INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

BLIND ADAPTIVE SENSOR-ARRAY PROCESSING TECHNIQUES FOR EXTRACTING HIGHLY CORRUPTED RADIO COMMUNICATION SIGNALS

By

Eric April, B.A.Sc.

A THESIS SUBMITTED TO THE SCHOOL OF
GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

OTTAWA-CARLETON INSTITUTE FOR ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING
OTTAWA UNIVERSITY

September 1996

Abstract

This thesis examines the techniques for blindly extracting highly corrupted radio communication signals with the use of sensor arrays. Initially, existing work is reviewed. The blind signal extraction techniques reviewed do not require known *a priori* training signals, signals direction of arrival, sensor array geometry or calibration. However they are found generally to be dependent upon some prior signal assumptions made as their basis for separating signals. A new blind adaptive filtering technique called the *Self-Adapting sensor array MUlti-Signal blind Extractor*, or S-AMUSE, is proposed by the author to provide a general truly blind adaptive filtering framework based on a multi-stage approach. The technique is described in detail and results are presented *via* Monte Carlo simulations. The performance results obtained are highly promising.

Acknowledgements

En tout premier lieu, je désire remercier ma très chère épouse, Maryse, pour son amour, ses encouragements, ses suggestions et ses petites attentions à mon égard. Aussi, mes parents, qui ont et continuent toujours de m'apporter un soutien indéfectible dans toutes mes entreprises, quelle qu'elles soient. Cette capacité de ne pas abandonner, c'est à vous que je la dois.

I wish to gratefully thank my supervisor, Dr. Abbas Yongaçoglu, for being so kind and comprehensive, for providing me the desired lattitude in the work, and finally, for his guidance, review and suggestions made for this thesis. Thank you Abbas.

Finally, I would like to thank my working colleagues at the Communication Research Center (CRC) and the Defence Research Establishment Ottawa (DREO) who gave me their help and support. In particular, I wish to sincerely thank my closest and respected colleague, Mr. Will Read, for his suggestions and for sharing his valuable knowledge with me. I also wish to express my deepest gratitude towards Mr. Pierre Yansouni, who greatly encouraged me and gave me his full support to complete this thesis.

Abbreviations and Acronyms

AM	amplitude modulation
BASAP	blind adaptive sensor array processor
BER	bit error rate
BPSK	binary phase-shift keying
CMA	constant-modulus algorithm
CMAR	constant-modulus array
CW	continuous-wave
DOA	direction of arrival
FFT	fast Fourier transform
FIR	finite impulse response
FM	frequency modulation
FOSLD	first-order spectral line detector
FRESH	frequency-shift filtering
HOSLD	higher-order spectral line detector

IIR	infinite impulse response
LMS	least-mean square
LS-CMA	least-square constant-modulus algorithm
LTI	linear time-invariant
m-FSK	m-ary frequency-shift keying
m-PSK	m-ary phase-shift keying
m-QAM	m-ary quadrature amplitude modulation
MMSE	minimum mean-squared error
MSE	mean-squared error
NATO	North Atlantic treaty organization
NMSE	normalized mean-squared error
NMSER	normalized mean-squared error ratio
NSINR	normalized signal to interference and noise ratio
MVDR	minimum variance distortionless response
PCCA	programmable canonical correlation algorithm
QPSK	quadrature phase-shift keying
RF	radio frequency
S-AMUSE	self-adapting sensor array multi-signal blind extractor
SCORE	spectral self-coherence restoral algorithm

SER	symbol error rate
SINR	signal to interference and noise ratio
SNOI	signal not of interest
SNR	signal to noise ratio
SOI	signal of interest
STANAG	NATO standard agreement
SVD	singular value decomposition

List of Symbols

$\mathbf{a}(\underline{\theta}, f)$	array response vector to a signal coming from direction $\underline{\theta}$ and RF f
$\mathbf{d}(\mathbf{n})$	control reference path data vector
f_c	center frequency
f_o	RF center frequency
f_s	sampling frequency
f_{symb}	symbol rate
g_l	amplitude of the l^{th} received signal
l	signal number
L	number of signal sources received
m	number of levels for digitally modulated signals
m_a	sampled audio signal waveform
m_s	message symbol stream
M	number of sensors
M_a	maximum audio signal amplitude

n	sampling time instant
N	number of time samples
N_β	number of cycle frequencies
$N_{\overline{\beta}}$	number of conjugate cycle frequencies
N_p	number of subpaths
N_s	block size in samples
N_{symb}	number of symbols contained in a transmitted message
N_τ	number of lags
$\mathbf{P}$	sensor array position matrix expressed in meters
$\mathbf{P}(f_o)$	sensor array position matrix expressed in wavelengths
q_{r-c}	raised-cosine pulse shape
q_{rect}	rectangular pulse shape
ϕ_o	arbitrary phase offset
$\mathbf{R}_{\mathbf{xd}}$	cross-correlation matrix of vectors $\mathbf{x}(n)$ and $\mathbf{d}(n)$ evaluated at lag $\tau = 0$
$\mathbf{R}_{\mathbf{xx}}$	auto-correlation matrix of vector $\mathbf{x}(n)$ evaluated at lag $\tau = 0$
$\mathbf{s}(n)$	complex baseband signals source vector
$s_l(n)$	l^{th} complex baseband signal source waveform
$s(n)$	some complex baseband signal source waveform

$s_{am}(n)$	AM modulated signal waveform
$s_{cw}(n)$	complex CW signal waveform
$s_{fm}(n)$	FM modulated signal waveform
$s_{m-fsk}(n)$	m-ary FSK modulated signal waveform
$s_{m-qam}(n)$	m-ary QAM modulated signal waveform
$s_{m-psk}(n)$	m-ary PSK modulated signal waveform
$s_{st}(n)$	STANAG4285 modulated signal waveform
$\mathbf{S}$	signal source data matrix
$\hat{\mathbf{s}}(n)$	estimated signals vector
$\hat{s}_l(n)$	l^{th} estimated signal waveform
$\hat{\mathbf{S}}$	estimated signal data matrix
t	continuous time instant
T_s	sampling period
T_{symb}	symbol period
$u(t)$	radio signal source waveform
$\mathbf{v}(n)$	sensor received complex noise vector snapshot
$\mathbf{w}_l$	filter weights vector to extract l^{th} signal
$\mathbf{W}$	filter weights matrix
$\mathbf{x}(n)$	sensor received snapshot complex vector

$x_i(n)$	i^{th} sensor received complex waveform
$\mathbf{X}$	sensor received complex data matrix
$\alpha_{l,p}$	subpath relative gain for l^{th} signal and p^{th} subpath
β	cycle frequency
$\overline{\beta}$	conjugate cycle frequency
Δf_p	peak frequency deviation
μ	amplitude modulation index
$\underline{\theta}$	direction of arrival
θ_{az}	azimuth bearing DOA
θ_{el}	elevation bearing DOA
ρ	constant modulus measure
τ	time lag
$\tau_{l,p}$	subpath relative delay for l^{th} signal and p^{th} subpath

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Statement and Motivation

A radio communication surveillance receiver system typically intercepts a specific band of the radio frequency (RF) spectrum from which signal activity is being detected. This intercepted RF spectral band potentially contains a number of desired signals referred to as signals of interest (SOIs), a number of interferers known as signals not of interest (SNOIs), and finally some background noise. Consequently, all these signals are superimposed in the sense that they coexist in both the temporal and the spectral domain. This phenomenon may be referred to as co-channel interference. In addition to this co-channel interference situation, each transmitted signal is individually subject to a distinctive and unknown channel distortion effect (e.g. multipath propagation). In this context, SOIs are not only unknown and varying, but the undesired interfering signals and masking noise also fluctuate unpredictably. In short, each individually and originally transmitted signal source is highly corrupted by its environment when received at the radio surveillance receiver system.

For obvious reasons, SOIs cannot be exploited directly in this context. To successfully exploit individual SOIs, advanced blind adaptive signal processing techniques,

able to blindly extract each desired signal from all other signal sources in some optimum way, are essential. The blind constraints primarily refer to the unavailable prior knowledge of training signals, direction of arrival (DOA), and array geometry or array calibration. This problem is referred to herein as blind adaptive signal extraction but may also be termed blind adaptive waveform estimation, signal recovery, signal separation, or signal copy.

The purpose of this thesis is to present blind adaptive sensor-array processing techniques applicable to an unknown highly corrupted radio communication signal environment. The objective of the techniques is to extract each of the SOIs as best as possible in a non-cooperative receiving context, i.e. with the above-stated blind constraints along with only minimal assumptions of the SOI characteristics. To achieve this objective, the work is delimited in scope to the use of:

1. a multi-sensor digital coherent receiver system providing a sampled baseband complex data stream received by the sensors disposed in any arbitrary geometry,

2. a general spatio-temporal filtering structure able to exploit either or both spatial and temporal information, and

3. an adaptive technique deriving its signal knowledge requirements from the received data.

A conceptual block diagram of the receiving system can be viewed in Fig. 1.1. M sensors, individually composed of an antenna and an analog front-end receiver tuned to a common RF and bandwidth, are used to provide the analog received signals to a multi-channel digitizer. The digitizer simultaneously digitizes all the analog signals coming off the receivers at some arbitrary uniform sampling rate. The output provided by the digitizer is a vector $\mathbf{x}$ of M complex samples coming out at each sampling time instant, denoted n. This digital data stream is then passed to the

blind adaptive sensor array processor (BASAP) block. This block is responsible for analyzing and filtering the incoming data by properly adapting the filter weights such that each of the L_d filtered outputs, denoted $\hat{s}_l(n)$, comes out as the best possible copy of one of the L originally transmitted signal sources, denoted $s_l(n)$. The bulk of this thesis is primarily concerned with best defining this last block.

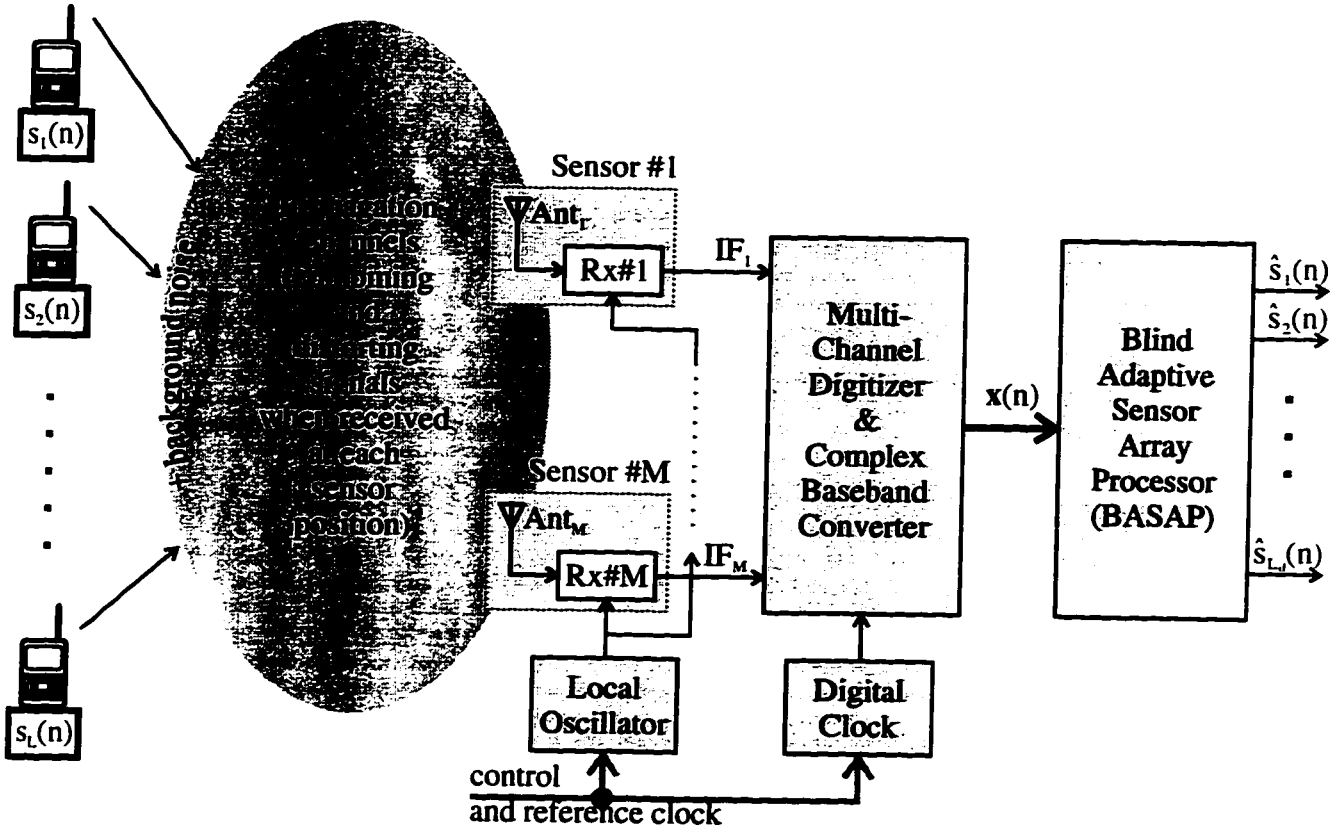

Figure 1.1: System conceptual block diagram.

As pointed out previously, the filter structure considered is limited to a spatio-temporal filter. This is indeed a very general filter structure by nature. Although filter structures will be discussed in much more details in the next chapter, two examples of such spatio-temporal filters are shown in Figs 1.2 and 1.3. The filter structures shown combine a linear spatial beamforming filter with a frequency shift (FRESH)

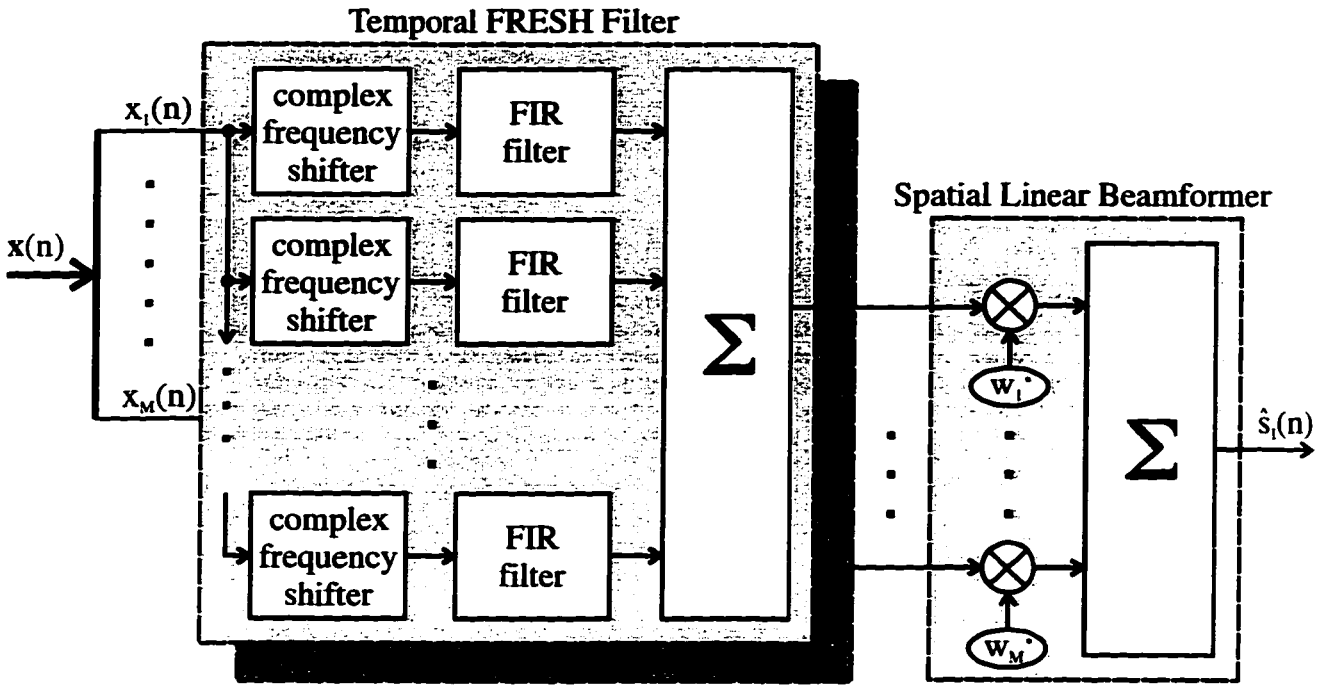

Figure 1.2: Spatio-temporal filter structure - spatial linear beamformer last.

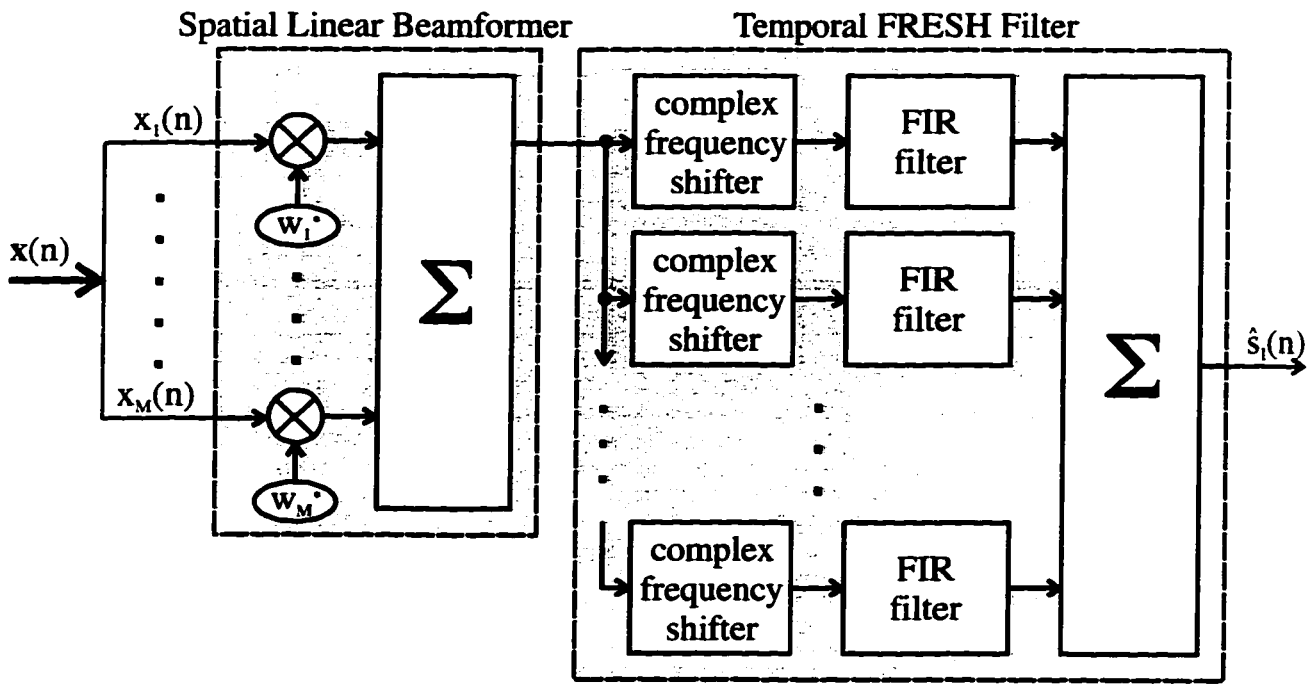

Figure 1.3: Spatio-temporal filter structure - spatial linear beamformer first.

temporal filter[1]. In practice, however, the simpler restricted spatial filters have been much more popular to solve blind signal extraction problems. The difficulty to blindly adapt combined spatio-temporal filters in some joint optimum way explains their lack of popularity. Not only the problem is difficult to solve and the filters more difficult to apply but, in the blind context of interest, existing solutions are not known to exist for these filter structures. As a result, the techniques reviewed tackle the problem of interest with a simple linear spatial beamformer. For all these reasons and also to limit the scope of this thesis, the focus will be given to blind adaptive techniques aimed at updating the filter weights of a linear spatial beamformer. Nonetheless, considerations of the spatio-temporal filters will be made to some extent in Chapter 2. Choosing the filter structure is an important element in limiting the best possible level of performance attainable, but blindly and adaptively updating the filter weights in an optimum way constitutes, surely, the key issue in this thesis.

1.2 Thesis Organization

The body of the dissertation is divided into three chapters, preceded by this introduction and followed by a chapter containing conclusions and suggestions for further work.

Chapter 2 provides the mathematical background, necessary to understand the context in which blind adaptive sensor-array processing is applied. The first section establishes the general definitions and notation rules used throughout the dissertation. In the next section, expressions for selected analogically and digitally modulated signal source types are defined in a sample-based mathematical formulation. A description of the sensor array signal models applicable to various conditions follows. In

[1]FRESH filters can be seen as extensions of linear finite-impulse response (FIR) temporal filters also known as periodically time-varying FIR filters (cf [11, 22]). Note however that FRESH filters are not limited to the use of FIR filters.

the fourth section, the potential filter structures to achieve blind signal extraction are considered. The following section defines a number of possible performance measures and factors used to evaluate the performance of any blind adaptive signal extraction technique. The next section of the chapter reviews selected existing conventional and blind adaptive signal extraction techniques in some details, with the results of a common signal scenario analyzed for each technique. Some of the techniques are tested in a version slightly modified by the author. Finally, the last section provides a summary of the Chapter.

Chapter 3 presents a newly designed blind adaptive sensor array processing techniques aimed at blindly extracting signals. The blind adaptive signal extraction technique is referred to as the S-AMUSE[2] technique and is the result of the author's work. The various aspects of the algorithm are initially considered. A detailed example is then provided. At last, implementation and coding issues are briefly discussed.

In Chapter 4, meaningful signal scenarios are defined and the performance of S-AMUSE is analyzed via Monte Carlo simulations. A discussion concerning the performance results completes the chapter.

Finally, Chapter 5 contains a summary of the main conclusions of the work comprised in this dissertation, and some suggestions for further work.

1.3 Contributions of this Thesis

The following lists from the thesis work what the author considers to be additional contributions to the field:

1. two new measures called the normalized signal to interference and noise ratio (NSINR) and the normalized mean-squared error ratio (NMSER) as defined in Section 2.5;

[2]S-AMUSE stands for *Self-Adapting sensor array Multi-Signal blind Extractor.*

2. the details of the S-AMUSE technique, a new blind adaptive signal extraction algorithm designed by the author, as provided in Chapter 3;

3. evaluation of the S-AMUSE performance via a detailed example and Monte Carlo simulations, as provided in Chapter 3 and Chapter 4.

Note that the second and third items of this list are considered to be the major contributions of this thesis.

Chapter 2

Mathematical Foundation

2.1 General Definitions and Notation

Let $s(t)$ be the baseband complex envelope[1] of some radio signal source, $u(t)$, transmitted at a continuous time instant t and an arbitrary radio frequency (RF), f_o. Mathematically, we have [43]

$$u(t) \triangleq \Re\left\{s(t)e^{j2\pi f_o t}\right\} \tag{2.1}$$

where $\Re\{\cdot\}$ specifies the real portion of its argument. The sampled representation of $s(t)$ is then denoted $s(n)$ where n is an integer specifying the sampling time instant. More precisely,

$$s(n) = s(nT_s)\,, \tag{2.2}$$

provided uniform sampling is used and $s(t)$ occupies a bandwidth[2] less than $f_s = 1/T_s$ where T_s is the sampling period in seconds and, conversely, f_s the sampling frequency in Hertz (Hz). This latter condition satisfies the Nyquist sampling criteria for a complex analytic waveform (recall that for a real signal, the bandwidth must be less

[1] Other terms, such as baseband analytic version for instance, can alternatively be used.

[2] The term bandwidth refers to a band of frequency large enough to recover the information conveyed by the signal.

than $f_s/2$ instead of f_s). Using matrix notation, the sampled complex waveform is conveniently assembled in a $1 \times N$ vector

$$\mathbf{s}(n) = \begin{bmatrix} s(n) & s(n+1) & \cdots & s(n+N-1) \end{bmatrix}, \tag{2.3}$$

or equivalently,

$$\mathbf{s}(n) = s(n : n+N-1). \tag{2.4}$$

The work presented in this thesis is strictly based on analysis and processing of digital data. The matrix notation introduced above is therefore important and merits further precisions. The following list defines the notation rules used throughout this dissertation.

1. Non-bolded letters represent scalar values. Bolded or underlined lower-case and upper-case letters specify vectors and matrices, respectively.

2. $\mathcal{R}$ and $\mathcal{C}$ denote, respectively, the ensemble of real numbers and the ensemble of complex numbers. Using $\mathcal{C}$ as an example, $\mathbf{v}_r \in \mathcal{C}^{1\times N}$ specifies an N-sized row vector of complex valued elements, $\mathbf{v}_c \in \mathcal{C}^{N\times 1}$ an N-sized column vector of complex valued elements, and $\mathbf{A} \in \mathcal{C}^{M\times N}$ an $M \times N$ matrix of complex valued elements.

3. The character ":" is used to generate vectors. For example, $[1 : N]$ defines an N-sized row vector composed of $[1, 2, \ldots, N]$ while $[1 : \Delta : N]$ is $[1, 1+\Delta, \ldots]$ up to a number that is less or equal to N.

4. When employed in a matrix expression delimited with brackets, the characters ";" and "," are used to delimit rows and columns, respectively. For instance, given a 3×3 square matrix $\mathbf{A}$, the matrix $[\mathbf{A}, \mathbf{A}]$ defines a 3×6 matrix while $[\mathbf{A}; \mathbf{A}]$ is a 6×3 matrix. Note, however, that the character "," is also sometimes

used to separate matrix or vectors of different sizes when present on the left side of an equation[3].

5. A scalar function of a vectorized or matricized argument becomes a vector or matrix of the same size. A combination of one row and one column vector arguments in a scalar function creates a matrix of corresponding size.

6. The hat character on top of a variable generally specifies its estimate value.

Additional mathematical definitions (e.g. matrix functions and others) can be found in Appendix A. The reader is invited to refer to it since these will be used later on.

2.2 Modulated Signal Source

This section presents specific signal modulation types used below in this dissertation. A discrete sample-based mathematical formulation is used for both analogically and digitally modulated signal types.

Generally, a modulated signal source can be mathematically expressed as

$$s(n) = \text{LPF}\left\{m(n)e^{j\phi(n)}\right\} e^{j\left[\frac{2\pi f_c n}{f_s}+\phi_o\right]} \tag{2.5}$$

where $m(n)$ and $\phi(n)$ are the signal modulation in amplitude and phase, respectively, and ϕ_o is some arbitrary constant phase offset. Since $s(n)$ corresponds to a baseband complex signal, the term f_c denotes in reality the resulting carrier frequency after reception, equal to the difference between the RF carrier frequency and the receiver tuning frequency. This carrier frequency must fall into the $-f_s/2$ to $f_s/2$ range. The acceptable range for the carrier is however further limited by the signal bandwidth requirement. For the sake of brevity, f_s is often removed from the equation as long as f_c constitutes a normalized frequency. Finally, in practice, modulated signals must be

[3]Special matrix functions may generate multiple matrices of different sizes.

limited in bandwidth and this is reflected by the inclusion of a low-pass filter indicated by the notation LPF{·}. This filter could be of any type (FIR filter or others).

2.2.1 CW and Analog Modulations

The first signal considered is the pure complex continuous-wave (CW) signal. It constitutes indeed the non-modulated signal case. This trivial signal is defined as

$$s_{cw}(n) \triangleq e^{j\left[\frac{2\pi f_c n}{f_s}+\phi_o\right]}. \tag{2.6}$$

Two analog modulation types known as frequency modulation (FM) and amplitude modulation (AM) may also be expressed in a digital formulation. Specifically, an FM signal is defined as

$$s_{fm}(n) \triangleq e^{j\left(\frac{2\pi\Delta f_p}{M_a f_s}\sum_{n_s=-\infty}^{n} m_a(n_s)\right)} e^{j\left[\frac{2\pi f_c n}{f_s}+\phi_o\right]} \tag{2.7}$$

where $m_a(n)$ represents an analog audio signal digitized at f_s Hertz with bandwidth limited to less than f_s, Δf_p is the FM peak frequency deviation in Hertz, and M_a is the maximum absolute value of $m_a(n)$ that can be found at any n. The AM signal, on the other hand, is defined as

$$s_{am}(n) \triangleq (m_a(n) + \mu M_a)\, e^{j\left[\frac{2\pi f_c n}{f_s}+\phi_o\right]}. \tag{2.8}$$

where μ is a real number between 0 and 1 denoting the amplitude modulation index.

2.2.2 Digital Modulations

The following four signal types are strictly digital modulations. Those are: m-ary phase-shift keying (m-PSK), m-ary quadrature amplitude modulation (m-QAM), m-ary frequency shift keying (m-FSK), and finally the more specific STANAG-4285 8-PSK based standard signal type.

The expression

$$s_{m-psk}(n) \triangleq \text{LPF}\left\{\sum_{n_k=1}^{N_{symb}} \text{map}_{psk}(m_s(n_k), m)q(n - n_{st} - (n_k - 1)T_{symb})\right\} e^{j[\frac{2\pi f_c n}{f_s} + \phi_o]} \tag{2.9}$$

defines an m-PSK signal where

$$\text{map}_{psk}(m_s(n_k), m) = e^{\left(j\frac{2\pi m_s(n_k)}{m}\right)} \tag{2.10}$$

is the m-PSK mapping, n_{st} denotes the starting modulation sample instant, N_{symb} specifies the total number of symbols in the message consisting of $\log_2(m)$ bits per symbol transmitted, $T_{symb} = 1/f_{symb}$ is the symbol period, $m_s(n_k)$ is the symbol sequence message containing any integer between 0 and $(m-1)$, and, finally, $q(n)$ is the pulse shape applied to each symbol. Two pulse shapes are considered: the rectangular pulse shape and the family of raised-cosine pulse shapes[4]. Mathematically, these are respectively given by

$$q_{rect}(n_k) \triangleq \begin{cases} 1/T_{symb}, & 1 \leq n_k \leq T_{symb} \\ 0, & \text{elsewhere} \end{cases} \tag{2.11}$$

and

$$q_{r-c}(n_k) \triangleq \begin{cases} \text{sinc}(\tilde{n})\frac{\cos(r_o\pi\tilde{n})}{(1-4r_o^2\tilde{n}^2)}, & 1 \leq n_k \leq T_{pulse} \\ 0, & \text{elsewhere} \end{cases} \tag{2.12}$$

where

$$\text{sinc}(r) \triangleq \begin{cases} \frac{\sin(\pi r)}{\pi r}, & r \neq 0 \\ 1, & r = 0 \end{cases}, \tag{2.13}$$

r_o is the roll-off factor, T_{pulse} denotes the period in samples over which the pulse sequence is not zero, and $\tilde{n} = (n_k - T_{pulse}/2 - n_{shift})/T_{symb}$ with n_{shift} permitting, if not zero, unaligned pulses with respect to the first pulse sampling instant. The lowpass filtering operation may or may not be used and can be either an Infinite

[4]Raised-cosine pulse shapes are also often referred to as Nyquist pulse shapes.

Impulse Response (IIR) or Finite Impulse Response (FIR) filter, as desired. Note that the raised-cosine pulse shape inherently incorporates lowpass filtering and no additional filtering would normally be used in this case.

The m-QAM signal type closely follows the m-PSK since

$$s_{m-qam}(n) = \text{LPF}\left\{\sum_{n_k=1}^{N_{symb}} \text{map}_{qam}(m_s(n_k)+1)q(n-n_{st}-(n_k-1)T_{symb})\right\} e^{j\left[\frac{2\pi f_c n}{f_s}+\phi_o\right]}. \tag{2.14}$$

The QAM constellation mapping, map_{qam}, is a different mapping than the one used for PSK modulation. The m-PSK scheme maps a set of m finite number of possible symbol levels to the complex unit circle with each symbol having its own distinct phase state equispaced by $2\pi/m$. In the case of the QAM, a rectangular mapping as shown in Fig. 2.1 for up to $m = 64$ is used. This mapping can easily be put into a complex vector such as

$$\mathbf{map}_{qam_{64}} = [\text{map}_{qam}(1), \ldots, \text{map}_{qam}(64)] \tag{2.15}$$

This vector could be extended to accommodate higher values of m than 64.

The third digital modulation type presented is the m-FSK. The mathematical expression for the m-FSK can be given as

$$s_{m-fsk}(n) \triangleq \text{LPF}\left\{e^{j\frac{2\pi\Delta f_p}{f_s}\sum_{n_s=0}^{(n-n_{st})}\left(\frac{m_s(\lfloor n_s/T_{symb}\rfloor+1)}{(m-1)}-0.5\right)}\right\} e^{j\left[\frac{2\pi f_c n}{f_s}+\phi_o\right]}. \tag{2.16}$$

The term Δf_p follows the same conditions as for FM.

The last modulation type discussed is a specific and practical signal type used for HF modems. This signal belongs to the general PSK family signal type discussed above and implements a particular NATO standard [62] referred herein to as STANAG4285 or s_{st}.

The STANAG4285 signal takes the same expression as for $s_{m-psk}(n)$ but for the specific case where $m = 8$ and the pulse shape is raised-cosine with $r_o = 0.25$. In

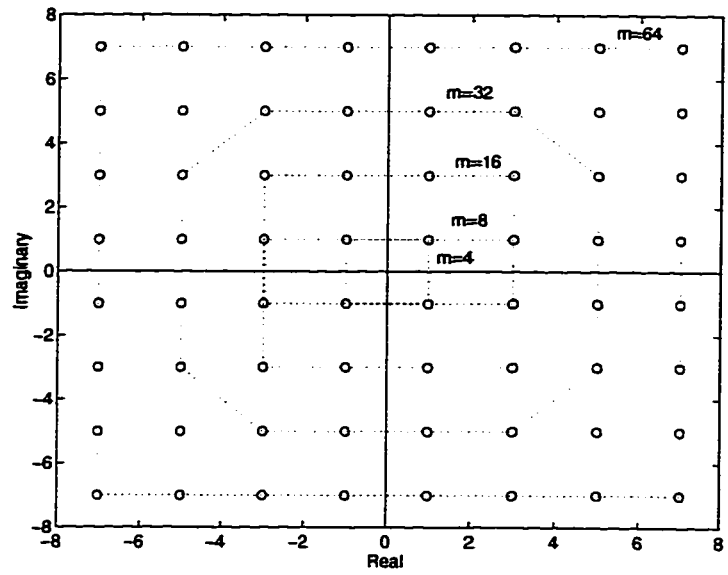

Figure 2.1: Constellation mapping of m-QAM.

addition to these, the specificity of STANAG4285 primarily lies in its message formating and frame structure used to form $m_s(n)$. Half of the symbols sent are overhead since for each frame structure composed of 256 symbols, 80 are fixed synchronization symbols, followed by 4 blocks of 32 data symbols interleaved by blocks of 16 reference symbols (all zeros), totalling 176 symbols. A scrambling sequence of length 176 is used to scramble the last 176 symbols of each frame, i.e. the portion containing the data and reference symbols. In more details, the synchronization sequence is[5]

$$c_{sync}(k) = \mathrm{mod}_8\{c_{sync}(k-3) + c_{sync}(k-5)\}, k = [6:80] \tag{2.17}$$

for the initial values

$$c_{sync}(1:5) = [0,4,4,0,4]; \tag{2.18}$$

the scrambling sequence is indirectly given by

$$c_{sc}(k) = \mathrm{mod}_2\{c_{sc}(k-5) + c_{sc}(k-9)\}, k = [10:3\cdot 176] \tag{2.19}$$

[5]Note that $\mathrm{mod}_N\{\cdot\}$ represents the modulo N operation, i.e. the argument of the function is wrapped into the 0 to N-1 integer range value.

with

$$c_{sc}(1:9) = [1,1,1,1,1,1,1,1,1] \tag{2.20}$$

since

$$c_{scramble}(n) = c_{sc}(3n-2) + 2c_{sc}(3n-1) + 4c_{sc}(3n)\,, n = [1:176] \tag{2.21}$$

defines the scrambling sequence of 176 symbols; the data sequence is given by the incoming block of either 1, 2, or 3 bits that are combined to respectively form a number between 0 and 1 equal to b_0, or between 0 and 3 equal to $(b_0 + 2b_1)$, or between 0 and 7 equal to $(b_0 + 2b_1 + 4b_2)$, subject to the following transcoding vectors

$$\begin{aligned} \mathbf{t}_{(2)} = t_{(2)}(0:1) &= [0,4] \\ \mathbf{t}_{(4)} = t_{(4)}(0:3) &= [0,6,2,4] \\ \mathbf{t}_{(8)} = t_{(8)}(0:7) &= [1,6,2,5,0,7,3,4] \end{aligned} \tag{2.22}$$

depending on how many bits form a symbol. Defining

$$\mathbf{c}_{sync} \triangleq c_{sync}(1:80)\,, \tag{2.23}$$

$$\mathbf{c}_{scramble} \triangleq c_{scramble}(1:176)\,, \tag{2.24}$$

$$\mathbf{c}_z \triangleq \mathbf{0}_{1,16}\,, \tag{2.25}$$

and letting $m_d(n)$ denote the nth data symbol after transcoding has been applied. the first full frame symbol stream is given by

$$\begin{aligned} \mathbf{m}_{st}(1) = \; & [\mathbf{c}_{sync}, \mathrm{mod}_8\,\{[m_d(1:32), \\ & \mathbf{c}_z, m_d(33:64), \mathbf{c}_z, \\ & m_d(65:96), \mathbf{c}_z, \\ & m_d(97:128)] + \mathbf{c}_{scramble}\}]\,. \end{aligned} \tag{2.26}$$

The overall symbol rate obligatory set to 2400 symbols per second yields into a 1200 symbols/sec for the data message portion only which results into 1200 bits/sec, 2400

bits/sec, or 3600 bits/sec for respectively 1, 2, or 3 bit symbols. Fig. 2.2 presents a single typical message frame. It can be observed that the synchronization sequence produces a 2-PSK type of signals which is then followed by an 8-PSK signal type for the remaining 176 symbols, regardless of the data coding scheme. Finally, note that the error control coding scheme was not included.

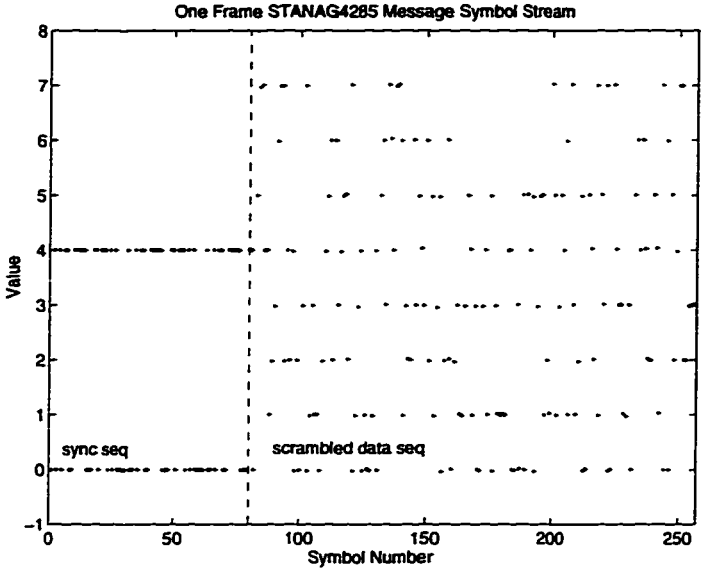

Figure 2.2: Typical STANAG4285 symbol frame message.

This completes the description of the signal types to consider later in this dissertation. These signal definitions will be particularly useful to describe the signal scenarios used in the computer simulations of Chapter 4.

2.3 Sensor Array Signal Modeling

2.3.1 Background

For this dissertation, the receiver platform is assumed to include an antenna array composed of M sensors disposed in any two-dimensional arbitrary geometry. The

two-axis cartesian coordinate system, denoted (p_1, p_2), is used to specify the position of each sensor. The position can be expressed either in wavelengths or in meters. When specified in meters, the sensor array position matrix is defined by

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 \end{bmatrix} = \begin{bmatrix} p_1(1) & p_2(1) \\ \vdots & \vdots \\ p_1(M) & p_2(M) \end{bmatrix} \tag{2.27}$$

where $\mathbf{P} \in \mathcal{R}^{M \times 2}$. To transform the sensor array position matrix into wavelengths, the transformation

$$\mathbf{P}(f_o) = \begin{bmatrix} \mathbf{p}_1(f_o) & \mathbf{p}_2(f_o) \end{bmatrix} = \frac{f_o}{c}\mathbf{P} \tag{2.28}$$

is used, where f_o is the RF center frequency, and c is the speed of the signal wave travelling in the propagation medium. Most often, Eq. (2.28) is conveniently reduced to

$$\mathbf{P}(f_o) = \frac{f_o}{300}\mathbf{P} \tag{2.29}$$

with f_o expressed in MHz.

When an ideal narrowband signal received at some radio frequency f_o is impinging at the antenna array with a direction of arrival (DOA) specified by $\underline{\theta} = [\theta_{az}, \theta_{el}]$, the sensor array position translates into an array response vector[6] [54] . Note that the DOA includes both azimuth (θ_{az}) and elevation (θ_{el}) bearings although, in several situations, elevation is often assumed to be zero. However for the sake of completeness, elevation is included in the expression for the array response vector provided by

$$\mathbf{a}(\underline{\theta}, f) = e^{j2\pi[\mathbf{p}_1(f)sin(\theta_{az}) + \mathbf{p}_2(f)cos(\theta_{az})]cos(\theta_{el})} \tag{2.30}$$

where $\mathbf{a}(\underline{\theta}, f) \in \mathcal{C}^{M \times 1}$. In the above expression for the array response vector, one assumption being made is that the transmission source is distant enough from the

[6]The terms steering vector, aperture vector, array vector, array manifold vector, and direction vector are also found in the literature to express this same entity.

receiver platform to allow the use of a single DOA with respect to each sensor[7]. Another assumption is that the sensors used are ideal identical omni-directional isotropic antennas. In practice, antenna mismatches, imperfect antenna responses, and other practical effects will cause discrepancies with the ideal array response vector expression given above. Nevertheless, this will not jeopardize the validity of the treatment made in this dissertation.

2.3.2 Multipath-free Basic Sensor Array Signal Model

Considering now L narrowband signals impinging at the M-sensor array, the $M \times 1$ received snapshot complex vector, in the absence of multipath, is well known (e.g. [55]) to be given by

$$\mathbf{x}(n) = \sum_{l=1}^{L} g_l \mathbf{a}(\underline{\theta}_l, f_o) s_l(n) + \mathbf{v}(n) \tag{2.31}$$

where g_l is the amplitude of the l^{th} signal, $\mathbf{a}(\underline{\theta}_l, f_o)$ is the array response vector to the l^{th} signal coming from direction $\underline{\theta}_l$ at RF f_o, $s_l(\cdot)$ is the l^{th} complex baseband signal sampled waveform[8], and $\mathbf{v}(\cdot)$ is composed of zero mean independently identically distributed (iid) white Gaussian noise. Since, in reality, multipath is always present to some degree, this received sensor array signal model needs to be extended to include the effect of multipath.

2.3.3 More Comprehensive Models Considering Multipath

Multipath, in essence, results from the superposition of a single transmitted signal being received via multiple paths, or equivalently, multiple directions with various

[7] This is known as the far-field assumption.

[8] All signals are assumed to have been normalized to a squared-root mean power of unity (i.e., $\sqrt{\frac{s_l(1:N)s_l(1:N)^H}{N}} = 1$).

gains. Mathematically, for the L-signal environment stated above, this translates to

$$\mathbf{x}(n) = \sum_{l=1}^{L} g_l \sum_{p=1}^{N_p(l)} \alpha_{l,p}\mathbf{a}(\underline{\theta}_{l,p}, f_o)s_l(n - \tau_{l,p}) + \mathbf{v}(n) \tag{2.32}$$

where $N_p(l)$ is the number of subpaths for the l^{th} signal, and $\alpha_{l,p}$ and $\tau_{l,p}$ are, respectively, the attenuation and time delay corresponding to the p^{th} subpath. If the propagation delays associated with these paths are assumed to be much smaller than the inverse bandwidth of the signals, the simplifications

$$\mathbf{x}(n) = \sum_{l=1}^{L} \mathbf{a}_l s_l(n) + \mathbf{v}(n) \tag{2.33}$$

where

$$\mathbf{a}_l = g_l \sum_{p=1}^{N_p(l)} \alpha_{l,p} e^{-j2\pi f_o \tau_{l,p}} \mathbf{a}(\underline{\theta}_{l,p}, f_o) \tag{2.34}$$

can be made, since delays can be modeled as phase-shifts under this assumption, referred to as the narrowband assumption[9] .

Further extending the sensor array signal model given in Eq. (2.33) to supersede the narrowband signal assumption, the received M-sensor outputs finally become

$$\mathbf{x}^f(k,n)_N = \sum_{l=1}^{L} \mathbf{a}_l(k) s_l^f(k,n)_N + \mathbf{v}^f(k,n)_N \tag{2.35}$$

where k denotes the discrete frequency index. The notation $\mathbf{x}^f(k,n)_N$, $s_l^f(k,n)_N$, and $\mathbf{v}^f(k,n)_N$ represent, respectively, the N-point Discrete Fourier Transform (DFT) of $\mathbf{x}(n)$, $s_l(n)$, and $\mathbf{v}(n)$. With

$$\mathbf{X}^f(n) \triangleq \text{fft}\left\{\mathbf{x}(n - N/2 : n + N/2 - 1)^T\right\}^T \ , \ \mathbf{X}^f(n) \in \mathcal{C}^{M \times N} \ , \tag{2.36}$$

$$\mathbf{s}_l^f(n) \triangleq \text{fft}\left\{\mathbf{s}(n - N/2 : n + N/2 - 1)\right\} \ , \ \mathbf{s}_l^f(n) \in \mathcal{C}^{1 \times N} \ , \tag{2.37}$$

[9]Normally, the narrowband signal assumption refers to the unvariability of the array response vector over the entire received signal bandwidth for some given DOA. This is satisfied when the bandwidth is much smaller than the RF. The narrowband assumption used in this thesis adds the additional constraints on the propagation delays.

$$\mathbf{V}^f(n) \triangleq \text{fft}\left\{\mathbf{v}(n-N/2:n+N/2-1)^T\right\}^T\,,\ \mathbf{V}^f(n) \in \mathcal{C}^{M\times N}\,, \tag{2.38}$$

and

$$\mathbf{A}_l^f \triangleq \mathbf{a}_l(0:N-1)\,,\ \mathbf{A}_l^f \in \mathcal{C}^{M\times N}\,, \tag{2.39}$$

Eq. (2.35) can be reformulated as

$$\mathbf{X}^f(n) = \sum_{l=1}^{L} \mathbf{A}_l^f \otimes \left(\mathbf{1}_{M,1}\mathbf{s}_l^f(n)\right) + \mathbf{V}^f(n) \tag{2.40}$$

with the frequency-dependent array response vector for signal l, $\mathbf{a}_l(k)$, given by

$$\mathbf{a}_l(k) \triangleq g_l \sum_{p=1}^{N_p(l,k)} \alpha_{l,p}(k) e^{-j2\pi(f_o+f_k)\tau_{l,p}(k)} \mathbf{a}(\underline{\theta}_{l,p}, f_o+f_k)\,. \tag{2.41}$$

Comparing the wideband with the narrowband sensor array signal model easily reveals that the order of the problem is at least increased by a factor N.

The particular choice of model and its parameters used in the computer simulations will be more detailed in Chapter 4.

2.4 Filtering Structures

The objective of extracting and equalizing a signal in a corrupted multi-signal environment requires a filter to be applied to the composite received signal. The more sophisticated the filter is, the more costly it will be to apply. However, an improvement in interference signal rejection would normally be expected. This is therefore often a trade-off and one has to evaluate whether the extra complexity in the filters is worth the level of improvement in performance. It all depends on the specific needs and applications. Two types of filters have commonly been applied and those are known as temporal[10] and spatial filters. In this dissertation, non-linear filters are not considered.

[10]Note that spectral filters can be made equivalent to temporal filters.

2.4.1 Temporal Filtering

The temporal filter only requires a single sensor receiving system and is, as a result, much cheaper to implement. The most common temporal filter used is the linear time invariant (LTI) finite impulse response (FIR) filter [44]. The FIR filtered signal estimate is mathematically expressed as

$$\hat{s}_l(n) = \mathbf{w}_l(n)^H \begin{bmatrix} x(n-\tau_1) \\ x(n-\tau_2) \\ \vdots \\ x(n-\tau_{N_\tau}) \end{bmatrix} \tag{2.42}$$

where N_τ is the number of filter taps, and $\mathbf{w}_l \in \mathcal{C}^{N_\tau \times 1}$ is the FIR temporal vector filter taps with each element corresponding to a specific lag value, τ_i, for the l^{th} signal extracted. Such a filter, shown in Fig. 2.3, could be well suited for mitigating multipath as long as the signal bandwidth is much greater than the inverse of multipath spread in time[11]. However, for spectrally and temporally overlapping signals, this is not a very efficient filter. In other words, where single-signal equalization is required, the FIR filtering structure could be used.

A slightly more complex temporal filter known as FRESH filter consists of frequency shift operations prior to FIR filtering (note that other types of filtering may be employed as well). The principle of operation of the FRESH filter is to combine frequency shifted portions of the spectrum in order to enhance a desired signal and suppress other signals. FRESH filters exploit spectral redundancy, or equivalently second-order cyclostationarity, a very common property in most digitally modulated signals (see [14, 16, 17, 19, 20] for more details). As reported for instance in [22], the FRESH filtering technique can be shown to separate temporally and spectrally

[11] When the bandwidth is smaller than the inverse of the multipath spread in time, no real need to mitigate multipath prevails (for instance, it would not cause intersymbol interference to an unacceptable level for a digitally modulated signal).

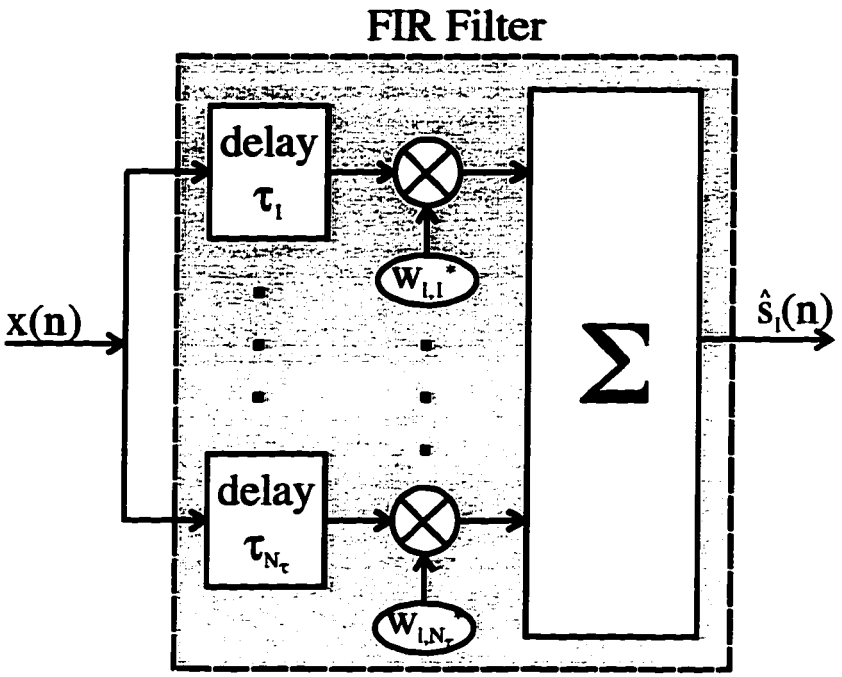

Figure 2.3: Temporal FIR filtering structure.

overlapping signals, including AM, 2-PSK (BPSK), and 4-PSK (QPSK) modulated signals. In this thesis, the FRESH filtered signal estimate is given by

$$\hat{s}_l(n) = \mathbf{w}_l(n)^H \begin{bmatrix} x(n-\tau_1)e^{j2\pi\beta(1)n} \\ x(n-\tau_2)e^{j2\pi\beta(1)n} \\ \vdots \\ x(n-\tau_{N_\tau})e^{j2\pi\beta(1)n} \\ x(n-\tau_1)e^{j2\pi\beta(2)n} \\ \vdots \\ x(n-\tau_{N_\tau})e^{j2\pi\beta(N_\beta)n} \\ x(n-\tau_1)^* e^{j2\pi\bar{\beta}(1)n} \\ \vdots \\ x(n-\tau_{N_\tau})^* e^{j2\pi\bar{\beta}(N_{\bar{\beta}})n} \end{bmatrix}, \tag{2.43}$$

where $\mathbf{w}_l \in \mathcal{C}^{N_\tau \cdot (N_\beta + N_{\bar{\beta}}) \times 1}$ is the FRESH filter weight vector. In the FRESH filter

equation, β and $\overline{\beta}$ are used to denote the scalar value of frequency shift[12] applied to the received sensor data and correspond in fact to the cycle and conjugate cycle frequencies respectively. Fig. 2.4 shows the block diagram of this FRESH filter.

2.4.2 Spatial Filtering

The spatial filter presented next is the well-known linear beamformer [3]. Although simple, this type of filter requires a multi-sensor receiver system which cannot always be afforded. The spatially filtered signal estimate is expressed as

$$\hat{s}_l(n) = \mathbf{w}_l(n)^H \mathbf{x}(n)\,. \tag{2.44}$$

where $\mathbf{x}(n) \in \mathcal{C}^{M\times 1}$ is the received sensor snapshot, and $\mathbf{w}_l \in \mathcal{C}^{M\times 1}$ is the spatial filter weight vector. This type of filter can typically be set up to spatially null out undesired signals while beamforming the desired one. The filter remains able to achieve signal separation this way as long as the signal array response vectors are reasonably uncorrelated. This filter structure is shown in Fig. 2.5.

[12]These frequencies are assumed to have been normalized with respect to the sampling frequency.

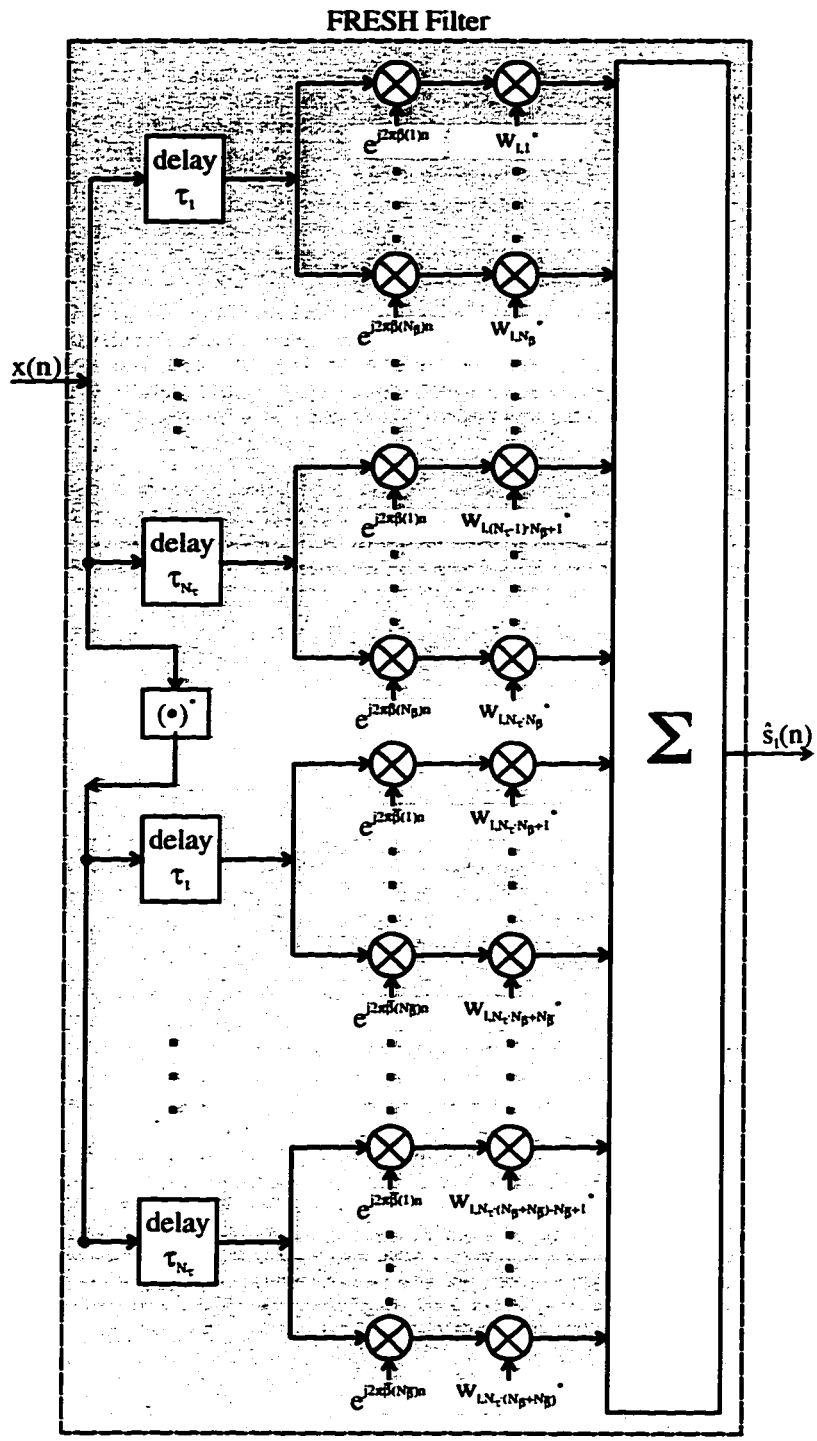

Figure 2.4: Temporal FRESH filtering structure.

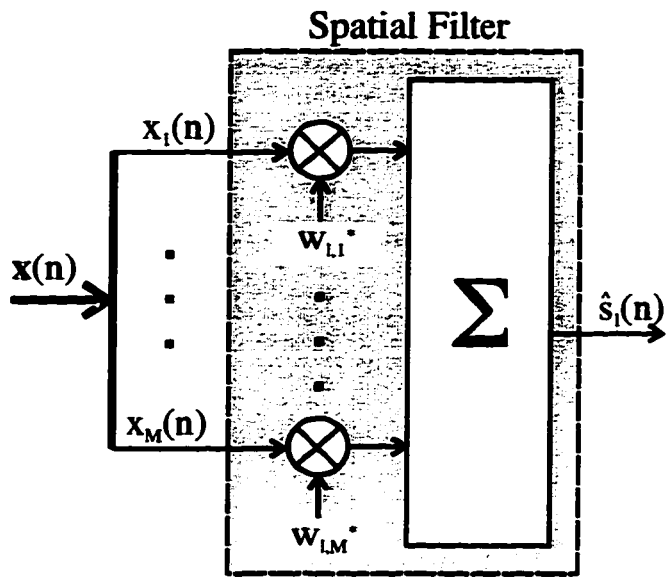

Figure 2.5: Spatial linear filtering structure.

2.4.3 Spatio-Temporal Filtering

A combination of the spatial and temporal filters, known as spatio-temporal filters [54], would be expected to perform even better. Defining

$$\mathbf{x}_a(n) = \begin{bmatrix} \mathbf{x}(n-\tau_1)e^{j2\pi\beta(1)n} \\ \mathbf{x}(n-\tau_2)e^{j2\pi\beta(1)n} \\ \vdots \\ \mathbf{x}(n-\tau_{N_\tau})e^{j2\pi\beta(1)n} \\ \mathbf{x}(n-\tau_1)e^{j2\pi\beta(2)n} \\ \vdots \\ \mathbf{x}(n-\tau_{N_\tau})e^{j2\pi\beta(N_\beta)n} \\ \mathbf{x}(n-\tau_1)^* e^{j2\pi\bar{\beta}(1)n} \\ \vdots \\ \mathbf{x}(n-\tau_{N_\tau})^* e^{j2\pi\bar{\beta}(N_{\bar{\beta}})n} \end{bmatrix}, \quad \mathbf{x}_a(n) \in \mathcal{C}^{M\cdot N_\tau\cdot(N_\beta+N_{\bar{\beta}})\times 1}, \tag{2.45}$$

as the augmented received sensor data vector, a spatio-temporal filter structure can then be used to form the spatio-temporal filtered estimates

$$\hat{s}_l(n) = \mathbf{w}_l(n)^H \mathbf{x}_a(n)\,, \tag{2.46}$$

where $\mathbf{w}_l(n) \in \mathcal{C}^{M \cdot N_\tau \cdot (N_\beta + N_{\overline{\beta}}) \times 1}$. Every previous filter is, in fact, a special case of this new, and more general, filter structure shown in Fig. 2.6. The flexibility necessary

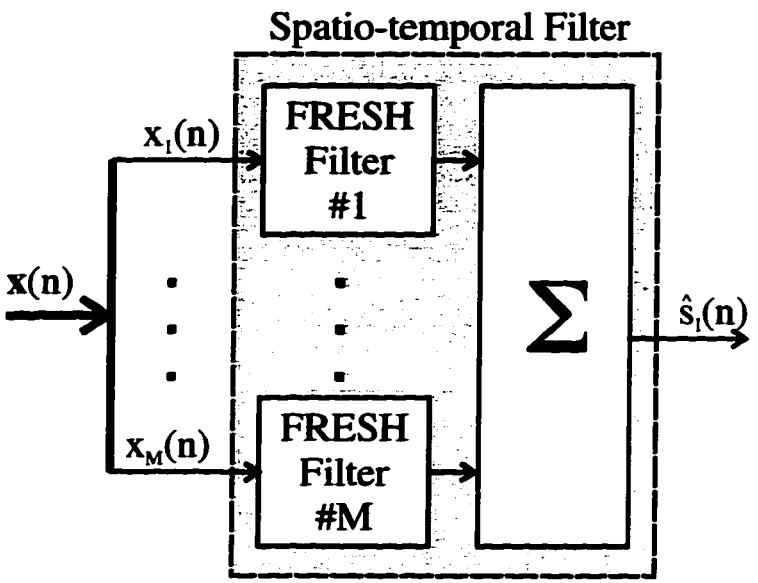

Figure 2.6: Spatio-temporal filtering structure.

to evaluate the various trade-offs mentioned previously is provided with this filter structure. A possible configuration of interest would be to let $N_\beta = 1$, $\beta = 0$, and $N_{\overline{\beta}} = 0$ which corresponds to a spatio-temporal filter combining temporal FIR and linear spatial filters. Eq. (2.46) can be extended to extract multiple signals simultaneously. This is possible with

$$\hat{\mathbf{s}}(n) = \mathbf{W}(n)^H \mathbf{x}_a(n)\,,\ \hat{\mathbf{s}}(n) \in \mathcal{C}^{L_d \times 1}\,, \tag{2.47}$$

where

$$\mathbf{W}(n) = \mathbf{w}_{1:L_d}(n) \tag{2.48}$$

is the spatio-temporal filter weights array, and $L_d \leq L$ is the number of simultaneously extracted SOI. Although Eq. (2.46) and Eq. (2.47) constitute the filtering structure initially considered for the work in Chapter 3, the special case consisting of a spatial filter structure only was finally retained, primarily for its simplicity, its applicability, and the existence of blind adaptive solutions.

2.5 Performance Measures

The problem of interest in this dissertation, pictured in Fig. 2.7, can be re-expressed as follows: given L simultaneously transmitted source signals combined, distorted, and received through an RF sensor array channel, what are the optimum blind adaptive processing techniques that could be applied to the corrupted received signal sensor vector in order to obtain an estimate as close as possible to the originally transmitted signal source for each $L_d \leq L$ SOIs? The section establishes the means that can and will be used to measure the level of performance of any particular technique. Those performance measures generally require prefectly known signal sources[13]. It should be pointed out that two performance measures presented next, namely the NSINR and NMSER, are new performance measures defined by the author.

2.5.1 The MSE, NMSE and SINR Measures

Given the source signal $s_l(n)$ and its estimate, denoted $\hat{s}_l(n)$, the mean-squared error (MSE) is a primary measure of the quality of the estimate defined as [23]

$$\mathrm{MSE}(\hat{s}_l, s_l) \triangleq \left\langle \|\hat{s}_l(n) - s_l(n)\|^2 \right\rangle_N . \tag{2.49}$$

Another common measure is the signal to interference and noise ratio (SINR). This measure is defined as the ratio of the power of the desired signal in $\hat{s}_l(n)$ to the power

[13]In practical situations, other types of measures would be required.

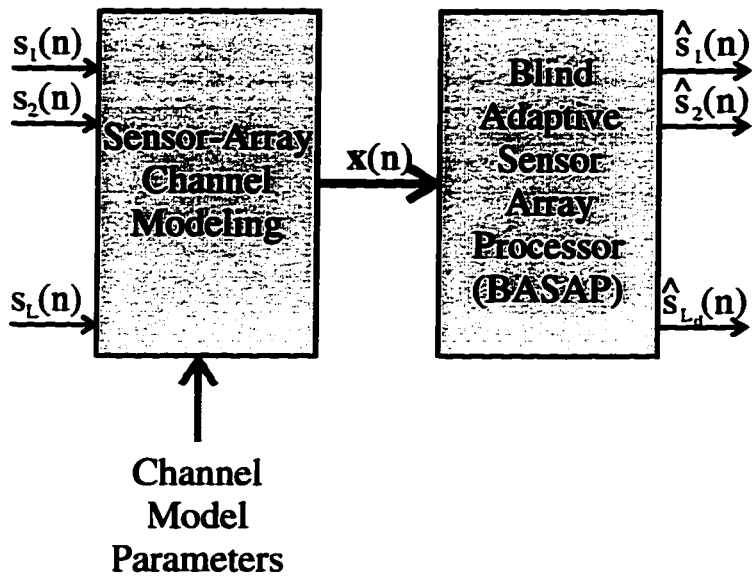

Figure 2.7: Problem statement block diagram.

of everything else in $\hat{s}_l(n)$. In mathematical terms, we have [54]

$$\text{SINR}(\hat{s}_l, s_l) \triangleq \frac{\left|\hat{R}_{\hat{s}_l s_l}\right|^2 / \hat{R}_{s_l s_l}}{\hat{R}_{\hat{s}_l \hat{s}_l} - \left|\hat{R}_{\hat{s}_l s_l}\right|^2 / \hat{R}_{s_l s_l}} = \frac{|\hat{\rho}_{\hat{s}_l s_l}|^2}{1 - |\hat{\rho}_{\hat{s}_l s_l}|^2} \tag{2.50}$$

where

$$\hat{\mathbf{R}}_{\mathbf{v_1 v_2}} = \langle v_1(n) v_2(n)^* \rangle_N \tag{2.51}$$

and

$$|\hat{\rho}_{\hat{s}_l, s_l}|^2 \triangleq \frac{\left|\hat{R}_{\hat{s}_l s_l}\right|^2}{\hat{R}_{s_l s_l} \hat{R}_{\hat{s}_l \hat{s}_l}}. \tag{2.52}$$

One may note that the correlation coefficient $|\hat{\rho}_{\hat{s}_l s_l}|$ (which could be considered as a performance measure by itself) goes to 1 in the limit when the estimate becomes highly correlated with the original signal. This causes the SINR to increase with no bound. In other words, the SINR equals infinity when the estimate equals exactly the signal source. Note that this would not happen in practice due to the presence of noise.

Using the correlation coefficient, a more convenient measure than the MSE known as the normalized mean-squared error (NMSE) [28] measure is defined as

$$\text{NMSE}(\hat{s}_l, s_l) \triangleq 1 - |\hat{\rho}_{\hat{s}_l s_l}|^2 \ . \tag{2.53}$$

It is obvious from this definition that the NMSE is bounded between 0 and 1, with 0 meaning no error, and 1, maximal error.

The MSE, NMSE, and SINR measures can be each averaged over multiple data segments for longer time-series.

2.5.2 The NSINR and NMSER Measures

Two new performance measures are next defined by the author. Those closely related measures are named the normalized mean-squared error ratio (NMSER) and the normalized SINR (NSINR). The commonality between the two lies in the use of the optimum non blind signal estimates denoted $s_{l,mmse}$ and presented in the next section. As will be seen, this signal estimate yields the maximum attainable SINR solution for a particular filter structure. The NMSER and NSINR measures definitions follow.

Firstly, the NMSER is defined as

$$\begin{aligned} \text{NMSER}(l) &\triangleq 10 \cdot \log_{10} \left\{ \frac{1-\left|\hat{\rho}_{\hat{s}_{l,mmse},s_l}\right|^2}{1-\left|\hat{\rho}_{\hat{s}_l,s_l}\right|^2} \right\} \\ &= 10 \cdot \log_{10} \left\{ \frac{1-c_{opt}(l)}{1-c_{est}(l)} \right\} \end{aligned} \tag{2.54}$$

where

$$c_{opt}(l) = \left|\hat{\rho}_{\hat{s}_{l,mmse},s_l}\right|^2 \tag{2.55}$$

and

$$c_{est}(l) = |\hat{\rho}_{\hat{s}_l,s_l}|^2 \ . \tag{2.56}$$

Secondly, the NSINR is defined as the ratio of the SINR obtained with some signal estimate and the SINR obtained with the optimal estimate, with respect to a common

signal source. Mathematically, we have

$$\begin{aligned} \text{NSINR}(l) &\triangleq 10 \cdot \log_{10}\left\{\frac{\text{SINR}_{est}(l)}{\text{SINR}_{opt}(l)}\right\} \\ &= 10 \cdot \log_{10}\left\{\frac{c_{est}(l)/(1-c_{est}(l))}{c_{opt}(l)/(1-c_{opt}(l))}\right\} \\ &= 10 \cdot \log_{10}\left\{\frac{1-c_{opt}(l)}{1-c_{est}(l)} \cdot \frac{c_{est}(l)}{c_{opt}(l)}\right\}. \end{aligned} \tag{2.57}$$

Both the NMSER and NSINR measures are in dB and are always less or equal to 0, with equality holding if and only if the estimate corresponds exactly to the optimal solution. The NMSER and NSINR measures have the simple relationship

$$\text{NSINR}(l) = \text{NMSER}(l) + 10 \cdot \log_{10}\left\{\frac{c_{est}(l)}{c_{opt}(l)}\right\} \tag{2.58}$$

pictured in Fig. 2.8. In this plot, $c_{est}(l)$ is varied over several values constrained to be smaller or equal to $c_{opt}(l)$, which is kept fixed to five values selected as 0.2, 0.5, 0.9, 0.99, and 0.999 (as shown on the graph). By investigating those five curves and the simple mathematical relationship, we conclude that:

- there is a minimum NMSER that strictly depends on the optimum solution and corresponds to the maximum SINR attainable (with the sign inversed);
- when the optimum solution only provides a poor estimate (with low values of $c_{opt}(l)$), the NSINR will continue to be affected while the NMSER will remain fairly constant to a value approaching zero; and finally
- the two measures will be almost equal for an optimum solution providing a reasonably good estimate since as $c_{opt}(l) \to 1$, $\text{NSINR}(l) \to \text{NMSER}(l)$.

Since the two measures are found to be almost equal, only one of the two measures is really useful. The NSINR will be the favored measure because of its closest relationship with the SINR.

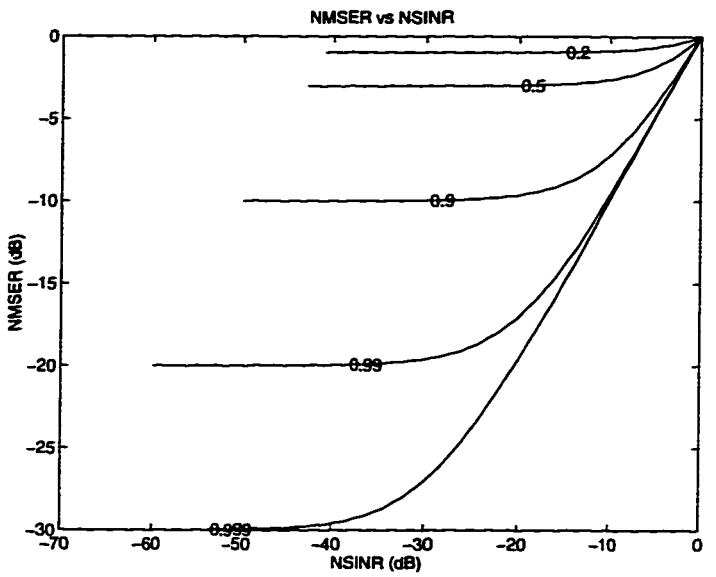

Figure 2.8: Relationship between the NMSER and NSINR measures.

2.5.3 The BER and SER Measures

In all cases dealing with digital communication signals, the bit and symbol error rate (BER and SER respectively) are other measures extremely useful and popular to characterize the performance of the bit and symbol estimation process. These measures require demodulation and bit/symbol estimation to be achieved. Despite the fact that they constitute the important end result in the case of digital signals, these measures depend on several factors related to the demodulation scheme and bit/symbol estimation. Also, the BER/SER measures cannot be used in the case of non-digitally modulated signals. Consequently, the BER/SER measures will not be selected to evaluate the techniques in this dissertation.

2.5.4 Other Performance Factors

Over and above these signal quality measures, other important factors (cf [28]) must also be considered when evaluating various blind adaptive filtering techniques. They

are enumerated next.

1. *Rate of convergence.* Defined as the number of iterations required for the algorithm to converge "close enough" to a MMSE solution.

2. *Tracking.* Defined as the ability to track the variations in the environment.

3. *Robustness.* Refers to the ability of the algorithm to remain successful in the most difficult environment.

4. *Computational requirements.* Refers to the algorithm requirements in terms of number of operations, size of memory, and programming complexity.

5. *Structure.* Refers to the hardware implementability of the algorithm structure.

6. *Numerical properties.* Refers to numerical stability and numerical accuracy.

7. *Validity.* Refers to the conditions and assumptions to satisfy for the algorithm to be successful.

2.6 Review of Signal Extraction Techniques

This section first reviews conventional sensor array processing techniques suitable to adaptively extract signals. The methods are not blind in the sense previously defined as they require either training signals and/or knowledge of the direction of arrival and sensor array manifold. An overview of selected blind methods follows this review of conventional techniques. These methods are often based on restoring some signal properties instead of optimizing some signal quality measures as for the conventional methods. For this reason, blind methods are often referred to as property restoral techniques. It is to be noted that all techniques presented next assume

the narrowband signal model to be valid and are set up to adapt a spatial linear beamformer, unless stated otherwise.

As a visual example, each technique presented will be applied as required to a specific arbitrary signal scenario consisting of :

1. three modulated signal sources:
   - *source 1* - a **4-PSK (QPSK) signal** with $f_c = 0$ Hz, random bit stream message (uniformly distributed over the possible values), symbol rate equal to 1320 symbols per second, rectangular pulse shape, low-pass filter (digital Butterworth filter of fifth order) having the 3 dB cut-off frequency at 2420 Hz, relative power of 0 dB, and $\theta_{az} = -32$ degrees;
   - *source 2* - an **FM signal** with $f_c = 321$ Hz, $\Delta f = 4530$ Hz, analog random noise modulating source (limited to the 300 to 3000 Hz range), relative power of 2 dB, and $\theta_{az} = 2$ degrees;
   - *source 3* - a **STANAG4285 signal** with $f_c = -220$ Hz, ASCII text file message coded into 1 bit symbols, symbol rate equal to 2400 symbols per second, Nyquist pulse shape with a roll-off factor of 0.25 and pulse length of 16 samples, relative power of 4 dB, and $\theta_{az} = 28$ degrees;
2. an SNR of 15 dB relative to the 0 dB signal power reference;
3. $\theta_{el} = 0$ degrees for all signals;
4. 16384 sample snapshots generated at a sampling frequency of 10 kHz;
5. as shown in Fig. 2.9, a sensor array with 5 sensors disposed in an L-shape and spaced by half a wavelength; and
6. no multipath present.

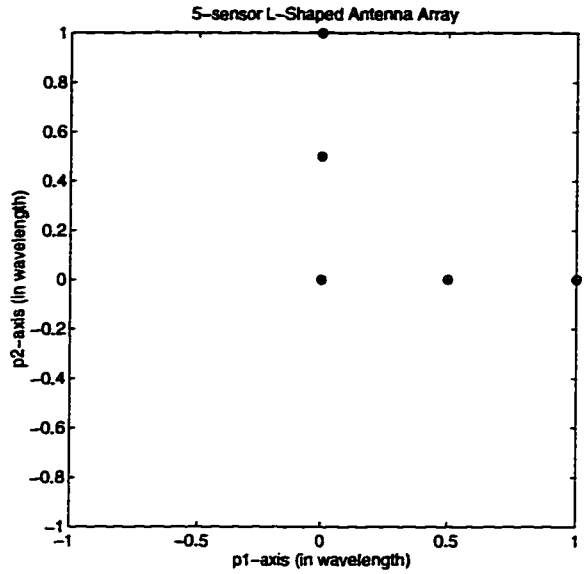

Figure 2.9: Layout of the 5-sensor L-shaped antenna array used in the example signal scenario.

Note that the choice of parameters such as data rate and 3 dB cut-off frequency is set arbitrarily. However, special care not to have the sampling rate an integer multiple of the data rate was ensured.

The Welch periodogram [44] for each of the three signal sources is shown in Fig. 2.10. The periodogram was computed by averaging FFT windows of 1024 samples without overlap over all the 16384 samples. The periodogram of the combined signals at the sensors is shown in Fig 2.11 for the three first sensors, all calculation parameters being equal. Another useful visual tool that can be used is the I-Q constellation plot which consists of plotting the real portion against the imaginary portion of the data. The I-Q constellation plots of the three signal sources and the three first sensors data are respectively shown in Fig. 2.12 and Fig. 2.13. To give a quantitative measure of the signal quality received at the sensors, the mean (over all sensors) SINR for each signal has been computed. The result is as follows: *signal 1* has a mean SINR of -6.21 dB, *signal 2*, -3.42 dB, and *signal 3*, -0.37 dB. The techniques presented next

are aimed at improving these figures as much as possible.

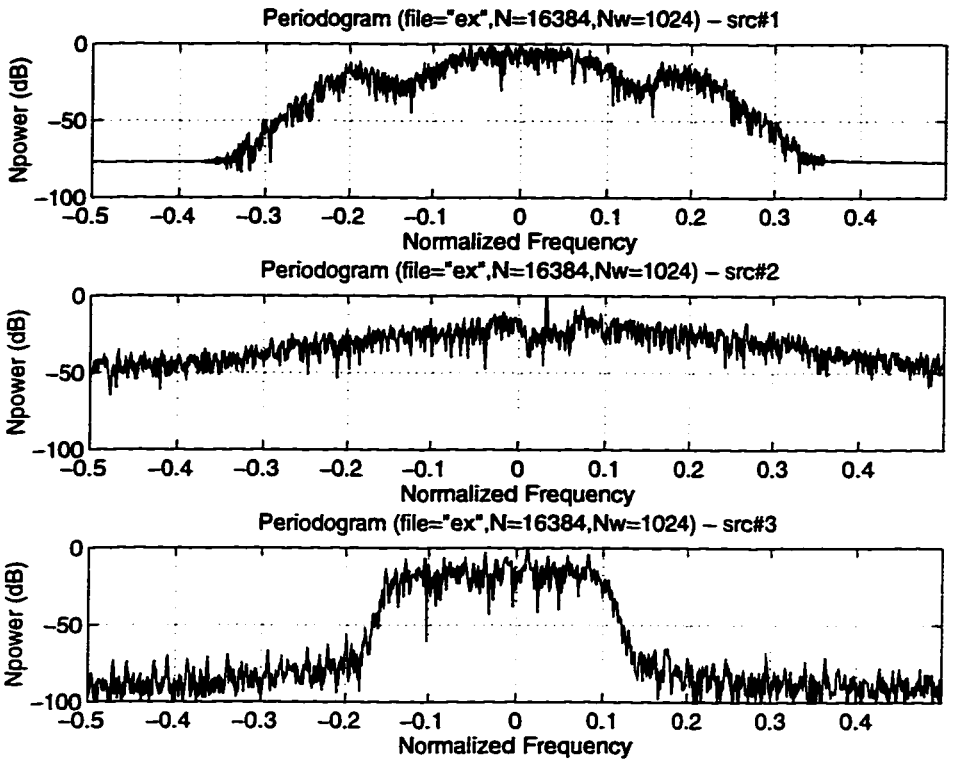

Figure 2.10: Welch periodogram of the three signal sources (4-PSK, FM, and STANAG4285) in the example signal scenario. A total of 16384 samples are considered, the window size is 1024, and no overlap is used.

2.6.1 Conventional Techniques

Two conventional techniques are reviewed. Those are the minimum mean-squared error (MMSE) solution, and the minimum variance distortionless response (MVDR) solution[14].

[14]The MVDR is also known as the Capon's method.

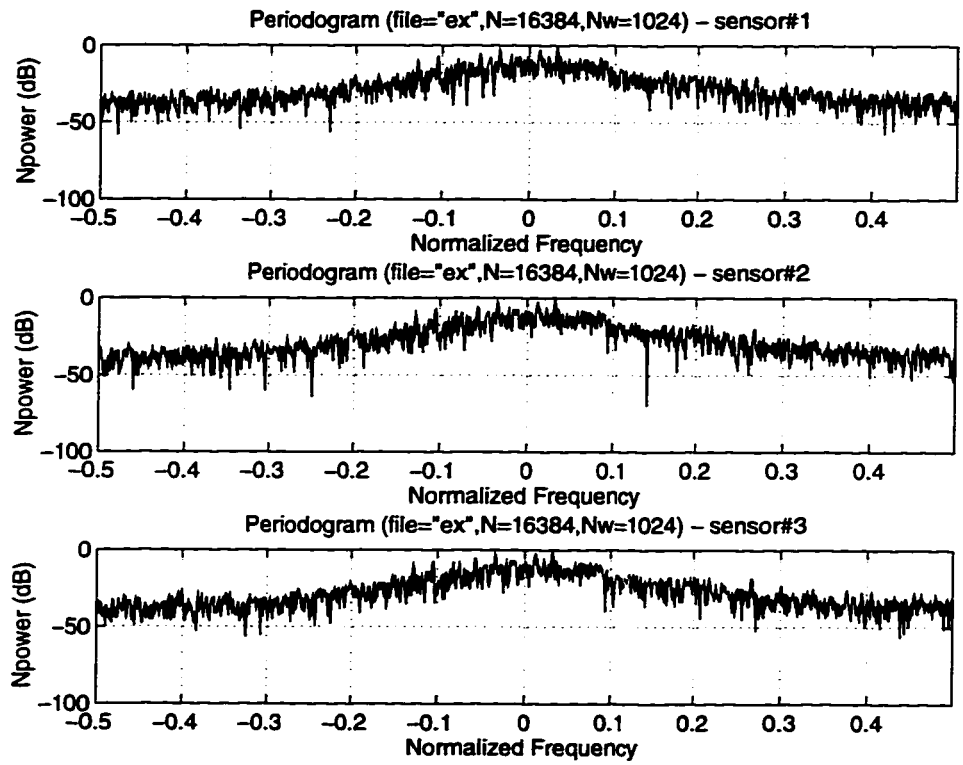

Figure 2.11: Welch periodogram of the three first sensor received signals in the example signal scenario. A total of 16384 samples are considered, the window size is 1024, and no overlap is used.

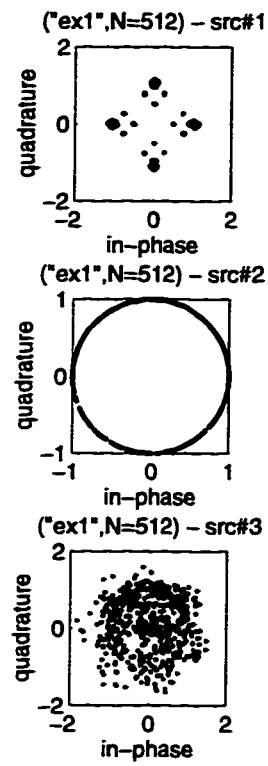

Figure 2.12: I-Q constellation plot of the signal sources (4-PSK, FM, and STANAG4285) in the example signal scenario. Only the first 512 samples are shown.

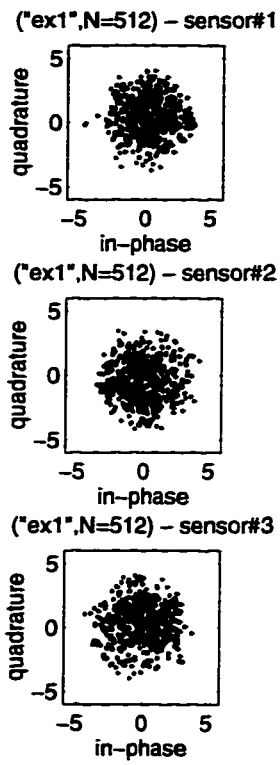

Figure 2.13: I-Q constellation plot of the first three sensor received signals in the example signal scenario. Only the first 512 samples are shown.

2.6.1.1 Minimum Mean-Squared Error (MMSE)

The least mean-square (LMS) algorithm (e.g. [47]) presented next leads to the MMSE solution since its objective is to minimize

$$\min \left\langle \|\hat{s}_l(n) - s_l(n)\|^2 \right\rangle_N . \tag{2.59}$$

Dropping momentarily most arguments for convenience, this equation may be re-written as

$$\min_{\mathbf{w}} \left\langle \left\| \mathbf{w}^H \mathbf{x} - \mathbf{s} \right\|^2 \right\rangle_N . \tag{2.60}$$

The equation to minimize then becomes

$$\begin{aligned} \varepsilon = & \left(\mathbf{w}^H \mathbf{X} - \mathbf{s}\right) \left(\mathbf{w}^H \mathbf{X} - \mathbf{s}\right)^H \\ = & \ \mathbf{w}^H \mathbf{X}\mathbf{X}^H \mathbf{w} - \mathbf{s}\mathbf{X}^H \mathbf{w} - \mathbf{w}^H \mathbf{X}\mathbf{s}^H + \mathbf{s}\mathbf{s}^H . \end{aligned} \tag{2.61}$$

ε can be minimized by operating a differentiation[15] with respect to $\mathbf{w}^*$, i.e.

$$\frac{\partial \varepsilon}{\partial \mathbf{w}^*} = \mathbf{X}\mathbf{X}^H \mathbf{w} - \mathbf{X}\mathbf{s}^H \tag{2.62}$$

which, when put equal to zero and extended to multiple signals, leads to the MMSE solution

$$\begin{aligned} \mathbf{W}_{mmse} &= (\mathbf{X}\mathbf{X}^H)^{-1} \mathbf{X}\mathbf{S}^H \\ &= \mathbf{R}_{\mathbf{xx}}{}^{-1} \mathbf{R}_{\mathbf{xs}} . \end{aligned} \tag{2.63}$$

A block diagram of the MMSE adaptive filter is shown in Fig. 2.14.

The MMSE solution requires known training signals. As long as the training signals correspond exactly to the signal sources, this solution provides an estimate $\hat{s}_{l,mmse}(n) = \mathbf{w}_l^H \mathbf{x}(n)$ which achieves the maximum SINR attainable (SINR_{opt}) for any possible linear filter structure. In other words, given a specific filter structure, no other methods will be able to exceed this SINR_{opt}. This is the reason why the

[15] The following well-known identities are used in the transformation: $\frac{\partial(\mathbf{w}^H \mathbf{K})}{\partial \mathbf{w}^*} = \mathbf{K}$, and $\frac{\partial(\mathbf{w}\mathbf{K})}{\partial \mathbf{w}^*} = 0$.

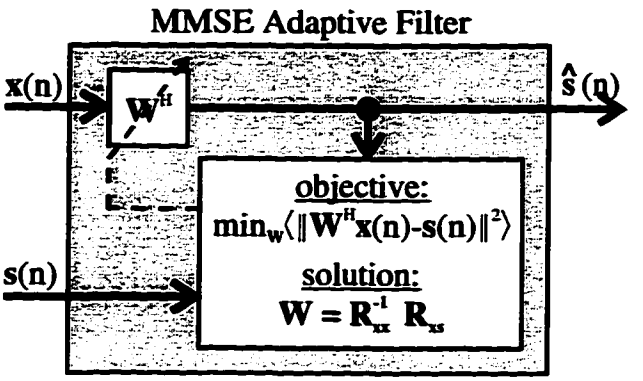

Figure 2.14: Block diagram of the MMSE adaptive filter.

performance measures NMSER and NSINR are always smaller or equal to 0 for all methods, with 0 resulting only from the optimum filter weights.

The MMSE solution is applied to the signal scenario example. All the 16384 snapshots have been used to generate the MMSE signal estimates. The resulting SINRs are found to be

$$\begin{aligned} \text{SINR}_{opt}(1) &= 15.66\text{dB} \\ \text{SINR}_{opt}(2) &= 18.88\text{dB} \\ \text{SINR}_{opt}(3) &= 19.94\text{dB} \end{aligned} \quad . \tag{2.64}$$

Figures 2.15, 2.16, and 2.17 respectively show the periodogram, the I-Q constellation plot, and the resulting filter weights antenna pattern[16]. In the case of the periodogram plots, the original signal source periodograms are overlaid to the signal estimate ones which are shown as dotted lines. In both the periodogram and I-Q plots, one may observe that the white noise remains present since it is now the dominant source of signal distortions[17].

[16]The filter weights antenna pattern, also known as beam pattern, provides the spatial filter gain response as a function of the DOA.

[17]Since the noise is uncorrelated from one sensor to another, the spatial filter is not expected to actually remove it. At best, it will raise the SNR by a factor that will depend on the antenna array.

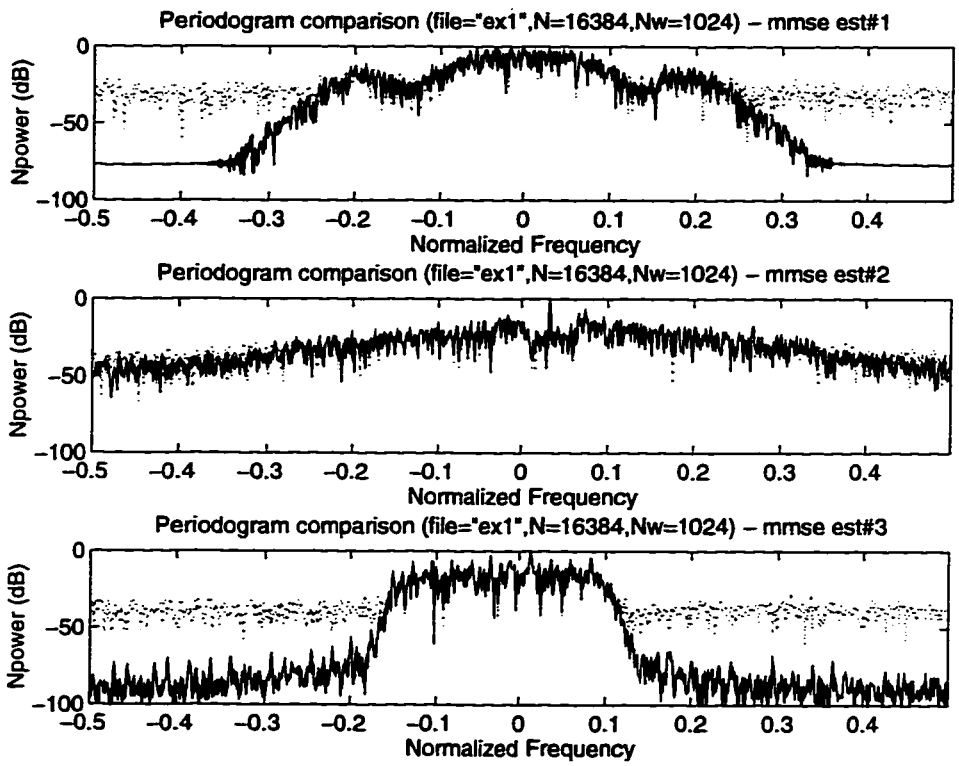

Figure 2.15: Welch periodogram of the signal sources (solid lines) compared to the MMSE signal estimates (dotted lines) in the example signal scenario. A total of 16384 samples are considered, the window size is 1024, and no overlap is used. The three signals correspond to 4-PSK, FM, and STANAG4285.

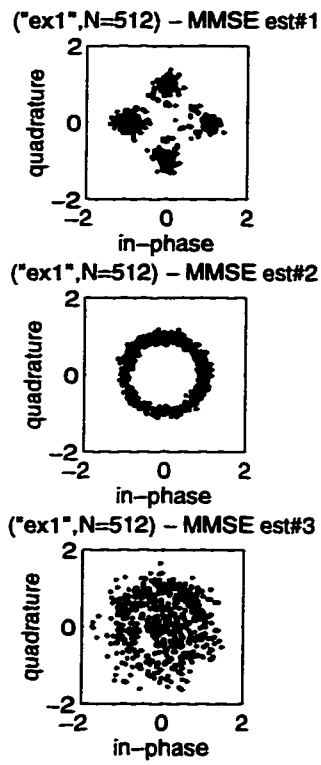

Figure 2.16: I-Q constellation plots of the MMSE signal estimates in the example signal scenario. Only the first 512 samples are shown. The three signals correspond to 4-PSK, FM, and STANAG4285.

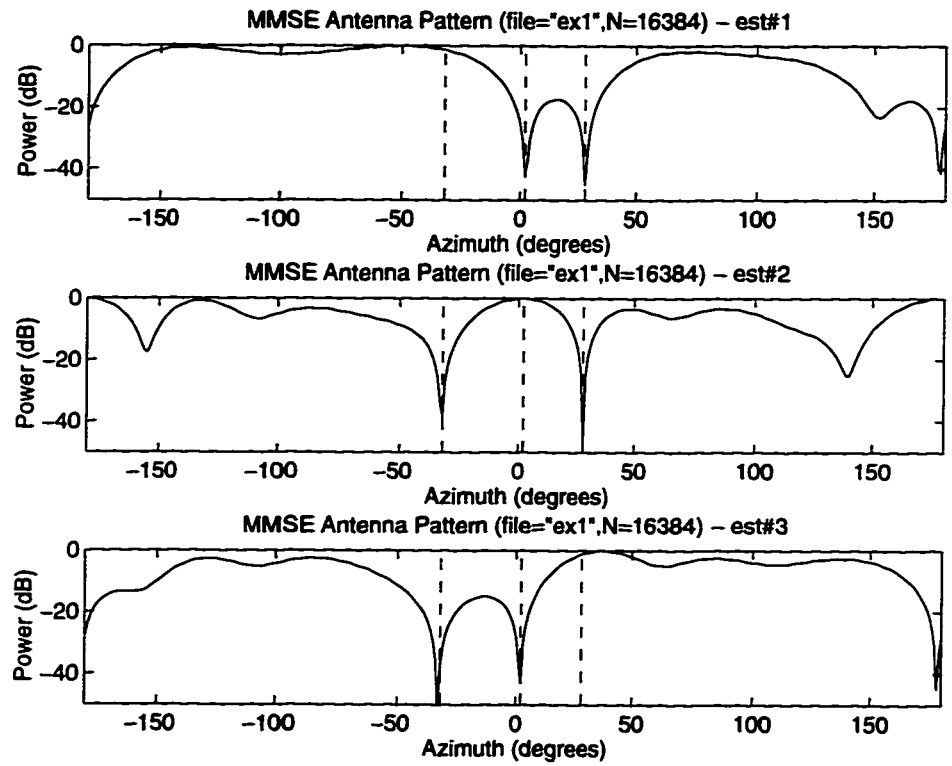

Figure 2.17: MMSE filter weights antenna pattern in the example signal scenario. The dashed vertical lines indicate the true DOAs of the three signals. The three signals correspond to 4-PSK, FM, and STANAG4285.

2.6.1.2 Minimum Variance Distortionless Response (MVDR)

If no training signals are available, the minimum variance distortionless response (MVDR) solution can be used, provided signal DOAs and calibrated array manifold are known. The MVDR objective is to beamform in the direction of the desired signal and put nulls in all other signal a priori known DOAs at the same time. To achieve this objective, the spatial filter weights are chosen to minimize the average output signal power while maintaining unity array gain in the direction of the signal of interest θ_l. Mathematically, the objective is given by

$$\min_{\mathbf{w}_l \text{ s.t. } \mathbf{w}_l \mathbf{a}_{\underline{\theta}_l}=1} \left\langle \|\hat{s}_l(n)\|^2 \right\rangle , \tag{2.65}$$

and the MVDR solution can be shown to be given by

$$\mathbf{w}_{mvdr}(\underline{\theta}_l) = \left(\mathbf{a}(\underline{\theta}_l)^H \mathbf{R}_{xx} \mathbf{a}(\underline{\theta}_l)\right)^{-1} \mathbf{R}_{xx}{}^{-1} \mathbf{a}(\underline{\theta}_l) . \tag{2.66}$$

The solution converges to an output signal estimate with SINR_{opt}, as long as each SOI arrives from a single DOA. This would not be the case in a multipath environment. A block diagram of the MVDR adaptive filter is shown in Fig. 2.18.

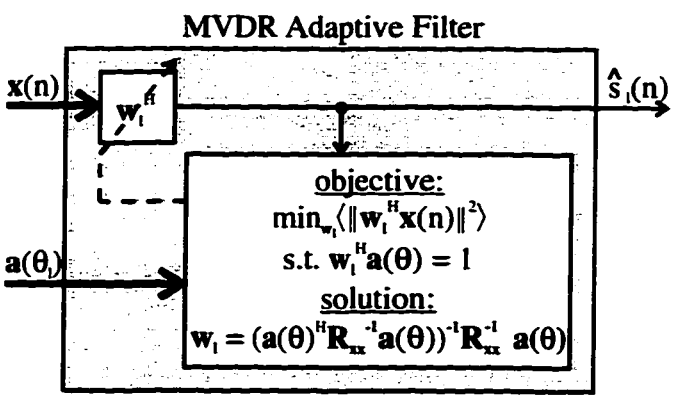

Figure 2.18: Block diagram of the MVDR adaptive filter.

The MVDR adaptive filter has been applied to the example signal scenario and all 16384 snapshots. The exact DOA, number of signals, and array manifold were

provided to the algorithm. The resulting NSINR is found to be -0.37, -0.64, and -0.26 dB for signal source estimates 1, 2, and 3, respectively. This verifies that it converges indeed to $SINR_{opt}$. Fig. 2.19 compares the MVDR and MMSE filter weight estimates antenna pattern. In this case, since the nulls are significantly deeper than the noise level, the crucial point is the intersection with the extracted signal DOAs, shown as dashed lines. At these points, the two filters are found to have almost identical gains. This explains why the results are so close.

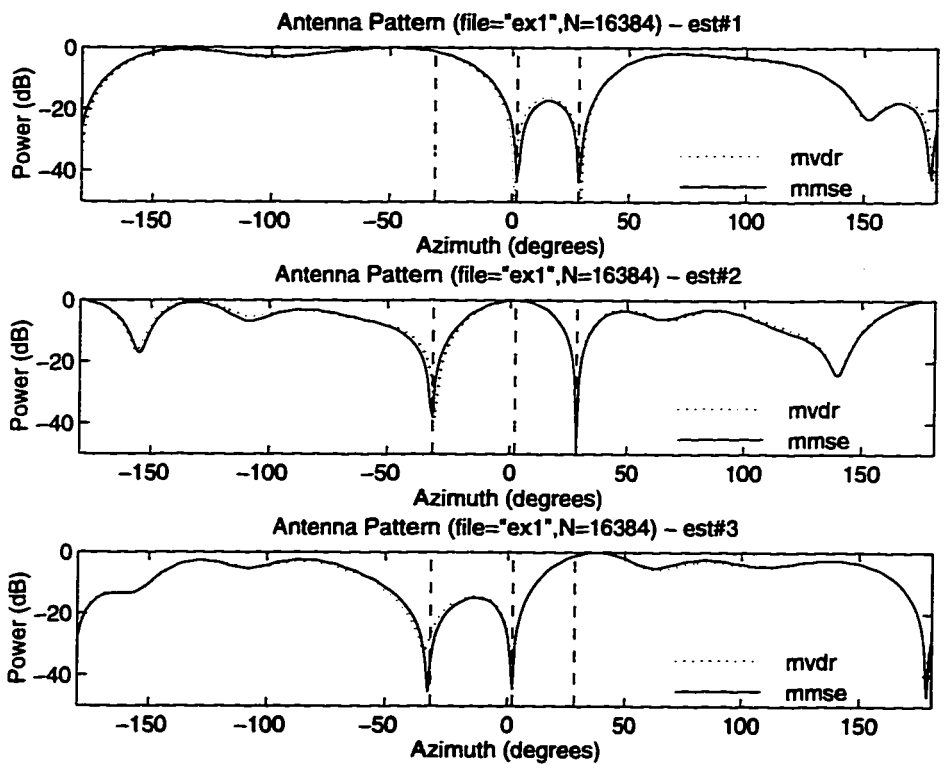

Figure 2.19: MVDR vs MMSE filter weights antenna pattern in the example signal scenario. The dashed vertical lines indicate the true DOAs of the three signals. The three signals correspond to 4-PSK, FM, and STANAG4285.

When DOAs are unknown, direction finding methods have been suggested to estimate them. However, in addition to increasing the computational burden, the calibrated sensor array manifold is still required. Furthermore, no robustness against

multipath is provided.

For an unintended receiver used in radio surveillance and reconnaissance applications, conventional techniques cannot be afforded. This brings the need for blind methods.

2.6.2 Blind Techniques

A blind signal extraction technique achieves its goal without prior knowledge of training signals, DOAs, array geometry, and calibration data. The strategy of the blind techniques is to exploit a selected or a set of signal properties. Those properties may correspond to spatial and/or temporal signal characteristics. Three existing families of blind techniques have been selected for review. They are:

- the constant modulus algorithm (CMA) based techniques;
- the Spectral self-COherence REstoral (SCORE) based techniques; and
- the Programmable Canonical Correlation Analysis (PCCA) based techniques.

2.6.2.1 CMA-based Techniques

The constant modulus algorithm (CMA) based techniques are certainly, at present, the most popular of the blind techniques and have received most of the attention since its first derivation in the early 80's [66]. Many forms of CMA exist but, in general, the CMA objective is to restore the constant modulus property found in many modulated signal types (e.g. FM, FSK, PSK,). This objective may be expressed in mathematical terms as

$$\min_{\mathbf{w}_l} \langle \| |\hat{s}_l(n)|^p - 1 \|^q \rangle \ , \tag{2.67}$$

or equivalently,

$$\min_{\mathbf{w}_l} \left\langle \left\| |\mathbf{w}_l^H \mathbf{x}(n)|^p - 1 \right\|^q \right\rangle \ . \tag{2.68}$$

One particular case termed the least-square CMA (LS-CMA) solves for $(p, q) = (1, 2)$ [1]. Precisely, the LS-CMA objective is to minimize

$$\left\langle \left\| |\mathbf{w}_l^H \mathbf{x}(n)| - 1 \right\|^2 \right\rangle \tag{2.69}$$

subject to the constraint $\mathbf{w}_l^H \mathbf{R}_{xx} \mathbf{w}_l = 1$. A possible solution is found to be an iterative solution given by

$$\begin{aligned} r(n) &= \mathbf{w}_{l,i}^H \mathbf{x}(n) / |\mathbf{w}_{l,i}^H \mathbf{x}(n)| \\ \mathbf{w}_{l,i+1} &= \mathbf{R}_{xx}^{-1} \mathbf{R}_{xr} \end{aligned} \tag{2.70}$$

Refer to [1] for the details on the derivation leading to this solution. The LS-CMA adaptive filter solution is pictured in Fig. 2.20. To apply the LS-CMA technique to

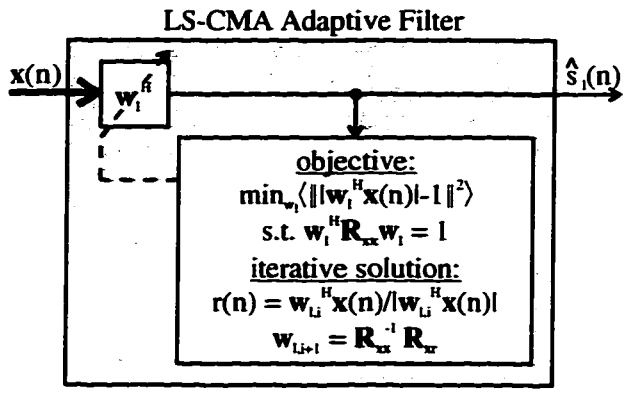

Figure 2.20: Block diagram of the LS-CMA adaptive filter.

the example signal scenario and allow the extraction of multiple signals, a structure named the constant modulus array (CMAR) described in [25] is used. With this multi-stage structure, the CMA is used to extract each signal, one at a time. This is achieved by running CMA (or LS-CMA in this case) to the input sensor data matrix, obtaining the first signal estimate, and removing this estimate from the sensor data. This new sensor data matrix then forms the input to the second CMA stage which extracts the second signal, removes it from the sensor data matrix, and pass it to the next stage. This CMAR structure will form orthogonal output signal estimates since,

at each stage, the signal estimate is removed from the sensor data matrix by applying

$$\begin{aligned} \mathbf{u} &= \mathbf{X}\hat{\mathbf{s}}_l^H/(\hat{\mathbf{s}}_l\hat{\mathbf{s}}_l^H) \\ \mathbf{X} &= \mathbf{X} - \mathbf{u}\hat{\mathbf{s}}_l \end{aligned} \tag{2.71}$$

Using the CMAR structure with the LS-CMA technique, results are obtained for the example signal scenario. A three stage structure is used with an even number of iterations at each stage varied from 1 to 20. The results are shown in Fig. 2.21 as a plot of NSINR versus the number of iterations. In this plot, a curve is shown for each of the estimated signals as well as for the mean results. It should be noted that the number of iterations required is highly dependent on the initial filter weights vector estimate which, in this example, was chosen to be equal to [1,1,1,1,1] for all three stages. With more than about 13 iterations, the algorithm reaches signal estimates NSINR within 2 dB of the optimal solutions for all signals, although the first and third were not truly constant modulus signals. In fact, investigations of this technique show that, as a blind beamformer, it can lock onto non-constant modulus signals as well. One known problem of this multi-stage approach is that, for short data sequences, the signals are not orthogonal yet, and therefore, the orthogonality constraint imposed on the output signal estimates may lead to limited performance in all stages subsequent to the first one. This problem has lead to other CMA approaches aimed at relaxing the orthogonality constraint by using a parallel CM architecture such as the Multi-Target CMA (MT-CMA) technique [4].

2.6.2.2 SCORE-based Techniques

The Spectral self-COherence REstoral (SCORE) algorithms for blind adaptation of sensor arrays were first introduced in 1987 [2] (subsequent work can be found in [3, 5, 23, 54]). A SCORE technique objective function tries to exploit and restore a property known as cyclostationarity.

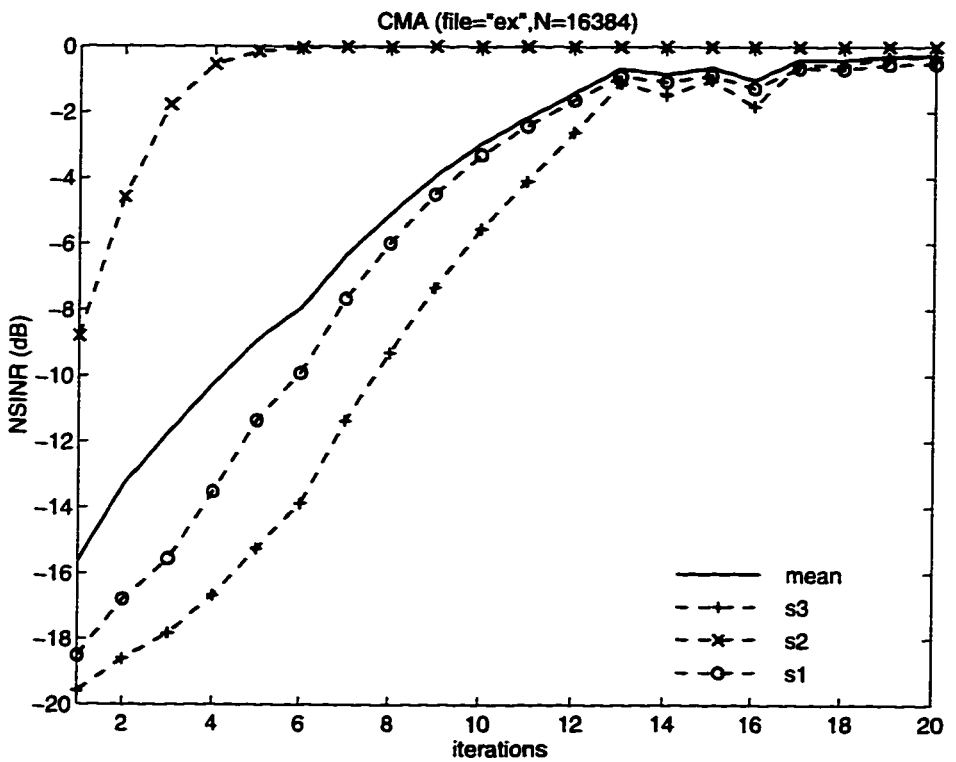

Figure 2.21: Performance results of the LS-CMA with a three-stage CMAR structure. The NSINR is plotted against the number of iterations equally set for all three stages. The result of each signal is shown as well as the mean. A block size of 16384 snapshots is used. The three signals correspond to 4-PSK, FM, and STANAG4285.

A signal $s(n)$ is said to be spectrally self-coherent at frequency separation β if the correlation[18] between itself and a frequency shifted version of itself is not zero for some lag τ. Similarly, the same signal is said to be spectrally conjugate self-coherent at frequency separation $\overline{\beta}$ if the correlation between $s(n)$ and a frequency shifted version of the conjugate of $s(n)$ is not zero for some lag τ. The frequencies, denoted β and $\overline{\beta}$, are respectively used to specify the cycle frequency and conjugate cycle frequency. A signal which is spectrally self-coherent or conjugate self-coherent at these cycle frequencies is said to exhibit second-order cyclostationarity. A single cyclostationary feature, or cyclic feature, can be defined as a pair of (β, τ) or $(\overline{\beta}, \tau)$. The objective function of SCORE techniques is setup to maximize the level of spectral (or conjugate) self-coherence at some cyclic feature chosen to discriminate with some other undesired signals not sharing this feature. The most well known examples of baseband signal cyclic features are multiple of the baud rates, doubled carrier frequencies, and differences of these. The cyclic feature distribution depends on the particular signal type. Many SCORE techniques are presented in [5]. One of them, the cross-SCORE algorithm, is presented and tested here against the example signal scenario.

The objective of the cross-SCORE algorithm is to maximize the strength of the cross-correlation coefficient between $\hat{\mathbf{s}}_l(n)$ and $\hat{\mathbf{r}}(n) = \mathbf{w}_d^H \mathbf{d}(n)$ where $\mathbf{d}(n)$ is some delayed and frequency shifted version of the input sensor data $\mathbf{x}(n)$. Precisely, the objective is

$$\max_{\mathbf{w}_l} \left\{ \frac{|\mathbf{w}_l^H \hat{\mathbf{R}}_{\mathbf{xd}} \mathbf{w}_{d,l}|^2}{(\mathbf{w}_l^H \hat{\mathbf{R}}_{\mathbf{xx}} \mathbf{w}_l)(\mathbf{w}_{d,l}^H \hat{\mathbf{R}}_{\mathbf{xx}} \mathbf{w}_{d,l})} \right\}, \tag{2.72}$$

where the control data vector is defined as

$$\mathbf{d}(n) = \mathbf{x}(n - \tau) e^{j2\pi\beta n} \tag{2.73}$$

[18]This represents the correlation taken in the limit.

for the non-conjugate cycle frequency, or

$$\mathbf{d}(n) = \mathbf{x}^*(n-\tau)e^{j2\pi\overline{\beta}n} \tag{2.74}$$

in the case where the conjugate cycle frequency is the feature of interest. Imposing the constraints $\mathbf{w}_l^H\hat{\mathbf{R}}_{\mathbf{xx}}\mathbf{w}_l = 1$ and $\mathbf{w}_{d,l}^H\hat{\mathbf{R}}_{\mathbf{dd}}\mathbf{w}_{d,l} = 1$, the solution is found by computing the eigenvectors corresponding to the strongest eigenvalues of the following two equations:

$$\lambda\hat{\mathbf{R}}_{\mathbf{xx}}\mathbf{w}_l = \left(\hat{\mathbf{R}}_{\mathbf{xd}}\hat{\mathbf{R}}_{\mathbf{dd}}^{-1}\hat{\mathbf{R}}_{\mathbf{xd}}^H\right)\mathbf{w}_l \tag{2.75}$$

and

$$\lambda\hat{\mathbf{R}}_{\mathbf{dd}}\mathbf{w}_{d,l} = \left(\hat{\mathbf{R}}_{\mathbf{xd}}^H\hat{\mathbf{R}}_{\mathbf{xx}}^{-1}\hat{\mathbf{R}}_{\mathbf{xd}}\right)\mathbf{w}_l\,. \tag{2.76}$$

Those two equations are known to as the cross-SCORE eigenequations. The block diagram of the cross-SCORE adaptive filter is shown in Fig. 2.22. To use the cross-SCORE algorithm, the values of lag τ and cycle frequency β or cycle frequency $\overline{\beta}$ must be known and provided. It should be noted that this algorithm can easily be extended to accommodate multiple values of lags and cycle and/or conjugate cycle frequencies.

Once again, the example signal scenario is used to test out the algorithm. This time, the cycle frequencies and lags need to be selected. Examining the signal sources, a number of exploitable cycle frequencies can be identified. Those are:

- for *source 1*, $\beta = \pm 1320$ Hz (symbol rate);
- for *source 2*, $\overline{\beta} = 642$ Hz (doubled carrier frequency); and
- for *source 3*, $\overline{\beta} = -440$ Hz (doubled carrier frequency) and $\beta = \pm 2400$ Hz (symbol rate).

All these features may be combined with $\tau = 0$. Furthermore, multiple values of τ can be used. Many combinations have been chosen for comparison. Multiple values

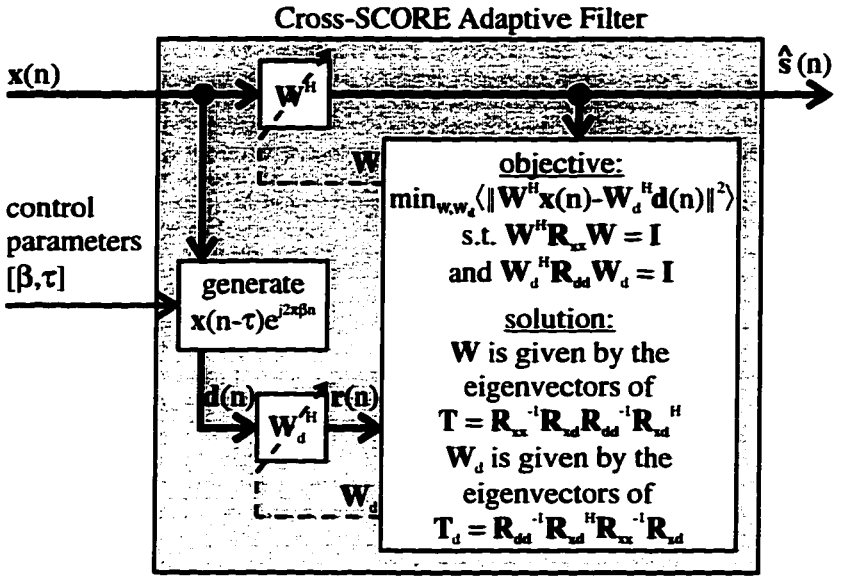

Figure 2.22: Block diagram of the Cross-SCORE adaptive filter.

of lags and cycle frequencies are also considered. For all these runs, the algorithm has been modified by the author to include feedback. Fig. 2.23 shows the results obtained when the features are selected to extract the signal *source 1* (see indicated features). Clearly, the use of multiple features results in superior performance. Fig. 2.24 demonstrates extraction of the other two signals by carefully setting the desired features. The limitations of this algorithm can be summarized as follows: it requires a priori knowledge of the cyclic features and the number of signals, the features strengths have to be unequal if multiple signals share a same feature, the output signal estimates sharing a same feature are made orthogonal, and finally, it needs a particular set of cyclic features for each extracted signal not sharing the same feature.

2.6.2.3 PCCA-based Techniques

The Programmable Canonical Correlation Analysis (PCCA) algorithm [54] is based on the Canonical Correlation Analysis work originally developed by Hotelling [29] in the 1930's. The general framework provided with PCCA allows most previous

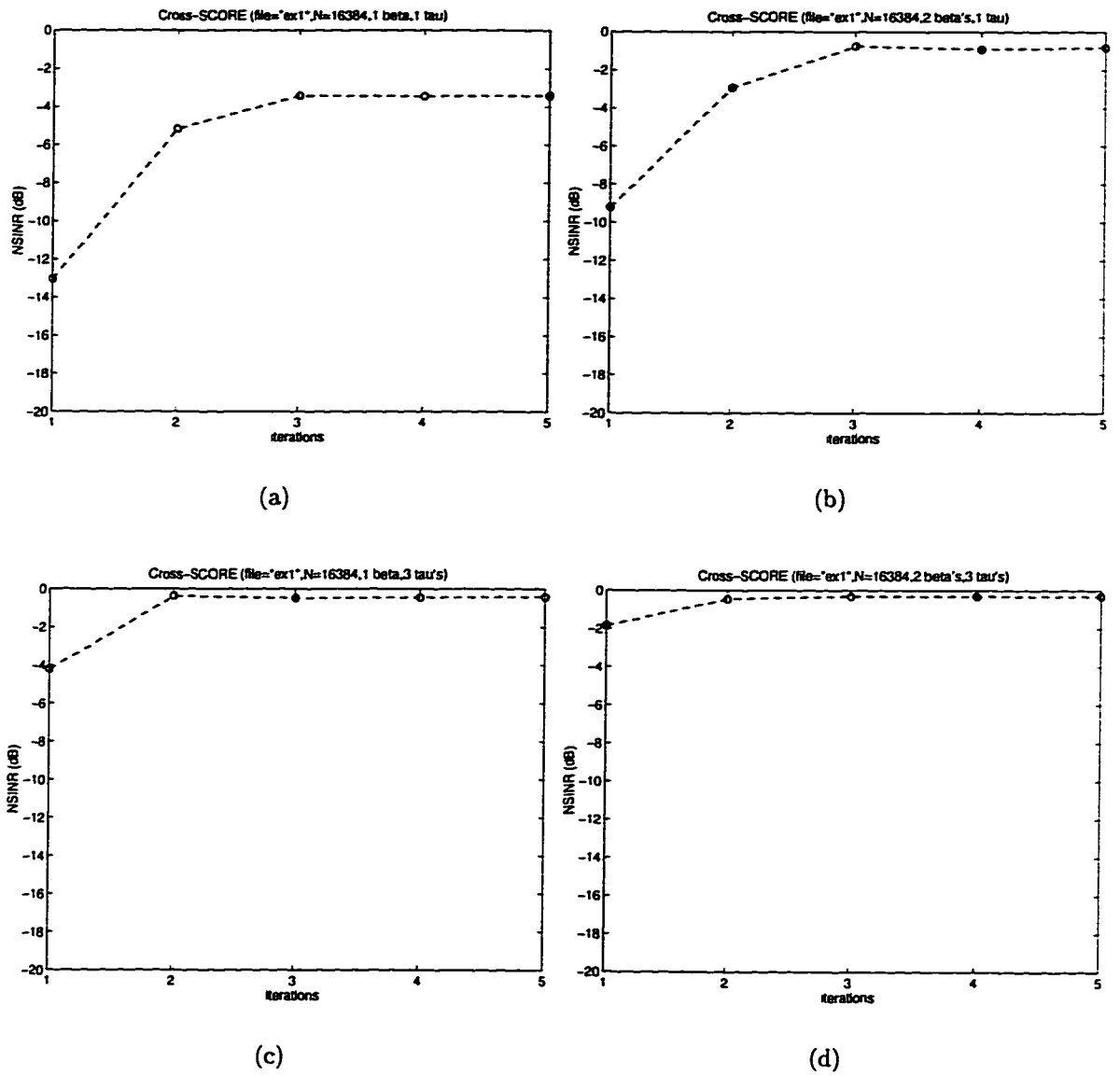

Figure 2.23: Results of the Cross-SCORE technique setup to extract the signal corresponding to the first signal source (4-PSK) in the example signal scenario. The results are shown for various cyclic features selected: (a) $\beta = 1320$ Hz and $\tau = 0$, (b) $\beta = [1320, -1320]$ Hz and $\tau = 0$, (c) $\beta = 1320$ Hz and $\tau = [-1 : 1]$, and finally (d) $\beta = [1320, -1320]$ Hz and $\tau = [-1 : 1]$.

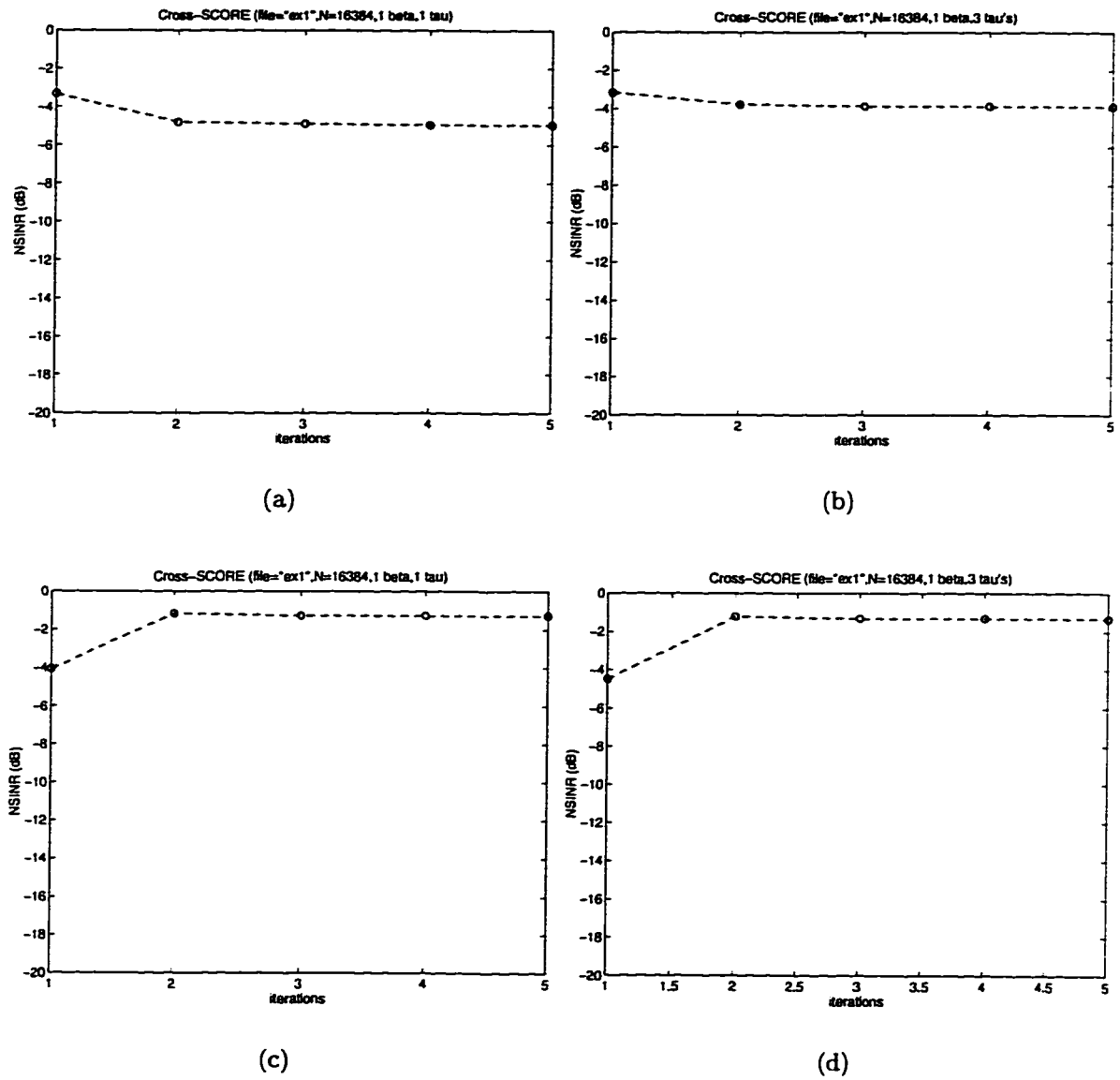

Figure 2.24: Results of the Cross-SCORE technique setup to extract the other two signals (FM and STANAG4285) in the example signal scenario. (a) $\overline{\beta} = 642$ Hz and $\tau = 0$, and (b) $\overline{\beta} = 642$ Hz and $\tau = [-1:1]$ are setup to extract the second source while (c) $\overline{\beta} = -440$ Hz and $\tau = [0]$, and (d) $\overline{\beta} = [-440]$ Hz and $\tau = [-1:1]$ are for estimating the third signal source.

techniques to be a special case of PCCA-based techniques. The PCCA approach is reviewed next.

Let $\mathbf{d}(n)$ be the output of a user-programmable transformation of $\mathbf{x}(n)$ (generally chosen to be correlated with the SOIs but uncorrelated with the SNOIs). The two data sets, $\mathbf{x}(n)$ and $\mathbf{d}(n)$, may be used to provide two signal estimate sets obtained from linear combinations of each set, i.e. $\hat{\mathbf{s}}(n) = \mathbf{W}^H\mathbf{x}(n)$ and $\hat{\mathbf{s}}_d(n) = \mathbf{W}_d^H\mathbf{d}(n)$. The PCCA seeks to minimize the MSE between $\hat{\mathbf{s}}(n)$ and $\hat{\mathbf{s}}_d(n)$, subject to the constraints that each set of signal estimates produces uncorrelated signals (in other words, uncorrelated sources are assumed). The mathematical expression of this PCCA optimization criterion is given by

$$\min_{\mathbf{W},\mathbf{W}_d} \left\langle \left\| \mathbf{W}^H\mathbf{x}(n) - \mathbf{W}_d^H\mathbf{d}(n) \right\|^2 \right\rangle_N \tag{2.77}$$

subject to the constraints[19] that $\mathbf{W}^H\hat{\mathbf{R}}_{\mathbf{xx}}\mathbf{W} = \mathbf{I}$ and $\mathbf{W}_d^H\hat{\mathbf{R}}_{\mathbf{dd}}\mathbf{W}_d = \mathbf{I}$. This problem can be solved with the method of Lagrange multipliers. Defining

$$J = \mathrm{tr}\left\{ \mathbf{W}^H\mathbf{R}_{\mathbf{xx}}\mathbf{W} - \mathbf{W}^H\mathbf{R}_{\mathbf{xd}}\mathbf{W}_d - \mathbf{W}_d^H{\mathbf{R}_{\mathbf{xd}}}^H\mathbf{W} + \mathbf{W}_d^H{\mathbf{R}_{\mathbf{dd}}}^H\mathbf{W}_d \right\}, \tag{2.78}$$

and transforming the two constraints into

$$C = \mathrm{tr}\left\{ \mathbf{W}^H\mathbf{R}_{\mathbf{xx}}\mathbf{W} + \mathbf{W}_d^H\mathbf{R}_{\mathbf{dd}}\mathbf{W}_d - c\mathbf{I} \right\} = 0, \tag{2.79}$$

the solution for the optimum filter weights may be obtained by solving

$$J_1 = \frac{\partial J}{\partial \mathbf{W}^*} - \frac{\partial C}{\partial \mathbf{W}^*}\mathbf{\Lambda} = 0 \tag{2.80}$$

and

$$J_2 = \frac{\partial J}{\partial \mathbf{W}_d^*} - \frac{\partial C}{\partial \mathbf{W}_d^*}\mathbf{\Lambda} = 0. \tag{2.81}$$

These two equations respectively result in[20]

$$\mathbf{W} = {\mathbf{R}_{\mathbf{xx}}}^{-1}\mathbf{R}_{\mathbf{xd}}\mathbf{W}_d(\mathbf{I} - \mathbf{\Lambda})^{-1} \tag{2.82}$$

[19] Effectively, these constraints force the estimated signals to be orthonormal.

[20] The following identities are used to arrive at the results: $\frac{\partial \mathrm{tr}\{\mathbf{W}\}}{\partial \mathbf{W}^*} = 0$, $\frac{\partial \mathrm{tr}\{\mathbf{W}^H\mathbf{A}\}}{\partial \mathbf{W}^*} = \mathbf{A}$.

and

$$\mathbf{W}_d = \mathbf{R_{dd}}^{-1}\mathbf{R_{xd}}^H\mathbf{W}(\mathbf{I}-\mathbf{\Lambda})^{-1}, \tag{2.83}$$

which then becomes[21]

$$\mathbf{R_{xx}}^{-1}\mathbf{R_{xd}}\mathbf{R_{dd}}^{-1}\mathbf{R_{xd}}^H\mathbf{W} = \mathbf{W}\mathbf{\Lambda} \tag{2.84}$$

and

$$\mathbf{R_{dd}}^{-1}\mathbf{R_{xd}}^H\mathbf{R_{xx}}^{-1}\mathbf{R_{xd}}\mathbf{W}_d = \mathbf{W}_d\mathbf{\Lambda}. \tag{2.85}$$

The solution for $\mathbf{W}$ and $\mathbf{W}_d$ are therefore the eigenvectors of

$$\mathbf{T_x} = \mathbf{R_{xx}}^{-1}\mathbf{R_{xd}}\mathbf{R_{dd}}^{-1}\mathbf{R_{xd}}^H \tag{2.86}$$

and

$$\mathbf{T_d} = \mathbf{R_{dd}}^{-1}\mathbf{R_{xd}}^H\mathbf{R_{xx}}^{-1}\mathbf{R_{xd}} \tag{2.87}$$

respectively. In [55], this solution was also derived from another perspective to show that it is in addition the results of a constrained conditional maximum likelihood optimization. One may note the equivalence with the cross-SCORE solution, with the difference that, in this case, $\mathbf{d}(n)$ is not limited to a delayed frequency shifted version of the input but may be any linear or non-linear transformation of the input (as well as the output if desired). The only real restriction on this transformed signal is that it does not equal the input data matrix, otherwise any solution would satisfy the eigenequations. A non-exhaustive but very interesting list of useful transformations were proposed in [54]. Those are repeated here for reference:

1. $\mathbf{d}(n)$ is a frequency-shifted and delayed version of $\mathbf{x}(n)$ or $\mathbf{x}(n)^*$, yielding into the Cross-SCORE algorithm described previously (this results in selecting signals exhibiting cyclostationarity or conjugate cyclostationarity and could be generalized to include multiple frequency shift, multiple delays, and pre-filtering);

[21] Note that the following reassignment is made: $\mathbf{\Lambda} = (\mathbf{I}-\mathbf{\Lambda})(\mathbf{I}-\mathbf{\Lambda})$.

2. $\mathbf{d}(n)$ is the output of some LTI filter applied to $\mathbf{x}(n)$ (this results in selecting signals that have spectral support as defined by the filter);

3. $\mathbf{d}(n)$ is simply a delayed version of $\mathbf{x}(n)$ (this results in selecting signals for which coherence time is greater than or equal to the delay used);

4. $\mathbf{d}(n)$ is the output of a temporal gating device applied to $\mathbf{x}(n)$ (this results in selecting signals that are active outside the stop interval);

5. $\mathbf{d}(n)$ is the narrowband (or wideband) output of an adaptive spectral line enhancer applied to $\mathbf{x}(n)$ (this results in selecting signals that are relatively narrowband (or wideband));

6. $\mathbf{d}(n)$ is the enhanced (or degraded) output of a spectral correlation enhancer (a blind FRESH filter) applied to $\mathbf{x}(n)$ (this results in selecting signals that exhibit (or don't exhibit) cyclostationarity at a specified cycle frequency);

7. $\mathbf{d}(n)$ is the constant modulus (or non-constant modulus) output of an LTI filter (or LTI canceller) adapted by the CMA (this results in selecting signals that have (or do not have) constant modulus);

8. $\mathbf{d}(n)$ is the output of a demodulation/remodulation device that is applied to $\mathbf{x}(n)$ (this results in selecting signals that have the particular modulation scheme used).

9. $\mathbf{d}(n)$ is some higher order element-wise power or Kronecker product of $\mathbf{x}(n)$ (this results in selecting signals that have higher-order stationarity or higher-order cyclostationarity selected for by the chosen transformation).

As stated previously, the key in the selection of the transformation is to have a transformed signal that correlates with the desired signals and not with others.

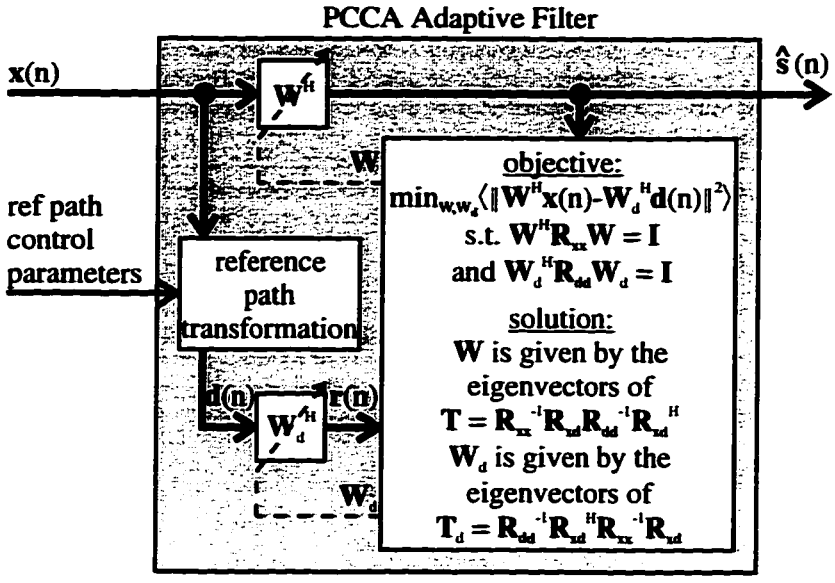

Figure 2.25: Block diagram of the PCCA adaptive filter.

The block diagram of the PCCA adaptive filter is shown in Fig. 2.25. In reality, there is no need to solve for $\mathbf{W}_d$ since only $\mathbf{T}_x$ is required to solve for $\mathbf{W}$. Also, as suggested for instance in [30], an iterative block power method [26] may alternatively be used instead of the eigenvalue (or singular value) decomposition. This solution is given by

$$\begin{aligned} \mathbf{W}_{d,k} &= \mathbf{R_{dd}}^{-1}\mathbf{R_{xd}}^{H}\mathbf{W}_{k-1} \\ \mathbf{W}_k &= \mathbf{R_{xx}}^{-1}\mathbf{R_{xd}}\mathbf{W}_{d,k} \end{aligned} \tag{2.88}$$

where $\mathbf{W}_k$ and $\mathbf{W}_{k-1}$ denote respectively the actual and previous iterative block update solution. Note that a Gram-Schmidt orthogonalization procedure is required between each update[22].

A PCCA transformation has been selected for each signal to extract in the example signal scenario. Obviously, PCCA could have been setup just as Cross-SCORE and inevitably would have given the same results. This would have been too trivial, and

[22]In addition, $\mathbf{W}_{d,k}$ must be orthogonalized prior to being inserted in the update equation for $\mathbf{W}_k$.

therefore, other transformations are found. The first transformation is selected to be the fourth-order conjugate cyclostationarity (at 0 Hz) exhibited by the first signal source (due to its carrier frequency). Specifically,

$$d_1(n) = [x_s(n)^*]^3 e^{j2\pi 0n/10000} \tag{2.89}$$

where

$$x_s(n) = \text{sum}\{\mathbf{x}(n)\} \tag{2.90}$$

is used to extract the first signal. The results are shown in Fig. 2.26. The sec-

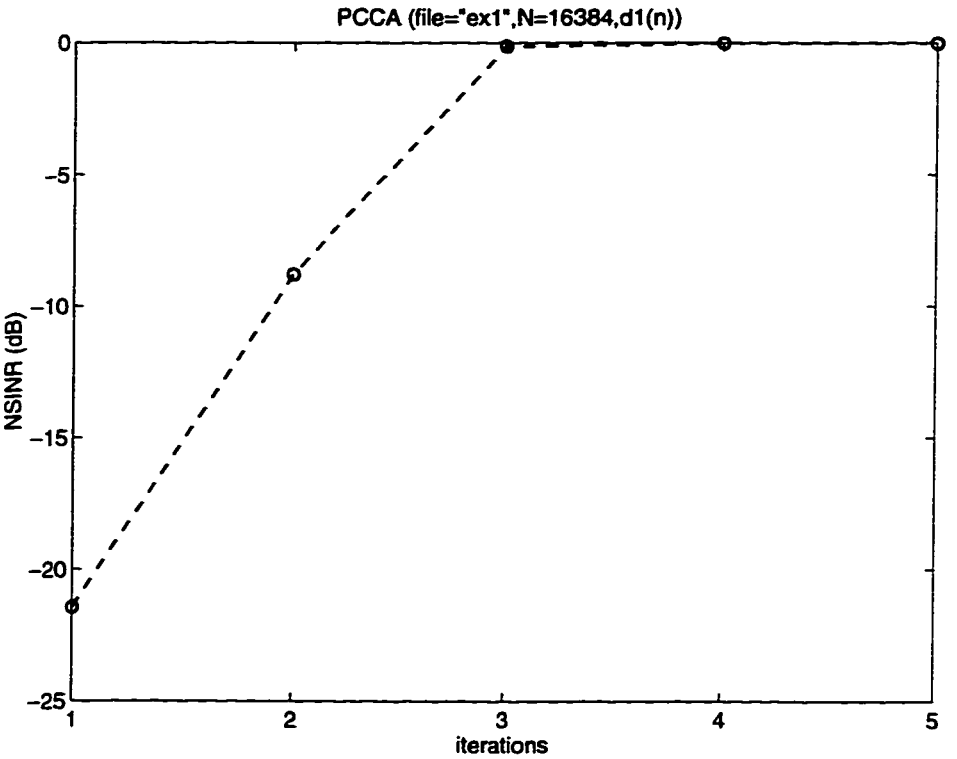

Figure 2.26: Performance results of the PCCA for the example signal scenario. The PCCA transformation is $d_1(n)$ to extract the first signal (4-PSK). The NSINR for up to five iterations are shown. A block size of 16384 snapshots is used.

ond transformation is chosen to extract the strongest constant modulus signal. The

transformation is

$$d_2(n) = x_s(n)/|x_s(n)| \,. \tag{2.91}$$

The performance results for this transformation are provided in Fig. 2.27. One may

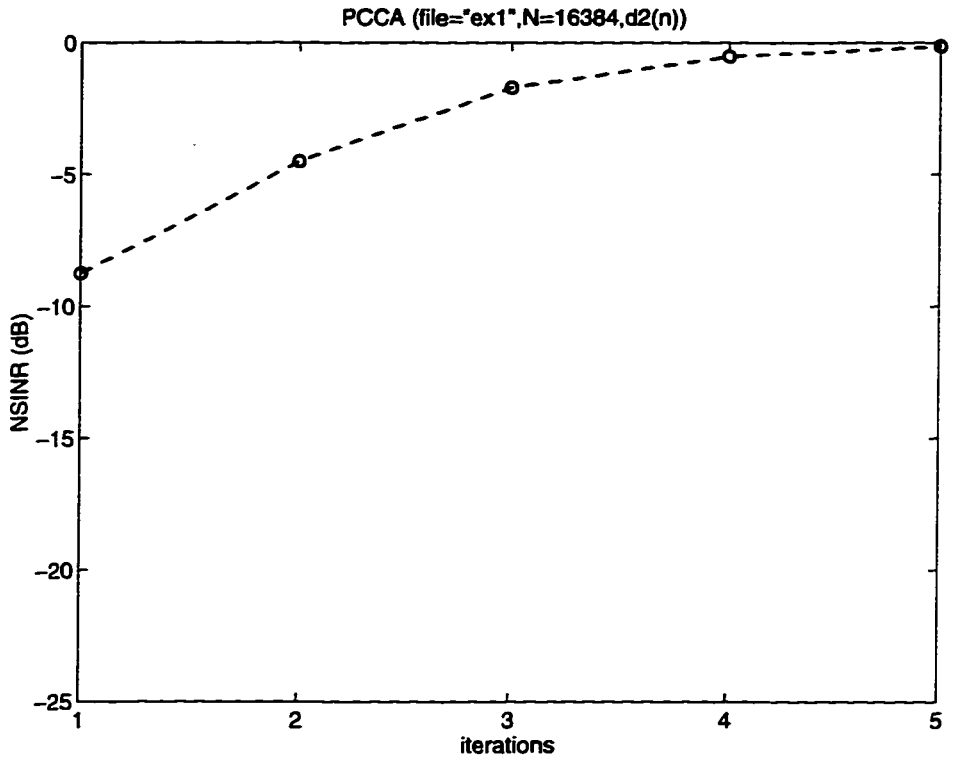

Figure 2.27: Performance results of the PCCA for the example signal scenario. The PCCA transformation is $d_2(n)$ to extract the second signal (FM). The NSINR for up to five iterations are shown. A block size of 16384 snapshots is used.

note that the convergence time is much faster than for the CMA technique tested earlier. Finally, the third transformation is

$$d_3(n) = [x_s(n)^*]^2 x_s(n) e^{j2\pi(-440)n/10000} \tag{2.92}$$

which selects the fourth-order cyclostationary frequency belonging to the third signal. The results of this last transformation are shown in Fig. 2.28. Again, the results are highly satisfactory. It should be noted that, as in the cross-SCORE case, the PCCA

algorithm was modified to include recursion after the first iteration (specifically, $x_s(n)$ was replaced by $\hat{s}(n)$ after the first iteration).

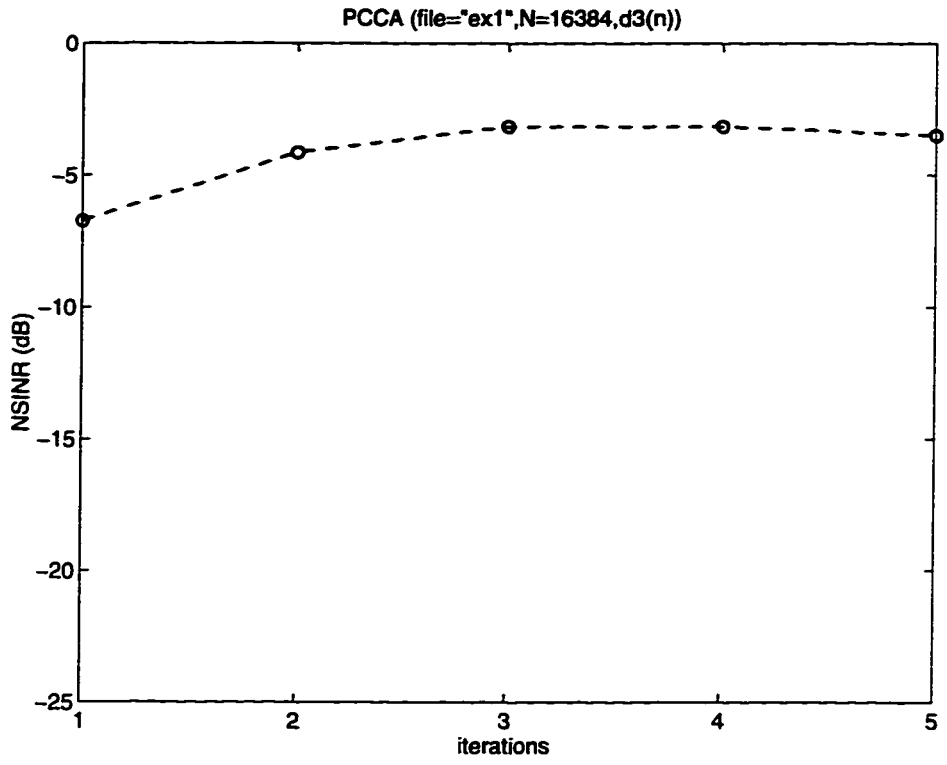

Figure 2.28: Performance results of the PCCA for the example signal scenario. The PCCA transformation is $d_3(n)$ to extract the third signal (STANAG4285). The NSINR for up to five iterations are shown. A block size of 16384 snapshots is used.

A major limitation of PCCA for blind beamforming is that it needs to establish which transformation and parameters to use as a prior knowledge of the signal of interest. This cannot always be possible and it does not actually fulfill the objective of not making any prior assumption about the incoming signals. The technique presented in the next chapter attempts to tackle this problem.

2.7 Summary

This chapter has brought the necessary mathematical background to motivate and understand the content of the next chapters. A summary of the material covered in the present chapter follows.

The three first sections have respectively defined mathematical notation rules and generalities, sample-based expressions for selected analogically and digitally modulated signal source types (CW, AM, FM, m-PSK, m-QAM, m-FSK, and STANAG4285), and sensor array signal models applicable to various conditions including multipath.

In the fourth section, presentations of temporal, spatial, and spatio-temporal filters have lead to the selection of the spatial linear filtering structure for the work described in the following chapters. The reasons for this choice are the filter simplicity, its applicability to the problem of interest, and the existence of blind adaptive solutions.

To measure the performance and merit of any particular adaptive sensor array signal extraction techniques, specific performance measures have been identified in the fifth section. Apart from the existing MSE, NMSE, SINR, BER, and SER performance measures, primarily providing an indication of the signal estimate quality, two new measures proposed by the author, namely the NMSER and NSINR, have been described. In the normalization process, both the NMSER and NSINR use the signal estimate resulting from the non-blind optimal filtering solution (reaching the maximum attainable SINR for a given filter structure). Thus, the new measures give an indication of the optimality of the solution rather than of the signal estimate quality. Since the NMSER and NSINR provide almost identical results in most conditions, the NSINR, due to its closer relationship with the commonly used SINR, has been selected among the two new measures as the performance measure used throughout the dissertation. Other factors such as rate of convergence, tracking, robustness,

computational requirements, structure, numerical properties, and validity were also mentionned as useful indicators of the merit of a technique.

The major portion of this chapter has been devoted to the review of selected signal extraction techniques, including both the conventional and the blind methods. As conventional methods, the MMSE and MVDR techniques have been reviewed. Those non-blind techniques require knowledge of the training signals in the case of the MMSE, and knowledge of the DOA, array geometry, and calibration array manifold in the case of the MVDR technique. For a given filter structure, the MMSE provides the maximum attainable SINR solution and is therefore used to compute the NSINR performance measure. As blind methods, the CMA, SCORE, and PCCA-based signal extraction techniques have been reviewed. They all adapt a spatial linear beamformer, but the CMA is based under the assumption that the modulated signals to extract exhibit constant modulus, the SCORE that they exhibit second-order non-conjugate (or conjugate) cyclostationarity properties, and finally, the PCCA is based under the assumption that a given desired signal to extract best correlates with some transformed version of the received signal without correlating with others. All signal extraction techniques presented have been tested against an example signal scenario containing three signals of different types, with background noise but no multipath, and received at an L-shaped 5-sensor antenna array. Although generally providing reasonably adequate results in the selected signal environment conditions, the methods have important drawbacks leading to limited performance in the context considered for this thesis. Those limitations are listed next.

1. The reviewed techniques depend upon some prior signal characteristic assumptions made as their basis to separate signals (e.g. the signal source must be constant modulus, or the signal must have such and such second-order cyclic features, or the best transformation to use must be such and such transformations, etc.).

2. Signal sources which are in reality independent are assumed to be orthogonal. Orthogonal would also mean uncorrelated signal sources if the signals had zero mean (this is a common assumption made in practice). Even in the case of uncorrelated signals (a necessary condition for independence), independence would only be satisfied if the signal sources were Gaussian (communication signal sources are generally non-Gaussian and therefore uncorrelated signal sources do not necessarily lead to independence).

3. The number of signals sharing the same exploited features need to be known.

4. The prior assumptions made about the signal do not necessarily apply to all signals.

The new blind signal extraction technique, proposed by the author and presented in the next two chapters, attempts to address and overcome those limitations.

Chapter 3

Development of the S-AMUSE Technique

3.1 Approach and Motivation

S-AMUSE stands for the *Self-Adapting sensor array MUlti-Signal blind Extractor* and is a new algorithm designed by the author to blindly and adaptively extract signals. Its derivation comes from a mixture of ideas with some inspired from existing techniques and others, coming out of the author's own analysis, knowledge, and experience in dealing with the blind signal extraction problem.

The main objectives set to arrive at the S-AMUSE algorithm architecture were to overcome the limitations of the existing techiques. More specifically:

- to setup an algorithm that is capable of blindly and adaptively detecting signals;
- to ensure that extracted signals are properly tracked;
- to use a multi-stage architecture, extract one signal at a time, but avoid the orthogonal signal constraints, or at least, allow this constraint to be relaxed;

- to operate with no a priori knowledge of the signals and sensor array geometry or calibration;
- to be robust against multipath and various signal conditions;
- to remain valid against as many types of signal modulations as possible;
- to have some level of flexibility.

The novelty of the S-AMUSE algorithm resides primarily in its particular architecture, and the way signals are detected and control reference signals produced. Other particularities include the use of iterative algorithms with feedback.

The S-AMUSE algorithm is described in the next section. A detailed example is then provided and a discussion on the implementation and coding issues completes the chapter.

3.2 Algorithm Description

3.2.1 Overview

The overall architecture of S-AMUSE is depicted in Fig. 3.1. It primarily consists of L_m stages where each stage is responsible for blindly detecting and extracting up to one signal. This implies that the maximum number of signals that can be extracted (filtered) is limited by the number of stages. Each of these stages is referred to as the S-AMUSE single signal extraction core processor. The operation is sequential since the subsequent stage requires the results of the previous ones. Furthermore, the first stage requires the results obtained at the last stage for the previous block processed results. The complex sensor data block matrix, denoted $\mathbf{X}$, is passed to all stages since one particularity of this algorithm is that the filter weights are applied directly to the sensor data. This contrasts with the usual multi-stage approach, such

as the CMAR described previously, in which the filter weights are applied to the data matrix with all previous stage filtered signals removed. This matrix however, denoted $\mathbf{X}_r$ and called the reference data block matrix, is still produced at each stage, but is used for the sole purpose of detecting new signals and producing new initial signal estimates.

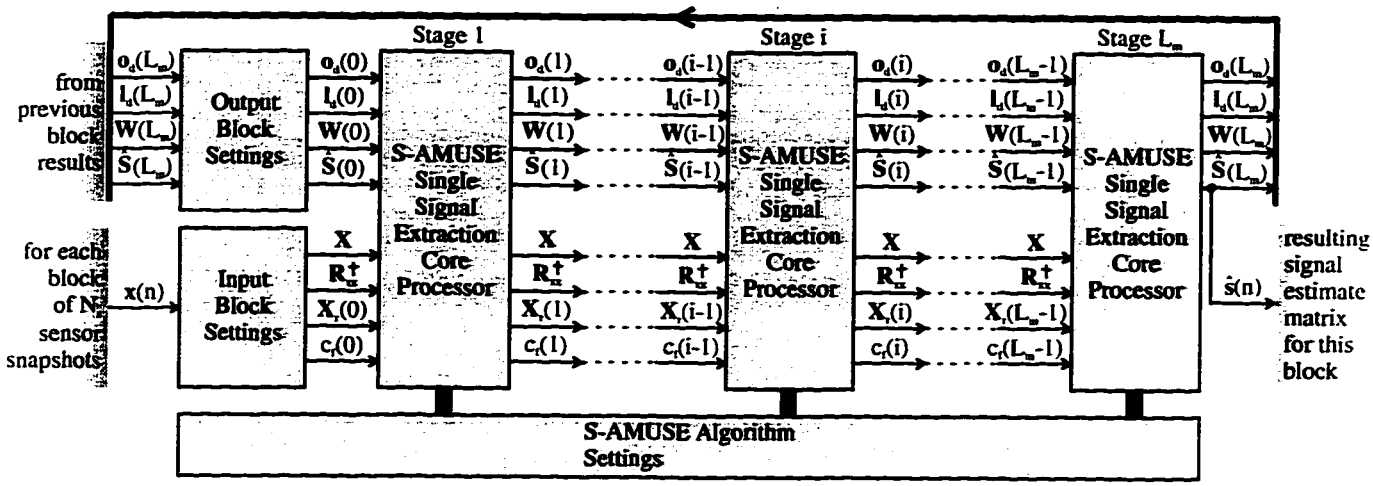

Figure 3.1: Overview of the S-AMUSE architecture.

To ensure that signals detected in the previous block are processed prior to detect new signals, two vectors are defined. The first one, denoted $\mathbf{o}_d$, is called the detection order vector. The second one, denoted $\mathbf{l}_d$, is called the signal ordering vector and may be any permutation of $[1 : L_m]$. The i^{th} element of the vector $\mathbf{l}_d$ indicates to the i^{th} stage which signal to process with respect to the row of the output signal estimate block matrix, $\hat{\mathbf{S}} \in \mathcal{C}^{L_m \times N_s}$, and column of the filter weight matrix, $\mathbf{W} \in \mathcal{C}^{M \times L_m}$. In other words, the i^{th} stage does not automatically process the i^{th} signal, but rather the signal as specified by the signal ordering vector. If the i^{th} element of $\mathbf{o}_d$ is zero, it indicates that no signal was extracted in the previous block number. In this case, a new signal would need to be detected, and if so, extracted. In the event of no detection, the output flag denoted c_f shall be assigned a value of zero. This flag is

used at all stages to determine whether to continue trying to detect new signals or simply stop and wait until the next block arrives. The concept is that, once a stage declares that no further signal has been detected, there is no need to further process the data since it signifies that no more signals are present in the data (at least within the limitation of the detection scheme which is identical at all stages).

Before presenting the details of each of the blocks shown in Fig. 3.1, the various considerations made during the development of S-AMUSE are discussed.

3.2.2 Considerations

This section presents the main considerations made by the author during the development of S-AMUSE in order to arrive at its detailed description provided below in Section 3.2.3. The primary concerns discussed next include:

1. blind signal detection,
2. initial signal estimation, and
3. blind adaptive filter weights determination.

3.2.2.1 Blind Signal Detection

Several ways have been examined to achieve blind signal detection. One basic statement is clear: it is by far much easier to decide whether a signal (or multiple signals) is present than trying to confirm if and how many signals are present simultaneously. Given the objective of extracting signals, a sound strategy would be to treat signals one at a time. More specifically, the strategy would be to detect the presence of signals, extract a single signal, remove this signal from the data, and with this new data, try again to detect the signal presence, and so on and so forth until no signal presence is detected. This constitutes the principal reason for employing a multi-stage structure.

With this strategy, signal detection reduces to only detecting if there is any signal present or not, regardless of how many signals are present. Signal detection is based under the assumption that signals are different than white Gaussian noise. To decide if the sensor data contains more than white Gaussian noise, a number of avenues are possible.

The first avenue considered uses singular value decomposition (SVD). Given the reference data matrix $\mathbf{X}_r$, defined earlier as the data matrix with all previous stage filtered signals removed, the reference data spatial correlation matrix can be defined as

$$\hat{\mathbf{R}}_{\mathbf{x}_r\mathbf{x}_r} = \mathbf{X}_r\mathbf{X}_r^H . \tag{3.1}$$

If $\mathbf{X}_r$ contains only (ideal) white Gaussian noise, the singular values of $\mathbf{R}_{\mathbf{x}_r\mathbf{x}_r}$, denoted $\alpha_s(1:M)$[1], would all be equal. Anything else than white Gaussian noise would result in unequal singular value strengths. Therefore, a possible measure of data whiteness as defined by the author is

$$d_{val} = 10 \cdot \log_{10}\left\{\frac{mean\{\alpha_s(1:M-1)\}}{\alpha_s(M)}\right\} \tag{3.2}$$

given in dB. This detection value will always be greater than 0 dB with 0 dB indicating a perfectly white Gaussian noise. A threshold could be compared against this detection value to confirm or infirm the presence of signals other than white Gaussian noise. This method did not work successfully as it was found that, after a signal has been extracted, the smallest singular values converge to a zero value. This caused the detection value to always increase and therefore the method, as is, was found to be unusable. Using the spatial correlation matrix instead of the temporal correlation matrix was confirmed to cause the problem because of the way the reference data matrix is produced.

[1]Singular values are assumed to have been ordered in descending order.

To rectify this first technique to use the temporal correlation matrix instead, a single row of $\mathbf{X}_r$ is required. Denoting $x_r(a,b)$ as a single element of $\mathbf{X}_r$ at row a and column b, one may define an augmented detection matrix given by

$$\mathbf{X}_{det} = \begin{bmatrix} x_r(1,1:N_s) \\ x_r(1,2:N_s),\ 0 \\ \vdots \\ x_r(1,p:N_s),\ \mathbf{0}_{1,p-1} \end{bmatrix}. \tag{3.3}$$

This $p \times N_s$ augmented detection matrix is then taken to form the temporal correlation matrix used for detection and given by

$$\hat{\mathbf{R}}_{det} = \mathbf{X}_{det}\mathbf{X}_{det}^H. \tag{3.4}$$

Using the singular values of this matrix, the detection value may now be defined as

$$d_{val} = 10 \cdot \log_{10}\left\{\frac{mean\{\alpha_s(1:p-1)\}}{\alpha_s(p)}\right\} \tag{3.5}$$

where $\alpha_s(i)$ denotes the i^{th} singular value of $\hat{\mathbf{R}}_{det}$. The value of p is chosen to be greater than the number of stages which is normally made equal to the number of sensors M. Therefore, an adequate value for p would be $p = M + 1$. This detection technique was found to behave properly and provide adequate results.

Another way of detecting signal presence was considered with the help of the FFT computation. The technique is summarized here. Given the matrix $\mathbf{X}_r$, an FFT is computed for each row. The mean of the magnitude for each column then forms a vector where each of its element corresponds to a frequency bin. Multiple bins are then averaged to obtain a frequency smoothed spectrum. A detection value could then be defined as

$$d_{val} = 10 \cdot \log_{10}\{v_{max}/v_{min}\} \tag{3.6}$$

where v_{max} and v_{min} represent respectively the maximum and minimum value of the frequency smoothed spectrum vector. With sufficient averaging, white Gaussian

noise will cause a flat spectrum over all frequency while, if a signal is present, some deviation from the minimum to the maximum value found is expected. Although this technique is viable, the SVD technique (with temporal correlation matrix used) was selected for inclusion in S-AMUSE because it is expected to be faster since it works with a limited set of data, and requires less computations. Overall, the SVD method was deemed superior.

3.2.2.2 Initial Signal Estimation

If the presence of signal is detected and it represents a newly detected signal, a wise initialization of the filter weights vector (used later to filter and extract the signal) is highly desirable. This initialization will often make the difference between a faster and a slower convergence time compared to the case where only arbitrary random filter weights are selected. Furthermore, it will allow a particular blind extraction algorithm to lock on a signal that might not have been the case otherwise. This step is therefore an important one. The method proposed for inclusion into the S-AMUSE algorithm uses the First Order Spectral Line Detector (FOSLD) which consists of the following:

- given the reference data block matrix $\mathbf{X}_r \in \mathcal{C}^{M \times N_s}$, compute the FFT of each row;
- average the magnitude of the FFT data over each column to produce a single vector of spectral data;
- and finally detect the bin number, denoted k_{max}, corresponding to the peak value of the spectral data vector (k_{max} could be any value between 0 and $N_s - 1$).

The block diagram of the FOSLD is shown in Fig. 3.2. The value of k_{max} is then used to form a vector

$$\mathbf{d} = e^{j2\pi k_{max}[0:N_s-1]/N_s}, \tag{3.7}$$

a complex sine wave at normalized frequency k_{max}/N_s. Based on this vector, three ways to compute the initial filter weight vector can be set in S-AMUSE.

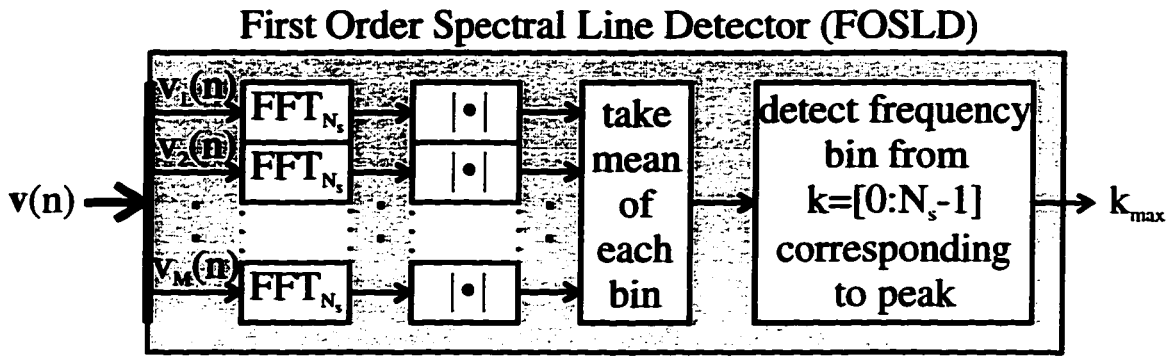

Figure 3.2: Block diagram of the First Order Spectral Line Detector (FOSLD).

The first one is referred to as the "BEAM" technique since it will maximize the power in the direction of the strongest signal found at the detected frequency bin. The "BEAM" initial filter weight vector is given by

$$\mathbf{w}_{l,0} = \hat{\mathbf{R}}_{\mathbf{x}_r\mathbf{d}}\,, \tag{3.8}$$

where

$$\hat{\mathbf{R}}_{\mathbf{x}_r\mathbf{d}} = \mathbf{X}_r\mathbf{d}^H\,. \tag{3.9}$$

The second method is referred to as the "MMSE" method . The "MMSE" filter weight initial vector is given by

$$\mathbf{w}_{l,0} = \hat{\mathbf{R}}^{\dagger}_{\mathbf{x}_r\mathbf{x}_r}\hat{\mathbf{R}}_{\mathbf{x}_r\mathbf{d}} \tag{3.10}$$

where

$$\hat{\mathbf{R}}^{\dagger}_{\mathbf{x}_r\mathbf{x}_r} = (\mathbf{X}_r\mathbf{X}_r)^{\dagger}\,. \tag{3.11}$$

The advantage of the "MMSE" over the "BEAM" method is that it nulls out other signals. The disadvantage is that it might not provide maximum power in the direction of the desired signal. An *had hoc* compromise between the two was found

by the author. The method is referred to as the "NOISY" filter weight initialization method. This method is similar to the "MMSE" method but ensures that maximum power is provided towards the direction of the desired signal (strongest signal at the peak frequency) by injecting some white noise in the data matrix. This is achieved by adding a diagonal matrix with equal valued elements to the spatial correlation matrix $\hat{\mathbf{R}}_{\mathbf{x}_r\mathbf{x}_r}$. The "NOISY" filter weight initial vector is given by

$$\mathbf{w}_{l,0} = \left(\hat{\mathbf{R}}_{\mathbf{x}_r\mathbf{x}_r} + m_f\mathbf{I}_M\right)^{\dagger}\hat{\mathbf{R}}_{\mathbf{x}_r\mathbf{d}}\,. \tag{3.12}$$

The multiplying factor, denoted m_f, provides some flexibility for adjusting the level of noise added. A value of

$$m_f = \sqrt{\text{mean}\{\text{diag}\{\hat{\mathbf{R}}_{\mathbf{x}_r\mathbf{x}_r}\}\}} \tag{3.13}$$

was found experimentally to be suitable. However, it is not claimed to constitute any optimum of any sort. More analysis would be required to determine a more optimum value. The choice of the particular initialization method is one of the algorithm setting.

Once an initial filter weights vector is determined, it is first normalized to $\mathbf{w}^H\mathbf{w} = 1$, i.e.

$$\mathbf{w}_{norm} = \mathbf{w}(\mathbf{w}^H\mathbf{w})^{-1/2}\,, \tag{3.14}$$

and the first signal estimate is given by

$$\hat{\mathbf{s}}_l = \mathbf{w}_{l,0}^H\mathbf{X}_r \tag{3.15}$$

where $\mathbf{w}_{l,0}$ now represents the normalized initial filter weights vector.

For any previously detected signal, the initial filter weights vector is taken from the results of the last block processed for the given signal. In this case, the first signal estimate would be equal to

$$\hat{\mathbf{s}}_l = \mathbf{w}_l^H\mathbf{X}\,. \tag{3.16}$$

where $\mathbf{w}_l$ is the previous block filter weights vector for the l^{th} signal. The filter is applied to the sensor data block matrix and not to the reference data matrix. As a result, only newly detected signals have an initial estimate derived from $\mathbf{X}_r$. The choice of $\mathbf{X}$ over $\mathbf{X}_r$ is justified later. In case the transmitter of this particular signal has stopped transmitting, this data is then checked to verify if the signal is still present (using the detection technique described above). If signal presence is detected, this signal estimate vector represents the initial vector signal estimate.

3.2.2.3 Filter Weights Determination Adaptive Techniques

At each stage of S-AMUSE, a single filter weights vector needs to be adapted. The adaptive technique chosen is partly inspired from the PCCA technique since it also uses a control vector $\mathbf{d}$ but is selected to best fit the data. The adaptive solution is iterative and a criterion for determining when convergence has been obtained is established. Provisions for constraining the output to be orthogonal (which also qualifies to be uncorrelated if signal sources exhibit zero mean) are a possibility.

First, the choice of $\mathbf{d}$ was carefully studied. Two different sets of control vectors were selected. They were found to effectively apply to any type of signals (at least those tested within this dissertation). The first control vector is

$$\mathbf{d} = (\hat{\mathbf{s}}_l^*)^{[o-2]} \otimes \hat{s}_l(n+\tau : n+\tau+N_s-1)^{(*)k_f} e^{j2\pi k_{max}[0:N_s-1]/N_s} \tag{3.17}$$

and the second one,

$$\mathbf{d} = \hat{\mathbf{s}}_l \oslash |\hat{\mathbf{s}}_l| \,. \tag{3.18}$$

They are respectively referred to as the "HOSLD" and "CMA" iteration mode.

The "HOSLD" iteration mode exploits the strongest higher-order spectral line detected in the data using the higher-order spectral line detector (HOSLD) shown in Fig. 3.3. Given a maximum order, denoted o_m, a vector $\mathbf{v}$ chosen to be equal to the initial signal estimate $\hat{\mathbf{s}}_l$, and a selected lag τ (typically chosen to be 0), a

frequency bin number k_{max} corresponding to the peak value of any computed order spectra from 2 to o_m in step of 2 can be detected along with the particular order $o = o_f$. Two types of higher-order spectral line are calculated for each order, the non-conjugate and conjugate higher-order spectral lines corresponding to the conjugate flag $k_f = 0$ and $k_f = 1$ respectively. The HOSLD effectively detects the strongest second and higher-order cycle and conjugate cycle frequency. Since no interpolation is used, it is noted that higher orders will lead to aliasing (with an higher-order cycle frequency at a possibly aliased value) but, nevertheless, the feature will remain usable. Extending this technique to include the use of multiple higher-order spectral lines may be achieved. In this case, great care in choosing features that are related shall be taken. It complicates and increases the amount of computations and it is not clear whether it is necessary. The same applies to the inclusion of multiple values of lag.

On the other hand, the "CMA" iteration mode exploits the constant modulus property of signals. However, it was found to apply to all signal types tested, provided the initial estimate is close enough to make the iterative algorithm lock onto the desired signal. Since the initial signal estimate is provided from either the previous block or the reference data block matrix as explained earlier, no real problem is expected. However, the level of performance is expected to be slightly dependent upon the level of constant modulus present in the signal. To measure the degree of constant modulus property present in the data, the constant modulus degree is defined as

$$\rho = \frac{|\mathbf{d}\hat{\mathbf{s}}^H|^2}{(\mathbf{d}\mathbf{d}^H)(\hat{\mathbf{s}}\hat{\mathbf{s}}^H)} \tag{3.19}$$

for any signal vector $\hat{\mathbf{s}}$ and $\mathbf{d}$ defined as in Eq. (3.18). This measure can take any value between 0 and 1 with 1 indicating a perfectly constant modulus signal. Among the advantages of the "CMA" iteration mode is its expected robustness against multipath.

A third iteration mode, called the "MIX" mode, is a mixture of the "HOSLD"

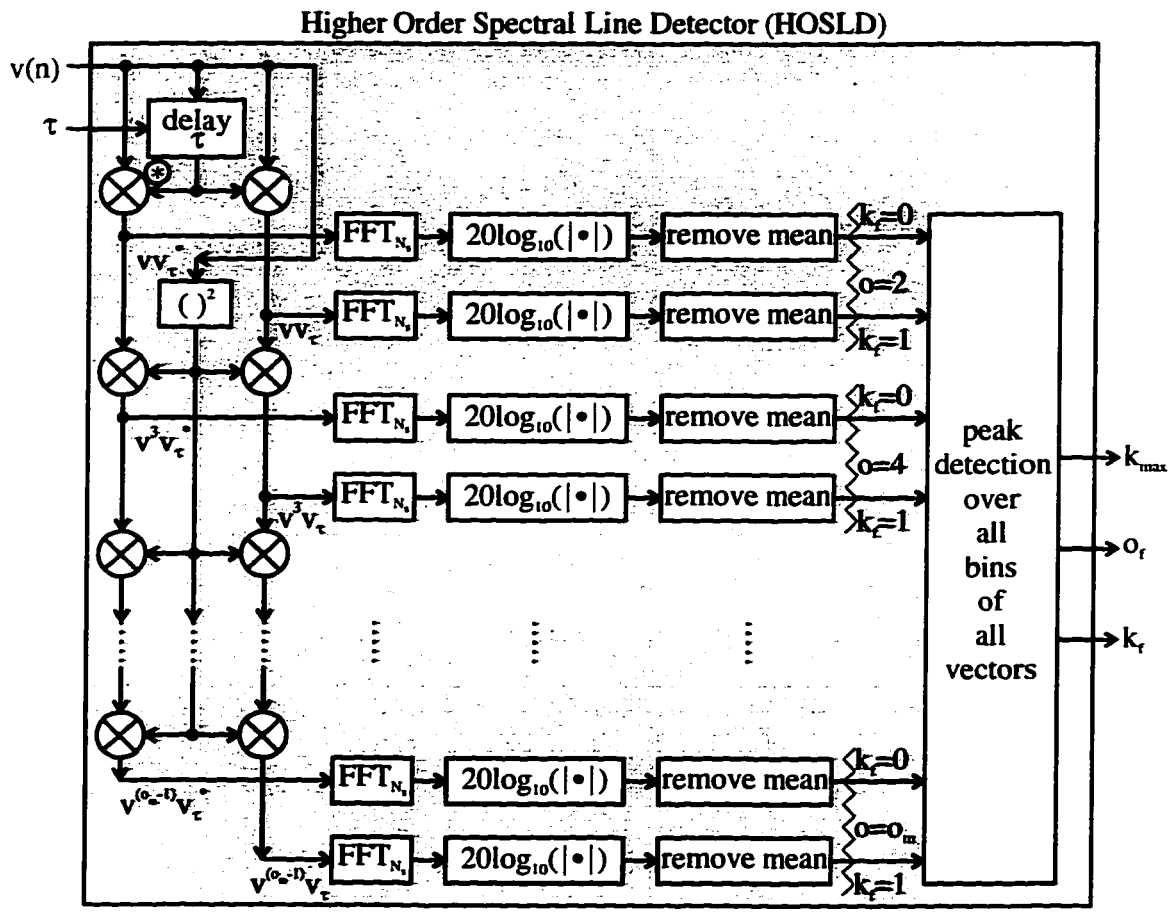

Figure 3.3: Block diagram of the Higher Order Spectral Line Detector (HOSLD).

mode and "CMA" mode. In "MIX" mode, the particular choice of control vector $\mathbf{d}$ depends on the value of ρ. Basically, if a signal exhibits a too low degree of constant modulus such as an AM signal type for example, the "HOSLD" mode control vector would be used[2]. A threshold on ρ, denoted ρ_{th}, could be set to determine a boundary value between the two control vectors. If $\rho < \rho_{th}$, then the "HOSLD" mode control vector would be used while, if $\rho \geq \rho_{th}$, the "CMA" mode control vector would be used. This measure of ρ could be made at each iteration if the starting control vector was determined to be the "HOSLD" mode control vector. If at some point, ρ goes over the threshold, then the control vector would switch to the "CMA" mode. A threshold for the number of iterations is also proposed. When started in "HOSLD" mode, if it does not converge before the number of iterations threshold, the control vector would be switched to the "CMA" mode control vector. Finally, it should be noted that if the mode is switched once, it should be prevented to switch back again for any particular stage and block of data (otherwise it might never converge properly by constantly changing modes).

At each iteration, the control vector produced is used to adapt the filter weights vector (a discussion on the adaptation techniques follows) and compute an updated signal estimate. The updated signal estimate is computed using

$$\hat{\mathbf{s}}_l = \mathbf{w}_l^H \mathbf{X} \tag{3.20}$$

since it avoids the problem of a multi-stage approach (such as CMAR for instance) which are typically dependent on the quality of the signal estimates provided in the previous stage. More precisely, if at any stage the filter weights vector does not produce maximum gain in the desired signal's direction, the reference data matrix is then subject to contain more noise than the original data matrix with respect to the remaining signals. In addition, this choice allows the orthogonal output constraint to

[2]Note that a too low degree of constant modulus might also signify a bad initial signal estimate.

be relaxed (i.e., $\mathbf{s}_i\mathbf{s}_j^H$ for $i \neq j$ is not constrained to be 0).

Four adaptation modes for updating the filter weights are defined in S-AMUSE. They are arbitrarily referred to as:

1. the "BEAM" adaptation mode with

$$\mathbf{w} = \hat{\mathbf{R}}_{\mathbf{xd}}\,; \tag{3.21}$$

2. the "MMSE" adaptation mode with

$$\mathbf{w} = \hat{\mathbf{R}}_{\mathbf{xx}}^{\dagger}\hat{\mathbf{R}}_{\mathbf{xd}}\,; \tag{3.22}$$

3. the "POWM"[3] adaptation mode with

$$\mathbf{w}_{i+1} = \hat{\mathbf{R}}_{\mathbf{xx}}^{\dagger}\hat{\mathbf{R}}_{\mathbf{xd}}\hat{\mathbf{R}}_{\mathbf{dd}}^{\dagger}\hat{\mathbf{R}}_{\mathbf{xd}}^{H}\mathbf{w}_i\,; \tag{3.23}$$

4. and finally the "SVD" adaptation mode with

$$\mathbf{w} = \text{slsv}\left\{\hat{\mathbf{R}}_{\mathbf{xx}}^{\dagger}\hat{\mathbf{R}}_{\mathbf{xd}}\hat{\mathbf{R}}_{\mathbf{dd}}^{\dagger}\hat{\mathbf{R}}_{\mathbf{xd}}^{H}\right\} \tag{3.24}$$

where slsv$\{\cdot\}$ is defined as the stronger left singular vector of its argument.

For all adaptation modes, the filter weights vector must be normalized to $\mathbf{w}^H\mathbf{w} = 1$ at each iteration. The criterion to determine whether the solution has converged enough to stop iterating is established from the differential filter weight measure defined as

$$w_{diff} = \text{mean}\{|\mathbf{w} - \mathbf{w}_o|\} \tag{3.25}$$

where $\mathbf{w}_o$ denotes the old previous iteration filter weights vector. A threshold for this value will be necessary. As well, a maximum number of iterations is required to prevent an infinite iterative loop in case of difficult convergence or too small a threshold.

[3]The name "POWM" was given in reference to the power method iterative solution.

Of all four adaptation modes, the favored one, at present, would be the "POWM" adaptation mode which is issued from the power method PCCA solution (for the case of a single signal to extract). The "SVD" mode (also a PCCA solution) is deemed overly complicated for an iterative method with only a single signal to extract at a time. It is however, equivalent in performance to the "POWM" mode. The "BEAM" mode is not expected to perform too well because it does not null out other signals. Nevertheless, it is left as an alternative to possibly face some particular conditions. Finally, the "MMSE" mode is somewhat similar to "POWM" but will surely exhibit slower convergence. The choice of using the pseudo-inverse instead of a simple matrix inverse is made because pseudo-inverses are more robust.

Although the matrix

$$\hat{\mathbf{R}}_{\mathbf{xd}} = \mathbf{X}\mathbf{d}^H \tag{3.26}$$

is normally used in S-AMUSE,

$$\hat{\mathbf{R}}_{\mathbf{xd}} = \mathbf{X}_r\mathbf{d}^H \tag{3.27}$$

could alternatively be used if it is desired to have orthogonal outputs (which once again, can be qualified as uncorrelated if the signal source means are zero). However, this is expected to result in inferior performance as the signals are in reality independent, and not necessarily orthogonal (as it will result from the orthogonality signal constraint). This is verified by taking any two modulated signals independently generated and computing the correlation coefficient between the two for a finite amount of data. It will never be 0 for any finite amount of data (although it will approach it as the sample size increases). For this reason, trying to constrain the signal outputs to be orthogonal is not the best solution in practical finite data analysis. The S-AMUSE strategy consists of, although making the initial signal estimate (only if newly detected) orthogonal with all previous stage filtered signals, not forcing, in any way, the updated signal estimates to be orthogonal during the iterative adaptation

scheme. The driving force is that each initial orthogonal signal estimate is then expected to best correlate with a different signal in the iterative process. In the case of Eq.(3.27) which is an alternative scheme offered in S-AMUSE, the iterative signal update scheme would ensure uncorrelated outputs.

Each extracted signal is normalized using

$$\hat{\mathbf{s}}_l = \hat{\mathbf{s}}(\hat{\mathbf{s}}\hat{\mathbf{s}}^H/N_s)^{-1/2} \tag{3.28}$$

to result in an averaged signal amplitude normalized to 1.

To update the reference data matrix after a signal has been extracted, the technique is mathematically given by

$$\begin{aligned} \mathbf{s}_r &= \mathbf{w}^H\mathbf{X}_r \\ \mathbf{u} &= \mathbf{X}_r\mathbf{s}_r^H(\mathbf{s}_r\mathbf{s}_r^H)^{-1} \\ \mathbf{X}_r &= \mathbf{X}_r - \mathbf{u}\mathbf{s}_r \end{aligned} \quad . \tag{3.29}$$

3.2.3 Detailed Description

In the following, each block found in the S-AMUSE architecture shown in Fig. 3.1 is given in detail. Since most of the explanations are provided in Section 3.2.2, it mainly consists of detailed block diagrams for each block.

3.2.3.1 Algorithm Settings

The various possible S-AMUSE algorithm settings are shown in Fig. 3.4. Other settings not mentioned in this figure are the block size, denoted N_s, which will affect the level of performance, and the number of stages, L_m, generally assumed to be equal to the number of sensors[4].

[4]It can generally be verified that a spatial beamformer will not be able to extract more signals than sensors. Addition of temporal filtering could change these restrictions however.

Details of the S-AMUSE Algorithm Settings

iteration mode
itermode = [CMA|HOSLD|MIX]
initialization mode
initmode = [BEAM|MMSE|NOISY]
adaptive mode
adaptmode = [BEAM|MMSE|POWM|SVD]
uncorrelated output flag
U_f = [CORR|UNCORR]
maximum HOSLD order
o_m (e.g. $o_m = 4$)
HOSLD mode lag value
τ (e.g. $\tau = 0$)
CMA threshold value for use in MIX mode
ρ_{th} (e.g. $\rho_{th} = 0.75$)
maximum # of iterations
N_{it} (e.g. $N_{it} = 20$ iterations)
iteration differential weight vector threshold
w_{dth} (e.g. w_{dth} = 1e-6)
signal detection threshold
D_{th} (e.g. $D_{th} = 5$ dB)
of iteration threshold value (used in MIX mode)
n_{th} (e.g. $n_{th} = 10$ iterations)

Figure 3.4: Details of the S-AMUSE algorithm setting parameters.

3.2.3.2 Output Settings

Fig. 3.5 shows the details of the S-AMUSE output block settings. Note that the function find{·} is used to denote all vector indices of its vector argument corresponding to non zero valued elements. The operator "!=" specifies logical inequality (instead of $\neq$). One may note that the *first block only initial values* for the data signal estimate matrix and filter weight matrix do not actually matter since they are not used.

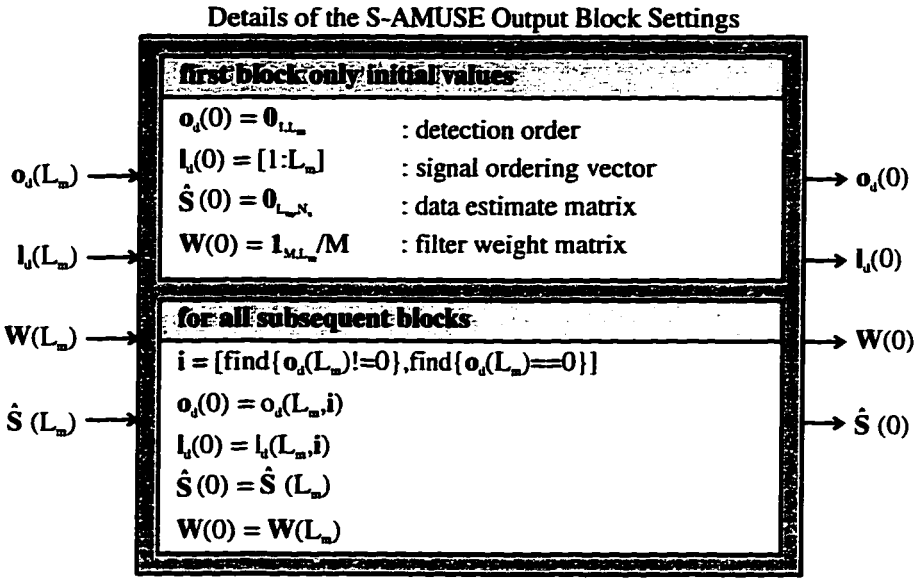

Figure 3.5: Details of the S-AMUSE output block setting parameters.

3.2.3.3 Input Settings

The details of the S-AMUSE input block settings are provided in Fig. 3.6. Perhaps the only detail to mention is the calculation of the pseudo-inverse spatial correlation matrix. This matrix need only to be computed once for all stages.

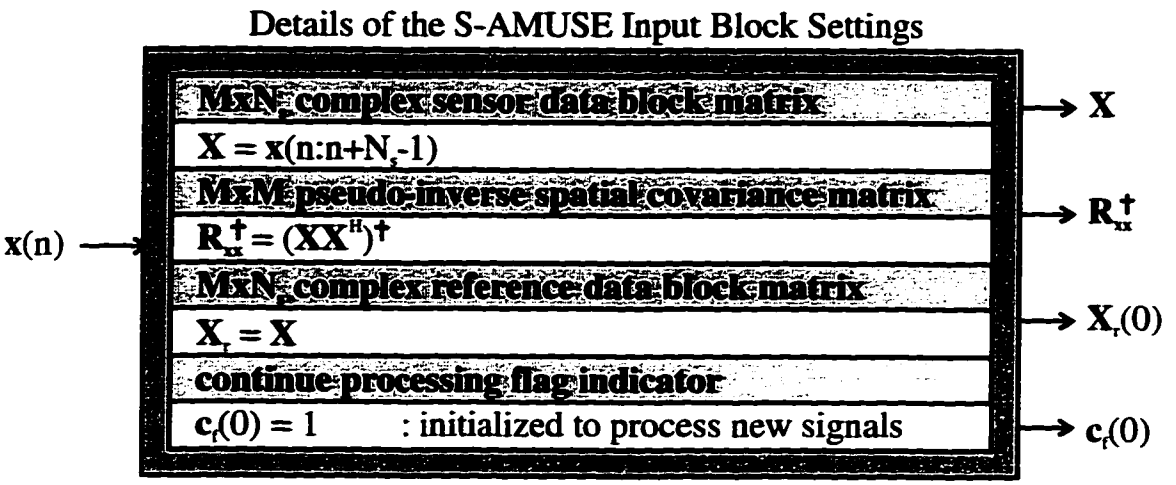

Figure 3.6: Details of the S-AMUSE input block setting parameters.

3.2.3.4 Single Stage Signal Extraction Core Processor

Fig. 3.7 portrays the S-AMUSE single signal extraction core processor in a block diagram. Note that it does not contain all the detailed equations discussed in Section 3.2.2 since it would have resulted in a figure too difficult to follow. A description of the block diagram is presented next.

First, the inputs are conditioned for the needs of the i^{th} stage processor. This means that the i^{th} vector elements of $\mathbf{l}_d(i-1)$ and $\mathbf{o}_d(i-1)$ are retained. They are noted to as $\bar{l}$ and $\bar{o}$ respectively. Also, the $\bar{l}^{th}$ row of $\hat{\mathbf{S}}(i-1)$ and $\bar{l}^{th}$ column of $\mathbf{W}(i-1)$ are respectively denoted as $\bar{\mathbf{s}}$ and $\bar{\mathbf{w}}$.

The flag c_f determines whether to extract a signal or not. If c_f is zero, then no signal is extracted and the "no signal conditions" are set. The "no signal conditions" block effectively sets the signal estimate vector $\bar{\mathbf{s}}$ to zeros, as well as $\bar{o}$ to zero and $\bar{\mathbf{w}}$ to all ones. The block entitled "set outputs" would then be the next and last step since it properly conditions the output data for the next processor stage.

In the case where c_f is not zero, the algorithm tries to detect if a signal is present.

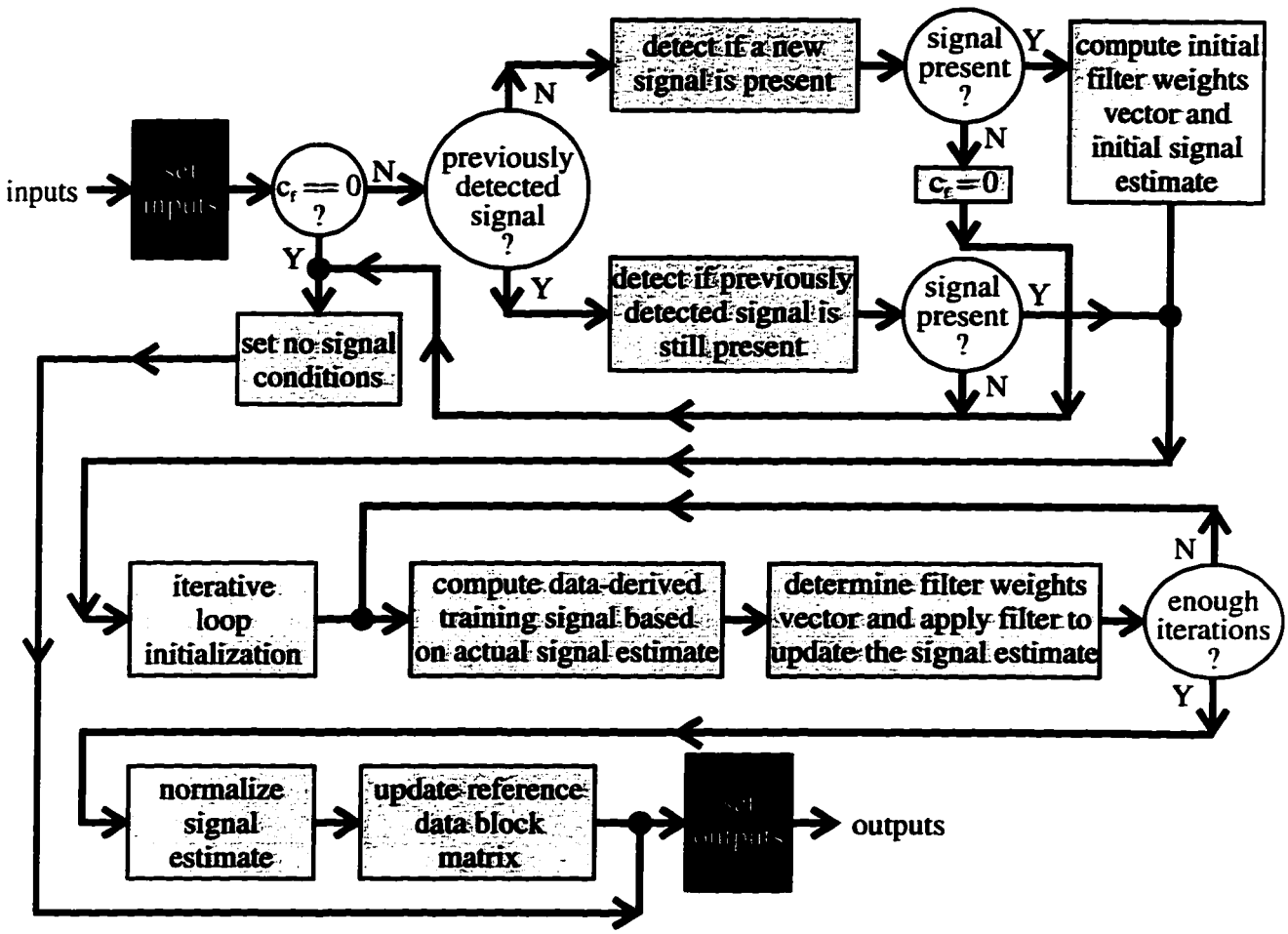

Figure 3.7: Detailed S-AMUSE core processor for single stage i.

The detection, as discussed previously, is different whether it is a known signal (detected in a previous block) or it is a new signal that needs to be detected.

In the case of a previously detected signal (case corresponding to $\bar{o} = 1$), an initial signal estimate is computed by applying the previously used filter weights vector to the actual sensor data matrix, i.e. $\bar{\mathbf{s}} = \bar{\mathbf{w}}^H \mathbf{X}$. Using this initial signal vector estimate, the SVD detection technique is used and produces a detection indicator d_{val} which is compared to the detection threshold value, D_{th}. If $d_{val} \leq D_{th}$, then the signal is declared absent and the algorithm proceeds to the "no signal conditions" and "set output" blocks. If on the opposite the signal is still present, the algorithm continues the processing by entering the iterative loop.

For a new signal to be detected (case corresponding to $\bar{o} = 0$), the computation of d_{val} uses the reference data matrix. The comparison of the resulting d_{val} with D_{th} provides a decision as to whether a new signal is detected or not. If d_{val} is smaller than D_{th} no new signal is detected and the flag c_f is set to 0 in order for the following stage not to continue trying to detect a new signal. In this event, the algorithm goes rapidly to an end by continuing through the "set no signal conditions" and "set outputs" blocks. If however a new signal is detected, $\bar{o}$ is set to 1 and the reference data matrix is used to compute a new filter weights vector and a new signal estimate as was described in 3.2.2.2. The algorithm setting *initmode* will determine the proper scheme to use in order to compute the initial filter weights vector. Once the initial signal vector estimate is computed, the algorithm continues to the iterative loop.

At the beginning of the iterative loop, some parameters are initialized. First, the actual loop number is initialized to 1. Second, depending on the iteration mode *itermode*, the HOSLD may have to be applied to $\bar{\mathbf{s}}$ to obtain o, k_{max}, and k_f which are respectively the resulting order, the bin number, and the conjugate flag corresponding to the peak. Following this initialization, the iterative process consists of computing a data-derived training signal $\mathbf{d}$, determining the filter weights vector $\bar{\mathbf{w}}$, and finally,

updating the signal vector estimate $\overline{\mathbf{s}}$.

Computing the data-derived training signal $\mathbf{d}$ depends on the iteration mode setting (*itermode*). The implication of the three possible choices ("CMA", "HOSLD", and "MIX") is well described in 3.2.2.3.

The filter weights vector is updated based on the data-derived training signal according to the adaptation mode setting *adaptmode*. The algorithm setting *adaptmode* can be any of the four possible settings: "BEAM", "MMSE", "POWM", and "SVD". Another factor affecting the computation of $\overline{\mathbf{w}}$ is the condition of the flag U_f which determines if the extracted signals are correlated or not. Once again, the detailed equations for each possible setting are given in 3.2.2.3. After the filter weights vector is updated and normalized, the signal vector estimate is computed. At this point, the differential filter weight measure w_{diff} is calculated and the loop number incremented by 1.

The next block is a decision block which determines if enough iterations have been achieved. Two criteria are used: the number of iterations (loop number) which are compared to a maximum number of iterations N_{it} (to prevent the algorithm to fall into an infinite loop), and the differential filter weight measure w_{diff} compared to the threshold w_{dth}.

If there has been enough iterations, the signal estimate is normalized, the reference data block matrix $\mathbf{X}_r$ is updated, and the outputs are conditionned in the "set outputs" block. All the outputs are passed to stage $i+1$.

3.3 Additional Remarks

The following are some additional general remarks that can be made about S-AMUSE:

- S-AMUSE primarily refers to its particular architecture;

- a possibly overly high level of flexibility is provided since some alternatives were deemed useful in the analysis and testing process but a number could potentially be eliminated in pratical applications;
- an S-AMUSE algorithm configuration considered typical by the author consists of the "NOISY" initialization mode, the "MIX" iteration mode, and "POWM" adaptive mode settings, with the output signal estimates not constrained to be uncorrelated.

3.4 Applying S-AMUSE to the Example Signal Scenario

In this section, S-AMUSE is applied to the example signal scenario presented in the last section of Chapter 2. Results are produced according to the following algorithm settings:

- iteration mode is set to "MIX";
- initialization mode is set to "NOISY";
- adaptive mode is set to "POWM";
- uncorrelated output flag is set to "CORR";
- maximum number of iterations is set to 30;
- iteration differential weight threshold is set to 1e-8;
- maximum HOSLD order is set to 4;
- HOSLD mode lag value is set to 0;

- CMA threshold value is set to 0.8;
- number of iteration threshold value is set to 10; and finally
- signal detection threshold is set to 5 dB.

The block size was varied from 512 to 16384 in power of 2. Results are shown in Figures 3.8 to 3.14. The main conclusions are:

1. the bigger the block size, the better the performance is;
2. even for the smallest block size chosen, the performance results are excellent for the two first signals which have a stronger degree of constant modulus (the block size did not seem to have much effect in these two cases);
3. in the case of the third signal, the level of performance results is not only worse[5], but also more variable for the 512 samples block size from one block to the other while the performance clearly improves for each subsequent higher block size;
4. since at each block the number of iterations is maximized until it converges sufficiently, no improvement is observed from one block to another, as was expected.

3.5 Coding and Implementation Issues

This section is not intended to provide an in depth analysis of the implementation of S-AMUSE. Rather, the intention is simply to raise a number of issues related to the coding and implementation of S-AMUSE.

[5]The third signal, a STANAG4285 signal, does not have a degree of constant modulus as strong as the other signals. S-AMUSE was setup to favor the "CMA" control vector since a constant modulus degree of more than 0.8 was met even by this third signal.

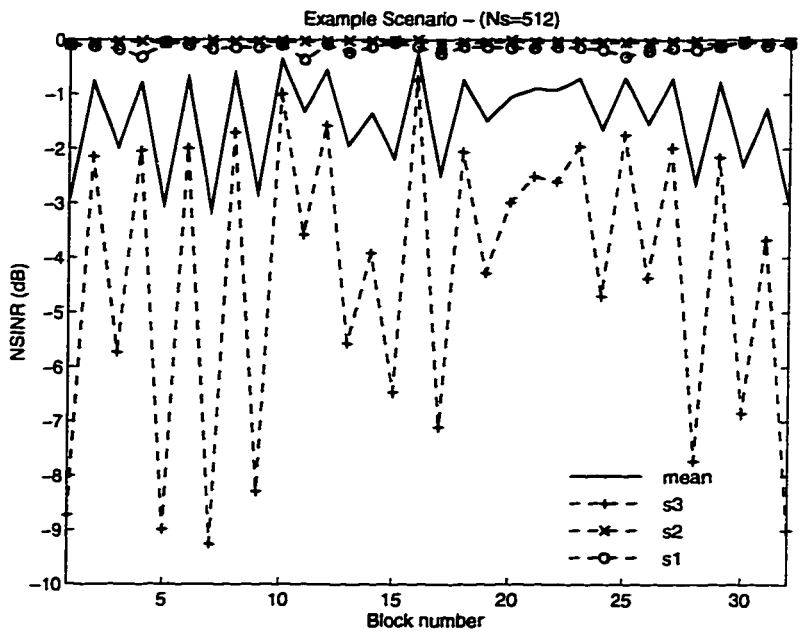

Figure 3.8: NSINR results for the S-AMUSE algorithm applied to the example signal scenario for a block size of 512 samples. Results are plotted against the block number (32 blocks are necessary to cover the entire 16384 snapshots). The three signal types are 4-PSK, FM, and STANAG4285.

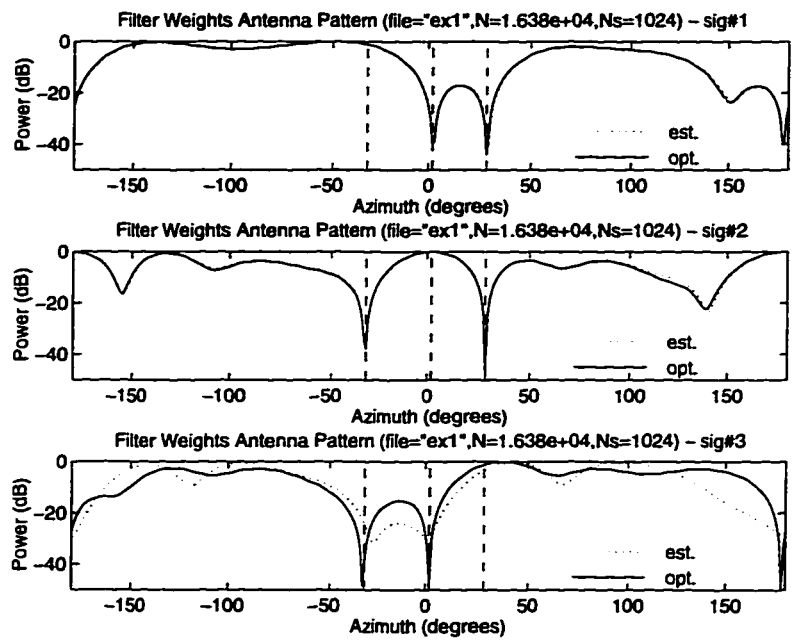

Figure 3.9: Final block filter weights antenna pattern for the S-AMUSE algorithm set to a block size of 1024 samples. The MMSE solution is also shown for comparison purposes. The three signal types are 4-PSK, FM, and STANAG4285. A slightly lower gain in the case of the third signal is observed for the estimated (S-AMUSE) results compared to the optimal (MMSE) results. This translates into a slightly lower NSINR, and correspondingly, lower SINR.

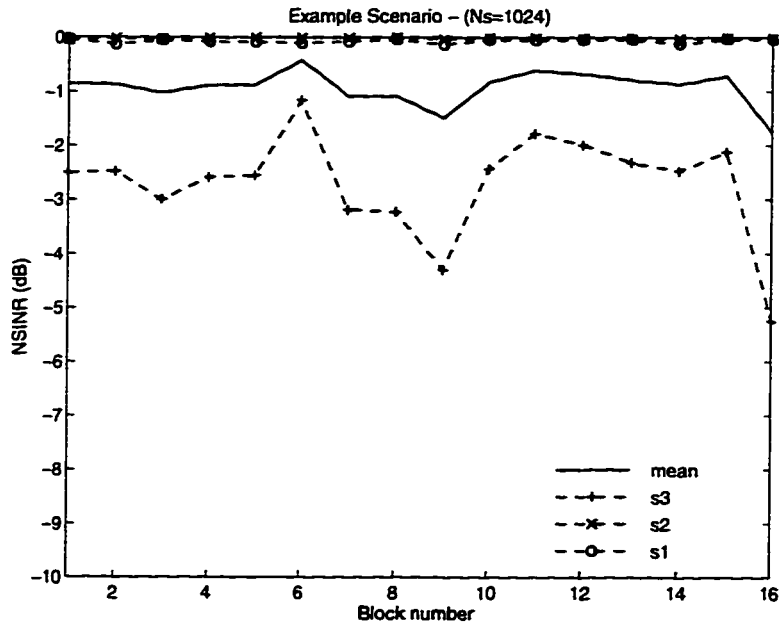

Figure 3.10: NSINR results for the S-AMUSE algorithm applied to the example signal scenario for a block size of 1024 samples. Results are plotted against the block number (16 blocks are necessary to cover the entire 16384 snapshots). The three signal types are 4-PSK, FM, and STANAG4285.

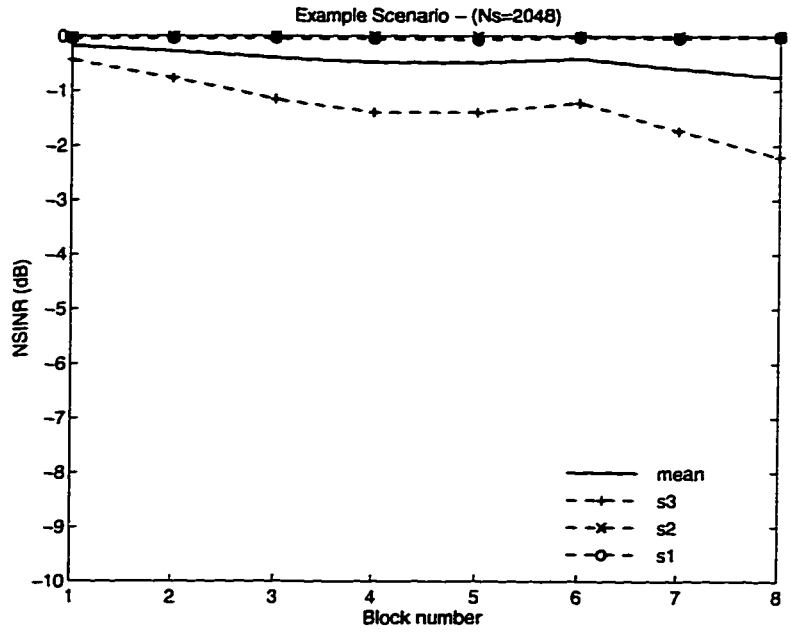

Figure 3.11: NSINR results for the S-AMUSE algorithm applied to the example signal scenario for a block size of 2048 samples. Results are plotted against the block number (8 blocks are necessary to cover the entire 16384 snapshots). The three signal types are 4-PSK, FM, and STANAG4285.

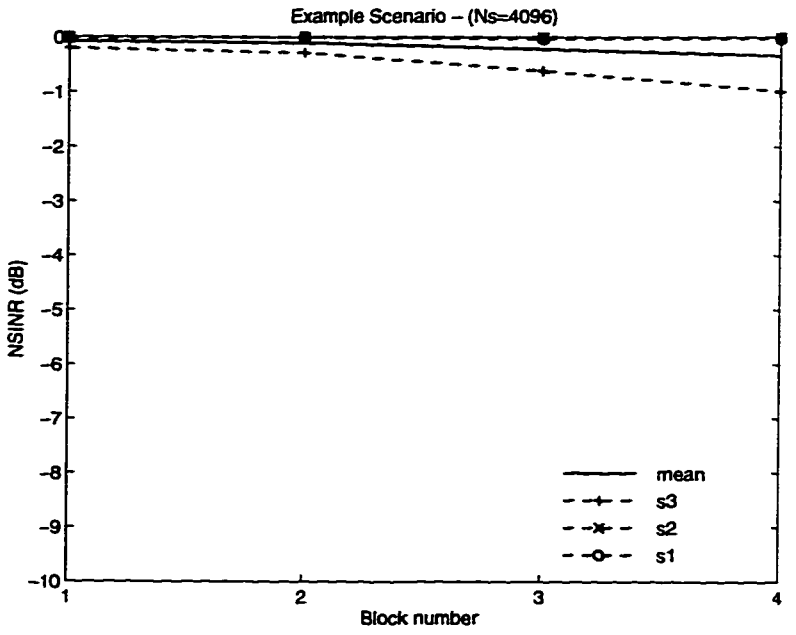

Figure 3.12: NSINR results for the S-AMUSE algorithm applied to the example signal scenario for a block size of 4096 samples. Results are plotted against the block number (4 blocks are necessary to cover the entire 16384 snapshots). The three signal types are 4-PSK, FM, and STANAG4285.

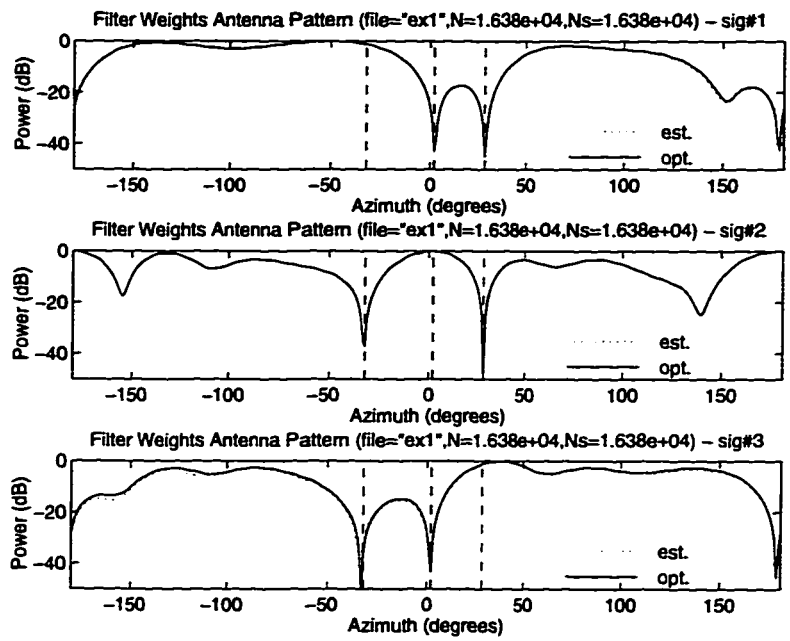

Figure 3.13: Final block filter weights antenna pattern for the S-AMUSE algorithm set to a block size of 16384 samples. The MMSE solution (opt.) is also shown for comparison purposes. The three signal types are 4-PSK, FM, and STANAG4285. In this plot, the estimated (S-AMUSE) results almost equal the optimal (MMSE) results, even in the case of the third signal. Thus, an NSINR of almost 0 for all three signals.

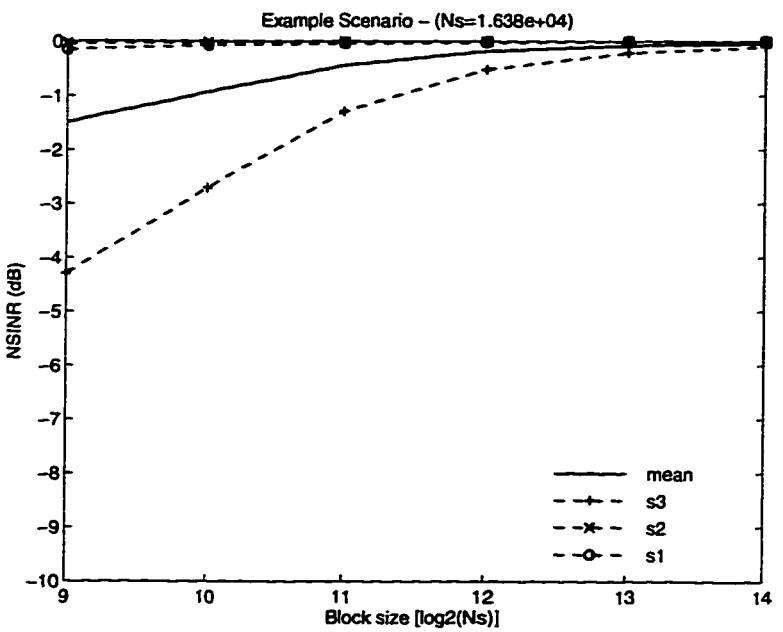

Figure 3.14: NSINR (averaged) results for the S-AMUSE algorithm applied to the example signal scenario as a function of the block size. The three signal types are 4-PSK, FM, and STANAG4285.

A short investigation of the requirements for S-AMUSE indicate that operations must be performed sequentially. However, many of the required operations may be speeded up with the help of parallel processing. The various matrix and vector manipulations and operations, FFT calculations, SVD calculations, and others can all benefit from parallelizing the computations.

In terms of practical implementation, S-AMUSE is likely to take the form indicated in Fig. 3.15 since the processor might not be fast enough to determine the best filter weight data matrix for all consecutive blocks of data. As indicated, in this case, a separate processor for filtering the data in real-time would be added. An appropriate maximum processing time delay would also be used to synchronize the incoming data as best as possible with the estimates. Note that the time taken for S-AMUSE to arrive at the solution would be highly variable. With this setup, data would be filtered continuously with, at worst, some degraded performance when the filter weight matrix update rate is slower and the environment is changing rapidly. This type of setup is common practice in adaptive filtering applications.

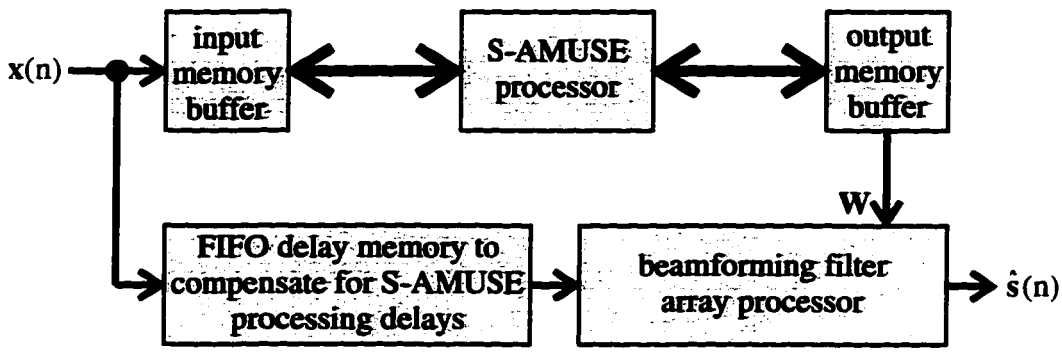

Figure 3.15: Simplified high-level block diagram of a practical implementation of S-AMUSE.

The memory requirements is not as huge as it first appears since the same processors could be used at all stages. This means the amount of memory required would be in the order of the quantity of the data shown to be passed between the various

stages in the overall picture (see Fig. 3.1). More specifically, this quantity of data is $N_s \cdot (2M + L_m) + M \cdot M + L_m \cdot (2 + M) + 1$ scalar values which could be approximated to $N_s \cdot (2M + L_m)$ since, generally, N_s is much greater than M and L_m.

Chapter 4

Monte Carlo Simulations

4.1 Introduction

The Monte Carlo simulations are aimed at evaluating the performance of the S-AMUSE blind signal extraction algorithm presented in the previous chapter by the author. To run the simulations, meaningful signal scenarios are required.

The signal scenarios depend on the signal sources as well as the particular sensor array signal model and its specific conditions. The scenarios have been divided into seven sets in which only a minimum number of parameters were varied for each given set. The objective of this strategy is to investigate more systematically the effect of each parameter variation on the performance results. Specifically, the seven signal sets were selected to study the effects of:

1. Set-A: varying the number of signals;
2. Set-B: varying the signal types;
3. Set-C: varying the differential signal power conditions;
4. Set-D: varying the noise conditions;

5. Set-E: varying the angular spacing between signals;

6. Set-F: varying the multipath conditions;

7. Set-G: varying the array of sensors in shape and number.

More details on each of these are provided in the next section. The results and a discussion of the results obtained are presented in Section 4.3.

4.2 Signal Scenarios

The signal scenarios are firstly based on the use of a selected number and types of signal sources. Once selected, signal sources are then combined and distorted under some assumed sensor array signal model.

This section presents the set of signal sources of which the signal scenarios are composed. The sensor array signal models under considerations are then reviewed. Finally, the seven sets of signal scenarios selected are described in detail.

4.2.1 Signal Sources

In the following, the signal sources are described using the parameters found in Table 4.1. A total of 35 independently generated signal sources were produced based on the seven signal source types presented in Chapter 2 (CW, AM, FM, m-PSK, m-QAM, m-FSK, and S4285). For each signal type numbered from 1 to 7, 5 actual signals were generated with arbitrarily chosen parameters. The details of the signal sources produced are provided in Tables 4.2-4.8. Each table contains one of the seven signal source types. In the tables, the character "-" is present whenever a parameter does not apply to the signal type. For instance, in the case of a CW signal, only the center frequency parameter is relevant and, consequently, all other parameters have

this special character. In addition, the following parameters are common to all signal sources.

1. The sampling frequency selected is 10 kHz.
2. A total of 16384 samples are generated.
3. Low-pass filters are generally made of low-pass digital Butterworth filters of fifth order with the cut-off frequency specifying the 3 dB cut-off point.

Table 4.1: Signal source parameters

	DESCRIPTION
type	Signal modulation type.
f_c	Signal carrier frequency in Hz.
modpar	Modulation parameters (modulation index in fraction if AM modulation and peak frequency deviation if FM or m-FSK).
modsrc	Modulation source type (either analog voice (AUD), analog random noise (RN), random bit stream (RBS), or ASCII text file (ATF)).
srcfilt	Modulation source filter parameters ([cut-off frequency 1, 2] if bandpass filter or a single cut-off frequency if lowpass filter expressed in Hz).
N_{bps}	Number of bits per symbol.
f_{symb}	Symbol rate in Hz.
pshape	Pulse shape (either rectangular (RCT) or Nyquist (NYQ)).
pparam	Pulse parameters if Nyquist pulse shape used ([roll-off, pulse length, pulse shift]).
filter	Overall filter parameters (specified as for "srcfilt").

Table 4.2: Source signal description: CW signals (type 1)

	– CW1 –	– CW2 –	– CW3 –	– CW4 –	– CW5 –
f_c	0.0	-192.0	1503.0	-577.0	264.0
modpar	-	-	-	-	-
modsrc	-	-	-	-	-
srcfilt	-	-	-	-	-
N_{bps}	-	-	-	-	-
f_{symb}	-	-	-	-	-
pshape	-	-	-	-	-
pparam	-	-	-	-	-
filter	-	-	-	-	-

Table 4.3: Source signal description: AM signals (type 2)

	– am_1 –	– am_2 –	– am_3 –	– am_4 –	– am_5 –
f_c	0.0	162.0	-404.0	1764.0	-1364.0
modpar	0.0	0.7	0.9	1.0	0.5
modsrc	RN	RN	RN	AUD_3	AUD_2
srcfilt	[300,3000]	[300,3000]	[300,3000]	[300,3000]	[300,3000]
N_{bps}	-	-	-	-	-
f_{symb}	-	-	-	-	-
pshape	-	-	-	-	-
pparam	-	-	-	-	-
filter	-	-	-	-	-

Table 4.4: Source signal description: FM signals (type 3)

	– fm_1 –	– fm_2 –	– fm_3 –	– fm_4 –	– fm_5 –
f_c	0.0	745.0	321.0	-125.0	-1025.0
modpar	4530	4530	4530	4530	4530
modsrc	RN	AUD_1	RN	AUD_2	RN
srcfilt	[300,3000]	[300,3000]	[300,3000]	[300,3000]	[300,3000]
N_{bps}	-	-	-	-	-
f_{symb}	-	-	-	-	-
pshape	-	-	-	-	-
pparam	-	-	-	-	-
filter	-	-	-	-	-

Table 4.5: Source signal description: m-PSK signals (type 4)

	– $mpsk_1$ –	– $mpsk_2$ –	– $mpsk_3$ –	– $mpsk_4$ –	– $mpsk_5$ –
f_c	0.0	-324.0	-324.0	100.0	-100.0
modpar	-	-	-	-	-
modsrc	RBS	ATF_1	RBS	RBS	ATF_2
srcfilt	-	-	-	-	-
N_{bps}	2	1	2	1	3
f_{symb}	1320	1580	830	1580	1230
pshape	RCT	RCT	RCT	NYQ	RCT
pparam	-	-	-	[1.0,5,0]	-
filter	2420	3200	-	-	2990

Table 4.6: Source signal description: m-QAM signals (type 5)

	– $mqam_1$ –	– $mqam_2$ –	– $mqam_3$ –	– $mqam_4$ –	– $mqam_5$ –
f_c	0.0	887.0	403.0	371.0	-624.0
modpar	-	-	-	-	-
modsrc	RBS	ATF_1	ATF_2	RBS	RBS
srcfilt	-	-	-	-	-
N_{bps}	3	2	4	3	3
f_{symb}	1125	920	830	640	1450
pshape	RCT	RCT	RCT	RCT	NYQ
pparam	-	-	-	-	[0.8,20,0]
filter	2030	1125	1690	1000	-

Table 4.7: Source signal description: m-FSK signals (type 6)

	– $mfsk_1$ –	– $mfsk_2$ –	– $mfsk_3$ –	– $mfsk_4$ –	– $mfsk_5$ –
f_c	0.0	225.0	-1324.0	-790.0	248.0
modpar	4300	4300	4300	4300	4300
modsrc	RBS	ATF_1	RBS	ATF_1	RBS
srcfilt	-	-	-	-	-
N_{bps}	1	1	2	3	1
f_{symb}	1200	980	980	625	780
pshape	RCT	RCT	RCT	RCT	RCT
pparam	-	-	-	-	-
filter	-	[2600]	-	-	-

Table 4.8: Source signal description: S4285 signals (type 7)

	– $s42_1$ –	– $s42_2$ –	– $s42_3$ –	– $s42_4$ –	– $s42_5$ –
f_c	0.0	-220.0	220.0	875.0	-1340.0
modpar	-	-	-	-	-
modsrc	RBS	ATF_1	RBS	RBS	ATF_2
srcfilt	-	-	-	-	-
N_{bps}	2	1	3	2	3
f_{symb}	2400	2400	2400	2400	2400
pshape	NYQ	NYQ	NYQ	NYQ	NYQ
pparam	[0.25,16,0]	[0.25,16,0]	[0.25,16,0]	[0.25,16,0]	[0.25,16,0]
filter	-	-	-	-	-

4.2.2 Sensor Array Signal Model Under Consideration

The choice of sensor array signal model is key to the elaboration of the signal scenarios, and particularly the signal sets defined later. From Chapter 2, expressions to model the propagation of multiple signal sources under various assumptions were provided. Although, in general, the narrowband signal assumption will prevail, a slightly more general expression to accommodate the range of situations of interest is defined next.

The new expression for the $M \times 1$ received sensor output vector is mostly based on the general case expression provided in Eq. (2.32). This expression is repeated here as

$$\mathbf{x}(n) = \sum_{l=1}^{L} g_l \sum_{p=1}^{N_p(l)} \alpha_{l,p}\mathbf{a}(\underline{\theta}_{l,p}, f_o)s_l(n - \tau_{l,p}) + \mathbf{v}(n)\,. \tag{4.1}$$

This is not really appropriate for the case where $\tau_{l,p}$ is not an integer multiple of the sampling period. In Chapter 2, the case for which $\tau_{l,p}$ is smaller than the symbol period (i.e. roughly the inverse of the signal bandwidth) was simplified by the transformation of time delays by phase shifts. Using once again this idea[1], the new sensor array signal model expression is given by

$$\mathbf{x}(n) = \sum_{l=1}^{L} g_l \sum_{p=1}^{N_p(l)} \alpha_{l,p}e^{-j2\pi f_o(\tau_{l,p} - \lfloor \tau_{l,p}/T_s \rfloor \cdot T_s)}\mathbf{a}(\underline{\theta}_{l,p}, f_o)s_l\left(n - \lfloor \tau_{l,p}/T_s \rfloor\right) + \mathbf{v}(n)\,. \tag{4.2}$$

The subpath propagation delay $\tau_{l,p}$ is divided into a first portion that is an integer multiple of the sampling period and a second portion corresponding to the remaining amount of delay, less than T_s. As can be seen in Eq. (4.2), the first portion is applied in terms of sample delay while the second portion is applied as a phase shift transformation. In all signal scenarios, the direct path of each signal is assumed to be the strongest path and corresponds to $p = 1$, or equivalently, $\tau_{l,1} = 0$, $\alpha_{l,1} = 0$ dB. As a result, the case for which $N_p(l)$ is 1 for all l (i.e. in the absence of multipath)

[1] With the difference being that it will apply when $\tau_{l,p}$ is smaller than the sampling period, $T_s = 1/f_s$ (the smallest possible usable signal bandwidth).

reduces effectively to

$$\mathbf{x}(n) = \sum_{l=1}^{L} g_l \mathbf{a}(\underline{\theta}_l, f_o) s_l(n) + \mathbf{v}(n) . \tag{4.3}$$

If multipath is present but $\tau_{l,p} < T_s$ then the sensor array signal model can be reduced to

$$\mathbf{x}(n) = \sum_{l=1} \mathbf{a}_l s_l(n) + \mathbf{v}(n) \tag{4.4}$$

where

$$\mathbf{a}_l = g_l \sum_{p=1}^{N_p(l)} \alpha_{l,p} e^{-j2\pi f_o \tau_{l,p}} \mathbf{a}(\underline{\theta}_{l,p}, f_o) . \tag{4.5}$$

For the sampling frequency of 10 kHz arbitrarily selected during the simulations, all multipath delays below Ts = 100 μsec could be dealt with this last simpler model while for delays greater than 100 μsec, Eq. (4.2) would be more appropriate. An RF of f_o = 8 MHz is chosen for all generated signal scenarios. At this RF, multipath delays of much greater than 100 μsec can often be encountered in practice due to the particular propagation channel conditions at HF [71].

The list of rules common to all the sets of signal scenarios presented in the next section is summarized in the following enumeration.

1. The direct signal path has unity relative gain and is assumed to correspond to the strongest path.

2. The RF, f_o, is 8 MHz.

3. The sampling frequency used is 10 kHz.

4. The number of sensor snapshots generated for each particular signal scenario is 16384.

5. Some parameters are described in terms of statistical distributions. The meaning of the notation used is provided next.

- $D_0\{min, max\}$ is a uniform distribution constrained to the $min \rightarrow max$ range.
- $D_1\{min, max, mean, var\}$ is a normal distribution with mean and variance equal respectively to *mean* and *var* but constrained to the $min \rightarrow max$ range.
- $D_2\{min, max, mean, var\}$ is the square of the normal distribution with mean and variance equal respectively to *mean* and *var*. The random values are constrained to range within the $min \rightarrow max$ range.
- $D_3\{min, max, mean, var\}$ is the absolute of the normal distribution with mean and variance equal respectively to *mean* and *var*. The random values are constrained to range within the $min \rightarrow max$ range.

6. When multipath is present, the longer delay values are always assumed to correspond to stronger attenuations. Furthermore, the values of subpath relative gain with respect to the direct path are always smaller than 1 (or 0 dB) and ordered in descending order, while the values of relative delay with respect to the direct path are always greater than 0 and ordered in ascending order.
7. When multipath is present, the angles of arrival for all subpaths are randomly selected from -180 to 180 degrees (according to a uniform distribution).

The meaning of each parameter used to describe the signal scenarios detailed next is presented in Table 4.9.

4.2.3 Description of the Set of Signal Scenarios Generated

A main signal scenario has been defined. Its description is provided in Table 4.10. Those are fixed parameters identical to the example signal scenario used previously. However, each particular signal scenario set will change one or more parameters. Each

Table 4.9: Signal scenario parameters

	DESCRIPTION
stype	Signal source type number (from 1 to 7).
snum	Signal source signal number (from 1 to 5).
g	Signal power level in relative dB.
θ	Azimuth angle of arrival in degrees (direct path).
N_p	Number of subpaths excluding the direct path.
α_p	Subpaths reflective power ratio in dB relative to the direct path signal.
τ_p	Subpaths time delay in usec.
SNR	Noise level in relative dB.
M	Number of sensors.
senstype	Sensor type (either linear, L-shape, circular, or randomly disposed).
senspace	Sensor spacing in wavelength.

set of signal scenarios will consist of 25 signal scenarios for a grand total of 175 signal scenarios for all seven sets.

Table 4.10: Main signal scenario settings

	SETTING
stype	[4, 3, 7] (i.e. 3 signals corresponding to m-PSK, FM, and S4285)
snum	[1, 3, 2] (refer to m-PSK_1, FM_3, and $S4285_2$ in the signal source tables)
g	[0, 2, 4] dB
θ	[-32, 2, 28] degrees
N_p	[0, 0, 0] (no multipath)
α_p	not applicable
τ_p	not applicable
SNR	15 dB
M	5
senstype	L-shape
senspace	0.5 wavelength

4.2.3.1 Set-A: Varying the Number of Signals

This set is defined to evaluate the effect of varying the number of signals. The differences from the main signal scenario are provided next.

1. The number of signals present is randomly chosen from 1 to 5.

2. The signals sources are randomly chosen from the set of 35 signal sources.

3. The minimum angular spacing is kept to a mean of 30 degrees with a normal variance of 2 degrees.

4. The relative signal power is set to have a fixed difference of 2 dB between the two closest signals in power (i.e. [0, 2, 4,] depending on the number of signals). It always start with a relative signal power of 0 dB such that the minimum SNR is kept to 15 dB.

Note that the number of signals is set not to exceed the number of sensors.

4.2.3.2 Set-B: Varying the Signal Source Types

This set is constructed to evaluate the effect of varying the signal source types. The only departure from the main signal scenario description is that the three source signals are randomly chosen from the set of 35 signal sources amongst the corresponding 7 signal types.

4.2.3.3 Set-C: Varying the Differential Signal Power Conditions

In set-C, the differential signal power is varied. Specifically, the differential power is varied according to a uniform distribution between 0 and 30 dB. Note that the three signals are then set to a relative signal power of $[0, \Delta g, 2\Delta g]$ where Δg specifies the minimum differential power in dB. Since the minimum signal power is always equal to 0 dB, the SNR is kept to 15 dB. All other conditions are identical to the main signal scenario.

4.2.3.4 Set-D: Varying the Noise Conditions

In this set, the SNR is varied according to a uniform distribution between -10 and 90 dB. All other main signal scenario conditions remain the same.

4.2.3.5 Set-E: Varying the Angular Spacing

In this set, the angular spacing is varied somewhat similar to the variation of differential signal power in set C. That is, the angular spacing is varied according to a uniform distribution between 0 and 50 degrees. The selected angular spacing, $\Delta\theta$ then gives rise to angles equal to $[-\Delta\theta, 0, \Delta\theta]$.

4.2.3.6 Set-F: Varying the Multipath Conditions

This set is defined to evaluate the effect ot two different cases of multipath conditions. They are:

- **Case 1:** The number of subpaths is varied from 1 to 12 with a maximum subpath delay smaller than 100 μsec. Specifically, $\tau_{l,p}$ is set according to $D_3\{2, 90, 0, 50\}$. The relative gain is set according to $-D_3\{0, \infty, 0, 1\}$.
- **Case 2:** The number of subpaths is varied from 1 to 12 with a subpath delay allowed to be greater than 100 μsec. Specifically, $\tau_{l,p}$ is set according to $D_3\{2, 1000, 0, 70\}$. Again, the relative gain is set according to $-D_3\{0, \infty, 0, 1\}$.

The last signal scenario of the 25 in this set is simply setup to have no multipath. All other parameters are identical to the main signal scenario conditions.

4.2.3.7 Set-G: Varying the Sensor Arrays

Set-G varies the number of sensors and their geometrical dispositions. To do this, a random shape antenna array is defined. The randomly shape antenna array is generated according to the following:

$$\begin{aligned}\mathbf{P} = \; & 0.5 \cdot [\text{randperm}\{M\} \otimes (\mathbf{1}_{M,1} + 0.2 \cdot \text{randn}\{M, 1)), \\ & \text{randperm}\{M\} \otimes (\mathbf{1}_{M,1} + 0.2 \cdot \text{randn}\{M, 1))]\end{aligned} \tag{4.6}$$

where randperm$\{M\}$ generates a random permutation of the vector $[1 : M]$ and randn$\{M, 1\}$ generates an $M \times 1$ normally distributed with a mean of 0 and variance of 1. The number of sensors, M, is randomly varied according to round $\{D_0\{3.5, 8.49\}\}$. A typical example for $M = 8$ is shown in Fig. 4.1

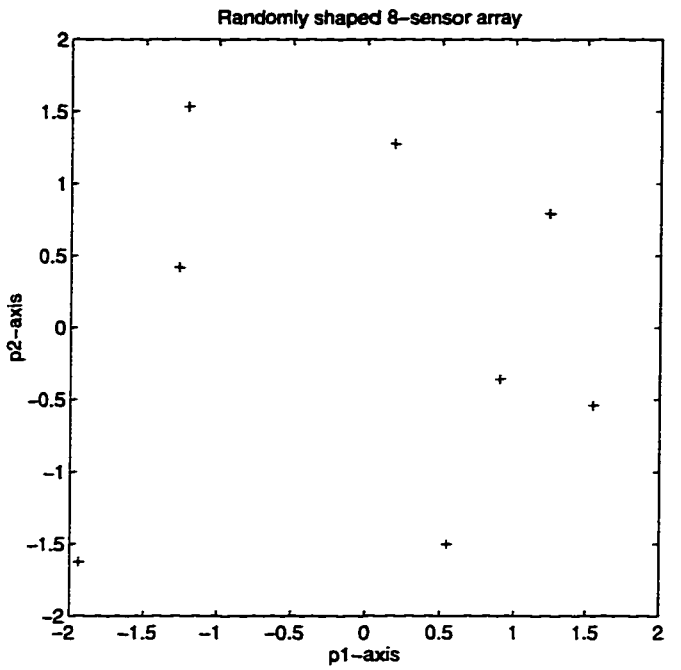

Figure 4.1: Example disposition of a randomly generated antenna array for 8 sensors.

4.3 Results and Discussion

To limit the amount of things to consider in this section, the results reported are for a single typical set of S-AMUSE algorithm settings. The settings are:

- iteration mode is set to "MIX";
- initialization mode is set to "NOISY";
- adaptive mode is set to "POWM";

- uncorrelated output flag is set to "CORR";
- maximum number of iterations is set to 30;
- iteration differential weight threshold is set to 1e-8;
- maximum HOSLD order is set to 4;
- HOSLD mode lag value is set to 0;
- CMA threshold value is set to 0.8;
- number of iteration threshold value is set to 10; and finally
- signal detection threshold is set to 5 dB.

With these settings, the S-AMUSE algorithm has been applied to all 175 signal scenarios contained in the 7 sets described in Section 4.2. The results provided next are for block sizes of $N_s = 16384$ (i.e., the entire sample size for each signal scenario)[2]. The measure of performance selected to present the results is the NSINR, as defined by the author in Chapter 2. Generally, results are plotted against the varying parameters in each set of scenarios. Whenever possible, the results belonging to each signal are shown separately, in which case the mean is then overlaid. The legend specifies the meaning of each of the curves.

For an adequate minimum angular spacing, adding signals did not seem to make a significant difference on the NSINR shown in Fig. 4.2. The resulting NSINR is roughly within 1 dB of the optimum solution in all cases. For the 5 signal results, the spread does not appear to be as wide as for other cases. This is simply because the first (weakest) signal was left undetected and was therefore not extracted. The 5 dB detection threshold was found to have been set too high. A smaller value would

[2]The analysis of shorter sets of data was made in the example provided in Chapter 3.

have been more adequate. Additionally, the detection function used was set to only consider the first 256 samples. More samples might have been more appropriate. Other test results, made by the author and not shown here, have demonstrated the capacity of the algorithm to satisfactorily extract as many signals as sensors.

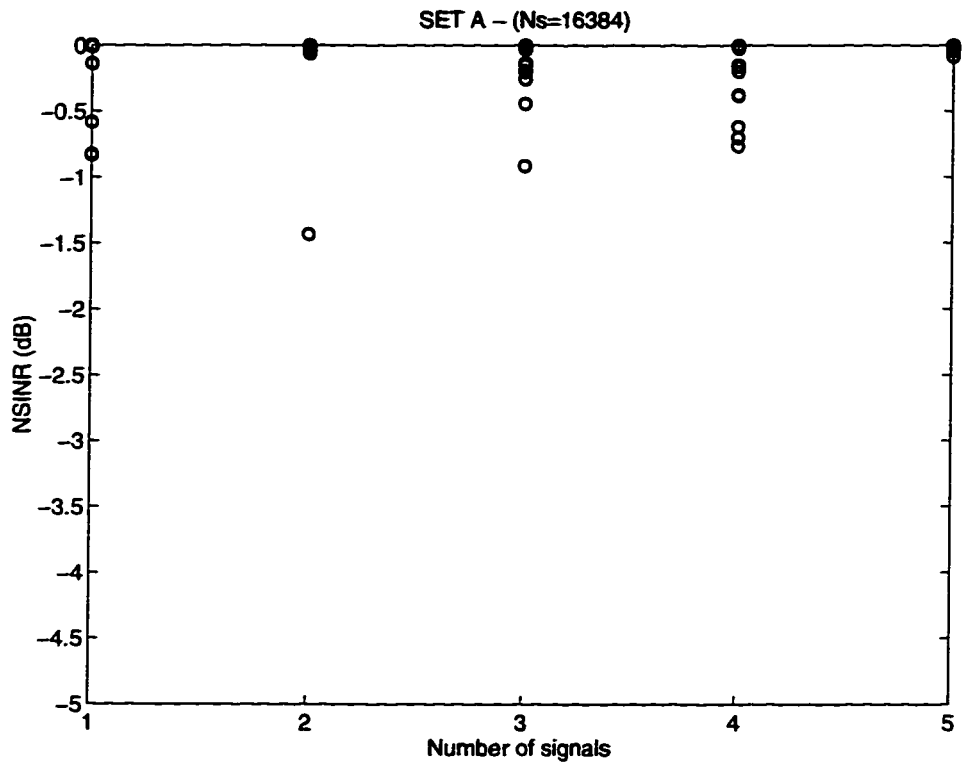

Figure 4.2: NSINR results when varying the number of signals from 1 to 5.

Since S-AMUSE was setup to strongly favor the "CMA" control vector (a value of 0.8 for ρ_{th} will be met for most signal types), the signal types that were found to cause slightly worse performance were the non-constant modulus ones. A high quality estimate was still maintained however even for a non-constant modulus signal. This can be verified in Fig. 4.3. Not too surprising, the constant modulus signals such as CW, FM, and FSK were found to be extracted almost optimally, with NSINR kept in the order of -0.01 dB. The most difficult signal to extract was found to be the

STANAG4285 signal. Nevertheless, the results were still fine.

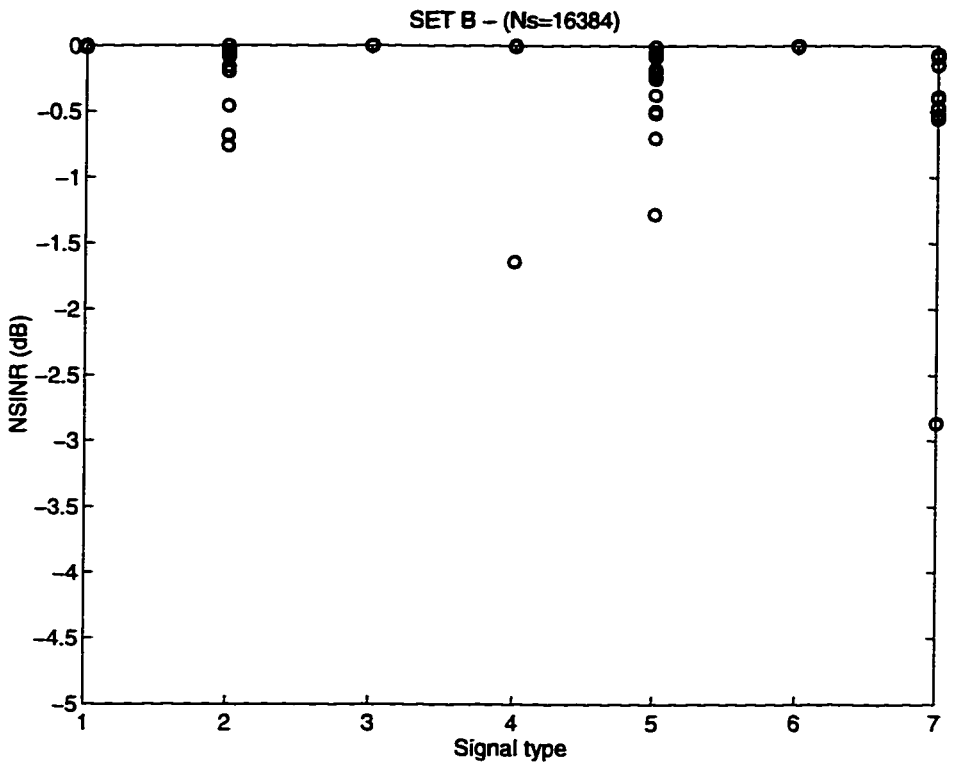

Figure 4.3: NSINR results when varying the signal types among 7 signal types (CW=1, AM=2, FM=3, m-PSK=4, m-QAM=5, m-FSK=6, S4285=7).

The effect of varying the minimum differential relative signal power on the performance results is observed in Fig. 4.4. Unfortunately once again, the detection threshold was set too high, resulting in the weakest signal not being detected for differential power conditions of more than roughly 15 dB. However, more investigations were made to obtain the results. The conclusions are that the differential signal power does not affect the optimality of the solution from the second strongest signal to the weakest but it does affect the strongest. The results found for the strongest signal indicate that the SINR remains approximately constant and even degrades instead of increasing as expected. Analyzing the behaviour of the filter weights antenna pattern shows that the solution for the strongest signal provides adequate interference nulling

but does not result in maximum gain towards the direction of the desired signal. The net result is that the level of noise (which comes from all directions) is then increased, causing the SNR (as well as SINR) to decrease.

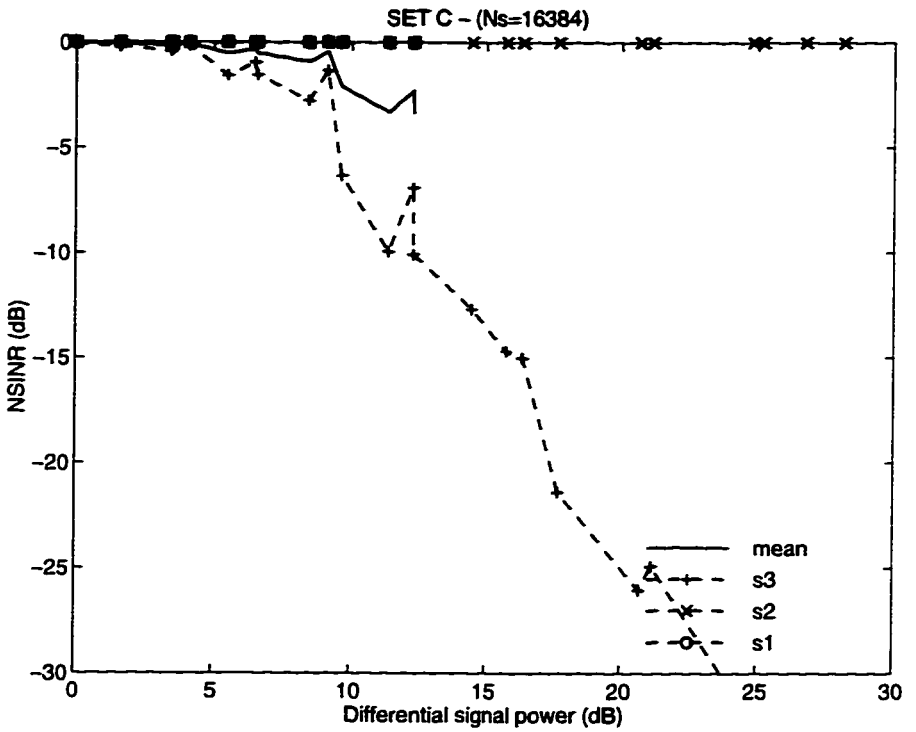

Figure 4.4: NSINR results when varying the minimum differential power from 0 to 30 dB.

The same behaviour is more or less observed in Fig. 4.5 where the minimum SNR is varied. However, in this case, the problem affects both the strongest and weakest signals (with approximately the same slope but a different drop starting point in each case). The reasons for the results are the same as in the precedent case. At present, what seems to be the solution to this problem could well be a technique similar to the "NOISY" initial signal estimate method where noise is injected in the data. However, this is not discussed further here. Note that, due to the particular settings, signals below approximately 5 dB SNR were not detected.

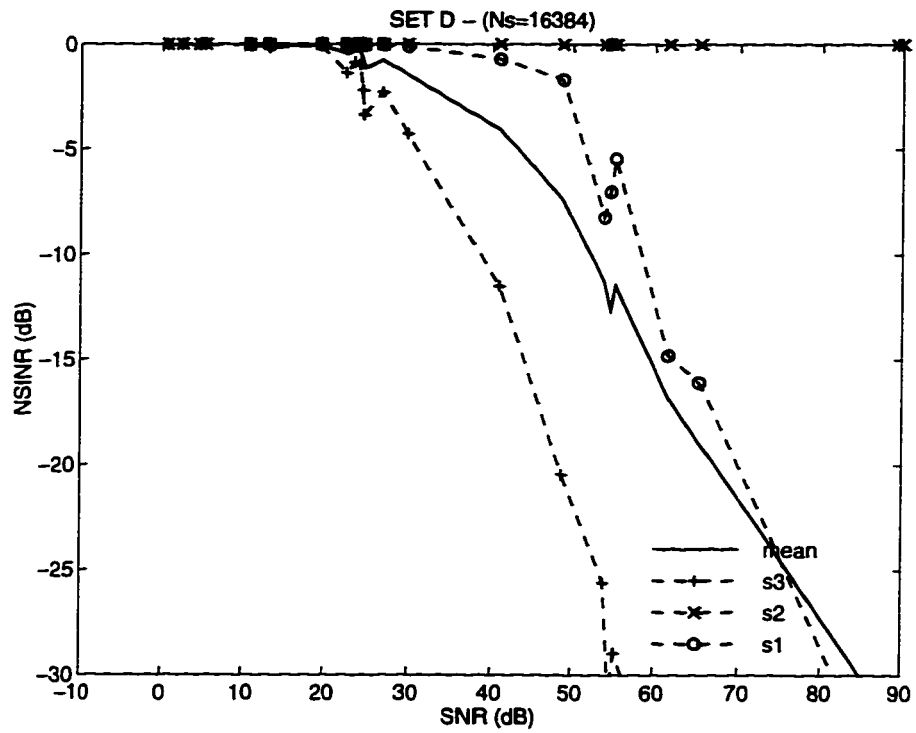

Figure 4.5: NSINR results when varying the minimum SNR from -10 to 90 dB.

Varying the angular spacing is found not to affect the optimality of the solution[3]. The effect of varying the angular spacing on the performance of the algorithm is shown in Fig 4.6. For less than about 12 degrees minimum angular spacing (for this particular antenna array), not all signals are detected since the extraction process leads to removing more than the desired signal. This is due to the physical limitations of the spatial filter. The only way to improve would be to include a temporal filtering dimension. It should be noted that multipath, which is present in practice, results in spatial diversity that may be beneficial to solve the problem. Specifically, if two signals originate from an identical direction but their transmitters are located at different ranges, the multipath will most likely come from different directions. This will have the effect of uncorrelating the resulting steering vectors of the two signals. In essence, this is what the spatial filter requires for separating the signals. The benefit of multipath in a small angular separation situation was verified by the author through additional tests not reported here.

Figures 4.7 and 4.8 show the effect of varying the number of subpaths to the NSINR results. Fig. 4.7 is for the case where all subpath delays are constrained to be smaller than the sampling period while Fig. 4.8 presents the case where delays are allowed to be greater[4]. In the first case, there seems to be a trend with the strongest signal curve which would indicate that increasing the number of subpaths worsens the NSINR. This trend is only observed with the strongest signal since the other two signals remain almost equal to the optimal solution. Another possibility conjectured more likely is that the power levels would have caused this trend since a change in the SNR and differential power level have previously shown to affect the performance. This could only be confirmed with more tests where, for instance, the set of scenarios would be re-constructed with the order of signal types interchanged and everything

[3]Note that it does affect the quality of the estimate however, since small angular spacings lead to poor optimal filters when constrained to simple linear beamformers.

[4]They were not always greater however since they were chosen randomly.

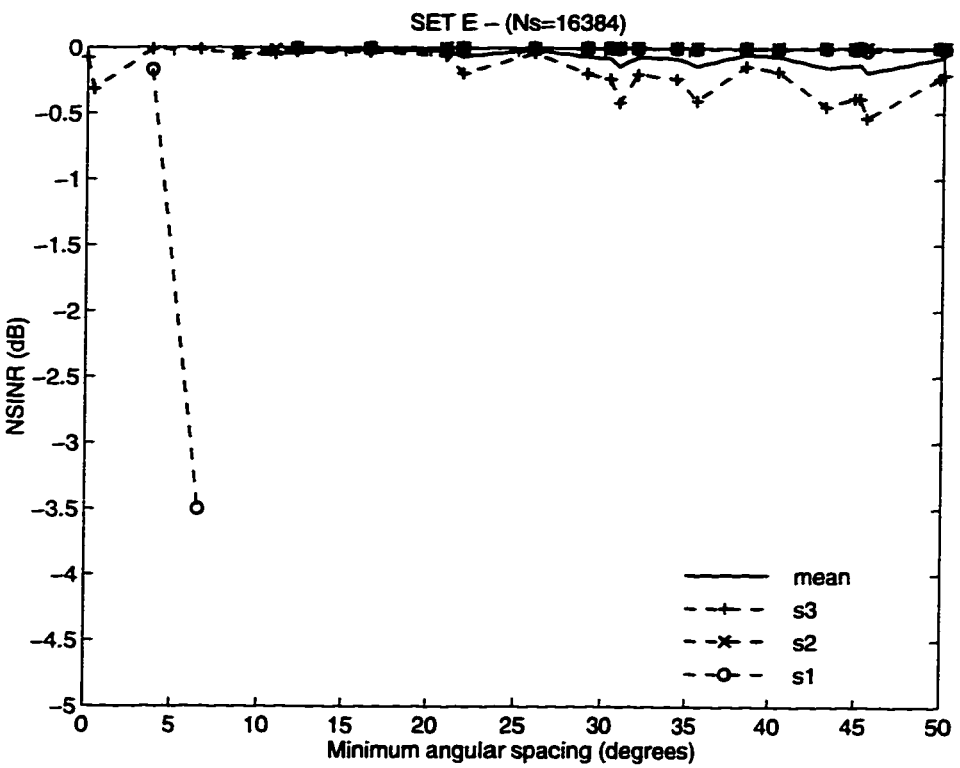

Figure 4.6: NSINR results when varying the angular spacing from 0 to 50 degrees.

else the same. Even without these additional tests, the results are satisfactory enough to conclude that S-AMUSE can be used successfully against multipath of this nature. The second case gives rise to a behaviour which appears even more difficult to interpret. The performance observed is rather erratic, especially for the third signal. This randomness in performance is in part explained by the use of a spatial filter which is not particularly well suited to solve the problem. Looking at the optimum SINR obtained with the MMSE estimate in Fig. 4.9, it shows how the performance vary in no particular order. Note that the selection process of the subpath time delays and relative gains (which was made randomly) has probably also contributed to this unpredictable behaviour. Still, it shows to provide some level of separability giving an SINR of more than 10 dB. This might be sufficient in some applications.

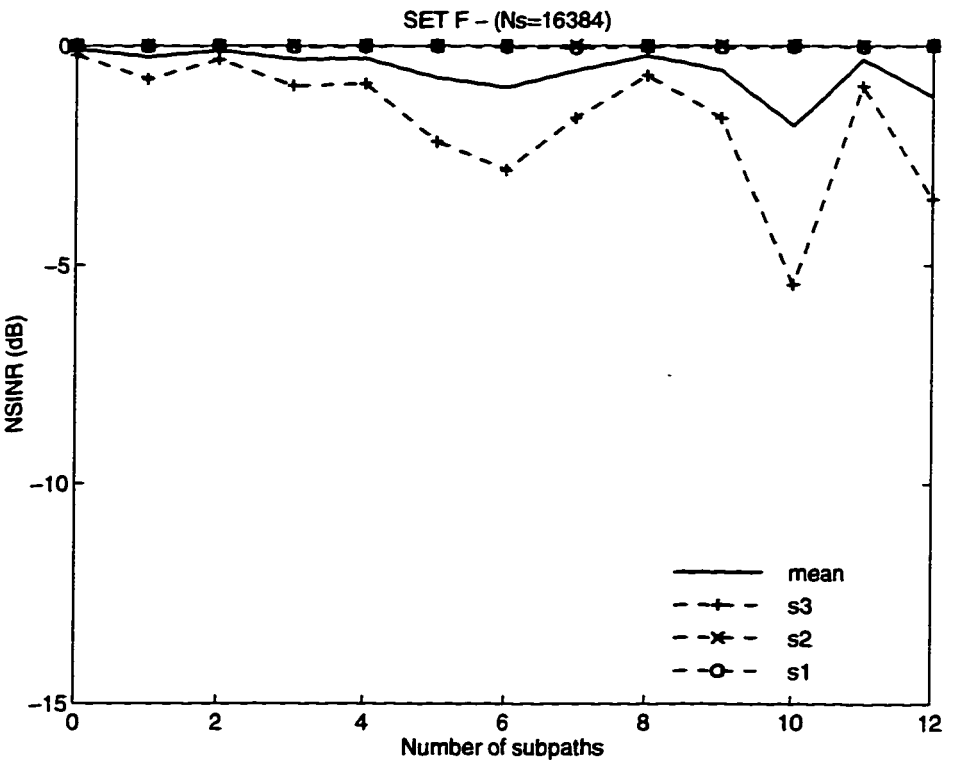

Figure 4.7: NSINR results when varying the multipath conditions from 0 to 12 subpaths. Case 1: subpath delays are smaller than the sampling period.

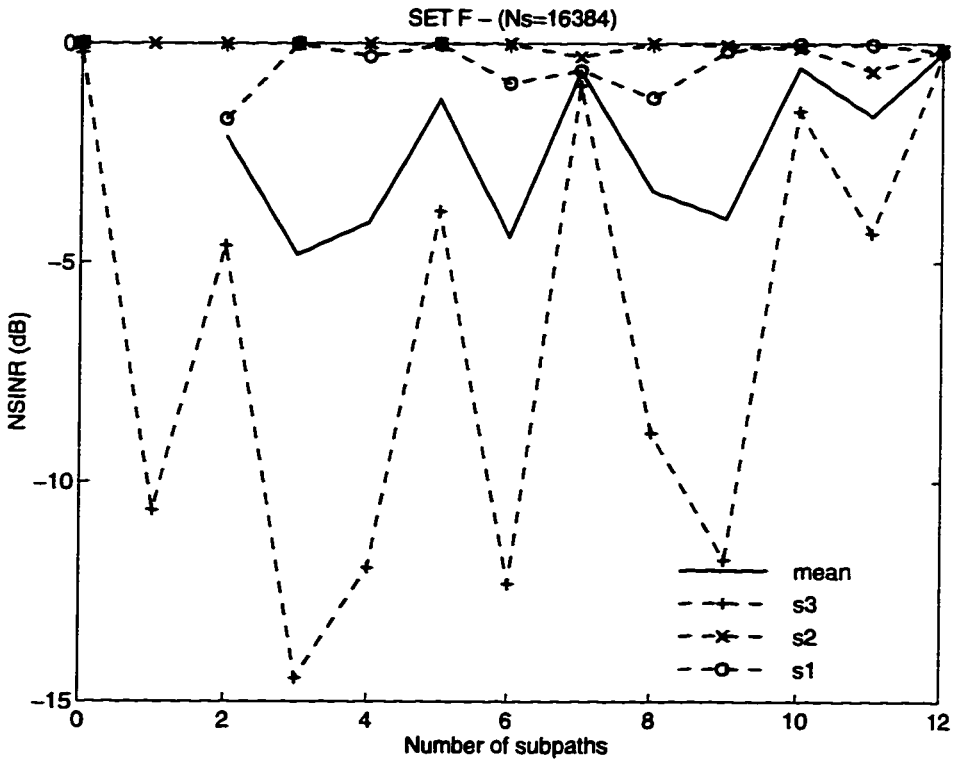

Figure 4.8: NSINR results when varying the multipath conditions from 0 to 12 subpaths. Case 2: subpath delays can be larger than the sampling period.

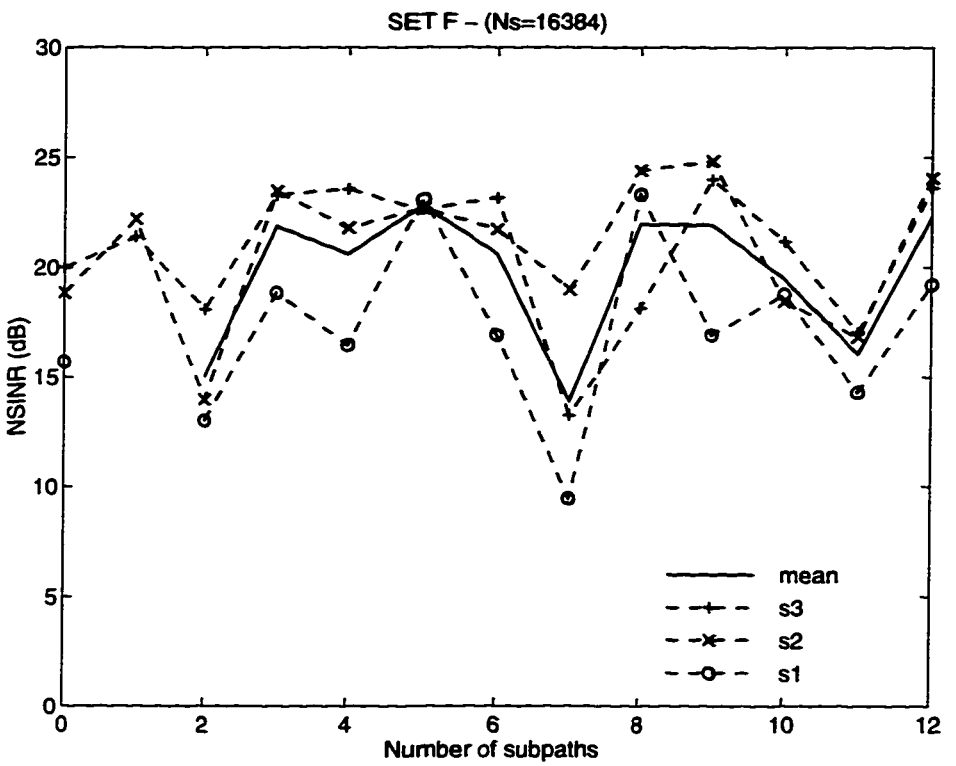

Figure 4.9: Optimal SINR results (using MMSE solution) when varying the multipath conditions from 0 to 12 subpaths. Case 2: subpath delays can be larger than the sampling period.

In the last set of scenarios, the sensor array was varied using a random number of sensors and a random shape. The blind signal extraction performance results are shown in Figures 4.10 and 4.11. The first figure shows the NSINR plotted against the number of sensors, and the second one, against the maximum sensor spacing. Once again, the two weakest signals are shown to be almost unaffected by the variations while the solution for the third (strongest) signal decreases in performance in comparison with the optimal solution by increasing the number of sensors or maximum sensor spacing. In reality, the SINR goes up, but less than for the optimal solution. The real advantage in increasing the number of sensors and spacing is to have a smaller antenna array beamwidth in order to achieve separation of signals that are closer in angular spacing. This does not show up in the results since the angular spacing was kept constant.

Overall, the results confirm that S-AMUSE can be successfully used to achieve blind signal extraction. From the results, the most disturbing behaviour is considered by the author to be the resulting lack of optimality for high SNR signals (at least for the strongest and weakest signals). Some work towards improving this troublesome behaviour has been started by the author along the lines of the "NOISY" initialization method (found in S-AMUSE, as proposed by the author). However, the results have also shown that this behaviour affected only modulated signals which are not truly constant modulus. The tests performed in this dissertation could not confirm whether it was just a coincidence. More work remains to be done to test out S-AMUSE in all possible settings (against an increased range of signal scenarios), to refine the S-AMUSE algorithmic details, and to potentially eliminate some unnecessary levels of flexibility.

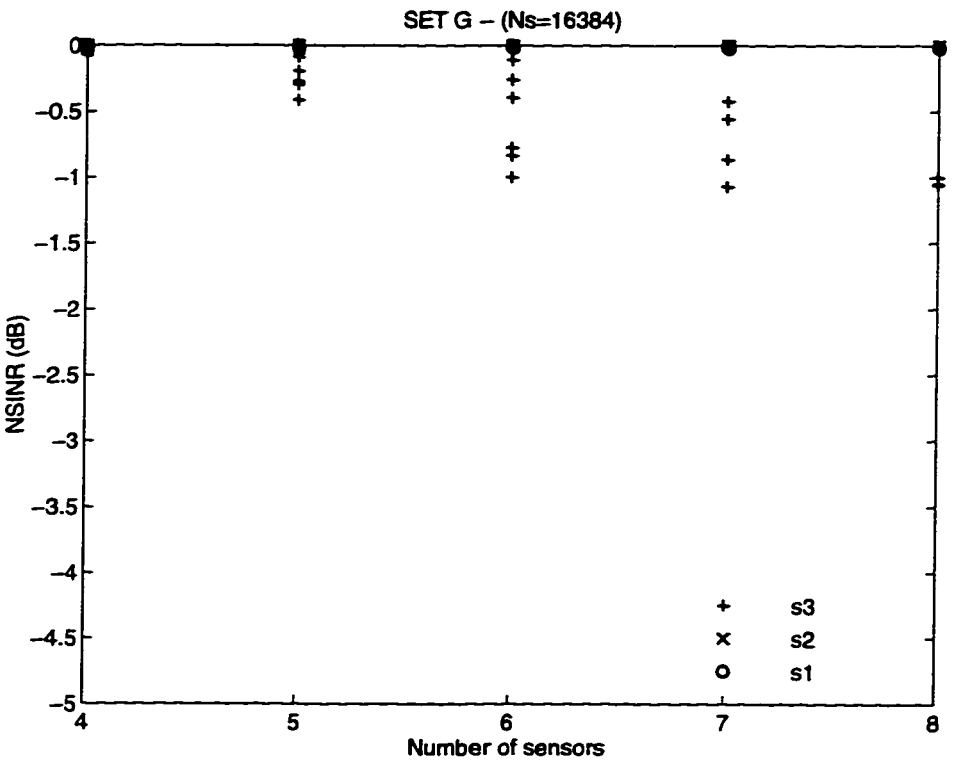

Figure 4.10: NSINR results when varying the number of sensors and array geometry. Results are plotted against the number of sensors.

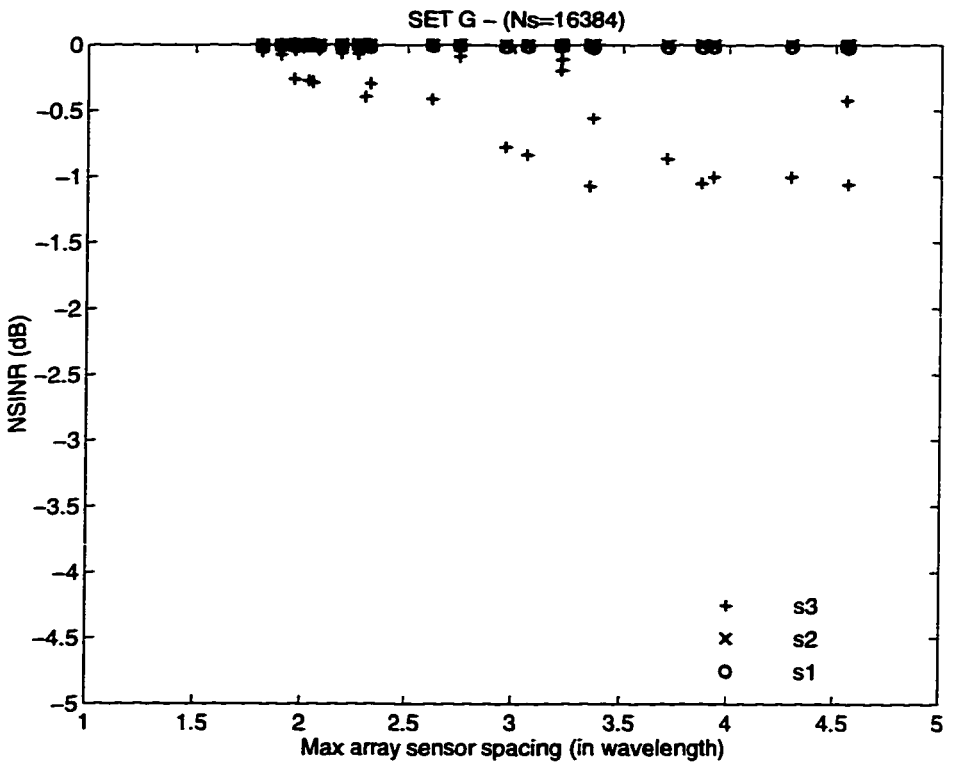

Figure 4.11: NSINR results when varying the number of sensors and array geometry. Results are plotted against the maximum baseline in wavelength.

Chapter 5

Conclusion

5.1 Summary and Concluding Remarks

In this thesis, the problem of blindly separating highly corrupted radio communication signals with the use of a sensor array was examined carefully. The review of existing techniques (including CMA, SCORE, and PCCA-based techniques) has shown that

- sources are generally assumed to be orthogonal (also qualify as uncorrelated if the means are zero);
- some prior knowledge (e.g. cycle frequency) is generally required for the algorithm to work;
- prior assumptions are made on the signal and do not necessarily apply to all types of signals.

In reality, the conditions are that:

- the signals are independent (for non-Gaussian signals, uncorrelatedness is a necessary but not sufficient condition); and

- the prior knowledge requirements may not be available or simply unusable.

To address the drawbacks of the existing techniques, the author has designed and presented the S-AMUSE technique. S-AMUSE blindly adapts a sensor array for extracting all signals present (up to some limit). S-AMUSE is partly inspired from already existing techniques but brings in many new features. The specificity of S-AMUSE lies in its architecture. The S-AMUSE architecture allows:

- multi-signal extraction with the orthogonality signal constraint relaxed;
- a true self-adaptation of the filter weights (no prior knowledge is necessary);
- inherent tracking of signals;
- initialization scheme favoring faster convergence (depending on environment);
- a convergence criterion automatically determining the adequate number of iterations;
- automatic detection of new signals coming up;
- robustness;
- ability to mitigate multipath;
- modularity (due to its multi-stage structure); and
- flexibility for further improvement.

The level of performance achieved throughout the detailed example and Monte Carlo simulations has clearly demonstrated the possibilities of this new technique. The biggest advantage of S-AMUSE over other existing techniques is that it does not require any prior knowledge of the signal environment. It detects and extracts automatically the signals as they come up. Although more work to refine and further test

out S-AMUSE is necessary, a field trial would be beneficial at this point to study the behaviour of S-AMUSE in a practical situation.

5.2 Suggestions for Further Work

The following could be considered for further work related to this thesis work:

1. testing of S-AMUSE with all possible settings, refinements to avoid undesirable behaviour, and elimination of settings that are deemed not viable or not useful;
2. testing of S-AMUSE against real off-air data;
3. investigation of the adaptability of the S-AMUSE structure to temporal and spatio-temporal filtering;
4. in-depth analysis and realization of the S-AMUSE algorithm hardware implementation;
5. definition of new performance measures useful to evaluate blind adaptive signal extraction techniques in practice (usual measures require perfectly known signals to evaluate the level of optimality - this is not available or difficult to obtain in practical tests).

Appendix A

Mathematical Definitions

In the following, mathematical definitions of operators, characters, matrix functions, and elementary matrices are provided. Specifically,

1. Table A.1 defines some special operators and characters used throughout the dissertation.

2. A number of convenient matrix functions are used in this dissertation. The notation and meaning of these functions are summarized in Table A.2.

3. Elementary matrices are defined in Table A.3.

Table A.1: Special operators and characters

	DESCRIPTION
$(\cdot)^T$	Transpose.
$(\cdot)^*$	Conjugate.
$(\cdot)^H$	Conjugate (Hermitian) transpose.
$(\cdot)^{-1}$	Matrix inverse.
$(\cdot)^\dagger$	Matrix pseudo-inverse.
$(\cdot)^{\|\|}$	Range space of a matrix.
$(\cdot)^\perp$	Null space of a matrix.
$(\cdot)^{[p]}$	Element-wise array power (to the power p).
$\otimes$	Element-wise array multiplication.
$\oslash$	Element-wise array division.
$\star$	Convolution.
$\|\| \cdot \|\|$	Matrix or vector norm.
$\langle \cdot \rangle$	Sample mean.
$\| \cdot \|$	Magnitude.
$\angle \cdot$	Phase.
$\lceil \cdot \rceil$	Rounds towards plus infinity.
$\lfloor \cdot \rfloor$	Rounds towards minus infinity.
|	Logical OR.
&	Logical AND.
!	Logical NOT.
==	Logical equality.

Table A.2: Summary of special matrix functions

	DESCRIPTION	
	if argument a vector	if argument a matrix
csum$\{\cdot\}$	Cumulative sum of the vector elements (returns a vector).	Cumulative sum over each column (returns a matrix).
cprod$\{\cdot\}$	Cumulative product of the vector elements (returns a vector).	Cumulative product over each column (returns a matrix).
diag$\{\cdot\}$	Forms a square matrix with diagonal elements equal to the vector elements and all other elements 0 (returns a matrix).	Forms a column vector with the elements equal to the diagonal elements of a matrix (returns a column vector).
eig$\{\cdot\}$	N/A	Eigenvalue decomposition of a square matrix (returns $[\mathbf{U}_e, \mathbf{D}_e]$ such that $(\cdot) = \mathbf{U}_e\mathbf{D}_e\mathbf{U}_e^H$).
fft$\{\cdot\}$	FFT of the vector elements (returns a vector).	FFT over each column (returns a matrix).
ifft$\{\cdot\}$	Inverse FFT of the vector elements (returns a vector).	Inverse FFT over each column (returns a matrix).
max$\{\cdot\}$	Maximum of the vector elements (returns a scalar).	Maximum over each column (returns a row vector).
median$\{\cdot\}$	Median of the vector elements (returns a scalar).	Median over each column (returns a row vector).
min$\{\cdot\}$	Minimum of the vector elements (returns a scalar).	Minimum over each column (returns a row vector).
prod$\{\cdot\}$	Product of the vector elements (returns a scalar).	Product over each column (returns a row vector).
mean$\{\cdot\}$	Mean of the vector elements (returns a scalar).	Mean over each column (returns a row vector).
tr$\{\cdot\}$	N/A.	Trace (sum of all diagonal elements) of a matrix (returns a scalar).
$\text{mod}_N\{\cdot\}$	Modulo N operation (wrapped into the 0 to $N-1$ integer range value) on each vector element (returns a vector).	Modulo N operation on each element of the matrix (returns a matrix).
$\text{resh}_{m,n}\{\cdot\}$	Reshape vector (of $m{\cdot}n$ elements) into an $m \times n$ matrix.	Reshape matrix (of $m \cdot n$ elements) into an $m \times n$ matrix, column-wise.
shift$\{\cdot\}$	Shift the vector by half the elements (returns a vector).	Shift each column by half the elements (returns a matrix).
sort$\{\cdot\}$	Sort the vector elements in ascending order (returns a vector).	Sort the element over each column in ascending order (returns a matrix).
std$\{\cdot\}$	Standard deviation of the vector elements (returns a scalar).	Standard deviation over each column (returns a row vector).
sum$\{\cdot\}$	Sum of the vector elements (returns a scalar).	Sum over each column (returns a row vector).
svd$\{\cdot\}$	Singular value decomposition of the vector (returns $[\mathbf{U}_v, \mathbf{S}_v, \mathbf{V}_v]$ such that $(\cdot) = \mathbf{U}_v\mathbf{S}_v\mathbf{V}_v^H$).	Singular value of the matrix (returns the same as for a vector).

Table A.3: Elementary matrices

	DESCRIPTION
$\mathbf{I}_{r,c}$	An $r \times c$ matrix with ones on the main diagonal.
$\mathbf{I}_n$	An $n \times n$ square matrix with ones on the main diagonal (n-sized identity matrix).
$\mathbf{1}_{r,c}$	An $r \times c$ matrix with all elements defined as 1.
$\mathbf{1}_n$	An $n \times n$ square matrix with all elements defined as 1.
$\mathbf{0}_{r,c}$	An $r \times c$ matrix with all elements defined as 0.
$\mathbf{0}_n$	An $n \times n$ square matrix with all elements defined as 0.

Bibliography

[1] B. G. Agee, "The least-squares CMA: a new technique for rapid correction of constant modulus signals," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 953–956, April 1986.

[2] B. G. Agee, S. V. Schell, and W. A. Gardner, "Self-Coherence Restoral: A new approach to blind adaptation of antenna arrays," in *Proceeding of the 21st Asilomar Conference on Signals, Systems and Computers*, pp. 589–593, 1987.

[3] B. G. Agee, *The property restoral approach to blind adaptive signal extraction.* PhD thesis, University of California, Davis, CA, 1989.

[4] B. G. Agee, "Blind separation and capture of communication signals using a multitarget constant modulus beamformer," in *Proc. IEEE MILCOM 1989*, pp. 19.2.1–19.2.6, April 1989.

[5] B. G. Agee, S. V. Schell, and W. A. Gardner, "Spectral self-coherence restoral: A new approach to blind adaptive signal extraction," *Proceedings of the IEEE*, vol. 78, no. 4, pp. 753–767, April 1990.

[6] B. G. Agee and R. B. Calabretta, "ARMA-like and ML-like Copy/DF approaches for signal-specific emitter location," in *Proceedings of the IEEE Fifth ASSP Workshop on Spectrum Estimation and Modeling*, (Rochester, NY), pp. 134–138, October 10–12 1990.

[7] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis.* New York, NY: John Wiley & Sons, Inc., 1958.

[8] M. Bellanger, *Analyse des signaux et filtrage numérique adaptatif.* Paris: Masson, 1989.

[9] T. E. Biedka and M. F. Kahn, "Methods for constraining a CMA beamformer to extract a cyclostationary signal," in *Proc. 2nd Workshop on Cyclostationary Signals* (S. V. Schell and E. C. M. Spooner, eds.), (Monterey, CA), pp. 19.1–19.10, August 1994.

[10] W. A. Brown, *On the theory of cyclostationary signals.* PhD thesis, University of California, Davis, CA, September 1987.

[11] W. A. Brown and W. A. Gardner, "Frequency-shift filtering theory for adaptive co-channel interference and fading," in *Proceedings of the Twenty-Fourth Annual Asilomar Conference of Signals, Systems, and Computers*, (Pacific Grove, CA), pp. 562–567, October 30–November 1 1989.

[12] J. A. Cadzow, "Signal processing via least squares error modeling," *IEEE Signal Processing Magazine*, pp. 12–31, October 1990.

[13] X. Cao and R. Liu, "General approach to blind source separation," *IEEE Transactions on Signal Processing*, vol. 44, no. 3, pp. 562–571, March 1996.

[14] C.-K. Chen, *Spectral correlation characterization of modulated signals with applications to signal detection and source location.* PhD thesis, University of California, Davis, CA, 1989.

[15] I. G. Clarke, "Blind deconvolution using non-gaussianity as a discriminant," in *Proc. IEEE International Symposium on Higher Order Statistics*, (South Lake Tahoe, CA), June 1993.

[16] W. A. Gardner, *Statistical Spectral Analysis: A Non-probabilistic Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[17] W. A. Gardner, "Signal interception: A unifying theoretical framework for feature detection," *IEEE Transactions on Communications*, vol. COM-36, no. 8, pp. 897–906, August 1988.

[18] W. A. Gardner, *Introduction to Random Processes with Applications to Signals and Systems.* New York: Second Edition, McGraw-Hill, 1990.

[19] W. A. Gardner, "Exploitation of spectral redundancy in cyclostationary signals," *IEEE Signal Processing Magazine*, pp. 4–35, April 1991.

[20] W. A. Gardner and C. M. Spooner, "Signal interception: Performance advantages of cyclic-feature detectors," *IEEE Transactions on Communications*, vol. 40, no. 1, pp. 149–159, January 1992.

[21] W. A. Gardner, "A unifying view of coherence in signal processing," *Signal Processing (EURASIP)*, vol. 29, no. 2, pp. 113–140, November 1992.

[22] W. A. Gardner, "Cyclic Wiener filtering: Theory and method," *IEEE Transactions on Communications*, vol. 41, no. 1, pp. 151–163, January 1993.

[23] W. A. Gardner, Editor, *Cyclostationarity in Communications and Signal Processing.* New York: IEEE Press, 1994.

[24] W. A. Gardner and C. M. Spooner, "Cyclostationary signal processing," in *Stochastic Techniques in Digital Signal Processing Systems (Part 2 of 2)* (C. T. Leondes, ed.), (San Diego, CA), pp. 1–92, 1994.

[25] R. Gooch and J. Lundell, "The CM array: An adaptive beamformer for constant modulus signals," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 2523–2526, April 1986.

[26] G. H. Golub and C. F. Van Loan, *Matrix Computations, 2nd edition.* Baltimore: John Hopkins University Press, 1989.

[27] S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1986.

[28] S. Haykin, *Adaptive Filter Theory – 2nd ed.* Englewood Cliffs, NJ: Prentice-Hall, 1991.

[29] H. Hotelling, "Relation between two sets of variables," *Biometrika*, pp. 321–377, 1936.

[30] M. F. Kahn, M. A. Mow, W. A. Gardner, and T. E. Biedka, "A recursive programmable canonical correlation analyzer," in *Proc. 2nd Workshop on Cyclostationary Signals* (S. V. Schell and E. C. M. Spooner, eds.), (Monterey, CA), pp. 18.1–18.12, August 1994.

[31] S. M. Kay, *Modern Spectral Estimation: Theory and Application.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[32] E. Kreyszig, *Advanced Engineering Mathematics – 5th ed.* New York, NY: John Wiley & Sons, Inc., 1983.

[33] B. P. Lathi, *Modern Digital and Analog Communications Systems.* New York, NY: HRW, 1983.

[34] W. A. Brown and H. H. Loomis, Jr., "Digital implementations of spectral correlation analyzers," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 703–720, February 1993.

[35] , *MATLAB User's Guide.* Natick, MA: The MathWorks Inc., 1992.

[36] , *MATLAB Reference Guide.* Natick, MA: The MathWorks Inc., 1992.

[37] , *Signal Processing Toolbox User's Guide.* Natick, MA: The MathWorks Inc., 1994.

[38] S. L. Marple, Jr, *Digital Spectral Analysis with Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[39] F. McCarthy, "Multiple signal direction-finding and interference reduction techniques," in *WESCON/93 Conf. Rec.*, (San Francisco, CA), pp. 354–361, September 1993.

[40] C. L. Nikias and J. M. Mendel, "Signal processing with higher-order spectra," *IEEE Signal Processing Magazine*, pp. 10–37, July 1993.

[41] B. Noble and J. W. Daniel, *Applied Linear Algebra – 3rd ed.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[42] A. Papoulis, *Probability, Random Variables, and Stochastic Processes – 2nd ed.* New York, NY: McGraw-Hill, 1984.

[43] J. G. Proakis, *Digital Communications – 2nd ed.* New York, NY: McGraw-Hill, 1989.

[44] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[45] R. S. Roberts, *Architectures for digital cyclic spectral analysis.* PhD thesis, University of California, Davis, CA, September 1989.

[46] R. S. Roberts, W. A. Brown and H. H. Loomis, Jr., "Computationally efficient algorithms for cyclic spectral analysis," *IEEE Signal Processing Magazine*, pp. 38–49, April 1991.

[47] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time-Series Analysis.* Reading, MA: Addison-Wesley, 1990.

[48] S. V. Schell and B. G. Agee, "Application of the SCORE algorithm and SCORE extensions to sorting in the rank-l spectral self-coherence environment," in *Proc. of the 22nd Asilomar Conf. on Signals, Systems, and Computers*, pp. 274–278, 1988.

[49] S. V. Schell, R. A. Calabretta, W. A. Gardner and B. G. Agee, "Cyclic MUSIC algorithms for signal-selective DOA estimation," in *Proceedings ICASSP*, (Glasgow, Scotland), pp. 2278–2281, May 1989.

[50] S. V. Schell and W. A. Gardner, "Signal-selective high-resolution direction finding in multipath," in *International Conference on Acoustics, Speech, and Signal Processing* (Albuquerque, ed.), pp. 2667–2670, April 1990.

[51] S. V. Schell and W. A. Gardner, "Progress on signal-selective direction finding," in *Proceedings of the IEEE Fifth ASSP Workshop on Spectrum Estimation and Modeling*, (Rochester, NY), pp. 144–148, October 10–12 1990.

[52] S. V. Schell, *Exploitation of spectral correlation for signal-selective direction finding.* PhD thesis, University of California, Davis, CA, December 1990.

[53] S. V. Schell and W. A. Gardner, "Maximum likelihood and common factor analysis-based blind adaptive spatial filtering for cyclostationary signals," in *Proc. IEEE International Conference on Acoutics, Speech, and Signal Processing*, (Minneapolis, MN), pp. IV–292–IV–295, 1993.

[54] S. V. Schell and W. A. Gardner, "Spatio-temporal filtering and equalization," in *Stochastic techniques in digital signal processing systems (Part 1 of 2)* (C. T. Leondes, ed.), (San Diego, CA), pp. 1–68, 1994.

[55] S. V. Schell and W. A. Gardner, "Programmable canonical correlation analysis: A flexible framework for blind adaptive spatial filtering," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 2898–2908, December 1995.

[56] J. L. Schenck and W. A. Gardner, "Blind adaptive spatial processing in a mobile radio environment," in *Proc. 2nd Workshop on Cyclostationary Signals* (S. V. Schell and E. C. M. Spooner, eds.), (Monterey, CA), pp. 17.1–17.7, August 1994.

[57] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, pp. 14–37, January 1992.

[58] J. J. Shynk and R. P. Gooch, "The constant modulus array for cochannel signal copy and direction finding," *IEEE Transactions on Signal Processing*, vol. 44, no. 3, pp. 652–660, March 1996.

[59] S. Sivanand, J. Yang, and M. Kaveh, "Focusing filters for wide-band direction finding," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 437–445, February 1991.

[60] M. R. Spiegel, *Formules et tables mathématiques - 12ème éd.* Paris: McGraw-Hill, 1986.

[61] C. M. Spooner, *Theory and application of higher-order cyclostationarity.* PhD thesis, University of California, Davis, CA, June 1992.

[62] Military Agency for Standardization (MAS), *NATO STANAG 4285 (2nd draft, ed. 1): Characteristics of 1200/2400/3600 Bits per Second Single Tone Modulators/Demodulators for HF Radio Links.* 1990.

[63] S. Stein, "Fading channel issues in system engineering," *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, no. 2, pp. 68–89, February 1987.

[64] S. Talwar, M. Viberg, and A. Paulraj, "Blind separation of synchronous co-channel digital signals using an antenna array – Part I: Algorithms," *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1184–1197, May 1996.

[65] L. Tong, G. Xu, and T. Kailath, "Fast blind equalization via antenna arrays," in *Proc. IEEE ICASSP*, pp. IV–272–IV–279, 1993.

[66] J. R. Treichler, and B. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, no. 2, pp. 459–472, April 1983.

[67] J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore, *Theory and Design of Adaptive Filters*. New York, NY: John Wiley & Sons, Inc., 1987.

[68] J. R. Treichler, I. Fijalkow, and C. R. Johnson, Jr., "Fractionally-spaced equalizers," *IEEE Signal Processing Magazine*, pp. 65–81, May 1996.

[69] M. Viberg, B. Otterson, and T. Kailath, "Detection and estimation in sensor arrays using weighted subspace fitting," *IEEE Transactions on Signal Processing*, vol. 39, no. 11, pp. 2436–2449, November 1991.

[70] A. J. van der Veen and A. Paulraj, "An analytic constant modulus algorithm," *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1136–1155, May 1996.

[71] C. C. Watterson, J. R. Juroshek, and W. D. Bensema, "Experimental confirmation of an HF model," *IEEE Transactions on Communication Technology*, vol. COM-18, no. 6, pp. 702–803, December 1970.

[72] G. Xu and T. Kailath, "A new array signal processing method via exploitation of cyclostationarity," in *Proceedings of the IEEE Fifth ASSP Workshop on*

Spectrum Estimation and Modeling, (Rochester, NY), pp. 94–98, October 10–12 1990.

[73] G. K. Yeung and W. A. Gardner, "Blind-adaptive linear frequency-shift filtering," in *Proc. 2nd Workshop on Cyclostationary Signals* (S. V. Schell and E. C. M. Spooner, eds.), (Monterey, CA), pp. 14.1–14.8, August 1994.

[74] G. D. Zivanovic and W. A. Gardner, "Degrees of cyclostationarity and their applications to signal detection and estimation," *Signal Processing*, vol. 22, pp. 287–297, April 1991.