad (12)$$

The singular value decomposition of $\mathbf{Y}$ has the following structure and dimensions

$$\mathbf{Y}_{m \times n} = \left[\begin{array}{c|c} \mathbf{U}_1 & \mathbf{U}_2 \\ \hline m \times r & m \times (m-r) \end{array} \right] \left[\begin{array}{c|c} \Sigma_1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \\ \hline (m-r) \times r & (m-r) \times (n-r) \end{array} \right] \left[\begin{array}{c} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ \hline r \times n \\ (n-r) \times n \end{array} \right]. \quad (13)$$

An underlying property of the SVD is that $\text{col}(\mathbf{A}) = \text{col}(\mathbf{U}_1)$ because the columns of $\text{col}(\mathbf{U}_1)$ are a basis for the column space of $\mathbf{Y}$. Using this property we can also show that,

$$\text{col}(\overline{\mathbf{A}}) = \text{col}(\overline{\mathbf{U}}_1)$$

$$\text{col}(\underline{\mathbf{A}}) = \text{col}(\underline{\mathbf{U}}_1)$$

$$\therefore \text{col}(\overline{\mathbf{U}}_1) = \text{col}(\underline{\mathbf{U}}_1)$$

This is an important result because it shows that the left-singular vectors in $\mathbf{U}_1$ are also shift-invariant, and are a basis for the $\text{col}(\mathbf{A})$. Unfortunately, with the more realistic case where noise is added to $\mathbf{Y}$, the singular vectors of the noisy matrix no longer retain the shift-invariant property. With this in mind, a noisy matrix is constructed that is the summation of a low-rank shift invariant noise-free data matrix $\mathbf{Y}$ of rank r , with additive independent and identically distributed (iid) white noise $\mathbf{N}$.

$$\tilde{\mathbf{Y}} = \mathbf{Y} + \mathbf{N}, \quad (14)$$

The SVD of $\tilde{\mathbf{Y}}$ is

$$\tilde{\mathbf{Y}}_{m \times n} = \left[\begin{array}{c|c} \tilde{\mathbf{U}}_1 & \tilde{\mathbf{U}}_2 \\ \hline m \times r & m \times (m-r) \end{array} \right] \left[\begin{array}{c|c} \tilde{\Sigma}_1 & \mathbf{0} \\ \hline \mathbf{0} & \tilde{\Sigma}_2 \\ \hline r \times r & r \times (n-r) \\ (m-r) \times r & (m-r) \times (n-r) \end{array} \right] \left[\begin{array}{c} \tilde{\mathbf{V}}_1^T \\ \tilde{\mathbf{V}}_2^T \\ \hline r \times n \\ (n-r) \times n \end{array} \right] \quad (15)$$

where $\text{col}(\tilde{\mathbf{U}}_1)$ is the perturbed signal subspace. Note that the perturbed signal subspace is not shift-invariant and the diagonal of $\tilde{\Sigma}$ is no longer 0 beyond the rank of the underlying matrix due to the additive noise. A basis $\hat{\mathbf{U}}_1$ for the unperturbed signal subspace can be written in terms of the perturbed singular vectors as,

$$\hat{\mathbf{U}}_1 = \tilde{\mathbf{U}}_1 + \tilde{\mathbf{U}}_2 \mathbf{C} \quad (16)$$

for some coefficient matrix $\mathbf{C}$. The OSE algorithm, which is based on a first-order matrix perturbation expansion, finds a matrix $\mathbf{C}$ of the following form that makes the matrix $\hat{\mathbf{U}}_1$ shift-invariant [7, 8]:

$$\mathbf{C} = -\hat{\mathbf{Z}}\tilde{\Sigma}^{-1} \quad (17)$$

where $\hat{\mathbf{Z}}$ is a matrix computed from the SVD of $\tilde{\mathbf{Y}}$. We now have an estimate of the noise-free left singular vectors,

$$\hat{\mathbf{U}}_1 = \tilde{\mathbf{U}}_1 - \tilde{\mathbf{U}}_2 \hat{\mathbf{Z}}\tilde{\Sigma}^{-1} \quad (18)$$

and this estimate provides an asymptotically efficient estimator of the signal subspace [1].

List of References

- [1] T. A. Palka and R. J. Vaccaro, “The differential geometry of asymptotically efficient subspace estimation,” in *Proc. IEEE 48th Asilomar Conference on Signals, Systems, & Computers*, Pacific Grove, CA, November 3-5 2014.
- [2] J. R. Guerci, *Space-Time Adaptive Processing for Radar*. Norwood, MA: Artech House, 2003.
- [3] B. Kang, V. Monga, and M. Rangaswamy, “Rank-constrained maximum likelihood estimation of structured covariance matrices,” in *IEEE Trans. on Aerospace and Electronic Systems*, vol. 50, no. 1, Jan 2014.
- [4] Y. I. Abramovich, B. A. Johnson, and N. K. Spencer, “Two-dimensional multivariate parametric models for radar applications - part i: Maximum entropy extensions for & for toeplitz-block matrices,” in *IEEE Trans. on Signal Processing*, vol. 56, no. 11, Nov 2008, pp. 5509–5526.
- [5] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Englewood Cliffs, NJ: PTR Prentice Hall, 1993.
- [6] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York, NY: Cambridge University Press, 2013.
- [7] R. J. Vaccaro, “A second-order perturbation expansion for the svd,” *SIAM Journal on Matrix Analysis and Applications*, vol. 15, no. 2, pp. 661–671, April 1994.
- [8] R. J. Vaccaro and T. A. Palka, “First-order matrix perturbations for asymptotically efficient estimation of shift-invariant subspaces,” Submitted to *2014 ASILOMAR Conference*, 2014.

CHAPTER 2

Beamforming

2.1 Adaptive Beamformer

The purpose of an adaptive beamformer is to preserve signals arriving in a desired direction while attenuating loud interferers in the sensor array response. The loud interferers must be attenuated because the desired signal is often very small in relation. This response is accomplished by modifying the received sensor array beampattern by estimating the direction of the interferer and creating a null in the signal response in that direction. For a single interferer, this problem is characterized by the ratio of the power of the interferer and the power of the background noise, defined as the interference-to-noise ratio (INR). As the INR decreases, the number of snapshots needed to obtain a good estimate of the interferer direction becomes unreasonable [1].

The minimum variance distortionless response beamformer (MVDR), otherwise known as a Capon beamformer, is used for the following simulations. The beamformer weight-vector calculation uses the inverse of the interference-plus-noise covariance matrix in its calculation. In practice this matrix is unknown and must be estimated. A common choice for this estimate is the sample covariance matrix (SCM), which gives very poor results in the realm of low sample support. For this reason, a better estimate is needed to give favorable results to the beamformer output. For the beamforming example the SCM is calculated as follows [2]

$$\mathbf{S}_{SCM} = \frac{1}{L} \sum_{l=1}^L \mathbf{y}_l \mathbf{y}_l^H = \frac{1}{L} \tilde{\mathbf{Y}} \tilde{\mathbf{Y}}^H \quad (19)$$

where L is the number of data snapshots and $\mathbf{y}_l$ is the l th data snapshot or signal received by the array.

The response of the MVDR beamformer for a planar wave impinging on the sensor array is given using the form found in [1, 3]. The $N \times 1$ steering vector, or array response, given on an element by element basis is

$$\mathbf{v}_i(\theta) = \exp(j\pi \cos \theta (i - (N - 1)/2)), \quad (20)$$

where the desired listen direction is $\theta = \theta_m$. The beamformer response in the listen direction should be unity, with deep nulls placed in the direction of interferers. The vector representing the direction of interest will be referenced to as $\mathbf{v}_m$. The actual beamformer output is computed with a weight vector that achieves this result and is given by

$$y = \mathbf{w}^H \mathbf{v}_m, \quad (21)$$

The theoretical MVDR weight-vector is computed using the ensemble covariance matrix ($\mathbf{S}_{ENS}$) which is made up of the known interference plus noise. This weight vector is given as

$$\mathbf{w}_{MVDR} = (\mathbf{v}_m^H \mathbf{S}_{ENS}^{-1} \mathbf{v}_m)^{-1} \mathbf{S}_{ENS}^{-1} \mathbf{v}_m, \quad (22)$$

In practice the weight-vector must be computed using some estimate of the ensemble covariance matrix. As previously mentioned, the ensemble covariance matrix cannot be replaced with the sample covariance matrix when a small number of data snapshots are available. The first improved estimate which is used for comparison to OSE is the dominant mode rejection beamformer (DMR).

2.2 Dominant Mode Rejection Beamformer

The dominant mode rejection beamformer generates a better estimate of the ensemble covariance matrix by replacing the small eigenvalues of the SCM with a weighted average. Following the examples for DMR outlined in [1, 3], the first step of the DMR algorithm is to take the eigendecomposition of the sample covariance

matrix

$$\mathbf{S}_{SCM} = \tilde{\mathbf{E}} \tilde{\mathbf{\Lambda}} \tilde{\mathbf{E}}^H, \quad (23)$$

where $\tilde{\mathbf{E}}$ contains the eigenvectors of the SCM and $\tilde{\mathbf{\Lambda}} = \text{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_N)$ is a diagonal matrix of eigenvalues. The D large eigenvalues which are related to the interferers of $\tilde{\mathbf{\Lambda}}$ are retained, with D being the number of interferers present. The remaining eigenvalues correspond to the additive noise, and are replaced by the average of the estimated noise variance. This gives the new covariance matrix estimate

$$\mathbf{S}_{DMR} = \sum_{i=1}^D \tilde{\lambda}_i \tilde{\mathbf{e}}_i \tilde{\mathbf{e}}_i^H + \sum_{i=D+1}^N \hat{\sigma}^2 \tilde{\mathbf{e}}_i \tilde{\mathbf{e}}_i^H, \quad (24)$$

where

$$\hat{\sigma}^2 = \frac{L}{L-1} \frac{1}{N-D} \sum_{i=D+1}^N \tilde{\lambda}_i. \quad (25)$$

Substituting equation 24 into equation 22 for $\mathbf{S}_{ENS}$ generates the DMR weight vector. This weight vector is used in the beamformer simulations to quantify the performance of the DMR algorithm. The performance is measured by using the DMR weight vector to calculate the notch depth created in the beampattern steered to the angle of interest. The weight vector is calculated with the angle of interest, θ_m , which is represented as $\mathbf{w}(\theta_m)$. For the case of a single interferer arriving from the angle θ_1 , the array response $\mathbf{v}(\theta_1)$ is generated. These two vectors are then used to generate the notch depth in decibels directed towards the angle of interest.

$$ND_i = 10 \log_{10} \left| \mathbf{w}^H(\theta_m) \mathbf{v}(\theta_i) \right|^2 \text{ dB}, \quad (26)$$

2.3 Optimal Subspace Estimation Beamformer

The OSE algorithm may be used to obtain an estimate of the “signal subspace” of the ensemble covariance matrix. The procedure is outlined as follows [4]. The OSE calculations begin with an eigendecomposition of the sample covariance matrix with eigenvalues and their respective eigenvectors sorted from largest to

smallest, is partitioned as follows:

$$\tilde{\mathbf{E}}_1 = [\tilde{\mathbf{e}}_1 \cdots \tilde{\mathbf{e}}_D], \tilde{\mathbf{\Lambda}}_1 = \text{diag}(\tilde{\lambda}_1, \cdots, \tilde{\lambda}_N), \tilde{\mathbf{E}}_2 = [\tilde{\mathbf{e}}_{D+1} \cdots \tilde{\mathbf{e}}_N], \quad (27)$$

The OSE subspace estimate is

$$\mathbf{X}_1 = \tilde{\mathbf{E}}_1 - \tilde{\mathbf{E}}_2 \hat{\mathbf{Z}} \tilde{\mathbf{\Lambda}}_1^{-1/2}, \quad (28)$$

where $\hat{\mathbf{Z}}$ is computed following the method outlined in Appendix A. Let $\mathbf{P}$ and $\mathbf{P}_\perp$ be the orthogonal projection matrices onto $\text{col}(\mathbf{X}_1)$ and its orthogonal complement, respectively:

$$\mathbf{P} = \mathbf{X}_1 (\mathbf{X}_1^H \mathbf{X}_1)^{-1} \mathbf{X}_1^H, \quad (29)$$

$$\mathbf{P}_\perp = \mathbf{I} - \mathbf{P}. \quad (30)$$

The OSE covariance estimate is then:

$$\mathbf{S}_{OSE} = \mathbf{P} \mathbf{S}_{SCM} \mathbf{P} + \hat{\sigma}^2 \mathbf{P}_\perp, \quad (31)$$

where $\hat{\sigma}^2$ is estimated in the same manner as DMR using equation 25. The notch depth is calculated by substituting the following weight vector into equation 26

$$\mathbf{w}_{OSE} = (\mathbf{v}_m^H \mathbf{S}_{OSE}^{-1} \mathbf{v}_m)^{-1} \mathbf{S}_{OSE}^{-1} \mathbf{v}_m, \quad (32)$$

2.4 Simulation Results

Simulations for the DMR and OSE algorithms were completed in MATLAB. Ideal arrays were first simulated to determine the maximum notch depths achievable for each algorithm. To allow comparison to the simulation results in [1, 3] the same parameters are used. A uniform linear array of $N = 50$ sensors at half-wavelength spacing, with the sensor array response from (20) and look direction $\theta_m = \frac{\pi}{2}$. A single interferer is introduced in the direction of $\theta_1 = \text{acos}(3/N)$ which places the interferer in the main side lobe of the array response using three interference to noise ratios of 10, 20, and 30 dB. Each INR level is computed using

1000 Monte Carlo simulations. The results of each INR level are plotted for both algorithms.

As can be observed in the following plots, the OSE algorithm produces a given notch depth with far fewer snapshots than the DMR algorithm. For each INR value, the OSE algorithm produces a given notch depth using up to two orders of magnitude fewer snapshots than the DMR algorithm.

Figure 1. Notch depths of ideal DMR and OSE beamformers with INR=10 dB.

Figure 2. Notch depths of ideal DMR and OSE beamformers with INR=20 dB

Figure 3. Notch depths of ideal DMR and OSE beamformers with INR=30 dB

To test the ability of the OSE algorithm to attenuate an interferer in more realistic scenarios, the sensor array is perturbed in terms of actual sensor position and gain and phase errors caused by the sensor circuitry.

Sensor position errors are simulated by perturbing each sensor position in the line array within a small circle by using a uniformly distributed random number for five different variance levels. This is visualized in Figure 4

Figure 4. Illustration of sensor array perturbation for $\sigma^2 = 0.03$

For each variance, 100 randomly perturbed arrays are generated. Notch depths are then calculated for each unique line array for 100 Monte Carlo simulations using the three INR values with a single interferer and additive complex Gaussian white noise. The averaged notch depths for each INR value are plotted with the varying levels of perturbation. Results are shown for the first 100 snapshots as we are most interested in OSE's performance in low sample support scenarios. Figures 4, 5, and 6 give the results of the sensor perturbation simulations for the three respective INR values. The OSE algorithm results are solid lines in the figures, with DMR's being dashed lines. The five variance levels are each represented by a different color for each algorithm.

Figure 5. Sensor position perturbation notch depths, INR=10 dB

Figure 6. Sensor position perturbation notch depths, INR=20 dB

Figure 7. Sensor position perturbation notch depths, INR=30 dB

Sensor array gain errors are simulated by multiplying each sensor by one plus a random real Gaussian number within a set variance. For each level of perturbation, the individual sensor response is multiplied by a random number within the desired variance. In the same manner as the sensor position perturbations, notch depths for 100 unique arrays with 100 Monte Carlo simulations are computed and plotted for each INR value using a single interferer.

Figure 8. Gain perturbation notch depths, INR=10 dB

Figure 9. Gain perturbation notch depths, INR=20 dB

Figure 10. Gain perturbation notch depths, INR=30 dB

Sensor array phase errors are simulated by multiplying each sensor by the exponential of an imaginary Gaussian number within a set variance. For each level of perturbation the individual sensor response is multiplied by a random number within the desired variance. In the same manner as the two former perturbation scenarios, notch depths for 100 unique arrays with 100 Monte Carlo simulations are computed and plotted for each INR value using a single interferer.

Figure 11. Phase perturbation notch depths, INR=10 dB

Figure 12. Phase perturbation notch depths, INR=20 dB

Figure 13. Phase perturbation notch depths, INR=30 dB

It is clear from the three perturbation scenarios that the OSE algorithm is sensitive to changes in the linear array structure, especially with increasing INR values. Due to perturbations, the array response vectors are no longer Vandermonde. Because OSE produces a (nearly) shift-invariant subspace estimate, it will produce deep notch depths only for (nearly) Vandermonde array response vectors. In all of the perturbation simulations the DMR algorithm is nearly insusceptible to disturbance. This makes DMR well suited to applications where the array is not well calibrated. However, for applications that require deep notch depths to be generated with a small number of snapshots (fast moving objects), OSE is far desirable. For this exceptional performance to be achieved, a well-calibrated array and sensor circuitry must be used.

List of References

- [1] R. J. Vaccaro, “Improved adaptive beamforming by optimal subspace estimation,” Submitted to *2014 ASILOMAR Conference*, 2014.
- [2] K. E. Wage and J. R. Buck, “Snapshot performance of the dominant mode rejection beamformer,” in *IEEE Journal Oceanic Engineering*, vol. 39, no. 2, April 2014, pp. 212–225.
- [3] J. R. Buck and K. E. Wage, “A random matrix theory model for the dominant mode rejection beamformer notch depth,” in *Proc. IEEE Statistical Signal Processing Workshop*, 2012, pp. 820–823.
- [4] R. J. Vaccaro, “Weighted subspace fitting using subspace perturbation expansions,” ESAT-SISTA/TR 1997-45, Katholieke Universiteit Leuven, Dept. of Elec. Eng., Tech. Rep., June 1997. [Online]. Available: <ftp.esat.kuleuven.ac.be/inpub/SISTA/guests/vaccaro.ps.Z>

CHAPTER 3

Space-Time Adaptive Processing

3.1 STAP Overview

Space-time adaptive processing (STAP) is a method commonly used to reject radar clutter or jamming that involves an airborne sensor array. The radar clutter has a much stronger return than the objects of interest and is most commonly generated by stationary objects on the ground such as buildings and vegetation. With a moving array it becomes increasingly difficult to filter out the zero Doppler objects (clutter) due to Doppler spreading which is induced by the array motion. The main goal for reducing this clutter is to gain the ability to detect slow moving ground based targets. The motivation behind using STAP for this thesis application is the continued work performed in this area of ground clutter returns without having prior knowledge of the true clutter covariance matrix [1, 2].

There are many methods used for simulating the radar clutter signal, this thesis looks at two simulation scenarios that have recently been used for . The simpler problem of nulling jammers (similar to the prior ABF application) involves a simulation technique that represents the true covariance matrix with jammers from various angles [3, 1]. The second simulation model commonly used to test covariance estimation techniques is known as KASSPER and involves the much more difficult issue of rejecting ground clutter to form a good estimate of the true covariance matrix [2, 4].

It was found early into using the KASSPER data that the OSE algorithm is computationally inefficient for use with STAP and due to real-world effects the data does not have the necessary structure to provide a good estimate in the region of interest. This led to Dr. Vaccaro developing a new variation of the algorithm that provides a sub-optimal estimate and is known as Subspace Averaging (SSA).

3.2 Introduction to Subspace Averaging

Consider two $n \times r$ matrices $\mathbf{X}$ and $\mathbf{Y}$, each having rank r . A necessary and sufficient condition for the $\text{col}(\mathbf{X}) = \text{col}(\mathbf{Y})$ is

$$\mathbf{P}_{\mathbf{Y}}(\mathbf{X}) = \mathbf{X} \quad (33)$$

where $\mathbf{P}_{\mathbf{Y}}(\mathbf{Y}) = \mathbf{Y}(\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}$ is the orthogonal projection matrix onto $\text{col}(\mathbf{Y})$.

This leads to the formulation of the Subspace Averaging Algorithm (SSA). Given a set of N matrices, $n \times r$, each of rank r : $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$, let $\mathbf{W}_1 = \mathbf{X}_1$ and $\mathbf{W}_k = \mathbf{P}_{\mathbf{W}_1}(\mathbf{X}_k)$, $k = 2, \dots, N$

A basis for the average subspace is given by

$$\mathbf{B} = \sum_{k=1}^N \mathbf{W}_k. \quad (34)$$

For a uniform line array, partition the $\mathbf{U}_1$ matrix (taken from the SVD of the SCM in the same manner as equation 15) into submatrices $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$, where $N = \text{shifts} + 1$. The “shifts” represent the number of rows that are to be left out of $\mathbf{U}_1$ in order to generate the submatrices. This is easily demonstrated by showing a couple of examples.

Given the case where $\text{shifts} = 1$, we have the same partitioning of $\mathbf{U}_1$ that is employed in the calculation of the OSE algorithm.

$$\mathbf{X}_1 = \overline{\mathbf{U}}_1, \quad \mathbf{X}_2 = \underline{\mathbf{U}}_1$$

For shifts = 3,

Figure 14. Example of Subspace selection for Shift=3

The basis matrix $\mathbf{B}$ will be $(M - \text{shifts}) \times r$. The method to to construct the full subspace estimate involves first forming $\mathbf{P} = \mathbf{B}(\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T$. The shifted subspaces are then recombined in the following fashion

$$\mathbf{V}_w = \begin{bmatrix} \mathbf{P} \cdot \tilde{\mathbf{X}}_1 \\ \hline 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \hline \mathbf{P} \cdot \tilde{\mathbf{X}}_2 \\ \hline 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \hline \mathbf{P} \cdot \tilde{\mathbf{X}}_3 \\ \hline 0 & 0 & \cdots & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \hline \mathbf{P} \cdot \tilde{\mathbf{X}}_3 \end{bmatrix}, \quad (35)$$

In this example, each row of $\mathbf{V}_w$ is computed using one, two, three, or four estimates. Each row of $\mathbf{V}_w$ must be divided by the number of nonzero estimates. Thus, the final subspace estimate is

$$\mathbf{V} = \mathbf{D}^{-1} \mathbf{V}_w, \quad (36)$$

where,

$$\mathbf{D}_{M \times M} = \begin{bmatrix} 1 & & & & & & & & & \\ & 2 & & & & & & & & \\ & & 3 & & & & & & & \\ & & & 4 & & & & & & \\ & & & & 4 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 4 & & & \\ & & & & & & & 4 & & \\ & & & & & & & & 3 & \\ & & & & & & & & & 2 \\ & & & & & & & & & & 1 \end{bmatrix}. \quad (37)$$

Multiple covariance estimation techniques found in [2, 1] are used for both simulations allowing comparison to the SSA algorithm. The techniques compared have simple implementations and are similar to the SSA algorithm in that they are applied directly to the SCM. There are numerous other examples which may exhibit excellent results but often involve multiple filtering steps before they are applied, further increasing computation time and are further removed from the SSA implementation.

3.3 Structured Covariance Matrix Simulation

The first covariance simulation method constructs a covariance matrix with similar form to the KASSPER data. The array snapshots come from an N-element array that also exhibits a similar structure to the beamforming application of Chapter 2. This covariance matrix is generated with any number of jammers and has the following form. [3]

$$\mathbf{r}_{nm} = \sum_{i=1}^J \sigma_i^2 \text{sinc}(0.5\beta(n-m)\phi_i) e^{j(n-m)\phi_i} + \delta_{nm} \quad (38)$$

where σ_i^2 is the normalized power of the i th jammer, β is the bandwidth of the desired signal and is assumed to be equal to the jammer signal's bandwidth. The i th jammer signal is constructed using $\phi_i = 2\pi d(\sin\theta)/\lambda_0$, where d is the array interelement spacing, θ is the angle off boresight, λ_0 is the jammer center frequency

wavelength. δ_{nm} represents the internal noise power of the array $\delta_{nm} = 1$ when $n = m$, and 0 otherwise.

Two values of β are simulated to show the effect the jammer bandwidth has on the SINR value. When $\beta = 0$ a narrow-band jammer signal is represented, $\beta = 0.03$ gives the jammer a 3% signal bandwidth making it more wideband. As the band becomes wider it becomes increasingly difficult to resolve the true covariance matrix.

It is of note that this simulation actually represents a one-dimensional example and does not suffer the same performance issues found with the ground clutter example.

3.4 Structured Covariance Matrix Results

Figure 15. Signal to Interference Noise Ratio for narrowband interference, INR=30 dB

Figure 16. Signal to Interference Noise Ratio for wideband interference, INR=30 dB

3.5 KASSPER Simulation

The covariance estimation techniques chosen for comparison are diagonal loading of the Sample Covariance Matrix, Fast Maximum Likelihood (FML), and Rank-Constrained Maximum Likelihood (RCML). The Sample Covariance Matrix is included in the comparisons because it is used in all of the estimation techniques and shows the improvement each additional step achieves. Diagonal loading is used with the SCM because we are only interested with low sample support scenarios which results in a rank-deficient matrix that is non-invertible. The examples shown in [2] are full-rank or slightly rank deficient. Using large sample support with the KASSPER data is not ideal because there are targets of interest in each range bin. This means the assumption that the adjacent ranges to the range bin of interest does not hold up and corrupted data is therefore used in the average. In the KASSPER data the clutter samples for each range bin are simulated as follows [4],

$$\mathbf{x}_l = \sum_{p=1}^{P_{cc}} \alpha_p \mathbf{v}(\theta_p, f_p) \quad (39)$$

where the space-time steering vector is $\mathbf{v}(\theta_p)$. The desired angle of arrival and Doppler frequency are θ_p and f_p , respectively. P_{cc} represents the number of scatterers in each range bin l . α_p is a random, complex, Gaussian number that has a variance that corresponds directly to the predicted powers in SCATS. (SCATS is the Information Systems Laboratories Splatter, Clutter, and Target Signal model which is a phenomenology modeling tool used to characterize the ground clutter in the KASSPER simulation).

The space-time steering vector is the Kronecker product of two vectors,

$$\mathbf{s}(\theta_p, f_p) = \mathbf{b}(f_p) \otimes \mathbf{a}(\theta_p) \quad (40)$$

where $\mathbf{b}(f_p)$ is the temporal steering vector for Doppler frequency f_p , and $\mathbf{a}(\theta_p)$ is

the array response at angle of arrival θ_p . These vectors have the following form,

$$\mathbf{a}(\theta_p) = [1 \ e^{j\phi(\theta_p)} \ \dots \ e^{j(N-1)\phi(\theta_p)}]^H \quad (41)$$

$$\mathbf{b}(f_p) = [1 \ e^{j2\pi f_p T_r} \ \dots \ e^{j(M-1)2\pi f_p T_r}]^H \quad (42)$$

where a uniform linear array is assumed, $\phi(\theta_p)$ is the relative phase shift to the first element of the array arriving at the angle of arrival θ_p . T_r is the pulse repetition interval (PRI), which is equal to the reciprocal of the pulse repetition frequency (PRF), for the KASSPER simulation the PRF = 1984 Hz. Each range clutter sample $\mathbf{x}_l$ is an $NM \times 1$ vector where $N = 11$ antenna elements with $M = 32$ pulses. H is the Hermitian transpose operator.

The true clutter covariance matrix is formed as follows,

$$\mathbf{R}_{TC} = \sum_{p=1}^{P_{cc}} |\alpha_p|^2 \mathbf{s}(\theta_p, f_p) \mathbf{s}^H(\theta_p, f_p). \quad (43)$$

The sample covariance matrix is computed as follows,

$$\mathbf{R}_{SCM} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_k \mathbf{x}_k^H = \frac{1}{K} \mathbf{X} \mathbf{X}' \quad (44)$$

where K is the number of range bins used for calculating $\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_K]$.

The Sample Matrix Inverse (SMI) algorithm directly uses the SCM in its calculations [5]. This algorithm has good results with large sample support but becomes non-invertible when the estimate is less than full rank. This issue is resolved by incorporating diagonal loading which adds a small constant value c , to the diagonal entries of the covariance estimate. The only constraint on this value is it must be large enough to make the matrix invertible.

$$\mathbf{R}_{SMIdl} = \frac{1}{K} \mathbf{X} \mathbf{X}' + c \mathbf{I}. \quad (45)$$

The FML algorithm is easily implemented following the procedure outlined in [3]. The first step in the FML algorithm is to take the eigen decomposition of the

SCM such that,

$$\mathbf{R}_{SCM} = \tilde{\mathbf{E}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{E}}^{-1}, \quad (46)$$

where $\mathbf{\Lambda}$ is ordered from largest to smallest eigenvalues.

The eigenvalues of the SCM are then manipulated to replace the less significant values with the noise floor. With the case of the KASSPER data the noise variance of 1 is known and the noise-floor does not need to be estimated. For radar applications, prior knowledge of the noise floor is not uncommon as it can be measured by placing the radar in receive only mode when it is first powered up [2]. Let $\mathbf{\Lambda}_{FML}$ be the matrix $\tilde{\mathbf{\Lambda}}$ from equation 45 but with all diagonal entries less than the noise variance σ^2 replaced by σ^2

$$\mathbf{\Lambda}_{FML} = \text{diag}(\tilde{\mathbf{\Lambda}}) < \sigma^2 = \sigma^2. \quad (47)$$

The FML covariance matrix is then computed such that,

$$\mathbf{R}_{FML} = \tilde{\mathbf{E}}\mathbf{\Lambda}_{FML}\tilde{\mathbf{E}}^{-1}. \quad (48)$$

The Rank Constrained Maximum Likelihood (RCML) covariance estimation algorithm [2] uses a very similar approach to FML. Rather than replacing values of the eigenvalue matrix that are less than the noise floor estimate with the estimate it, incorporates knowledge of the covariance matrix rank. All values in the estimated covariance eigenvalue matrix greater than the determined rank are replaced with the noise variance. The rank in this case is estimated using Brennan's rule where, $r = N + M - 1 = 11 + 32 - 1 = 42$ [6]. The new eigenvalue matrix is easily represented as

$$\mathbf{\Lambda}_{RCML} = \begin{cases} \max(\sigma^2, \tilde{\Lambda}_i) & \text{for } i = 1, 2, \dots, r, \\ \sigma^2, & \text{for } i = r + 1, r + 2, \dots, N. \end{cases}, \quad (49)$$

and,

$$\mathbf{R}_{RCML} = \tilde{\mathbf{E}}\mathbf{\Lambda}_{RCML}\tilde{\mathbf{E}}^{-1}. \quad (50)$$

Finally, the SSA algorithm is calculated similarly to the Adaptive Beamforming case where the estimate is taken by shifting and averaging. The singular value decomposition is computed such that, using MATLAB

$$\begin{aligned}\mathbf{X}_1 &= \text{subavg2D}(\tilde{\mathbf{\Lambda}}(:, 1 : r), M, N, 3) \\ \mathbf{P} &= \mathbf{X}_1(\mathbf{X}_1^H \mathbf{X}_1)^{-1} \mathbf{X}_1^H \\ \mathbf{P}_\perp &= \mathbf{I}_{M \times N} - \mathbf{P}\end{aligned}$$

$$\mathbf{R}_{SSA} = \mathbf{P} \mathbf{R}_{SCM} \mathbf{P} + \sigma^2 \mathbf{P}_\perp \quad (51)$$

3.6 KASSPER Simulation Results

The KASSPER dataset aims to realistically model the clutter effects of an actual Side-Looking Airborne Radar System (SLAR) within a specific region of the United States . This region is modeled as a mountainous area of California that provides large changes in the clutter power returns to the radar. This model includes real-world effects such as heterogenous terrain, sub-space leakage, array errors, and a multitude of ground targets [4]. The real-world effects incorporated into the model make it difficult to estimate the covariance matrices, providing an accurate representation of the effectiveness of the different techniques employed. The majority of the algorithms tested for the STAP processing method are computationally inefficient and/or require a large number of samples to generate a good estimate of the clutter. None of methods tested perform well in the real-world scenario simulated by the KASSPER dataset.

When computing the sample covariance matrix for STAP applications it is necessary to leave out the range cell being tested at a minimum. By leaving this range cell out it ensures the target of interest is not included in the estimate. It is also common to leave out adjacent cells to the range bin under test for the case

where the target may extend between multiple range bins. These left out range bins are referred to as guardcells.

The sample covariance matrix is modified to reflect this change,

$$\mathbf{R}_{SCM} = \frac{1}{K - (1 + g)} \sum_{k=\lfloor r_i - \frac{K}{2} \rfloor}^{r_i - \frac{g}{2}} \mathbf{x}_k \mathbf{x}_k^H + \sum_{k=r_i + \frac{g}{2}}^{\lfloor r_i + \frac{K}{2} \rfloor} \mathbf{x}_k \mathbf{x}_k^H. \quad (52)$$

where r_i is the range bin of interest, g is the number of guardcells (must be an even number), and K is the number of snapshots or range bins to be averaged (minus the target and guardcells).

The Normalized SINR vs. Angle and Doppler is commonly used in radar literature to evaluate the performance of estimation techniques and is calculated as follows [2],

$$\eta = \frac{|\mathbf{s}^H \hat{\mathbf{R}}^{-1} \mathbf{s}|^2}{|\mathbf{s}^H \hat{\mathbf{R}}^{-1} \mathbf{R}_{TC} \hat{\mathbf{R}}^{-1} \mathbf{s}| |\mathbf{s}^H \mathbf{R}_{TC}^{-1} \mathbf{s}|} \quad (53)$$

Figure 17. Diagonally Loaded SMI: Normalized SINR vs. Angle and Doppler, $c = 0.5$, 75 snapshots

Figure 18. Fast Maximum Likelihood: Normalized SINR vs. Angle and Doppler, 75 snapshots

Figure 19. Rank-constrained Maximum Likelihood: Normalized SINR vs. Angle and Doppler, 75 snapshots

Figure 20. Subspace Averaging: Normalized SINR vs. Angle and Doppler, 75 snapshots

It may be observed from the results that the diagonally loaded SMI, FML, and RCML have very similar results. None of which perform remarkably with this low number of snapshots. The SSA algorithm does an excellent job of estimating the true covariance matrix outside of the low Doppler range. Unfortunately, the low doppler range is the region of interest. However, it is unclear whether this may be a good result. The SINR measure compares the known true clutter covariance matrix to the estimate. Given the data used to generate the estimate is corrupted with slow moving ground targets it may make sense that there is poor SINR in the low doppler region.

Table 1. KASSPER Data Set 1 Parameters

Parameter	Value
Carrier Frequency	1240 MHz
Bandwidth	10 MHz
No. antenna elements	11
No. pulses	32
Pulse repetition frequency	1984 Hz
1000 range bins	35-50 km
91 azimuth angles	87, 89, ... , 267 deg
128 Doppler frequencies	-992, -976.38, ... , 992 Hz
Clutter power	40 dB
No. targets	226 (200 detectable targets)
Range of target Doppler frequency	-99.2 to 372 Hz

[4]

List of References

- [1] B. Kang, V. Monga, and M. Rangaswamy, “Estimation of structured covariance matrices for radar stap under practical constraints,” in *IEEE Signal Processing Radar Conference*, May 2014, pp. 2290–2305, cincinnati, OH.
- [2] B. Kang, V. Monga, and M. Rangaswamy, “Rank-constrained maximum likelihood estimation of structured covariance matrices,” in *IEEE Trans. on Aerospace and Electronic Systems*, vol. 50, no. 1, Jan 2014.
- [3] M. Steiner and K. Gerlach, “Fast converging adaptive processor or a structured covariance matrix,” in *IEEE Trans. on Aerospace and Electronic Systems*, vol. 36, no. 4, Oct 2000, pp. 1115–1126.
- [4] J. S. Bergin and P. M. Techau, “High-fidelity site-specific radar simulation: Kassper ’02 workshop datacube,” in *KASSPER Program - Data Set Documentation*, vol. Version 1.0, May 2002.
- [5] M. W. Ganz, R. L. Moses, and S. L. Wilson, “Convergence of the smi and the diagonally loaded smi algorithms with weak interference,” in *IEEE Trans. on Antennas and Propagation*, vol. 38, no. 3, March 1990.
- [6] M. Rangaswamy, S. Kay, C. Xu, and F. C. Lin, “Model order estimation for adaptive radar clutter cancellation,” in *IEEE Waveform Diversity & Design*, 2007, pp. 339–343.

CHAPTER 4

Future Work

The applications for this thesis were chosen because it was known the signal data is processed from the array in a manner which allows the OSE algorithm to use the shift-invariant property. Both of these applications apply directly to array processing but the application of OSE and the SSA algorithm may prove useful for any application where a subspace estimate is to be generated from the unperturbed signal subspace.

Within the beamforming application in Chapter 2 there may be further processing desired where OSE may still be desirable regardless of common performance metrics. It was shown in Chapter 2 that the OSE algorithm easily outperforms DMR when the signal matrix is formed from a truly linear array. The OSE algorithm has trouble estimating the subspace when a poorly calibrated array is introduced because the resulting signal matrix no longer has the structure the algorithm relies upon. However, it is of note that this does not mean that DMR is necessarily a better method. Some applications may require a covariance estimate that is closer to the actual subspace. In situations where an estimate of the actual underlying subspace is needed, OSE becomes a better choice. The subspace generated by OSE is shift-invariant and is a better estimate of the unperturbed signal space. DMR generates a better subspace of the perturbed array (recorded data) and should be used for applications where this property is desired.

Applying the OSE algorithm to the KASSPER data was immediately found to be computationally inefficient due to the size of the signal data and resulting number of computations needed for an estimate. The Subspace Averaging algorithm allows for much faster computation but was still found to have less than

favorable results in the Doppler region of interest. The true clutter covariance matrix used to generate the SINR measure does not contain targets. It is possible the SSA algorithm exhibits poor SINR due to the targets of adjacent range cells being included in the signal data. Due to the necessity of averaging adjacent range cells it may be that airborne-MTI radar is not an ideal application for SSA where direct processing on the signal data is needed. It may be that the SSA or OSE algorithm do have effective applications in STAP that may be identified by individuals with practical experience in this field.

In relation to both applications another avenue of research may focus on ensuring the data received by the array is shift-invariant. Array perturbations have an adverse effect on the performance of the OSE algorithm so it may be possible to minimize the impact of these array perturbations. This may involve performing some form of prefiltering to the received data to restore the shift-invariance that should have been present from the uniform line array.

APPENDIX A

Derivation of $\mathbf{X}_1$ for OSE Beamforming

The singular value decomposition of the sample covariance matrix which has been generated from the $N \times L$ snapshot matrix, N being the number of array elements, and L the number of snapshots, is taken such that

$$\mathbf{S}_{SCM} = \mathbf{U}\mathbf{S}\mathbf{V}^*, \quad (\text{A.54})$$

where $\mathbf{U}$ is $N \times N$, $\mathbf{S}$ is $N \times L$, and $\mathbf{V}$ is $L \times L$. $\mathbf{U}$ is then partitioned such that

$$\begin{aligned} \mathbf{U}_1 &= [\mathbf{u}_1 \cdots \mathbf{u}_D], & \mathbf{U}_2 &= [\mathbf{u}_{D+1} \cdots \mathbf{u}_N], \\ \overline{\mathbf{U}}_1 &= [\mathbf{u}_{1_1} \cdots \mathbf{u}_{1_{N-1}}], & \overline{\mathbf{U}}_2 &= [\mathbf{u}_{2_1} \cdots \mathbf{u}_{2_{N-1}}], \\ \underline{\mathbf{U}}_1 &= [\mathbf{u}_{1_2} \cdots \mathbf{u}_{1_N}], & \underline{\mathbf{U}}_2 &= [\mathbf{u}_{2_2} \cdots \mathbf{u}_{2_N}], \\ \underline{\mathbf{U}}_1^\dagger &= (\underline{\mathbf{U}}_1^* \underline{\mathbf{U}}_1)^{-1} \underline{\mathbf{U}}_1^*, \end{aligned}$$

and

$$\mathbf{S}_1 = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1D} \\ s_{21} & s_{22} & \cdots & s_{2D} \\ \vdots & & \ddots & \vdots \\ s_{D1} & s_{D2} & \cdots & s_{DD} \end{bmatrix}$$

$$\mathbf{P}_{\perp 1} = \mathbf{I} - \underline{\mathbf{U}}_1 \underline{\mathbf{U}}_1^\dagger, \quad (\text{A.55})$$

$$\tilde{\mathbf{H}}_{LS} = [\mathbf{S}_1^{-1} \otimes (\mathbf{W}^* \overline{\mathbf{U}}_2)] - [(\mathbf{S}_1^{-1} \underline{\mathbf{U}}_1^\dagger \overline{\mathbf{U}}_1)^T \otimes (\mathbf{W}^* \underline{\mathbf{U}}_2)], \quad (\text{A.56})$$

where $(\mathbf{W})$ is the orthonormal basis of $\mathbf{P}_{\perp 1}$ and $\otimes$ is the Kronecker product.

The rank of the constraint equation $\tilde{\mathbf{H}}_{LS}$ is then determined as [1]

$$r = \text{rank}(\tilde{\mathbf{H}}_{LS}) = D(N - D) - D, \quad (\text{A.57})$$

the singular value decomposition is then taken and partitioned as follows

$$\tilde{\mathbf{H}}_{LS} = \mathbf{U}_3 \mathbf{S}_2 \mathbf{V}_2^*, \quad (\text{A.58})$$

$$\begin{aligned}
\mathbf{U}_4 &= [\mathbf{u}_{3_1} \cdots \mathbf{u}_{3_r}] \\
\mathbf{S}_3 &= \begin{bmatrix} s_{2_1 1} & s_{2_1 2} & \cdots & s_{2_1 r} \\ s_{2_2 1} & s_{2_2 2} & \cdots & s_{2_2 r} \\ \vdots & & \ddots & \vdots * \vdots \\ s_{2_r 1} & s_{2_r 2} & \cdots & s_{2_r r} \end{bmatrix} \\
\mathbf{V}_3 &= [\mathbf{v}_{2_1} \cdots \mathbf{v}_{2_r}]
\end{aligned} \tag{A.59}$$

finally

$$\hat{\mathbf{Z}} = \mathbf{V}_3 \mathbf{S}_3^{-1} \mathbf{U}_4^* \mathbf{r}_{\text{LS}}, \tag{A.60}$$

where

$$\mathbf{r}_{\text{LS}} = \text{vect}(\mathbf{W}^* \overline{\mathbf{U}}_1), \tag{A.61}$$

the final calculations for the OSE algorithm are defined as follows:

$$\mathbf{X}_1 = \text{orthogonal basis of } \mathbf{X}_1 \text{ as defined in 28} \tag{A.62}$$

$$\mathbf{P}_{\perp 2} = \mathbf{I} - \mathbf{X}_1 \mathbf{X}_1^H \tag{A.63}$$

taking the eigenvalue decomposition

$$\mathbf{X}_1^H \mathbf{S}_{\text{SCM}} \mathbf{X}_1 = \mathbf{Q} \mathbf{E} \mathbf{Q}^{-1} \tag{A.64}$$

$$\mathbf{U}_{\text{OSE}} = \mathbf{X}_1 \mathbf{Q} \tag{A.65}$$

$$\mathbf{S}_{\text{OSE}} = \mathbf{U}_{\text{OSE}} \mathbf{E} \mathbf{U}_{\text{OSE}}^H + \sigma^2 \mathbf{P}_{\perp 2} \tag{A.66}$$

List of References

- [1] R. J. Vaccaro and T. A. Palka, “First-order matrix perturbations for asymptotically efficient estimation of shift-invariant subspaces,” Submitted to *2014 ASILOMAR Conference*, 2014.

APPENDIX B

One-dimensional OSE MATLAB script

```
function [ X1 ] = OSE_1D( signal, r, L )

%OSE_1D Calculates basis for subspace of one-dimensional array data
%   signal = signal received by linear array, r = rank of signal ...
%   matrix, L
%       L = # of snapshots
%       Singular Value Decomposition of signal (sample) matrix
[U,S,V] = svd(signal/sqrt(L)) ; % signal is NxL ...
%       matrix, U (left singular vectors is NxN
%               % S (singular values) is ...
%               NxL, V (right ...
%               singular) is LxL

N = length(U) ;
U1 = U(:,1:r) ;
    if r < N
        U2 = U(:, r+1:N) ;
    end

U1up = U1(1:N-1,:) ; %Takes upper portion of matrix
U1low = U1(2:N, :) ; % lower portion
U2up = U2(1:N-1,:) ;
U2low = U2(2:N, :) ;
U1lowdag = inv(U1low'*U1low)*U1low' ;
S1 = S(1:r, 1:r) ;
S1i = inv(S1) ;

P = U1low *U1lowdag ;
```

```

Pperp = eye(N-1) - P ;
W = orth(Pperp) ;

H_LS = kron(S1i,W'*U2up) - ...
        kron((S1i*U1lowdag*U1up).', W'*U2low) ;

q = r*(N-r) - r ; %eqn 14

[u,s,v] = svd(H_LS) ;
u1 = u(:,1:q) ;
s1 = s(1:q,1:q) ;
v1 = v(:,1:q) ;

r_LS = W'*U1up ;
r_LS = r_LS(:) ; %places r in vector (columns ...
                become stacked in rows)

z = v1*inv(s1)*u1'*r_LS ;

Zhat = reshape(z, N-r,r) ;

%X1 must be calculated from SVD of SCM not signal
X1 = U1 - U2*Zhat*inv(S1) ; % inv(S1) = Lam^.- (1/2)

% single interferer optimal subspace estimate
X1 = orth(X1) ;

end

```

APPENDIX C

Two-dimensional OSE MATLAB script

```
function X=OSE2D(Y,p,M,N)

[U,S,V]=svd(Y);
U1=U(:,1:p);
U2=U(:,p+1:end);
S1i=inv(diag(sqrt(diag(S(1:p,1:p))))); %changed from inv(S(1:p,1:p))
L=M*N;
L1=(M-1)*(N-1);
% p=length(S1i);
index=calc_indices(M,N)
U1up=U1(index(:,1),:); % changed "indices" to index
U2up=U2(index(:,1),:);
row1=p*(L1-p);
H=zeros(3*row1,p*(L-p));
rhs=zeros(3*row1,1);
for k=1:3
    U1down=U1(index(:,k+1),:);
    U2down=U2(index(:,k+1),:);
    rows=[(k-1)*row1+1:k*row1]';
    U1dd=(U1down'*U1down)\U1down';
    W=orth(eye(L1)-U1down*U1dd);
    H(rows,:)=kron(S1i,W'*U2up)-kron((S1i*U1dd*U1up).',W'*U2down);
    rhs(rows)=W'*U1up;
    rhs=rhs(:);
end
[u,s,v]=svd(H);
r=p*(L-p)-2*p;
```

```
u1=u(:,1:r);  
s1=s(1:r,1:r);  
v1=v(:,1:r);  
z=v1/s1*u1'*rhs;  
Z=reshape(z,L-p,p);  
X=U1-U2*Z*S1i;  
X=orth(X);  
end
```

APPENDIX D

Sub-Space Averaging MATLAB script

```
function X=subavg_2D(U1,M,N,s)
index=calc_inds(N,M,s);
count=zeros(M*N,1);
for k=1:3*s+1
v=zeros(M*N,1);
v(index(:,k))=1;
count=count+v;
end
D=diag(count);
[MN,r]=size(U1);
W1=U1(index(:,1),:);
W=W1;
for k=1:3*s
    U1down=U1(index(:,k+1),:);
    U1dd=(U1down'*U1down)\U1down';
    P=U1down*U1dd;
    Z=P*W1;
    W=W+Z;
end
W=orth(W);
X=zeros(M*N,r);
X(index(:,1),:)=W*(W'*U1(index(:,1),:));
for k=1:3*s
    X(index(:,k+1),:)=X(index(:,k+1),:)+W*(W'*U1(index(:,k+1),:));
end
X=orth(D\X);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function index=calc_inds(M,N,s)
index=zeros((M-s)*(N-s),3*s+1);
% create s horizontal, s vertical, and s diagonal shifts
x=[1:M-s]';
base=[x;zeros((M-s)*(N-s-1),1)];
    for k=1:N-s-1
        base(k*(M-s)+1:(k+1)*(M-s))=x+k*M*ones(M-s,1);
    end
index(:,1) = base; % Upper left subarray
    for k=1:s % create horizontal shifts
        index(:,k+1)=base+k*M;
    end
    for k=1:s % create vertical shifts
        index(:,s+1+k)=base+k;
    end
    for k=1:s % create diagonal shifts
        index(:,2*s+1+k)=base+k*M+k;
    end
end

```


BIBLIOGRAPHY

- Abramovich, Y. I., Johnson, B. A., and Spencer, N. K., “Two-dimensional multivariate parametric models for radar applications - part i: Maximum entropy extensions for & for toeplitz-block matrices,” in *IEEE Trans. on Signal Processing*, vol. 56, no. 11, Nov 2008, pp. 5509–5526.
- Abramovich, Y. I., Johnson, B. A., and Spencer, N. K., “Sample-deficient adaptive detection: Adaptive scalar thresholding versus cfar detector performance,” in *IEEE Trans. on Aerospace and Electronic Systems*, vol. 46, no. 1, Jan 2010, pp. 32–46.
- Abramovich, Y. I., Rangaswamy, M., Johnson, B. A., Corbell, P. M., and Spencer, N. K., “Performance analysis of two-dimensional parametric stap for airborne radar using kassper data,” in *IEEE Trans. on Aerospace and Electronic Systems*, vol. 47, no. 1, Jan 2011, pp. 118–139.
- Bergin, J. S. and Techau, P. M., “High-fidelity site-specific radar simulation: Kassper ’02 workshop datacube,” in *KASSPER Program - Data Set Documentation*, vol. Version 1.0, May 2002.
- Buck, J. R. and Wage, K. E., “A random matrix theory model for the dominant mode rejection beamformer notch depth,” in *Proc. IEEE Statistical Signal Processing Workshop*, 2012, pp. 820–823.
- Ganz, M. W., Moses, R. L., and Wilson, S. L., “Convergence of the smi and the diagonally loaded smi algorithms with weak interference,” in *IEEE Trans. on Antennas and Propagation*, vol. 38, no. 3, March 1990.
- Guerci, J. and Baranoski, E., “Knowledge-aided adaptive radar at darpa,” in *IEEE Signal Processing Magazine*, Jan 2006, pp. 41–50.
- Guerci, J. R., *Space-Time Adaptive Processing for Radar*. Norwood, MA: Artech House, 2003.
- Horn, R. A. and Johnson, C. R., *Matrix Analysis*, 2nd ed. New York, NY: Cambridge University Press, 2013.
- Johnson, D. H. and Dudgeon, D. E., *Array Signal Processing: Concepts and Techniques*. Englewood Cliffs, NJ: PTR Prentice Hall, 1993.
- Johnson, D. H. and Dudgeon, D. E., “Array signal processing: Concepts and techniques,” in *Prentice Hall Signal Processing Series*, vol. 1, 1993.

- Kang, B., Monga, V., and Rangaswamy, M., “Estimation of structured covariance matrices for radar stap under practical constraints,” in *IEEE Signal Processing Radar Conference*, May 2014, pp. 2290–2305, Cincinnati, OH.
- Kang, B., Monga, V., and Rangaswamy, M., “Rank-constrained maximum likelihood estimation of structured covariance matrices,” in *IEEE Trans. on Aerospace and Electronic Systems*, vol. 50, no. 1, Jan 2014.
- Melvin, W. L., “A stap overview,” in *IEEE Aerospace and Electronics Systems Magazine*, vol. 19, no. 1, Jan 2004, pp. 19–35.
- Palka, T. A. and Vaccaro, R. J., “The differential geometry of asymptotically efficient subspace estimation,” in *Proc. IEEE 48th Asilomar Conference on Signals, Systems, & Computers*, Pacific Grove, CA, November 3–5 2014.
- Palka, T. A. and Vaccaro, R. J., “The differential geometry of asymptotically efficient subspace estimation,” in *Proceedings IEEE Asilomar Conference on Signals, Systems & Computers*, November 3–5 2014, pp. 460–464.
- Rangaswamy, M., Kay, S., Xu, C., and Lin, F. C., “Model order estimation for adaptive radar clutter cancellation,” in *IEEE Waveform Diversity & Design*, 2007, pp. 339–343.
- So, H. C., Chan, K. W., Chan, Y. T., and Ho, K. C., “Linear prediction approach for efficient frequency estimation of multiple real sinusoid: Algorithms and analyses,” in *IEEE Trans. on Signal Processing*, vol. 53, no. 7, Jul 2005, pp. 2290–2305.
- Steiner, M. and Gerlach, K., “Fast converging adaptive processor or a structured covariance matrix,” in *IEEE Trans. on Aerospace and Electronic Systems*, vol. 36, no. 4, Oct 2000, pp. 1115–1126.
- University of Rhode Island. “A guide to producing your thesis with latex.” jun 2006. [Online]. Available: <http://www.ele.uri.edu/info/thesis/guide>
- Vaccaro, R. J., “A second-order perturbation expansion for the svd,” *SIAM Journal on Matrix Analysis and Applications*, vol. 15, no. 2, pp. 661–671, April 1994.
- Vaccaro, R. J., “Weighted subspace fitting using subspace perturbation expansions,” ESAT-SISTA/TR 1997-45, Katholieke Universiteit Leuven, Dept. of Elec. Eng., Tech. Rep., June 1997. [Online]. Available: <ftp.esat.kuleuven.ac.be/inpub/SISTA/guests/vaccaro.ps.Z>
- Vaccaro, R. J., “Improved adaptive beamforming by optimal subspace estimation,” Submitted to *2014 ASILOMAR Conference*, 2014.

- Vaccaro, R. J. and Owsley, N. L., “Robust direction finding,” in *Proc. IEEE 38th Asilomar Conference on Signals, Systems & Computers*, November 7-10 2004.
- Vaccaro, R. J. and Palka, T. A., “First-order matrix perturbations for asymptotically efficient estimation of shift-invariant subspaces,” Submitted to *2014 ASILOMAR Conference*, 2014.
- Wage, K. E. and Buck, J. R., “Snapshot performance of the dominant mode rejection beamformer,” in *IEEE Journal Oceanic Engineering*, vol. 39, no. 2, April 2014, pp. 212–225.