Department of Electrical and Computer Engineering

Adaptive Antenna Array Beamforming Using A Concatenation of Recursive Least Square and Least Mean Square Algorithms

Jalal Abdulsayed Srar

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

June 2011

DECLARATION

To the best of my knowledge and belief, this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for award of any other degree or diploma in any university.

Signature: ……

Date: ……..

Dedicated to my Parents, my Family and my Wife for their support and affection

ABSTRACT

In recent years, adaptive or smart antennas have become a key component for various wireless applications, such as radar, sonar and cellular mobile communications including worldwide interoperability for microwave access (WiMAX). They lead to an increase in the detection range of radar and sonar systems, and the capacity of mobile radio communication systems. These antennas are used as spatial filters for receiving the desired signals coming from a specific direction or directions, while minimizing the reception of unwanted signals emanating from other directions.

Because of its simplicity and robustness, the LMS algorithm has become one of the most popular adaptive signal processing techniques adopted in many applications, including antenna array beamforming. Over the last three decades, several improvements have been proposed to speed up the convergence of the LMS algorithm. These include the normalized-LMS (NLMS), variable-length LMS algorithm, transform domain algorithms, and more recently the constrained-stability LMS (CSLMS) algorithm and modified robust variable step size LMS (MRVSS) algorithm. Yet another approach for attempting to speed up the convergence of the LMS algorithm without having to sacrifice too much of its error floor performance, is through the use of a variable step size LMS (VSSLMS) algorithm. All the published VSSLMS algorithms make use of an initial large adaptation step size to speed up the convergence. Upon approaching the steady state, smaller step sizes are then introduced to decrease the level of adjustment, hence maintaining a lower error floor. This convergence improvement of the LMS algorithm increases its complexity from $2N$ in the case of LMS algorithm to $9N$ in the case of the MRVSS algorithm, where N is the number of array elements.

An alternative to the LMS algorithm is the RLS algorithm. Although higher complexity is required for the RLS algorithm compared to the LMS algorithm, it can achieve faster convergence, thus, better performance compared to the LMS algorithm. There are also improvements that have been made to the

RLS algorithm families to enhance tracking ability as well as stability. Examples are, the adaptive forgetting factor RLS algorithm (AFF-RLS), variable forgetting factor RLS (VFFRLS) and the extended recursive least squares (EX-KRLS) algorithm. The multiplication complexity of VFFRLS, AFF-RLS and EX-KRLS algorithms are $2.5N^2+3N+20$, $9N^2+7N$, and $15N^3+7N^2+2N+4$ respectively, while the RLS algorithm requires $2.5N^2+3N$.

All the above well known algorithms require an accurate reference signal for their proper operation. In some cases, several additional operating parameters should be specified. For example, MRVSS needs twelve predefined parameters. As a result, its performance highly depends on the input signal.

In this study, two adaptive beamforming algorithms have been proposed. They are called recursive least square - least mean square (RLMS) algorithm, and least mean square - least mean square (LLMS) algorithm. These algorithms have been proposed for meeting future beamforming requirements, such as very high convergence rate, robust to noise and flexible modes of operation. The RLMS algorithm makes use of two individual algorithm stages, based on the RLS and LMS algorithms, connected in tandem via an array image vector. On the other hand, the LLMS algorithm is a simpler version of the RLMS algorithm. It makes use of two LMS algorithm stages instead of the RLS – LMS combination as used in the RLMS algorithm.

Unlike other adaptive beamforming algorithms, for both of these algorithms, the error signal of the second algorithm stage is fed back and combined with the error signal of the first algorithm stage to form an overall error signal for use update the tap weights of the first algorithm stage.

Upon convergence, usually after few iterations, the proposed algorithms can be switched to the self-referencing mode. In this mode, the entire algorithm outputs are swapped, replacing their reference signals. In moving target applications, the array image vector, $\mathcal{F}$, should also be updated to the

new position. This scenario is also studied for both proposed algorithms. A simple and effective method for calculate the required array image vector is also proposed. Moreover, since the RLMS and the LLMS algorithms employ the array image vector in their operation, they can be used to generate fixed beams by pre-setting the values of the array image vector to the specified direction.

The convergence of RLMS and LLMS algorithms is analyzed for two different operation modes; namely with external reference or self-referencing. Array image vector calculations, ranges of step sizes values for stable operation, fixed beam generation, and fixed-point arithmetic have also been studied in this thesis. All of these analyses have been confirmed by computer simulations for different signal conditions. Computer simulation results show that both proposed algorithms are superior in convergence performances to the algorithms, such as the CSLMS, MRVSS, LMS, VFFRLS and RLS algorithms, and are quite insensitive to variations in input SNR and the actual step size values used. Furthermore, RLMS and LLMS algorithms remain stable even when their reference signals are corrupted by additive white Gaussian noise (AWGN). In addition, they are robust when operating in the presence of Rayleigh fading. Finally, the fidelity of the signal at the output of the proposed algorithms beamformers is demonstrated by means of the resultant values of error vector magnitude (EVM), and scatter plots. It is also shown that, the implementation of an eight element uniform linear array using the proposed algorithms with a wordlength of nine bits is sufficient to achieve performance close to that provided by full precision.

ACKNOWLEDGEMENTS

All praises go to Allah Almighty, Who provided me this opportunity to contribute a drop in the sea of knowledge which He blessed to the human being. I only desire betterment to the best of my power; and my success can only come from Allah. In Him I Trust, and unto Him I look.

It is my great honor to acknowledge the efforts of many peoples involved in this research to make it possible. First of all, my deepest gratitude goes to my supervisor, Prof. Kah-Seng Chung, for his passionate guidance, support, advice and encouragement throughout my studies at Curtin University. His supervision played a vital role to improve the quality and scientific vision of this research.

Secondly, I gratefully thank my Co-supervisor, Prof. Ali Mansour, for his supervision, help and support throughout all the struggling period of my research. He was always there to listen to my ideas and discuss them, especially for the mathematical side of my research.

Thirdly, I would like to record the role of my mother and my father in achieving this destination. Their prayers, encouragement, love and affection were a brilliant help to keep me focused during this study. I heartily acknowledge the prayers of my brothers, sisters and all other relatives as they played a vital role in the completion of this dissertation.

Fourthly, I also extend my sincere gratitude to my wife and children, who were so patient, and allowed me to spend extra time on this research. I have to say sorry to my wife since she lived a long time in very difficult conditions, and for my son, Ahmed, who lost four years of learning the book of Allah.

Finally, I am also very grateful to all the members in the Communication Technology Research Group for their friendship and support to help me finish my study. I am also thankful to all of my friends in Australia and Libya for their moral support during this study to boost my determination to achieve this goal.

PUBLICATIONS

1. J. A. Srar, K. S. Chung, and A. Mansour, "Adaptive Array Beamforming Using a Combined RLS-LMS Algorithm," Being reviewed for publication in IEEE Trans. on Signal Processing.

2. J. A. Srar, K. S. Chung, and A. Mansour, "Adaptive Array Beamforming Using a Combined LMS-LMS Algorithm," IEEE Trans. on Antennas and Propagation, vol. 58(11), pp. 3545-3557, November 2010.

3. Jalal SRAR, Kah-Seng CHUNG and Ali MANSOUR, “Analysis of the RLMS Adaptive Beamforming Algorithm Implemented with Finite Precision ," in the 16th Asia-Pacific Conference on Communications APCC 2010, Auckland, New Zealand, 31 Oct.-3 November 2010.

4. Jalal SRAR, Kah-Seng CHUNG and Ali MANSOUR, "A new LLMS Algorithm for Antenna Array Beamforming," in 2010 IEEE Wireless Communications and Networking Conference, Sydney, Australia, April 2010.

5. Jalal SRAR, Kah-Seng CHUNG and Ali MANSOUR, "Adaptive Array Beamforming using a Combined LMS-LMS Algorithm," in 2010 IEEE Aerospace Conference, Montana, USA, March 2010.

6. J. A. Srar and K.-S. Chung, "RLMS Algorithm for Fixed or Adaptive Beamforming," in IEEE 9th Malaysia International Conference MICC2009, Kuala Lumpur, Malaysia, December 2009.

7. J. A. Srar and K.-S. Chung, "Adaptive RLMS Algorithm for Antenna Array Beamforming," in IEEE Region 10 Conference TENCON2010, Singapore, November 2009.

8. J. A. Srar and K.-S. Chung, "Adaptive Array Beam Forming Using a Combined RLS-LMS Algorithm," in the 14th Asia-Pacific Conference on Communications APCC2008, Tokyo, Japan, October 2008.

9. J. A. Srar and K.-S. Chung, "Performance of RLMS Algorithm in Adaptive Array Beam Forming," in the 11th IEEE International Conference on Communication Systems ICCS2008, Guangzhou, China, November 2008.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AFFRLS	Adaptive forgetting factor recursive least square
AOA	Angle of arrival
AOD	Angle of departure
AWGN	Additive white Gaussian noise
BER	Bit error rate
bps	Bit per second
BPSK	Binary phase-shift keying
CCM	Constrained constant modulus
CCM-RLS	Constrained constant modulus-recursive least square
CDMA	Code division multiple access
CM	Constant modulus
CM-MASS	Combined constant modulus and modified adaptive step size algorithm
CM-TAASS	Constant modulus algorithm and time averaging adaptive step size algorithm combination
CMV	Constrained minimum variance algorithm
CMV-LMS	Constrained minimum variance-least mean square
CMV-RLS	Constrained minimum variance-recursive least square
CSLMS	Constraint stability LMS
dB	Decibel
DCT	Discrete cosine transform
DD	Decision directed
DFT	Discrete Fourier transform
DS/CDMA	Direct sequence code division multiple access

DWT	Discrete Walsh transform
EMSE	Steady-state excess mean-square error
EVD	Eigenvalue decomposition
EVM	Error vector magnitude
ESPRIT	Estimation of signal parameters via rotational invariance technique
EX-KRLS	Extended recursive least squares algorithm
FAEST	Fast a posteriori error sequential technique
FEDR	Fast Euclidian direction search
FFT	Fast Fourier transform
FM	Frequency modulation
FRLS	Fast recursive least square
FSK	frequency-shift keying
HoCA	Higher order cumulant algorithm
Hr	Hour
HRLS	Hierarchical recursive least square
Hz	Hertz
IFFT	Inverse fast Fourier transform
KLMS	The Kernal least mean square
Km/h	Kilometre per hour
LLMS	LMS-LMS combination which employs an adaptive array image vector
$LLMS_1$	LMS-LMS combination which employs a fixed array image vector
LMS	Least mean square
LMS_1	First LMS algorithm stage of the LLMS algorithm
LMS_2	Second LMS algorithm stage of the LLMS algorithm

LS	Least-squares
LTE	Long term evolution
Mbits	Mega bits
MCM	Modified constant modulus
MHz	Mega hertz
MIMO	Multiple-input and multiple-output
MMARY	Multi-modulus array
MMSE	Minimum mean square error
MRVSS	Modified robust variable step size
MSC	Most significant coefficient
MSE	Mean square error
MSINR	Maximum signal to interference plus noise signal
MUSIC	Multiple signal classification
MVDR	Minimum variance distortionless response
NAFFRLS	Normalized least mean square adaptive forgetting factor recursive least square
NLMS	Normalized LMS
NSVSSLMS	Normalized square variable step size LMS
OFDM	Orthogonal frequency-division multiplexing
OFDMA	Orthogonal frequency-division multiple access
PSK	Phase-shift keying
QAM	Quadrature amplitude modulation
RAMP	Recursive adaptive matching pursuit
RHS	Right hand side
rms	Root mean square
RLMS	RLS-LMS combination which employs an adaptive array

	image vector
$RLMS_1$	RLS-LMS combination which employs a fixed array image vector
RLS	Recursive least square
RLS-CM	Recursive least square- constant modulus
RVSS	Robust VSSLMS
SD-CM	Steepest-descent-constant modulus
SDD	Soft decision directed
SDMA	Space division multiple access
SINR	Signal-to-noise plus interference ratio
$SINR_o$	Output signal-to-noise plus interference ratio
SIR	Signal to interference ratio
SMI	Sample matrix inversion
SNR	Signal to noise ratio
SVSS	Sign variable step size
TDLMS	Transform-domain LMS
UCA	Uniform circular array
ULA	Uniform linear array
VOA	Variance oriented approach
VSSLMS	Variable step size LMS
WiMAX	Worldwide interoperability for microwave access

LIST OF SYMBOLS AND NOTATIONS

$\lvert\bullet\rvert$	Absolute value operator
θ_d	Angle of arrival of the desired signal
θ_i	Angle of arrival of the interfering signal
G	Array element gain
AF	Array factor
AF_{UCL}	Array factor of the uniform linear array
$\boldsymbol{\mathcal{F}}_{\mathrm{L}}$	Array image vector of the LLMS algorithm
$\boldsymbol{\mathcal{F}}_{\mathrm{R}}$	Array image vector of the RLMS algorithm
A_d	Array vector in the direction of the desired signal
A_i	Array vector in the direction of the interfered signal
ϕ	Azimuth angle
$\tilde{x}$	Average of the input signal samples
$\tilde{P}$	Average power of all symbols involved for the given modulation
f_c	Carrier frequency
$r_{l,ka}$	Coefficients of $\boldsymbol{R}_2$ of the LLMS algorithm
$r_{r,ka}$	Coefficients of $\boldsymbol{R}_{\mathrm{LMS}}$ of the RLMS algorithm
β	Constant between 0 and 1 associated with VSSLMS algorithm
υ	Constant between 0 and 1 associated with MRVSS algorithm
η_e	Constant between 0 and 1 associated with VSSLMS algorithm
q_v	Constant between 0 and 1 associated with NLMS algorithm
γ	Constant between 0 and 1 associated with VSSLMS algorithm
γ_{N}	Convergence factor of NLMS algorithm

$\gamma_{\mathrm{N,min}}$	Convergence factor lower limit of NLMS algorithm
$\gamma_{\mathrm{N,max}}$	Convergence factor upper limit of NLMS algorithm
γ'_{N}	Convergence updated factor of NLMS algorithm
η_{N}	Constant less than or equal to 1-ε associated with NLMS algorithm
η	Constant less than or equal to 1-ε associated with MRVSS algorithm
ε	Constant used in a NLMS algorithm, where its value equals the reciprocal of the number of used snapshots used to estimate the average
ε_{cs}	Constant used to prevent division by zero in CSLMS algorithm
ε_c	Constant equal to the reciprocal of the number of used samples employed for the estimation of the array image factor of the RLMS algorithm
$(*)$	Conjugate operator
$\rho_{x_{1,k},\boldsymbol{V}_1}$	Correlation coefficient between $x_{1,k}(t)$ and the error of LMS_1 algorithm $\boldsymbol{V}_1(n)$ of the LLMS algorithm
R^2_{ey}	Cross-correlation between the output signal (y) and the output error (e)
$\boldsymbol{R_e}$	Cross-correlation matrix of successive error samples
$\triangleq$	Denotes an equivalent
s_d	Desired signal
$\boldsymbol{\Lambda}_1$	Diagonal matrix of eigenvalues of $\boldsymbol{Q}_1$ of the LLMS algorithm
$\boldsymbol{\Lambda}_{\mathrm{R}}$	Diagonal matrix of eigenvalues of $\boldsymbol{Q}_{\mathrm{R}}$ of the RLMS algorithm
$\boldsymbol{\Lambda}_2$	Diagonal matrix of eigenvalues of $\boldsymbol{R}_2$ of the LLMS algorithm
$diag(\bullet)$	Diagonal of the matrix operator

$\boldsymbol{D}_X$	Diagonal of the matrix
$\boldsymbol{V}_1$	Difference between the estimated and actual tap weights for the LMS_1 algorithm stage of the LLMS algorithm
$\boldsymbol{V}_2$	Difference between the estimated and actual tap weights for the LMS_2 algorithm stage of the LLMS algorithm
$\boldsymbol{V}_{\mathrm{LMS}}$	Difference between the estimated and actual tap weights for the LMS algorithm stage of the RLMS algorithm
$\boldsymbol{V}_{\mathrm{RLS}}$	Difference between the estimated and optimal tap weights for the RLS stage of the RLMS algorithm
D_{L}	Difference between the current LMS_1 stage reference signal sample and the last LMS_2 stage error of the LLMS algorithm
D_{R}	Difference between the current RLS stage reference signal sample and the last LMS stage error of the RLMS algorithm
$\boldsymbol{S}$	Differentiation of the inverse of the correlation matrix with respect to the RLS forgetting factor (α_{RLS})
$\boldsymbol{\psi}$	Differentiation of the weight vector with respect to the RLS forgetting factor (α_{RLS})
f_d	Doppler frequency
R_2	Elements form of $\boldsymbol{R}_2$
E	Eigenvalue of $\boldsymbol{Q}_{\mathrm{R}}$
$\lambda_{\mathrm{L},2}$	Eigenvalue of $\boldsymbol{R}_2$
$\lambda_{\mathrm{R},2}$	Eigenvalue of $\boldsymbol{R}_{\mathrm{LMS}}$
$\mathcal{F}_l$	Elements of the array image factor vector, $\boldsymbol{\mathcal{F}}_{\mathrm{L}}$
$\mathcal{F}_r$	Elements of the array image factor vector, $\boldsymbol{\mathcal{F}}_{\mathrm{R}}$
$\mathcal{D}$	Elements spacing of the uniform linear array
$\mathcal{D}_x$	Elements spacing of the planar array in the x-direction
$\mathcal{D}_y$	Elements spacing of the planar array in the y-direction
$\boldsymbol{q}_1$	Eigenvectors matrix of $\boldsymbol{Q}_1$
$\boldsymbol{q}_{\mathrm{R}}$	Eigenvectors matrix of $\boldsymbol{Q}_{\mathrm{R}}$

$\overline{\xi}$	Ensemble average of the mean square error of the LLMS algorithm
$\overline{\xi}_{\mathrm{RLMS}}$	Ensemble average of the mean square error of the RLMS algorithm
e	Error
δe	Error difference associated with CSLMS algorithm
erf	Error function $\mathrm{erf}(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2/2}dt$
e_1	Error signal of the first LLMS algorithm stage
e_2	Error signal of the second LLMS algorithm stage
e_{RLS}	Error signal of the first RLMS algorithm stage
e_{LMS}	Error signal of the second RLMS algorithm stage
$\hat{\sigma}_e^2$	Estimation of the variance of the output error
$\hat{\sigma}_i^2$	Estimation of the variance of the i^{th} output signal of the transformation block
$E[\bullet]$	Expectation operator
α_{exp}	Exponential forgetting factor constant
$\boldsymbol{X}'$	Finite precision Input signal vector
$\boldsymbol{\alpha}$	Finite precision quantization error vector of the input signal
$\boldsymbol{\alpha}_1$	Finite precision quantization error vector of the input signal associated with the first LMS stage of LLMS algorithm
$\boldsymbol{\alpha}_2$	Finite precision quantization error vector of the input signal associated with the second LMS stage of LLMS algorithm
$\boldsymbol{\alpha}_{\mathrm{LMS}}$	Finite precision quantization error vector of the input signal associated with LMS stage of RLMS algorithm
$\boldsymbol{\alpha}_{\mathrm{RLS}}$	Finite precision quantization error vector of the input signal associated with RLS stage of RLMS algorithm
$\boldsymbol{\rho}_1$	Finite precision quantization error vector of the first LMS stage weights of LLMS algorithm
$\boldsymbol{\rho}_2$	Finite precision quantization error vector of the second LMS

	stage weights of LLMS algorithm
$\boldsymbol{\rho}_{\mathrm{LMS}}$	Finite precision quantization error vector of the LMS stage weights of RLMS algorithm
$\boldsymbol{\rho}_{\mathrm{RLS}}$	Finite precision quantization error vector of the RLS stage weights of RLMS algorithm
$\boldsymbol{\rho}$	Finite precision quantization error vector of the weight
e'_{LLMS}	Finite precision overall LLMS error signal
e'_{RLMS}	Finite precision overall RLMS error signal
η_1	Finite precision truncation and round-off errors associated with the first LMS stage of the LLMS algorithm
η_2	Finite precision truncation and round-off errors associated with the second LMS stage of the LLMS algorithm
η_{LMS}	Finite precision truncation and round-off errors associated with LMS stage of the RLMS algorithm
η_{RLS}	Finite precision truncation and round-off errors associated with RLS stage of the RLMS algorithm
α	Forgetting factor of the gradient algorithms
α_{RLS}	Forgetting factor of the RLS algorithm
$\alpha_{\max}$	Forgetting factor of the RLS algorithm, upper limit
$\alpha_{\min}$	Forgetting factor of the RLS algorithm, lower limit
κ	Gain constant associated with SD-CM algorithm
β_{I}	Gain constant of the MVDR beamforming algorithm
$\boldsymbol{K}$	Gain matrix of RLS algorithm
$(\bullet)^H$	Hermitian of matrix
e_I	Imaginary components of the error
A_S	Input desired signal amplitude
I_m	Input interference signal amplitude
$\boldsymbol{n}$	Input noise vector
$\boldsymbol{Q}$	Input signal correlation matrix

$\boldsymbol{Q}_1$	Input signal correlation matrix of the LLMS algorithm
$\boldsymbol{Q}_\mathrm{R}$	Input signal correlation matrix of the RMS algorithm
$\boldsymbol{R}_\mathrm{LMS}$	Input signal correlation matrix of the LMS algorithm stage of the RLMS algorithm
$\boldsymbol{R}_2$	Input signal correlation matrix of the LMS_2 algorithm stage of the LLMS algorithm
$\boldsymbol{Z}$	Input signal cross-correlation vector
$\boldsymbol{Z}_\mathrm{L}$	Input signal cross-correlation vector of the LLMS algorithm
$\boldsymbol{Z}'_\mathrm{L}$	Input signal cross-correlation vector using self-referencing mode of the LLMS algorithm
$\boldsymbol{Z}_\mathrm{R}$	Input signal cross-correlation vector of the RLMS algorithm
$\boldsymbol{Z}'_\mathrm{R}$	Input signal cross-correlation vector using self-referencing mode of the RLMS algorithm
$\boldsymbol{X}$	Input signal vector, $\left[x_1, x_2, ..., x_N\right]$.
$\boldsymbol{X}_1$	Input signal vector of the first stage of the LLMS algorithm
$\boldsymbol{X}_2$	Input signal vector of the second stage of the LLMS algorithm
$\boldsymbol{X}_\mathrm{LMS}$	Input signal vector of the LMS stage of the RLMS algorithm
$\delta\boldsymbol{X}$	Input signal difference of the CSLMS algorithm
s_i	Interfering signal
$\left[\bullet\right]_\mathrm{int}$	Integer operator
ψ_d	Inter-element phase shift with respect to the desired signal in radians
ψ_i	Inter-element phase shift with respect to the interfering signal in radians
ψ_c	Inter-element phase shift between the columns of the planar array in radians
ψ_x	Inter-element phase shift between the rows of the planar array in radians
i	Interference signal
$\boldsymbol{P}$	Inverse of the correlation matrix ($\boldsymbol{Q}$)

δ_l^m	Kronecker delta function which is defined as $\delta_l^m = \begin{cases} 0 & l \neq m \\ 1 & l = m \end{cases}$
$\tilde{e}_{\min}$	Lower bound of the time average of the error square signal
$\mu_{\max}$	Maximal allowable step size
$E_{\max}$	Maximum eigenvalue of the LMS_1 stage input signal covariance matrix of the LLMS algorithm
E_{RLS}	Maximum eigenvalue of the RLS stage input signal covariance matrix of the RLMS algorithm
$\Upsilon_{\max}$	Maximum tolerance in the inter-spacing of the array elements
$g_{\max}$	Maximum tolerance of the gain of the array elements
ξ	Mean-square error of an algorithm
ξ_{LLMS}	Mean-square error of the LLMS algorithm
ξ_{RLMS}	Mean-square error of the RLMS algorithm
z	Measurement noise
c_p	Mixing parameter of the affine LMS algorithm
$\xi_{\mathrm{LLMS,min}}$	Minimum mean square error of the LLMS algorithm
$\xi_{\mathrm{RLMS,min}}$	Minimum mean square error of the RLMS algorithm
$\mu_{\min}$	Minimal step size
K_{MSC}	Most significant coefficient index of the FFT output
N	Number of array elements
N_b	Number of bits
N_y	Number of the columns of the planar array elements
N_x	Number of the rows of the planar array elements
k	Number of signal snapshots used for estimating the average
N_{OFDM}	Number of OFDM subcarriers
$(\bullet)_n$	Normalized operator
$\boldsymbol{W}_{opt}$	Optimum weights vector of an algorithm

$\boldsymbol{W}_{opt1}$	Optimum weights vector of LMS_1 stage of the LLMS algorithm
$\boldsymbol{W}_{opt_{\mathrm{RLS}}}$	Optimum weights vector of RLS stage of the RLMS algorithm
P_d	Output desired signal power
P_i	Output interference signal power
P_n	Output noise power
y	Output signal of an algorithm
y_s	Output signal of the array due to the desired signal
y_1	Output signal of the first entire-output of the affine algorithm
y_{LMS1}	Output signal of the LMS_1 algorithm stage of the LLMS algorithm
y_{LLMS}	Output signal of the LLMS algorithm
y_{RLMS}	Output signal of the RLMS algorithm
y_{RLS}	Output signal of the RLS algorithm stage of RLMS algorithm
y_2	Output signal of the second entire-output of the affine algorithm
e_{LLMS}	Overall error signal of the LLMS algorithm
e_{RLMS}	Overall error signal of the RLMS algorithm
μ_l	Positive number related to the LMS step size
c_q	Quantization constant depends on how the inner product of a vector manipulation is implemented
R	Radius of the circular array
R_m	Radius of the QAM signal constellation area
e_R	Real components of the error
$\Re\{\bullet\}$	Real part operator
ε_r	Regularization parameter
d	Reference signal
d_{LMS}	Reference signal for the LMS stage of the RLMS algorithm

d_{RLS}	Reference signal for the RLS stage of the RLMS algorithm
d_1	Reference signal of the first LLMS algorithm stage
d_2	Reference signal of the second LLMS algorithm stage
s_f	Scaling factor
$sign(\bullet)$	Sign function operator
δ	Small positive constant to initiate $\boldsymbol{P}$ of the RLS algorithm
$\bar{\boldsymbol{g}}$	Smoothed gradient vector
c	Speed of light
$\|\bullet\|_2^2$	Squared Euclidean norm operator
μ_b	Step size of the bottom LMS stage of the affine LMS algorithm
μ_c	Step size of the combination parameter of the affine LMS algorithm
μ	Step size of the LMS algorithm
μ_1	Step size of the LMS_1 algorithm stage of the LLMS algorithm
μ_2	Step size of the LMS_2 algorithm stage of the LLMS algorithm
μ_{LMS}	Step size of the LMS stage of the RLMS algorithm
μ_{RLS}	Step size of the RLS stage of the RLMS algorithm
μ_t	Step size of the top LMS stage of the affine LMS algorithm
μ_r	Step size ratio constant
$\Delta\theta_{d,3\,\mathrm{dB}}$	The 3 dB beamwidth
x_m	The m^{th} element of the input signal vector
x'_k	The m^{th} element of the outputs of the individual taps of the RLS stage of the RLMS algorithm
n_k	The k^{th} element of $\boldsymbol{n}$
$w_{\mathrm{RLS},m}$	The m^{th} weight of the RLS stage vector of the RLMS algorithm
$w_{1,m}$	The m^{th} weight of the LMS_1 stage vector of the LLMS

	algorithm
$A_{d,k}$	The k^{th} element of $\boldsymbol{A}_d$
t	Time
$\tilde{e}$	Time average of the error square signal
$\tilde{e}_b$	Time average of the instantaneous error signal of the bottom LMS stage of the affine LMS algorithm
$\tilde{e}_t$	Time average of the instantaneous error signal of the top LMS stage of the affine LMS algorithm
τ	Time delay
n	Time index
$\hat{\boldsymbol{D}}$	Time-varying power normalization parameter matrix
g_r	Tolerance of the array elements
Υ	Tolerance of the inter-spacing of the array elements
$trace(\mathbf{.})$	Trace of the matrix $(\mathbf{.})$
x	Transmit signal
$\boldsymbol{U}$	Unitary transformation matrix of the second stage input signal
$\boldsymbol{I}$	Unity matrix
$\tilde{e}_{\max}$	Upper bounds of the time average of the error square signal
ρ	Variable step size adaptation parameter
$\sigma^2_{\boldsymbol{V}_1}$	Variance of $\boldsymbol{V}_1(n)$
σ^2_q	Variance of $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$ associated with the input signal and weight vectors quantization errors
σ^2_n	Variance of the input noise
σ^2_x	Variance of the input signal
$\sigma^2_{x_{1,k}}$	Variance of the input signal at the k^{th} element of the array
σ^2_1	Variance of the output signal of the LMS_1 stage of the LLMS algorithm
σ^2_{RLS}	Variance of the output signal of the RLS stage of the RLMS

	algorithm
σ_W^2	Variance of the weights
$\sigma_{\mathrm{L},r}^2$	Variance of the weight of the LLMS algorithm
$\sigma_{\mathrm{R},r}^2$	Variance of the weight of the RLMS algorithm
σ_z^2	Variance of the measurement noise
σ_η^2	Variance of the finite precision errors associated with either LMS or RLS stages of the RLMS algorithm
λ	Wavelength of the carrier signal
$\delta \boldsymbol{W}$	Weight difference of the CSLMS algorithm
$\boldsymbol{W}_{tb}$	Weight difference of the entire affine LMS algorithm weights.
$\boldsymbol{W}$	Weight vector
$\boldsymbol{W}_t$	Weight vector of the top LMS stage of the affine LMS algorithm
$\boldsymbol{W}_b$	Weight vector of the bottom LMS stage of the affine LMS algorithm
$\boldsymbol{W}_{0\,\mathrm{LMS}}$	Weight vector modeled by a random walk process of the LMS algorithm stage of the RLMS algorithm
$\boldsymbol{W}_{01}$	Weight vector as modeled by a random walk process of the LMS_1 stage of the LLMS algorithm
$\boldsymbol{W}_{02}$	Weight vector as modeled by a random walk process of the LMS_2 stage of the LLMS algorithm
$\boldsymbol{W}'$	Weight vector of the Finite precision
$\boldsymbol{W}_1$	Weight vector of the first stage of the LLMS algorithm
$\boldsymbol{W}_{\mathrm{LMS}}$	Weight vector of the LMS stage of the RLMS algorithm
$\boldsymbol{W}_{\mathrm{LLMS}}$	Weight vector of the LLMS algorithm
$\boldsymbol{W}_{\mathrm{RLMS}}$	Weight vector of the RLMS algorithm
$\boldsymbol{W}_2$	Weight vector of the second stage of the LLMS algorithm
$\boldsymbol{W}_{\mathrm{RLS}}$	Weight vector of the RLS stage of the RLMS algorithm
e''_{LMS}	Zero mean measurement noise of the LMS stage of the

	RLMS algorithm
e_1''	Zero mean measurement noise of the LMS_1 stage of the LLMS algorithm
e_2''	Zero mean measurement noise of the LMS_2 stage of the LLMS algorithm
$\boldsymbol{r}_{\text{LMS}}$	Zero mean white iid sequence vector of the LMS stage of the RLMS algorithm
$\boldsymbol{r}_1$	Zero mean white iid sequence vector of the LMS_1 stage of the LLMS algorithm
$\boldsymbol{r}_2$	Zero mean white iid sequence vector of the LMS_2 stage of the LLMS algorithm

CHAPTER 1

INTRODUCTION

1.1 Scope of the Thesis

In recent years, adaptive antenna arrays or smart antennas have become a key component for cellular mobile communications [1] including worldwide interoperability for microwave access (WiMax) [2, 3] and long term evolution (LTE) system [4, 5]. Its use leads to an increase in the capacity of mobile radio communication systems, to fulfill the rapid growth in demand for wireless services. These antennas are used as spatial filters for receiving the desired signals coming from specific direction or directions while minimizing the reception of unwanted signals emanating from other directions. The ability of these antennas to track their target signals quickly and accurately depends largely on the performance of the beamforming algorithm employed.

Among many different adaptive algorithms studied, the LMS algorithm offers simpler implementation and good tracking capability while the RLS algorithm provides relatively fast convergence [6, 7]. More recently, variants of these two algorithms have been investigated for enhancing the convergence speed and tracking ability in time varying operating environments. For the LMS algorithm family, there is always a trade off between the speed and the residual error floor when a given adaptation step size is used. Therefore, over the last three decades, several improvements have been proposed to speed up the convergence. Some recent examples are variable step size LMS algorithm (VSSLMS) [8], constrained-stability LMS (CSLMS) algorithm [9], and modified robust variable LMS (MRVSS) algorithm [10]. These algorithms make use of an initial large adaptation step size to speed up the convergence. Upon approaching the steady state, smaller step

sizes are then introduced to decrease the level of adjustment, hence maintaining a lower residual error floor.

On the other hand, several modifications have been proposed to improve the tracking ability of the RLS family of algorithms. These include the adaptive forgetting factor RLS (AFF-RLS) [11], variable forgetting factor RLS (VFFRLS) [12], and the extended kernel recursive least square (EX-KRLS) algorithm [13].

The beamforming algorithm is expected to be computationally simple, numerically robust, fast convergent [14], robust to noise and able to work with a noisy reference signal. Unfortunately, none of the adaptive algorithms developed so far are able to fulfill all these requirements [14]. Therefore, this project aims to research into new algorithms that could fulfill as many of these requirements as possible.

1.2 Objectives and Original Contributions

The primary objectives of this research are:

- The development of a low complexity and flexible adaptive beamforming algorithm, which has fast convergence in conjunction with a low residual error.
- The new algorithm should be tolerant to noisy reference signal while having good tracking capability.
- Formulation of a Matlab baseband simulation platform for evaluating the performance of the proposed algorithm operating under various conditions.

As a result of this study, a number of significant technical contributions have been made. These are briefly described as follows:

Two adaptive beamforming algorithms have been proposed in this thesis. They are called the recursive least square - least mean square (RLMS) algorithm, and the least mean square - least mean square (LLMS) algorithm. These algorithms have been proposed to meet future beamforming requirements, such as very high convergence rate, robustness to noise and flexible modes of operation. The RLMS algorithm makes use of two individual algorithm stages, based on the RLS and LMS algorithms, connected in tandem via an array image vector. The LLMS algorithm is a simpler version of the RLMS algorithm. It makes use of two LMS algorithm stages instead of the RLS – LMS combination as used in the RLMS algorithm. For both of these algorithms, the error signal of the second algorithm stage is fed back and combined with the error signal of the first algorithm stage to form an overall error signal used to update the tap weights of the first algorithm stage. Detailed analyses of the RLMS and LLMS algorithms are presented in Chapter 4 and Chapter 5 respectively.

The new common architecture of the RLMS and LLMS algorithms offers the flexibility of two different modes of operation. Normally, each of these algorithms operates with an external reference signal. Moreover, once the algorithm has converged, often after only a few iterations, it can be switched over to operate with self-referencing. In this case, the output of the first algorithm stage is used as the reference for the second algorithm stage. At the same time, the output of the second stage is fed back to be used as the reference signal for the first algorithm stage. These two modes of operation are analysed in Sections 4.3 and 5.3 for the RLMS algorithm and the LLMS algorithm, respectively.

Computer simulation results presented in Sections 4.7.3.1 to 4.7.3.4 for the RLMS algorithm, and in Sections 5.4.2.1 and 5.4.2.4 for the LLMS algorithm, confirm the superior performance of the proposed algorithms over the RLS, VFFRLS, LMS, CSLMS and MRVSS algorithms. The performance measures considered are the convergence rate, residual error floor, sensitivity to noisy reference signal, and tracking ability. Also, it is shown that the resulting steady

state mean square errors (MSE) of the RLMS and LLMS algorithms are quite insensitive to changes in input signal-to-noise ratio (SNR).

Both the proposed algorithms also allow two different application modes of beamforming operation; namely fixed or adaptive beamforming. With the former, an accurate fixed beam can be provided by prior setting the elements of the array image vector with the prescribed values for the required direction. On the other hand, adaptive beamforming is obtained when the array image vector is allowed to continuously update and track the user direction. A simple but effective method of calculating the element values of the array image vector for adaptive beamforming is presented in Sections 4.5 and 5.3.4.1.

The boundary values for the step sizes of the individual algorithm stages required to achieve stable operation have been derived analytically in Sections 4.3.1 and 4.4, Sections 5.3.3 and 5.3.4.2, for the proposed RLMS and LLMS algorithms, respectively. It is shown that a stable operation of the proposed algorithms could be achieved with a broad range of step size values.

For implementation of the proposed algorithms, some practical issues have been considered and studied in Chapter 6. These include:

- Practical implementations of the proposed algorithms are likely to make use of finite precision mathematical functions. As such, an analysis on the estimated overall MSE signal is presented in Section 6.2 to determine the minimum numerical precision, in terms of binary wordlength, required for achieving an adequate performance. It is shown in Section 6.2.3 that an eight element uniform linear array implemented with a wordlength of nine bits using either the RLMS or LLMS algorithm is able to achieve a performance close to that provided by full numerical precision.

- The influence of variations in inter-element spacing and element gain of the array is studied in Section 6.3. It is shown in Sections 6.3.1,

6.3.2 and 6.3.3 that these practical imperfections tend to only raise the sidelobe level.

- Finally, linear antenna arrays implemented with two and four elements are investigated in Section 6.4. It is observed that similar performance could be maintained when a slight reduction in the number of tap weights is used in the second algorithm stage of either the RLMS or LLMS algorithm.

1.3 Structure of the Thesis

This thesis is organized into seven chapters.

Chapter 2 provides a background introduction to adaptive array beamforming. A survey of some of the common array geometries is presented, including discussions on array ambiguity and grating lobes associated with uniform linear arrays (ULA). Then, a typical beamforming example is provided to show how the weights are calculated for a given angle of arrival (AOA) of the desired signal. Finally, a brief description of the need for simultaneous multiple beam forming to serve multiple users is given.

Next, in Chapter 3, a review of adaptive beamforming algorithms is presented with emphasis on the non-blind algorithms. A survey of blind algorithms is also provided, as well as the recently launched wireless communication systems such as WiMAX and LTE.

In Chapter 4, a new approach to adaptive array beamforming using a combined RLS-LMS algorithm or RLMS algorithm is proposed. The fast convergence and robust operation of this new RLMS algorithm are verified through a detailed analytical study, followed by extensive computer simulations.

The observations made in Chapter 4 provide an incentive to search for a simpler version of the RLMS algorithm while still maintaining its superior

performance. This leads to the replacement of the RLS algorithm in the RLMS algorithm with an LMS algorithm. The resultant scheme is referred to as the LLMS algorithm, which maintains the low complexity generally associated with an LMS algorithm. The LLMS algorithm is studied in detail in Chapter 5.

Chapter 6 considers the effects on the operation of the RLMS and LLMS algorithms due to the use of finite numerical precision. This leads to the determination of the minimum wordlength, in terms of the number of binary bits, required to achieve a minimum degradation in performance when compared with an implementation using full numerical precision. Following this, the influence on the performance of the proposed algorithms due to tolerances in inter-element spacing and element gain is also examined. Furthermore, the performance of a linear antenna array implemented with either two or four elements is investigated.

Finally, the major findings of this research are reviewed and recommend-dations for future studies are made in Chapter 7.

CHAPTER 2

FUNDAMENTALS OF ARRAY BEAMFORMING

2.1 Introduction

The demand for modern wireless communications systems is becoming progressively more complex. This is the result of attempting to meet ever growing demands for higher date rates, wider coverage and greater capacity, but without a corresponding increase in spectrum allocation. Therefore, adaptive or smart antennas have been introduced to exploit the spatial domain by minimizing interference in order to enhance system coverage and capacity. These antennas are able to automatically direct their beam patterns to the desired signals with nulls in the directions of interfering signals. An antenna array is a set of antenna sensors that are spatially distributed with reference to one of its elements. In the case of an antenna array used for transmission, the beam direction is steered by appropriately adjusting the phase and amplitude of the signal applied to each of the antenna elements. As a result, the angle of departure (AOD) of the main beam of the array can be steered towards the required direction.

There are different types of antenna array geometries, the most common being the linear, circular and planar arrays. Linear arrays consist of antenna elements which are aligned along a straight line, while circular arrays have the elements arranged in a circle. A planar array has its elements lie on a plane surface. If the spacing between adjacent elements is equal, then the array is often referred to as a uniformly spaced array, such as the uniform linear array (ULA) [15, 16]. A linear array is simpler to implement than the other two geometries. However, its radiation pattern is symmetrical about the endfire[1] axis, thus giving rise to direction ambiguity. On the other hand, both

[1] Endfire is the direction that is parallel to the line joining the antenna elements in a linear array.

circular and planar arrays do not suffer from such a disadvantage [17].

The radiation pattern of an antenna array is dependent on a number of factors, such as, the type and number of antenna elements used and the way these elements are configured. In the case of isotropic radiating elements, the radiation pattern depends only on what is commonly known as the array factor (AF), which is governed by the inter-element spacing and feeding signals [15]. On the other hand, for non-isotropic elements, it is possible to determine total AF using the principle of multiplying the field of a single element positioned at the origin, and the array factor of an isotropic radiating element [16, 18], which will be described in greater detail in Section 2.4.

This chapter provides a review of array geometries, including a discussion on array ambiguity and possible grating lobes associated with uniform linear arrays. Then, an example is introduced to show how the weights are calculated for achieving beamforming in a given direction. Finally, a brief description on beamforming with multiple beams to cater for multiple users is also discussed.

2.2 Uniform Linear Array

As mentioned in Section 2.1, a ULA is composed of a number of equally spaced elements. Consider an array consisting of N elements as shown in Figure 2.1, with an inter-element spacing of $\mathcal{D}$ and the AOA of the desired signal is θ_d. The array broadside is normal to the array axis, and element 1 is taken to be the reference element.

Let $x(t)$ be a signal emitted from far away and its wavefront impinges on the N linear array of Figure 2.1. In this case, the signal received by element 2 experiences a time delay of τ with respect to element 1. The resulting output, $y(t)$, of the array is given by [19]

$$y(t) = x(t) + x(t - \tau) + ... + x(t - (N - 1)\tau) \tag{2.1}$$

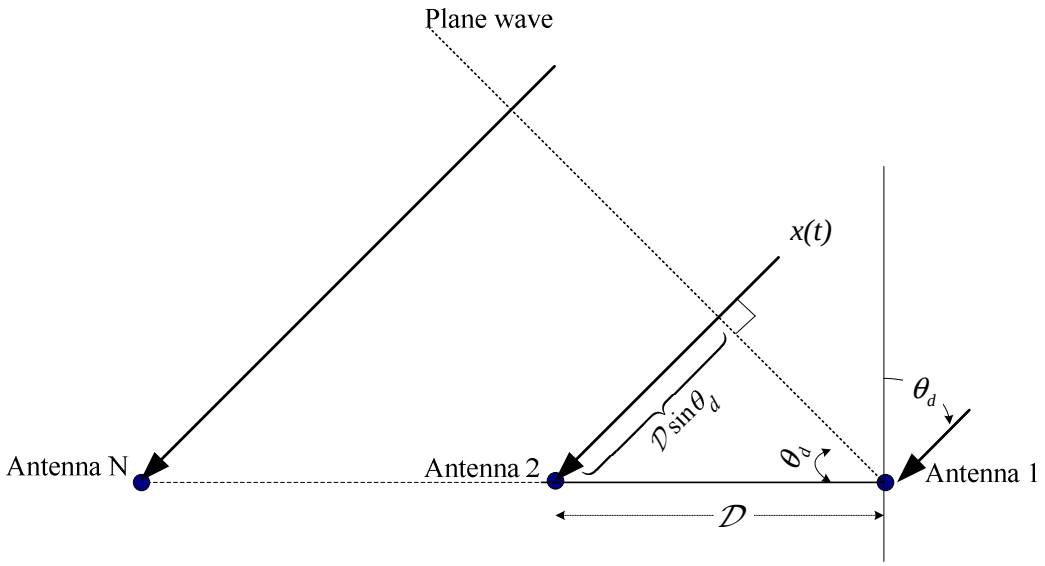


Figure 2-1 A linear array consisting of N identical omnidirectional antenna elements with the plane wavefront of $x(t)$.

where $\tau = \frac{\mathcal{D}\sin(\theta_d)}{c}$ with *c* being the velocity of light.

The time delay, τ, in equation (2.1) corresponds to a phase shift of $\psi_d = \frac{2\pi\mathcal{D}}{\lambda}\sin\theta_d$ radians [20], where λ is the carrier wavelength of frequency, f_c, such that

$$\lambda = \frac{c}{f_c} \tag{2.2}$$

The output signal of the array, $y(t)$, can also be expressed as

$$y(t) = \sum_{i=1}^{N} x(t)e^{-j(i-1)\psi_d} \tag{2.3}$$

where j is the complex operator defined as $j = \sqrt{-1}$.

Therefore, the directional pattern of the array, which defines the array sensitivity response to the received signal with an AOA of θ_d, can be expressed as [20]

$$AF(\theta_d) = \sum_{i=1}^{N} e^{j(i-1)\psi_d} \tag{2.4}$$

Equation (2.4) can be normalized with the maximum value at unity, so that [16]

$$\left(AF(\theta_d)\right)_n = \frac{\sin\left(\frac{N\psi_d}{2}\right)}{N\sin\left(\frac{\psi_d}{2}\right)} e^{\frac{j(N-1)\psi_d}{2}} \tag{2.5}$$

Equation (2.5) indicates that $\left(AF(\theta_d)\right)_n$ has the following characteristics [21]:

- Maximum values occur when $\psi_d = 0$, i.e. at the broadside angle, $\theta_d = \mp 2n_i\pi$, where $n_i = 0, 1, 2, \ldots$.
- Nulls of the array occur when $\psi_d = \mp 2n_i\pi/N$, where $n_i = 1, 2, 3, \ldots$, and $n_i \neq N, 2N, 3N, \ldots$.
- The 3 dB beamwidth, $\Delta\theta_{d,3\,\mathrm{dB}}$, of the array factor can be obtained from [21]

$$\Delta\theta_{d,3\,\mathrm{dB}} = 0.866\frac{\lambda}{N\mathcal{D}} \tag{2.6}$$

According to equation (2.6), the main beamwidth decreases as the number of array elements increases.

As an example, consider a linear array which is made up of eight isotropic antenna elements with uniform inter-element spacing $\mathcal{D}$ of $\lambda/2$. Figure 2-2 shows the modulus of the normalized array factor, $\left|\left(AF(\theta_d)\right)_{\mathrm{n}}\right|$, computed using equation (2.5), when the array is steered towards either 0°, 60°, or -30°. It can be observed from Figure 2.2 that the main beam of the array is directed to the correct specified direction, θ_d, and the width of the beam increases as the angle θ_d is deviating away from 0°.

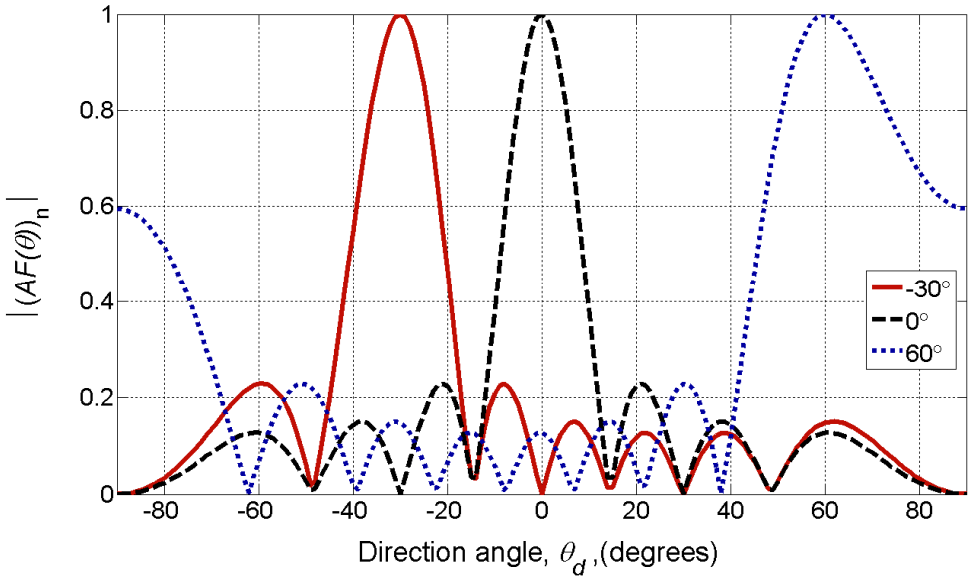


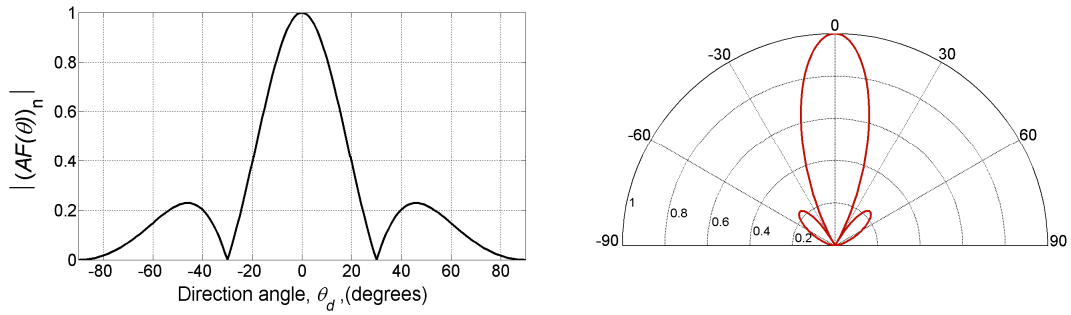
Figure 2-2 The beam patterns $\left|\left(AF(\theta_d)\right)_{\mathrm{n}}\right|$ of an eight element linear array obtained for θ_d equal to either 0°, 60°, or -30°. The inter-element spacing $\mathcal{D}$ is $\lambda/2$.

Next, when the inter-element spacing is varied, the resulting values of $\left|\left(AF(\theta_d)\right)_{\mathrm{n}}\right|$ are evaluated over a range of $-\pi/2 \le \theta_d \le \pi/2$. These are shown in Figure 2-3 (a-c) for $\mathcal{D}=\lambda/4$, $\mathcal{D}=\lambda/2$ and $\mathcal{D}=\lambda$, respectively. From Figure 2-3 (a-c), it is observed that the main beam width becomes larger when the inter-element spacing is reduced. On the other hand, when a larger element spacing is used, there will be an increase in the number of side lobes.

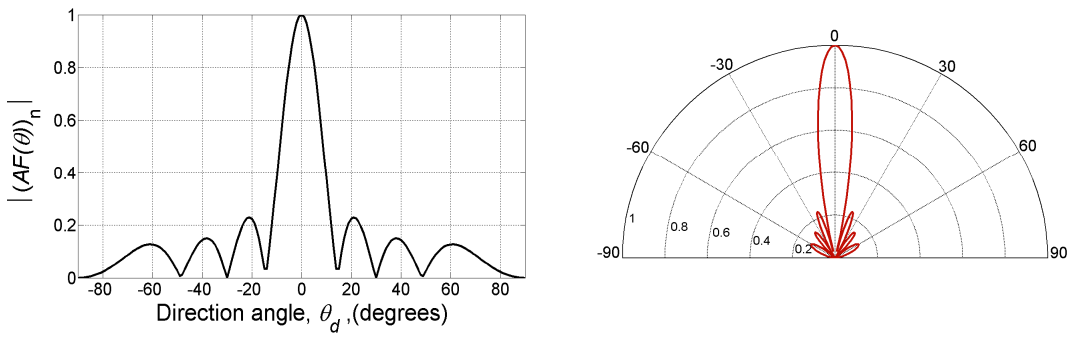
2.3 Array Ambiguity and Grating Lobes

From equation (2.5), the modulus of the normalised array factor, $\left|\left(AF(\theta_d)\right)_{\mathrm{n}}\right|$, can be expressed as

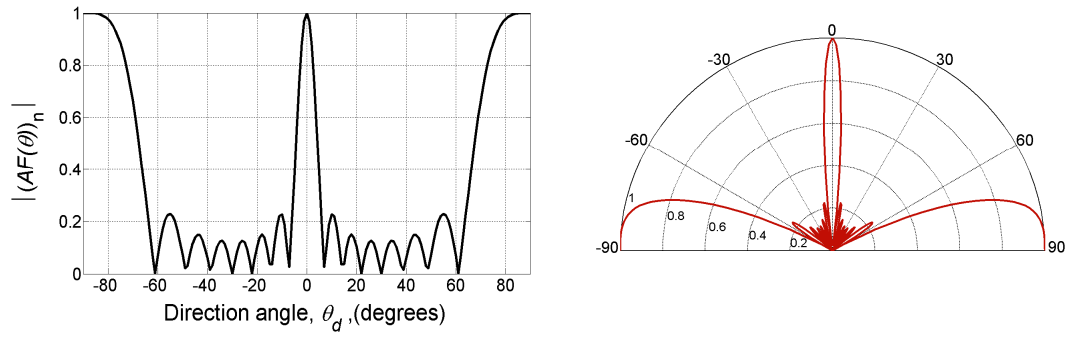
$$\left|\left(AF(\theta_d)\right)_{\mathrm{n}}\right| = \frac{\left|\sin\left(\frac{N\psi_d}{2}\right)\right|}{N\left|\sin\left(\frac{\psi_d}{2}\right)\right|} = \frac{\left|\sin\left(\frac{\pi\mathcal{D}N\sin(\theta_d)}{\lambda}\right)\right|}{N\left|\sin\left(\frac{\pi\mathcal{D}\sin(\theta_d)}{\lambda}\right)\right|} \qquad (2.7)$$



(a) $\mathcal{D} = \lambda/4$



(b) $\mathcal{D} = \lambda/2$



(c) $\mathcal{D} = \lambda$

Figure 2-3 The beam patterns of an eight element linear array obtained with θ_d set at 0^o and the inter-element spacing $\mathcal{D}$ is equal to (a) $\lambda/4$, (b) $\lambda/2$, and (c) λ .

According to equation (2.7), a maximum value of $\left|\left(AF(\theta_d)\right)_n\right|$ occurs whenever $\theta_d = \pm\frac{n_i\lambda}{\mathcal{D}}$ where $n_i = 0, 1, 2, \ldots$. In this case, when the inter-element

spacing $\mathcal{D}=\frac{\lambda}{2}$, a beam occurs at the desired direction of $\theta_d = 0^\circ$. In addition, it is noted that another beam also exits at $\theta_d = \pm 180^o$, as shown in Figure 2-4. This additional beam, which gives rise to ambiguity in the beam direction, is often referred to as a grating lobe.

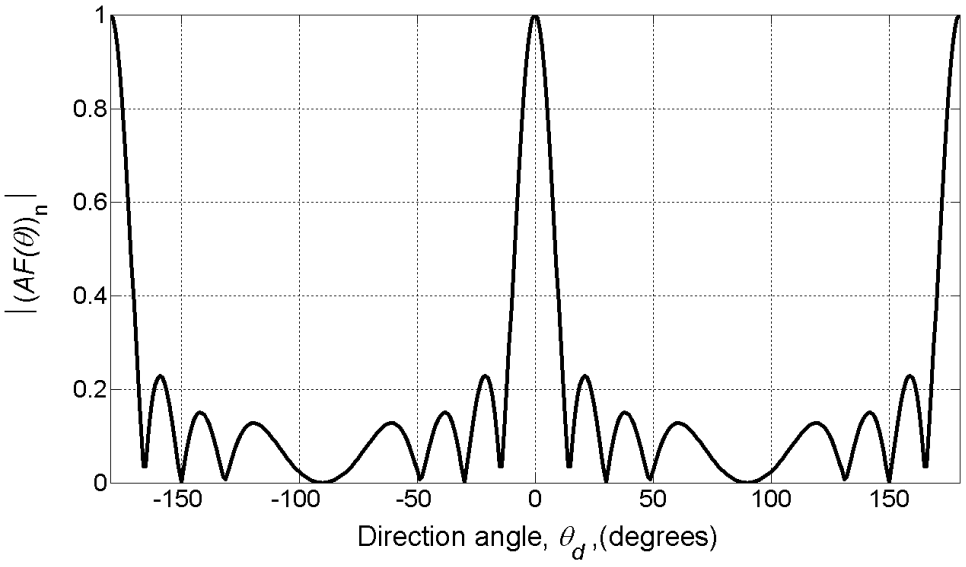
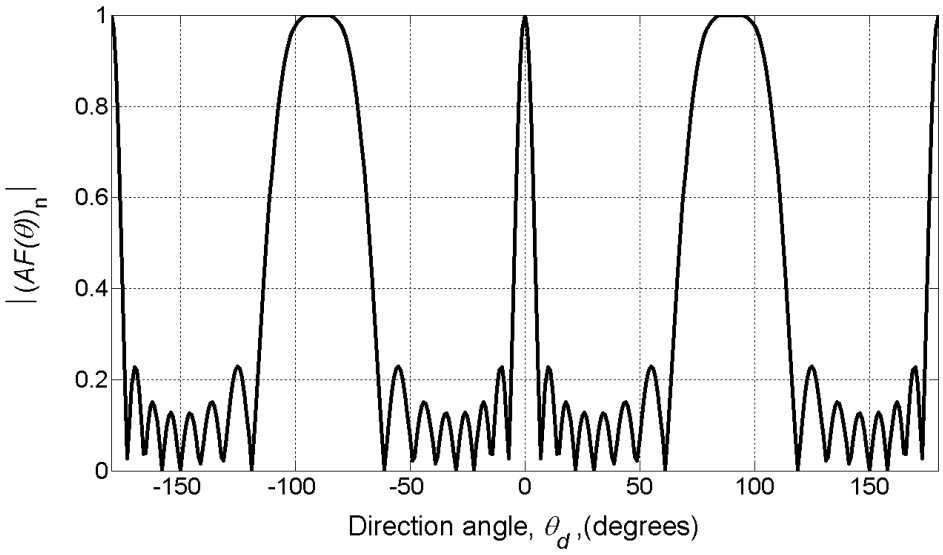
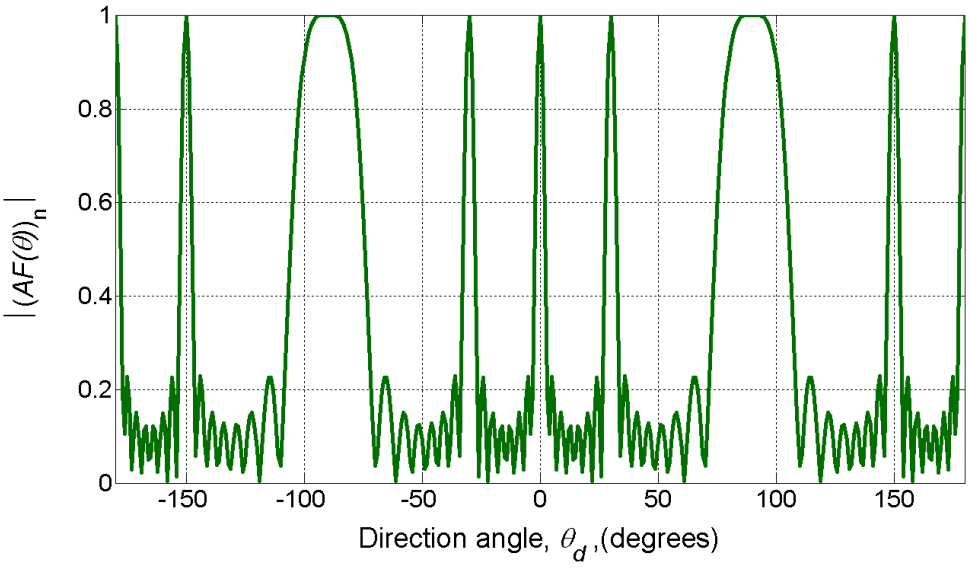


Figure 2-4 The beam patterns of an eight element uniform linear array obtained with $\mathcal{D}=\lambda/2$.

Now, for inter-element spacing larger than half wave length, i.e., $\mathcal{D}>\frac{\lambda}{2}$, more grating lobes may appear in addition to the desired main lobe. For instance, with an inter-element spacing of $\mathcal{D}=\lambda$, the main desired beam will appear at $\theta_d = 0^\circ$, while grating lobes would occur at $\pm 90^\circ$ and $\pm 180^\circ$, as shown in Figure 2-5a. In the case of $\mathcal{D}=2\lambda$, grating lobes occur at $\pm 30^\circ$, $\pm 90^\circ$, $\pm 150^\circ$ and $\pm 180^\circ$, as shown in Figure 2-5b. The presence of grating lobes makes it ambiguous to know for certain which is the correct angle to associate with the desired signal [22]. As a result, the beamformer is not able to distinguish between signals coming from the desired main lobe direction and those from the directions of the grating lobes.



(a) $\mathcal{D}=\lambda$



(b) $\mathcal{D}=2\lambda$

Figure 2-5 The presence of grating lobes when an eight element uniform linear array is implemented with (a) $\mathcal{D}=\lambda$ and (b) $\mathcal{D}=2\lambda$.

Now, to steer the main beam of an ULA to an angle $\theta=\theta_d$, the normalized array factor given in equation (2.7) can be rewritten as [22, 23]

$$\left|AF(\theta)\right| = \frac{\left|\sin\left(\frac{\pi \mathcal{D} N\left[\sin(\theta)-\sin(\theta_d)\right]}{\lambda}\right)\right|}{N\left|\sin\left(\frac{\pi \mathcal{D}\left[\sin(\theta)-\sin(\theta_d)\right]}{\lambda}\right)\right|} \tag{2.8}$$

From equation (2.8), the desired main lobe is pointed at an angle θ_d, while the grating lobes are located at those angles θ, which make the denominator equal to zero, i.e.,

$$\sin(\theta)-\sin(\theta_d) = \pm n_i \frac{\lambda}{\mathcal{D}} \tag{2.9}$$

where $n_i = 1, 2, 3, \ldots$.

A plot of equation (2.9) is shown in Figure 2-6. It shows that if either the element spacing or the wave length is changed, the angular locations where the grating lobes occur will also change. To illustrate the use of Figure 2-6, we consider the case of $\mathcal{D} = \lambda$ or $\lambda/\mathcal{D} = 1$. Under this condition, we observe that two curves intercept $\lambda/\mathcal{D} = 1$ to yield $\left|\sin\theta - \sin\theta_d\right| = 1$ (blue curve), and $\left|\sin\theta - \sin\theta_d\right| = 2$ (green curve). This means that, if $\theta_d = 0^\circ$, two grating lobes will appear at θ corresponding to 90^o and 180^o. This observation is verified by the case as shown in Figure 2-5a.

2.4 Planar Array

According to [20, 24, 25], a planar array consists of antenna elements arranged in an $x-y$ plane. A common configuration of the planar array, the rectangular array, is shown in Figure 2-7. Such a planar array has an additional degree of control over a linear array. As a result, it is now possible to steer the elevation angle, θ, as well as the azimuth angle, ϕ, of the beam of a planar array to form a pencil beam.

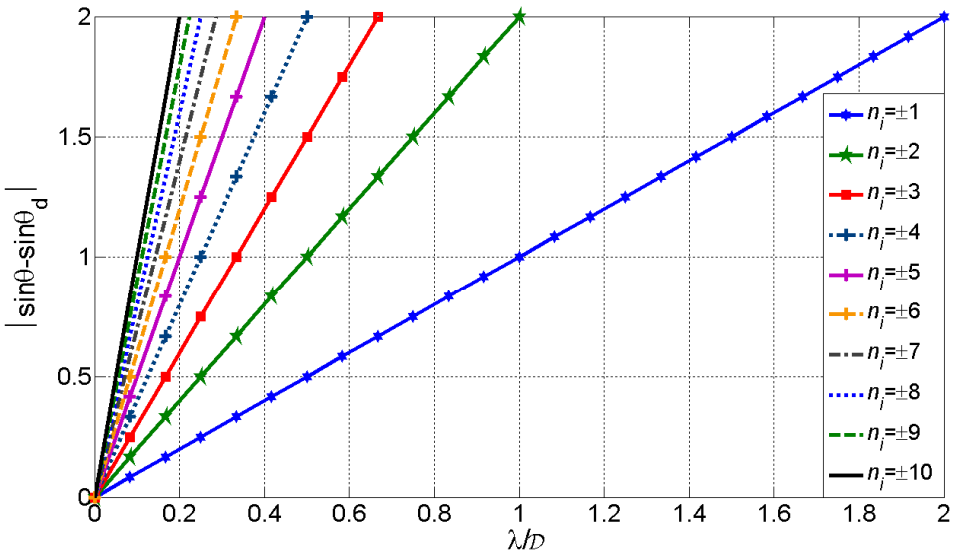


Figure 2-6 The angular locations, θ, of the grating lobes, θ, for an array with inter-element spacing $\mathcal{D}$ and wave length λ when the desired main beam is directed to an angle θ_d.

As shown in Figure 2-7, the rectangular planar array is made up of N_x rows and N_y columns of antenna elements, with a total of $N_x \times N_y$ elements. Each row or column represents a linear array, with inter-element spacing of $\mathcal{D}_y$ and $\mathcal{D}_x$, respectively.

Now, the phasor sum of the signals from each individual row of elements can be expressed as [20]

$$y_x(t) = \sum_{i=1}^{N_x} x(t) e^{j(i-1)\psi_x} \tag{2.10}$$

and the array factor for each row becomes

$$AF_x = \sum_{i=1}^{N_x} e^{j(i-1)\psi_x} \tag{2.12}$$

The same applies to the phasor sum of the signals from each individual column of elements, so that the array factor for each column is given by

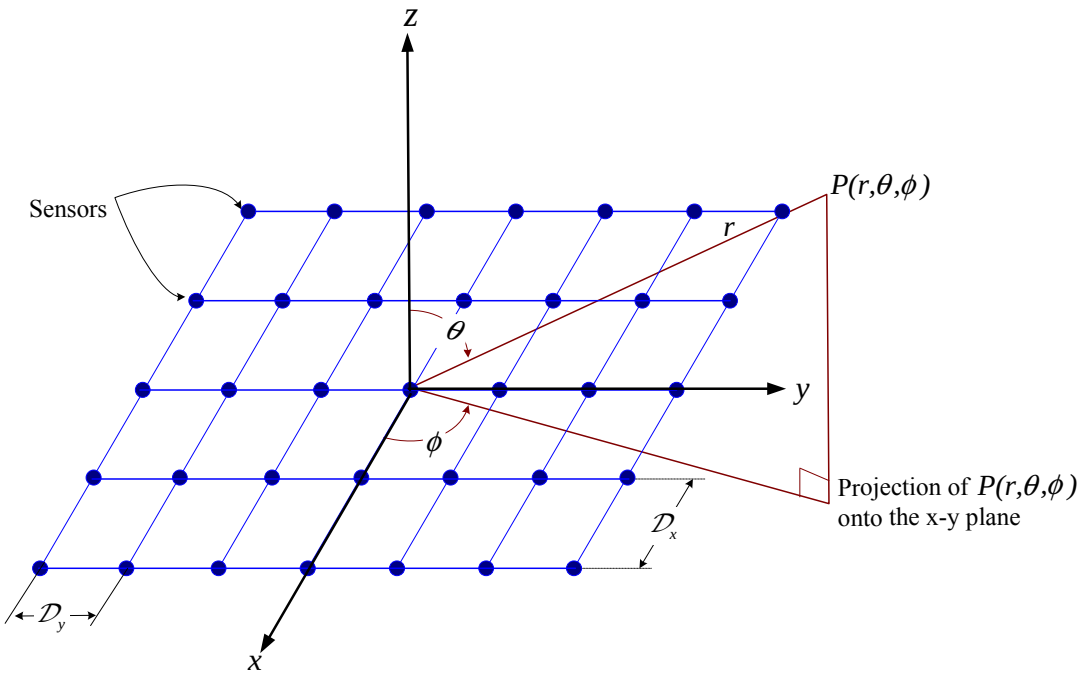


Figure 2-7 A rectangular planar array.

$$AF_y = \sum_{i=1}^{N_y} e^{j(i-1)\psi_y} \tag{2.13}$$

According to the principle of array multiplication, the overall array factor for the planar array is given by [20]

$$AF(\theta,\phi) = \sum_{i=1}^{N_x}\sum_{k=1}^{N_y} e^{j(i-1)\psi_x} e^{j(k-1)\psi_y} \tag{2.14}$$

where
$$\psi_x = \frac{2\pi \mathcal{D}_x}{\lambda}\sin(\theta)\cos(\phi)$$
$$\psi_y = \frac{2\pi \mathcal{D}_y}{\lambda}\sin(\theta)\sin(\phi) \tag{2.15}$$

Equation (2.15) shows that the array factor of a planar array is dependent on both the projected azimuth angle ϕ, and the elevation angle θ. Moreover, a planar array also suffers from array ambiguity and grating lobes, similar to

a linear array. Their occurrences depend on the inter-element spacings, $\mathcal{D}_x$ and $\mathcal{D}_y$. Figure 2-8 shows the beam pattern obtained from a 8×8 rectangular array with $\mathcal{D}_x = \mathcal{D}_y = \lambda/2$, and the beam is steered toward $\theta = \phi = 0^\circ$.

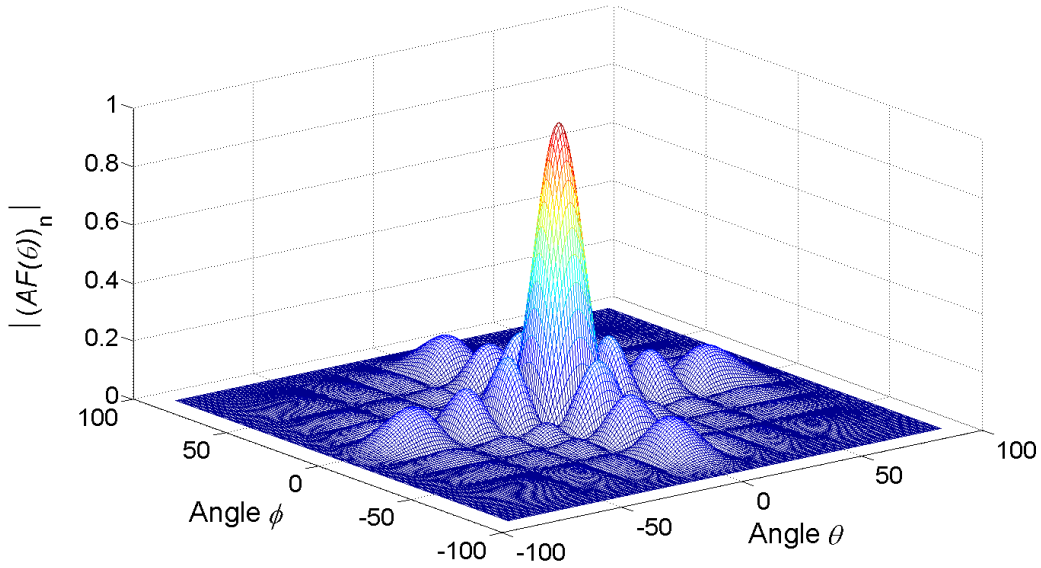


Figure 2-8 The beam pattern obtained with a rectangular planar array consisting of 8×8 elements with $\theta = \phi = 0^\circ$ and $\mathcal{D}_x = \mathcal{D}_y = \lambda/2$.

2.5 Circular Array

A circular array consists of N elements that are placed in a circular ring of radius *R*, as shown in Figure 2-9. For a uniform circular array (UCA), these elements are equally spaced [16]. When compared with an ULA of the same number of elements and inter-element spacing, the beam produced by a circular array is wider [25]. However, a UCA does not suffer from ambiguity in the beam direction and is able to provide a full azimuthal coverage [26]. This makes the UCA suitable for applications in surveillance and cellular communications where signals can arrive from any azimuth angle. On the other hand, circular arrays are normally associated with higher side lobe levels [24], and coupling between highly correlated multipath signals [26].

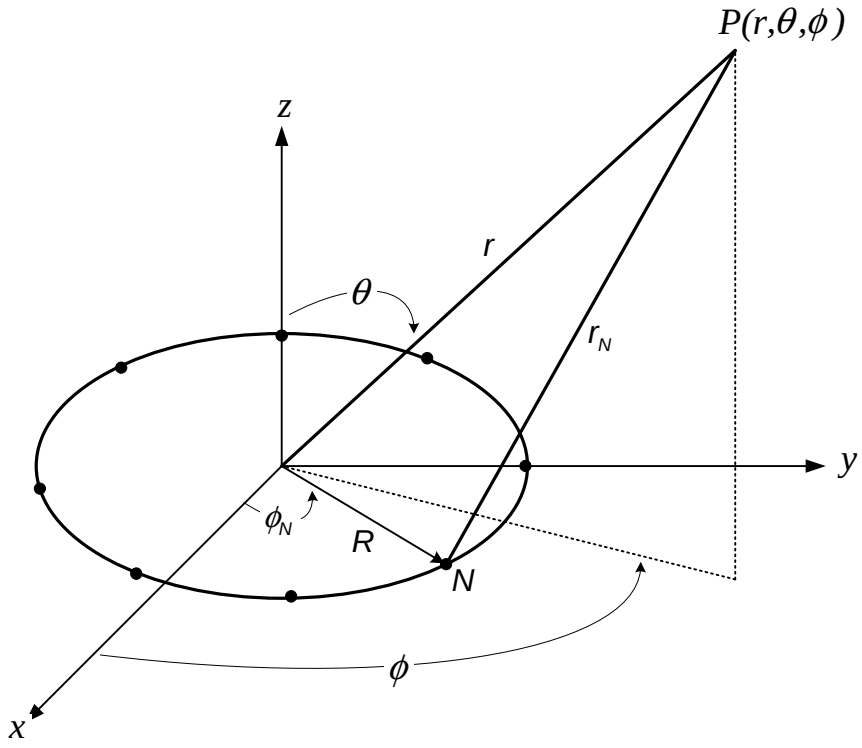


Figure 2-9 Geometry of an *N*-element circular array [24].

Now, consider the UCA of Figure 2-9. It has N elements equally spaced over a circle of radius *R* in the $x-y$ plane. As shown, $\theta \in [0, 2\pi]$ is the elevation angle measured from the z-axis, and $\phi \in [0, 2\pi]$ is the azimuth angle measured counterclockwise from the $x-$axis on the $x-y$ plane [26]. The array factor is then given by [27]

$$AF_{UCL}(\theta, \phi) = \sum_{k=1}^{N} e^{j2\pi \frac{R}{\lambda} \sin(\theta) \cos\left(\phi - \frac{2\pi k}{N}\right)} \tag{2.16}$$

If the elevation angle is $\theta = \pi/2$, then the array factor in (2.16) becomes [24]

$$AF_{UCL}(\phi) = \sum_{k=1}^{N} e^{j2\pi \frac{R}{\lambda} \cos\left(\phi - \frac{2\pi k}{N}\right)} \tag{2.17}$$

Figure 2-10 shows an example of the normalized array factor, $\left|\left(AF_{UCL}(\theta_d)\right)_{\mathrm{n}}\right|$, obtained with an 8-element uniform circular array of radius $R=\lambda$, where the beam is steered toward $\theta=30^{\circ}$ and $\phi=0^{\circ}$. It is observed that the maximum side lobe amplitude is 0.73 that of the main beam. This side lobe level is higher than for the planar array of Figure 2-8, which indicates that maximum side lobe level is 0.29 of the main beam.

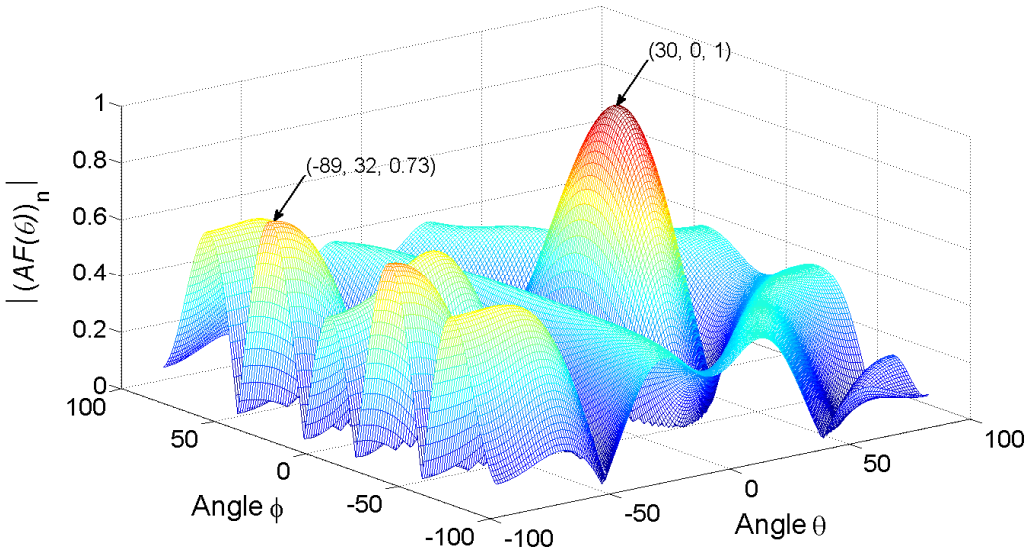


Figure 2-10 The beam pattern obtained with an 8-element uniform circular array of radius $R=\lambda$ and the main beam is directing toward $\theta=30^{\circ}$, $\phi=0^{\circ}$.

2.6 Beamforming and Spatial Filtering

A beamformer is a signal processor which can be used in conjunction with the array elements to provide a flexible form of spatial filtering [15]. The signal samples collected spatially by individual antenna elements are appropriately weighed so that the resultant beam is directed to the AOA of the desired signal, with nulls occurring at the directions of the interfering signals. The resulting output of the beamformer thus contains the desired

signal with the interfering signals greatly suppressed irrespective of their frequency and time [28].

In a mobile radio system, the AOA of a desired signal arriving at a base station is time varying. In this case, it requires that the individual weights of the beamformer to be automatically updated in order to adaptively steer its beam towards the desired direction. A detailed description on adaptive beamforming algorithms will be presented in Chapter 3.

Now, consider the following example of a beamformer consisting of a three-element linear array with an inter-element spacing of $\lambda/2$ as shown in Figure 2-11. It is assumed that the desired signal, $s(t)$, and two equal amplitude cochannel interfering signals, $i_1(t)$ and $i_2(t)$, are arriving from 0°, 30° and -60°, respectively. The output from each of the three antenna elements is passed through a complex weight, i.e., w_1, w_2 and w_3. The resultant output, $s(t)$, is obtained by summing the outputs of the three complex weights. Thus, the array output due to the desired signal can be expressed as

$$y_s(t) = A_s e^{j2\pi f_c t}\left(w_1 + w_2 + w_3\right) \tag{2.18}$$

where A_s is the signal amplitude, f_c is the frequency, and w_1, w_2 and w_3 are the three complex weights of the beamformer.

To recover the desired signal from (2.18), based on the null steering technique [18, 25, 29, 30], it is necessary that

$$w_1 + w_2 + w_3 = 1 \tag{2.19}$$

With respect to element 1, the interfering signals, $i_1(t)$ and $i_2(t)$, arrived at element 2 experience phase shifts of $2\pi d/\lambda \sin(30) = \pi/2$ and $2\pi d/\lambda \sin(-60) = -\sqrt{3}\pi/2$, respectively. Similarly, $i_1(t)$ and $i_2(t)$ at element 3 experience phase shifts of π and $-\sqrt{3}\pi$, respectively. As a result, the array output due to $i_1(t)$ is

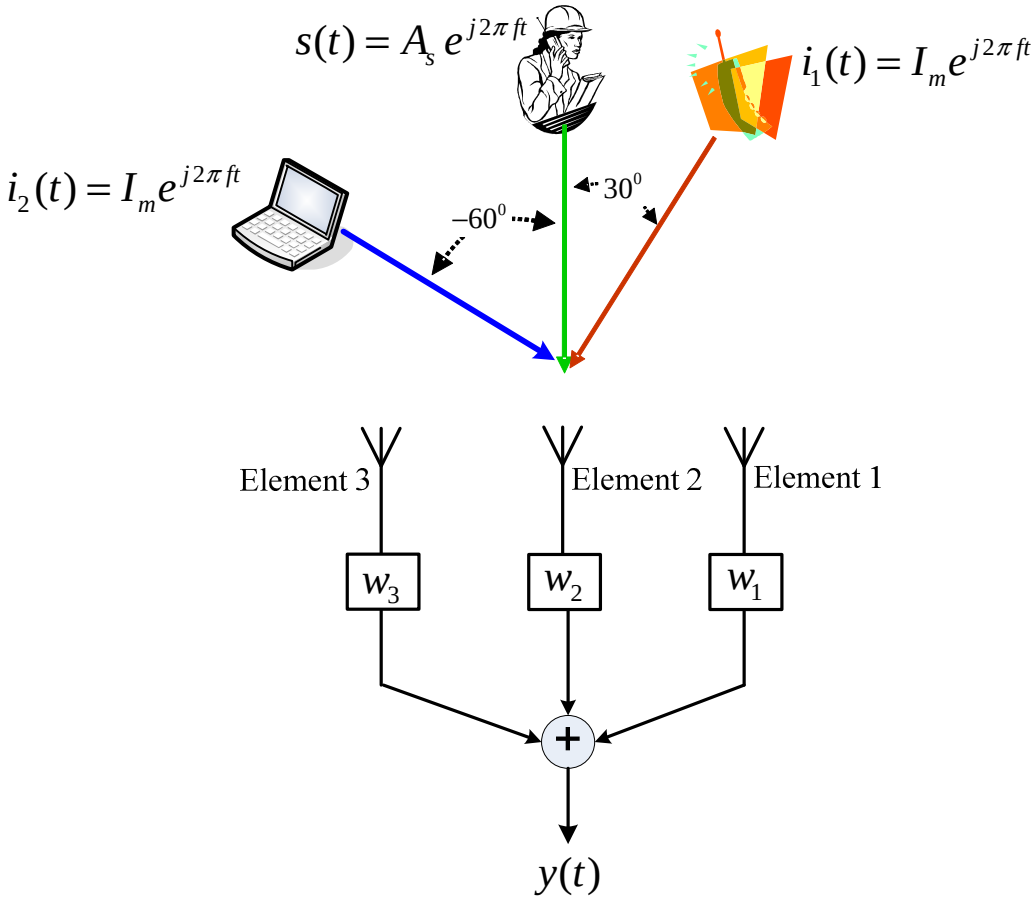


Figure 2-11 A three element array for interference suppression

$$\begin{aligned} y_{i1}(t) &= I_m e^{j2\pi ft} w_1 + I_m e^{j(2\pi ft+\pi/2)} w_2 + I_m e^{j(2\pi ft+\pi)} w_3 \\ &= I_m e^{j2\pi ft}\left[w_1 + jw_2 - w_3\right] \end{aligned} \tag{2.20}$$

and that due to $i_2(t)$ is

$$\begin{aligned} y_{i2}(t) &= I_m e^{j2\pi ft} w_1 + I_m e^{j(2\pi ft-\sqrt{3}\pi/2)} w_2 + I_m e^{j(2\pi ft-\sqrt{3}\pi)} w_3 \\ &= I_m e^{j2\pi ft}\left[w_1 + (-0.9127 - j0.4086)\, w_2 + (0.6661 + j0.7458)\, w_3\right] \end{aligned} \tag{2.21}$$

where I_m is the amplitude of $i_1(t)$ and $i_2(t)$.

In order to suppress both $i_1(t)$ and $i_2(t)$ in the array output, we need to set the weights of the beamformer such that

$$w_1 + jw_2 - w_3 = 0 \tag{2.22}$$

$$w_1 + (-0.9127 - j0.4086)w_2 + (0.6661 + j0.7458)w_3 = 0 \tag{2.23}$$

Solving equations (2.19), (2.22) and (2.23) yields

$$\begin{aligned} &w_1 = 0.3034 - j0.1966, w_2 = 0.3932 \quad \text{and} \\ &w_3 = 0.3034 + j0.1966 \end{aligned} \tag{2.24}$$

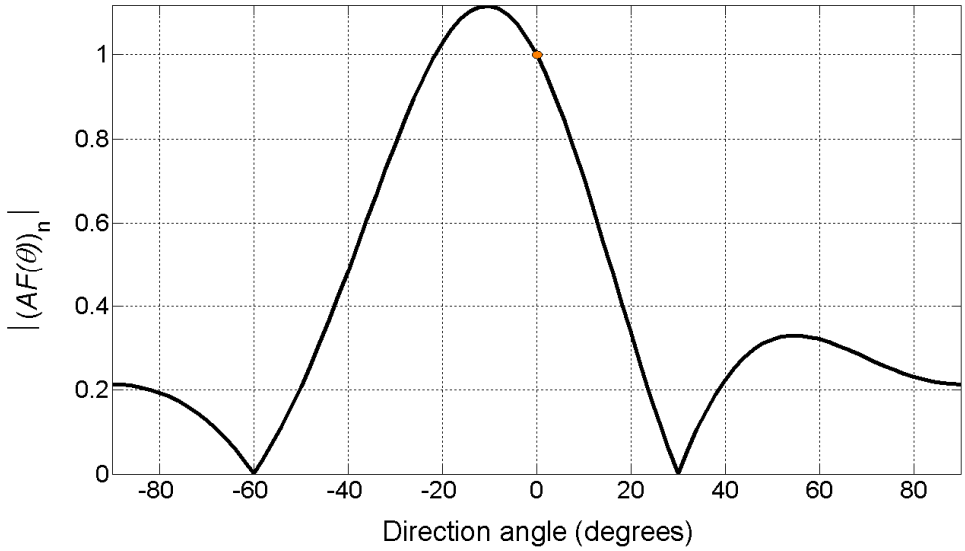


Figure 2-12 The beam pattern obtained from a 3-element uniform linear array of inter-element spacing $d = \lambda/2$. The angles of arrival of the desired, and two interfering signals are $\theta_s = 0^\circ$, $\theta_{i1} = 30^\circ$ and $\theta_{i1} = -60^\circ$, respectively.

The beam pattern of the array beamformer based on the weights given in equation (2.24) is shown in Figure 2-12. As expected, the beamformer produces unity gain for the desired user and a null at each of the directions of the two interferers. Moreover, this example also shows this type of

beamformer does not produce the maximum gain at the direction of the desired signal.

2.7 Summary

This chapter provides an introduction to the basic principles of antenna arrays and digital beamforming. It begins with a description of a uniform linear array and its characteristics, including the array factor, array ambiguity, and grating lobes. This is followed by a brief discussion on both the planar array and circular array. The latter is used in applications where array direction ambiguity is undesirable. Finally, it is shown that an array of antenna elements can be combined with a beamformer to electronically steer the beam to the direction of the desired signal while at the same time producing null response for the interfering signals. A detailed literature review of the adaptive beamforming algorithms is presented in Chapter 3.

CHAPTER 3

OVERVIEW OF ADAPTIVE BEAMFORMING ALGORITHMS

3.1 Introduction

During the past three decades, there has been much interest in adaptive systems. This led to the widespread use of adaptive techniques in various fields, such as wireless communications, signal processing, sonar, radar and biomedical engineering. An adaptive system is able to continuously adjust its system parameters in response to changes in the operating conditions so as to maintain its operation in an optimal manner following a reference signal. In the case of digital beamforming, the system parameters are the tap weights. Principally, the performance of an adaptive algorithm is highly dependent on the reference input and additive noise statistics. In the context of Wiener filter theory, there are assumptions of time invariance, linearity and Gaussian noise. Under these conditions, the mean square error criterion becomes an optimal cost function. These assumptions are often used to ease mathematical analysis, which often does not take into account the broader problems of non-Gaussian signals.

In digital communication systems, efficient bandwidth utilisation is economically important for maximising profits for the service providers, while at the same time it must be able to still maintain the required performance and reliability. Innovative techniques are being introduced into modern cellular mobile communication systems to meet these varied requirements. This includes the use of adaptive antenna arrays at base stations to provide space division multiple access (SDMA) [31] as a mean to realise increased system capacity. This chapter reviews some of the algorithms which are commonly used in adaptive array beamforming. Among the many algorithms,

the least mean square (LMS) based algorithms offer a relatively simple adaptive array beamforming solution. However, the performance of these algorithms often depends on the actual step size adaptation process. Also, since these algorithms make use of LMS processing, their operations are influenced by the characteristics of the input signals. On the other hand, adaptive beamformers based on recursive least square (RLS) algorithms tend to offer faster eigenvalue independent convergence. In other words, the convergence of the RLS algorithm, in term of the mean square error, is independent of the eigenvalues of the correlation matrix of the input signal vector [32-34]. Moreover, the proper operations of the LMS and RLS based algorithms require that a clean reference signal be provided.

In general, beamforming algorithms can be categorized into two classes, as shown in Figure 3-1, namely non-blind and blind algorithms. In the case of non-blind adaptive algorithms, a reference signal is used in the process of adjusting the array weights. On the other hand, no reference signal is used in blind adaptive algorithms. However, when compared with their non-blind counterparts, these algorithms tend to be more computation intensive, and often provide lower accuracy and slower convergence rate [24, 35]. Adaptive array beamforming has been adopted in some recently launched cellular systems, such as worldwide interoperability for microwave access (WiMAX) [3], and long term evolution (LTE) [5]. For these systems, a reference signal is already available in the form of a pre-ample sequence [36]. As such, the use of a non-blind algorithm for the array beamformer is thus an appropriate choice.

3.2 Non-Blind Algorithms

With non-blind algorithms, the weights of the array beamformer are usually adapted according to a specified criterion, such as minimization of mean square error (MMSE), or maximization of the signal to interference plus noise signal (MSINR). An error signal, produced by comparing the output signal with a reference signal, is used to iteratively adjust the weights of the beamformer to their optimal values, $\boldsymbol{W}_{opt}$, so as to obtain the minimum MSE

[37]. The trained algorithms could be classified according to their adaptive criterion: least-mean squares method (LMS), sample matrix inversion (SMI) or least-squares method (LS), and recursive least-squares method (RLS) [38, 39].

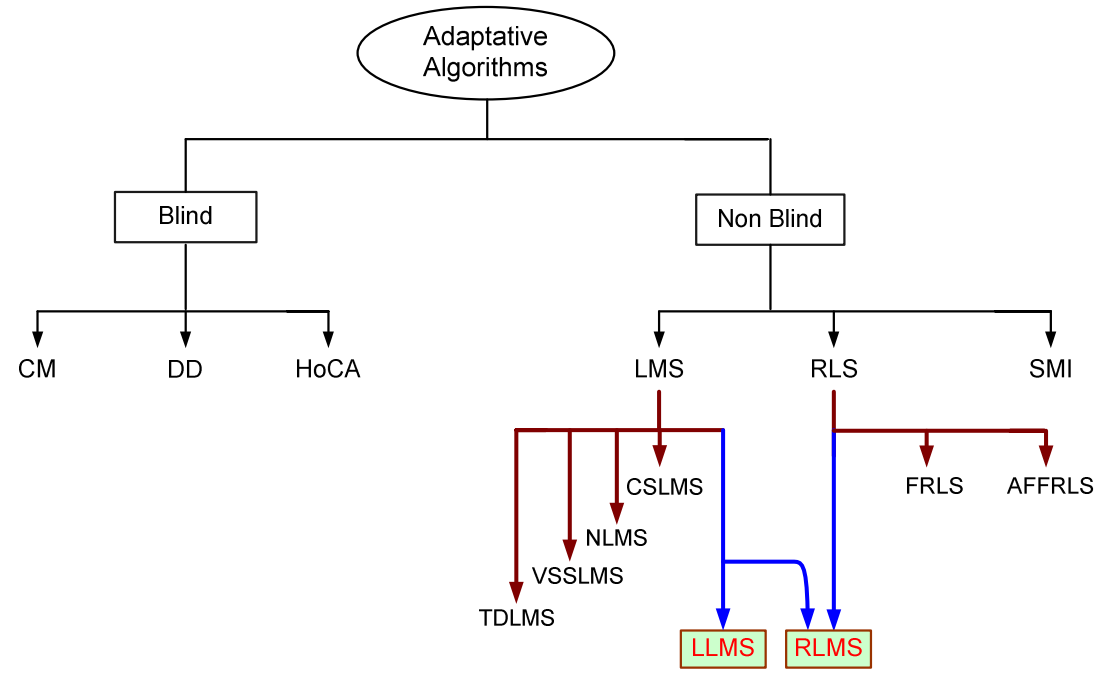


Figure 3-1 General classification of adaptive algorithms with some examples

3.2.1 LMS algorithm

The LMS algorithm was first proposed by Widrow and Hoff in [40] as an implementation of the steepest-descent based approach [41] to estimate the gradient of the error signal. It is computationally efficient, but is a bit slow in

convergence [34]. Also, its convergence is dependant on the eigenvalue spread variation of the input signal [6]. When an LMS algorithm is adopted in the implementation of an N-element array, the computation cost is in the order of $O(2N+1)$ multiplications [41].

Consider the LMS adaptive array as shown in Figure 3-2 [42]. According to the method of steepest-descent, the updating of the weight vector is carried out in such a way that minimises the error signal, given by

$$e(n) = d(n) - y(n) \tag{3.1}$$

where $d(n)$ is the zero-mean reference signal, and $y(n)$ is the output signal of the beamformer, such that

$$y(n) = \boldsymbol{W}^H(n)\boldsymbol{X}(n) \tag{3.2}$$

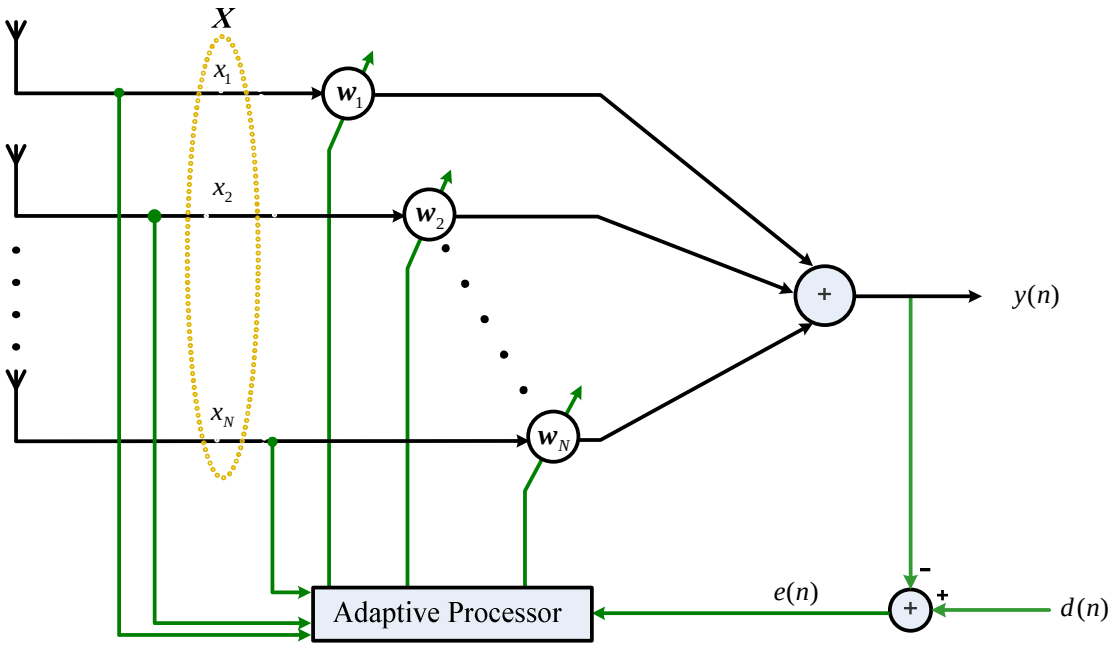


Figure 3-2 An LMS Adaptive Array

where $\boldsymbol{X}(n)$ is a N×1 complex vector of the received signal, which is assumed to have zero mean; $\boldsymbol{W}(n)$ is the complex weight vector having the same size as $\boldsymbol{X}(n)$, and $(\bullet)^H$ denotes the Hermitian (i.e., transpose and conjugate) matrix of $(\bullet)$.

For the LMS algorithm, the weight vector $\boldsymbol{W}(n)$ is adjusted based on the minimization of the cost function

$$\xi(n) = \mathrm{E}\left[e^2(n)\right] \tag{3.3}$$

where $\mathrm{E}[\ ]$ stands for the expectation operator.

Now, from equations (3.1) and (3.2), we can rewrite equation (3.3) as

$$\begin{aligned}\xi(n) &= \mathrm{E}\left[\left(d(n)-y(n)\right)^2\right] = \mathrm{E}\left[\left(d(n)-\boldsymbol{W}^H(n)\boldsymbol{X}(n)\right)^2\right] \\ &= \mathrm{E}\left[\left(d(n)-\boldsymbol{W}^H(n)\boldsymbol{X}(n)\right)\left(d^*(n)-\boldsymbol{X}^H(n)\boldsymbol{W}(n)\right)\right] \\ &= \mathrm{E}\left[\left|d(n)\right|^2\right]-\boldsymbol{Z}^H(n)\boldsymbol{W}(n)-\boldsymbol{W}^H(n)\boldsymbol{Z}(n)+\boldsymbol{W}^H\boldsymbol{Q}(n)\boldsymbol{W}(n)\end{aligned} \tag{3.4}$$

where * stands for conjugate operator, $\boldsymbol{Z}(n)$ corresponds to the input signal cross-correlation vector given by

$$\boldsymbol{Z}(n) = \mathrm{E}\left[d^*(n)\boldsymbol{X}(n)\right] \tag{3.5}$$

and $\boldsymbol{Q}(n)$ is an $N \times N$ correlation matrix of the input signals, that is

$$\boldsymbol{Q}(n) = \mathrm{E}\left[\boldsymbol{X}(n)\boldsymbol{X}^H(n)\right] \tag{3.6}$$

Equation (3.4) represents a quadratic cost function involving the weight vector $\boldsymbol{W}$. Hence, for the cost function of equation (3.3) to be minimized, we differentiate equation (3.4) with respect to $\boldsymbol{W}^H$ to obtain

$$\nabla\left(\xi(n)\right)=\frac{\partial}{\partial \boldsymbol{W}^{H}}\mathrm{E}\left[\left|e(n)\right|^{2}\right]=-2\boldsymbol{Z}+2\boldsymbol{Q}\boldsymbol{W}_{opt} \tag{3.7}$$

As a result, the optimal $\boldsymbol{W}$ is obtained when

$$-\boldsymbol{Z}+\boldsymbol{Q}\boldsymbol{W}_{opt}=0 \tag{3.8}$$

or equivalently

$$\boldsymbol{Q}\boldsymbol{W}_{opt}=\boldsymbol{Z} \tag{3.9}$$

Equation (3.9) is called the Wiener-Hopf equation. Multiplying both sides of the equation (3.9) by $\boldsymbol{Q}^{-1}$, the inverse of the correlation matrix, we obtain the optimal weight vector as

$$\boldsymbol{W}_{opt1}=\boldsymbol{Q}^{-1}\boldsymbol{Z} \tag{3.10}$$

Equation (3.10) shows that the computation of the optimum weight vector $\boldsymbol{W}_{opt}$ would require knowledge of the correlation matrix $\boldsymbol{Q}$ of the input data vector $\boldsymbol{X}(n)$, and the cross-correlation vector $\boldsymbol{Z}$, between the input data vector $\boldsymbol{X}(n)$ and the reference signal $d(n)$. Moreover, the main disadvantage of using equation (3.10) to obtain the optimal weight vector is the complexity of obtaining the inverse of the covariance matrix $\boldsymbol{Q}$, particularly for the case involving a large number of antenna elements. It is also possible that the covariance matrix $\boldsymbol{Q}$ may be singular or ill-conditioned.

A more attractive way to find $\boldsymbol{W}_{opt}$ is to use an iterative approach, in which the weight vector is updated according to [19]

$$\boldsymbol{W}(n+1)=\boldsymbol{W}(n)-\mu\boldsymbol{\nabla}\left(\xi(n)\right) \tag{3.11}$$

where μ is a small constant, usually referred to as the step size.

By substituting equation (3.7) into equation (3.11), we obtain

$$\boldsymbol{W}(n+1)=\boldsymbol{W}(n)-\mu\left[\boldsymbol{Q}(n)\boldsymbol{W}(n)-\boldsymbol{Z}(n)\right] \tag{3.12}$$

Until now, the correlation matrix $\boldsymbol{Q}$ and the cross correlation $\boldsymbol{Z}$ are assumed well known. In real world applications, the exact values of these quantities are not available. Therefore, these two quantities have to be estimated from available signals; i.e., based on current samples. The LMS criterion [43] makes use of simplified estimations of $\boldsymbol{Q}$ and $\boldsymbol{Z}$, such that

$$\begin{aligned} \boldsymbol{Q}(n) &= \boldsymbol{X}(n)\boldsymbol{X}^H(n) \\ \boldsymbol{Z}(n) &= d^*(n)\boldsymbol{X}(n) \end{aligned} \tag{3.13}$$

With these approximations, equation (3.7) becomes

$$\begin{aligned} \nabla\left(\xi(n)\right) &= \boldsymbol{X}(n)\boldsymbol{X}^H(n)\boldsymbol{W}(n) - d^*(n)\boldsymbol{X}(n) \\ &= \boldsymbol{X}(n)\left(\boldsymbol{X}^H(n)\boldsymbol{W}(n) - d^*(n)\right) \\ &= -e^*(n)\boldsymbol{X}(n) \end{aligned} \tag{3.14}$$

Form equations (3.14) and (3.12), the LMS algorithm may be defined as

$$\begin{aligned} y(n) &= \boldsymbol{W}^H(n)\boldsymbol{X}(n) \\ e(n) &= d(n) - y(n) \\ \boldsymbol{W}(n+1) &= \boldsymbol{W}(n) + \mu e^*(n)\boldsymbol{X}(n) \end{aligned} \tag{3.15}$$

3.2.2 LMS family of algorithms

The low computation complexity and robustness of the LMS algorithm have made it very popular in various applications, including adaptive antenna arrays. However, with an LMS algorithm, it is not possible to enhance both the convergence speed and lower the steady state error floor simultaneously [44]. This is due to the fact that when a larger step size is chosen, the algorithm converges quicker but with a larger residual error floor. On the other hand, the use of a smaller step will lead to slower convergence and lower steady state error floor. Since then, many modifications have been proposed in the literature to try to overcome the compromise between convergence speed and error floor of the conventional LMS algorithm. Most

of these modified LMS algorithms make use of some criteria to regulate the step size value. For example, an initial large adaptation step size could be used to speed up the convergence. When close to the steady state, smaller step sizes are then introduced to decrease the level of adjustment, hence maintaining a low error floor. Table 3-1 tabulates some of the modified algorithms belonging to the LMS family of algorithms, and the criteria used in calculating the step sizes.

Table 3-1 Some of the LMS family of algorithms

Category	**Algorithm Reference**	**Step Size Criteria**	**Complexity of Step size calculation**
Conventional LMS	LMS [45]	Constant Step Size	*N multiplications*
Transform Domain	DCT and DFT [46, 47]	Ortho-transformation of $\boldsymbol{X}$	*N multiplications*
Adaptive Step Size	NLMS [41, 48-51]	$\sigma_x^2(n)$ and $\sigma_W^2(n)$	*N+2 multiplications* *5 divisions* *4 additions*
	KLMS [48]	$\sigma_x^2(n)$ and $e(n)$	*N+1 multiplications* *2 divisions* *1 additions*
	Affine LMS [52, 53]	$\sigma_x^2(n)$, $\sigma_W^2(n)$ and σ_n^2	*N+2 multiplications* *1 divisions*
Error based step size adaptation	[54, 55]	$e^2(n)$	*N+5 multiplications* *1 divisions* *1 additions*
	[10, 56]	$R_e{}^2(n)$	*8N multiplications* *3 additions*
	[57]	$R_e{}^2(n)$ for all *N* Lags errors	*8N multiplications* *3N additions*
	[58]	$sign(e(n))$	*2N multiplications* *2 additions*
Error and Input Signal based step size adaptation	[59, 60]	$\boldsymbol{X}\, e$	*N+2 multiplications* *1 additions*
	[61-63]	$e(n)e(n-1)\times \boldsymbol{X}^H(n-1)\boldsymbol{X}(n)$	*N+3 multiplications* *1 additions*
Error, Input Signal, and Weights based step size adaptation	[9]	$\delta\boldsymbol{W}(n)=\boldsymbol{W}(n)-\boldsymbol{W}(n-1)$ $\delta e^{[n]}(n)=e^{[n]}(n)-e^{[n]}(n-1)$	*N multiplications* *1 divisions* *2 additions*

3.2.2.1 Normalized least mean square algorithm

In normalized least mean square algorithm (NLMS) [64], the step size μ is adjusted in accordance with the input signal power, which is estimated according to the power of the input signal through the autocorrelation of the signal [49, 50]. In [41, 48], the step size μ at n^{th} iteration is given by

$$\mu(n) = \frac{\gamma_{\mathrm{N}}}{\left| \boldsymbol{X}^{H}(n)\boldsymbol{X}(n) \right|^{2}} \tag{3.16}$$

where γ_{N} is a convergence factor in the range $(0,1)$ introduced to insure the stability of the algorithm.

Another form of the NLMS to improve the convergence rate of the algorithm is presented in [51]. In this case, the convergence factor, γ_{N}, is updated according to

$$\gamma_{\mathrm{N}}(n+1) = \begin{cases} \gamma_{\mathrm{N,max}} & if \quad \gamma'_{\mathrm{N}}(n+1) > \gamma_{\mathrm{N,max}} \\ \gamma_{\mathrm{N,min}} & if \quad \gamma'_{\mathrm{N}}(n+1) < \gamma_{\mathrm{N,min}} \\ \gamma'_{\mathrm{N}}(n+1) & otherwise \end{cases} \tag{3.17}$$

where $\gamma_{\mathrm{N,max}} = 1$ is chosen to assure fast convergence, $\gamma_{\mathrm{N,min}}$ is a fixed value specified to avoid too slow a convergence, and $\gamma'_{\mathrm{N}}(n+1)$ is calculated for the first element, x_1, of the input signal vector, $\boldsymbol{X}(n)$, as

$$\gamma'_{\mathrm{N}}(n+1) = \frac{R^{2}_{ey}(n+1)}{\overline{x}^{2}_{1}(n+1)} \tag{3.18}$$

where R^{2}_{ey} is the square of the cross-correlation between the output signal, $y(n)$, and the error signal, $e(n)$. $R^{2}_{ey}(n+1)$ and $\overline{x}^{2}_{1}(n+1)$ are updated according to

$$R^{2}_{ey}(n+1) = \eta_{\mathrm{N}} R^{2}_{ey}(n) + \varepsilon^{2} e(n) y(n) \tag{3.19}$$

$$\overline{x}^{2}_{1}(n+1) = \eta_{\mathrm{N}} \overline{x}^{2}_{1}(n) + \varepsilon x^{2}_{1}(n) \tag{3.20}$$

where ε and η_{N} are constants, the values of which are related to the number of the signal snapshot, k. For example, $\varepsilon = 1/k$, and $\eta_{\mathrm{N}} \leq 1-\varepsilon$.

In [48], yet another form of NLMS is proposed by combining the classical Kalman algorithm with the previous NLMS algorithm in an attempt to improve the stability of the latter. In this case, the step size is updated based on both the input signal and the weights variation, such that

$$\mu(n) = \frac{1}{\sigma_x^2(n) + q_v / \sigma_W^2(n)}, \tag{3.21}$$

$$\sigma_W^2(n+1) = \sigma_W^2(n)\left(1 - \mu(n)\sigma_x^2(n)/N\right) + \sigma_n^2 \tag{3.22}$$

where $\sigma_x^2 = \boldsymbol{X}^H \boldsymbol{X}$, is the time varying estimation of the input signal power, σ_W^2 is the weight variance, σ_n^2 is the AWGN noise power, and q_v is a constant.

It is shown in [50] that the NLMS algorithm potentially converges faster than the conventional LMS algorithm because its parameters have been very carefully chosen, based on the statistics of a given input signal. Furthermore, the algorithm used for updating its step size, as given in equations (3.21) and (3.22), has a higher noise immunity than that presented earlier in equations (3.18) and (3.20).

From equations (3.16), (3.19), (3.20), (3.21) and (3.22), it is quite obvious that the convergence of the NLMS algorithm is highly influenced by the proper choice of many parameters [48]. In addition, it also performs poorly when the input signal power is low [48].

3.2.2.2 Affine combination of two LMS algorithms

It is proposed in [52, 53] that the outputs of two LMS algorithm stages, operating in parallel but with two different step size values, are linearly

combined together to form a final output. In this way, the output is able to converge quicker while reducing the steady-state excess mean-square error (EMSE). One such scheme is shown in Figure 3-3, in which two LMS stages operate independently with two different fixed step sizes. As shown, the top LMS stage is operating with a large step size, μ_t, and the bottom LMS stage makes use of a smaller step size, μ_b. These step size values are obtained according to

$$\mu_t = \frac{1}{(N + \mu_{\max})\sigma_x^2}$$
$$\mu_b = \mu_r \mu_t \tag{3.23}$$

where $\mu_{\max}$ represents the LMS step size for maximum convergence rate and μ_r controls the ratio between μ_t and μ_b. Stable operation of the algorithm is achieved whenever $0 < \mu_r < 1$ [52].

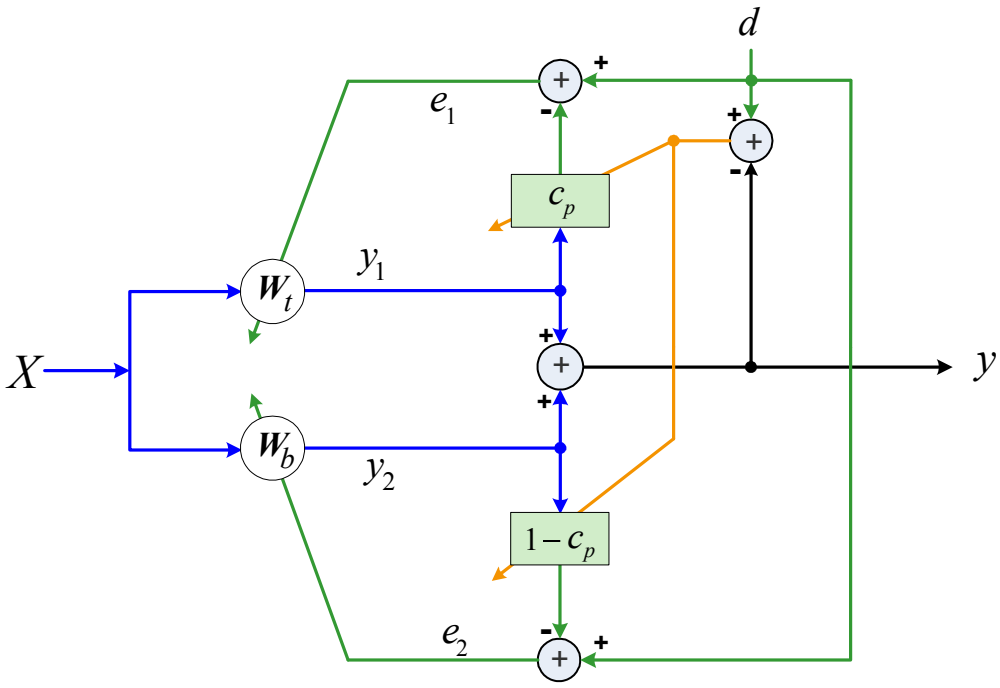


Figure 3-3 An affine combination of two LMS algorithm stages as proposed in [52]

The overall output, $y(n)$, is the affine combination of the individual outputs $y_1(n)$ and $y_2(n)$ of the two LMS stages, such that

$$y(n) = c_p(n)y_1(n) + \left[1 - c_p(n)\right]y_2(n) \tag{3.24}$$

where $c_p(n)$ is a combination parameter that can be updated using either a stochastic gradient method or an error power based scheme [52].

- Stochastic gradient method for updating $c_p(n)$:

  With this method, $c_p(n)$ is updated based upon a stochastic gradient search for the optimal $c_p(n)$, so that

  $$c_p(n+1) = c_p(n) + \mu_c \left[d(n) - \tilde{\boldsymbol{W}}_{tb}^H(n)\boldsymbol{X}(n) \right] \boldsymbol{W}_{tb}^H(n)\boldsymbol{X}(n) \tag{3.25}$$

  where μ_c is the adaptation step size of the combination parameter, and

  $$\tilde{\boldsymbol{W}}_{tb}(n) = c_p(n)\boldsymbol{W}_t(n) + \left[1 - c_p(n)\right]\boldsymbol{W}_b(n) \tag{3.26}$$

  where $\boldsymbol{W}_t$ and $\boldsymbol{W}_b$ are two different weight vector sets of length N each. $\boldsymbol{W}_{tb}(n)$ is given by

  $$\boldsymbol{W}_{tb}(n) = \boldsymbol{W}_t(n) - \boldsymbol{W}_b(n) \tag{3.27}$$

  Note that $c_p(n)$ obtained from the linear first order equation (3.25) is a scalar parameter. Moreover, a tradeoff between the stability and the tracking capability of the algorithm depends on the choice of μ_c. With $\mu_c < 1$, the operation of the algorithm is stable. However, the use of such a small value of μ_c is not efficient to track the adaptation of $\boldsymbol{W}_t(n)$ and $\boldsymbol{W}_b(n)$. The step size, μ_c, is also a function of the input signal to noise ratio (SNR), in which a larger SNR is associated with a larger μ_c, and

vice versa. This suggests that a proper choice of μ_c can only be made when an accurate estimate of the noise power could be obtained [52].

- Error power based scheme for updating $c_p(n)$:

  As discussed earlier, the ability to accurately estimate the value of μ_c places a limit on the usefulness of the stochastic gradient method. In order to avoid the need for an accurate estimate of the noise power, which affects the choice of μ_c, an error based method is used to update $c_p(n)$. In this case, $c_p(n)$ is calculated as a function of time averaged error powers of each of the two adaptive outputs, such that

$$c_p(n+1) = \mathrm{k}\left(\mathrm{erf}\left\{ \frac{\tilde{e}_t^2(n)}{\tilde{e}_b^2(n)} \right\} \right) \tag{3.28}$$

  where $\tilde{e}_t(n)$ and $\tilde{e}_b(n)$ are the time average of the instantaneous error signals of the two individual LMS stages. The constants $\mu_{\max}$ and μ_r are as defined in equation (3.23), and k is a parameter given by

$$\mathrm{k} = 1 - \frac{\mu_r}{2(\mu_r - 1)} \tag{3.29}$$

  Note that the errors $\tilde{e}_t$ and $\tilde{e}_b$ associated with equation (3.28) are readily available and do not need the estimation of the additive noise power.

In general, the calculation of $c_p(n)$ with either equation (3.25) or equation (3.28) calls for at least $3N+6$ multiplications, $N+1$ divisions and 8 additions, where N is the number of array elements [65]. As such, the use of an affine combination of two LMS algorithm would require $N^2+7N+12$ multiplications,

$N+2$ divisions and $2N+13$ additions. Furthermore, it is observed that the resultant MSE at steady state worsens with an increase in the number of elements, N [65]. Moreover, the results presented in [52] show that the operation of the affine combination of two LMS algorithms is very sensitive to the choice of the combination parameter c_p used, which in turn is dependent on the real signal signatures.

3.2.2.3 VSSLMS algorithms

With variable step size LMS (VSSLMS) algorithms, a large step size is initially used to accelerate the convergence rate. Then, as the algorithm approaches closer to the steady state, the step size is gradually reduced in order to achieve a lower residual error floor [10]. The update of the step size during iterations is normally carried out according to some signal parameters, such as signal power, error signal magnitude, or cross correlation of the error and input signals [61]. The strategies adopted for updating the step size in some recently published VSSLMS algorithms are discussed next.

The gradient adaptive VSSLMS algorithms proposed in [59, 60] make use of a time varying convergence parameter based on the cross-correlation between the input signal and the adaptation error, to adjust the step size value. A high correlation will result in a large step size and vice versa. For all these algorithms, the step size is updated a according to

$$\mu(n+1) = \frac{s_f}{\sigma_x^2}\rho(n) \tag{3.30}$$

and

$$\rho(n) = \alpha\rho(n-1) + (1-\alpha)\tilde{x}(n)e(n) \tag{3.31}$$

where σ_x^2 is the input signal power as defined in equation (3.21), α is a forgetting factor, s_f is the scaling factor, and $e(n)$ is the error given in

equation (3.15), $\tilde{x}(n)$ is the average of the input signal samples obtained from all the *N* antenna elements, and is defined as

$$\tilde{x}(n) = \frac{1}{N}\sum_{i=1}^{N} x_i(n) \tag{3.32}$$

One major drawback of these algorithms is their complexity, that could become too high for use in some applications, such as array beamforming [65]. Therefore, in order to simplify the implementation of equation (3.30), Kwong and Johnston [54], modify the equation so that it is only based on the estimated error between the algorithm output and the reference signal. For example, the step size adaptation process is based on

$$\mu(n+1) = \alpha\mu(n) + \gamma e^2(n) \tag{3.33}$$

with $0 < \gamma < 1$, and

$$\mu(n+1) = \begin{cases} \mu_{\max} \text{ ; if } \mu(n+1) > \mu_{\max} \\ \mu_{\min} \text{ ; if } \mu(n+1) < \mu_{\min} \\ \mu(n+1) \end{cases} \tag{3.34}$$

where $\mu_{\max}$ and $\mu_{\min}$ are the upper and lower bounds of the step size μ, respectively. Initially, the algorithm begins with the step size $\mu_{\max}$, which is the maximum allowable value for the MSE to remain stable. $\mu_{\max}$ is given in [66] as

$$\mu_{\max} \leq \frac{2}{3tr(\boldsymbol{Q})}, \tag{3.35}$$

where $tr(\bullet)$ denotes the trace of the matrix $(\bullet)$.

Note that equation (3.33) has the same form as equation (3.16). The two differ in that the former updates the step size based on the last step size value, while the latter, updates the step size based on the current signal information and the previous error.

The VSSLMS algorithm described above is highly sensitive to the additive noise, hence its performance deteriorates at low SNR [57]. Also, the performance of the algorithm is highly dependent on the choice of parameters, which in turn are influenced by input data [61]. Therefore, to overcome some of these limitations, another form of VSSLMS algorithm has been proposed in [61] that is less sensitive to sudden changes in the error level. In this case, step size adaptation is carried out based on the covariance of two successive error and input signal samples as given below:

$$\mu(n+1) = \mu(n) + \alpha e(n)e(n-1)\boldsymbol{X}^H(n-1)\boldsymbol{X}(n) \tag{3.36}$$

From the results published in [61], the constant α should be chosen within the range of 10^{-7} to 10^{-4} in order to ensure convergence of the algorithm. It is proposed in [62] that equation (3.36) may be simplified by taking only the real part of its second term to yield

$$\mu(n+1) = \mu(n) + \alpha \Re\left\{e(n)e(n-1)\boldsymbol{X}^H(n-1)\boldsymbol{X}(n)\right\} \tag{3.37}$$

Despite the modifications, the algorithm is still suffering from degradation at high noise levels [57, 67]. At low SNR condition, it is necessary to keep the adaptation parameter α used to update the step size small, and this in turn will slow down the convergence speed [44, 62].

In an attempt to overcome the slow convergence rate, Tyseer A. and K. Mayyas published in [56] a modification to the VSSLMS algorithm proposed by Kwong [54]. It involves the use of the square of a time-averaged estimate of the autocorrelation ($R_e(n)^2$) of successive error samples, instead of $e^2(n)$, to control the step size, so that

$$\mu(n+1) = \alpha\mu(n) + \gamma R_e^2(n) \tag{3.38}$$

$$R_e(n+1) = \beta R_e(n) + (1-\beta)e(n+1)e(n) \tag{3.39}$$

where the parameters α and γ, which are as defined in equation (3.33), are constants chosen to be equal to 0.97, 10^{-3}, respectively, and $\beta = 0.99$,

the same values as given in [56]. With this algorithm, it is assumed that the noise samples are non-correlated. However, for some applications, where the input data is correlated, the autocorrelation function between $e(n)$ and $e(n-1)$ is a poor indicator of convergence closeness. This may result in the use of a smaller step size, which will in turn affect the convergence rate [57]. Also, the steady state errors associated with the above algorithms tend to increase in the presence of uncertain measurement noise [68], thus limiting their applications [69].

The convergence rate and noise immunity of the above algorithm could be improved by calculating the autocorrelation of successive error samples based on all past *M* errors [57], such that

$$R_e(n+1) = \beta R_e(n) + (1-\beta)\sum_{i=1}^{M}\left[e(n)e(n-i)\right]^2 \tag{3.40}$$

The updating of the step size is the same as that of equation (3.38). Here, the values of α and β are as given in [54, 56], while γ takes on the value of 2×10^{-6}. It is shown in [57] that the performance, in terms of convergence rate and excess MSE, of this algorithm is superior to that obtained in [54] and [56]. However, this is achieved at the expense of a large increase in computation complexity.

More recently, the modified robust variable step size (MRVSS) algorithm, has been proposed in [10] to further improve both the noise immunity and tracking ability of the robust VSSLMS algorithms (RVSS) presented in [54] and [56]. In this case, the step size $\mu,$ is updated according to

$$\mu(n+1) = \begin{cases} \mu_{\max}\ ; & \text{if } \mu(n+1) > \mu_{\max} \\ \mu_{\min}\ ; & \text{if } \mu(n+1) < \mu_{\min} \\ \alpha\mu(n) + \gamma R_e^2(n) \end{cases} \tag{3.41}$$

with

$$R_e(n+1) = (1-\tilde{e}(n))R_e(n) + \tilde{e}(n)e(j)e(n-1) \tag{3.42}$$

and

$$\tilde{e}(n+1)=\begin{cases}\tilde{e}_{\max}\ ; & \text{if } \tilde{e}(n+1)>\tilde{e}_{\max}\\ \tilde{e}_{\min}\ ; & \text{if } \tilde{e}(n+1)<\tilde{e}_{\min}\\ \eta_e\,\tilde{e}(n)+\upsilon e^2(n)\end{cases} \tag{3.43}$$

where $0<\eta_e<1,$ and $0<\upsilon<1.$ $\tilde{e}$ is the time average of the square of the error signal with its upper and lower bounds given by $\tilde{e}_{\max}$ and $\tilde{e}_{\min}$, respectively. It is used to control the effect of the instantaneous error correlation (R_e). The use of the MRVSS algorithm requires five different parameters to be specified. The values of these parameters depend on particular system environments.

For all the different versions of the VSS algorithm discussed above, the magnitude of the estimated error plays a significant role in the step size adaptation process. Moreover, in the sign variable step size (SVSS) algorithm proposed in [58] only the sign of the error signal, $e(n)$, is considered for updating the step size, such that

$$\mu(n+1)=\alpha_{\exp}\mu(n)+\gamma sign(e(n)) \tag{3.44}$$

where $\alpha_{\exp}$ is an exponential forgetting constant with $0<\alpha_{\exp}<1$, and γ is a small positive constant that governs the amount of the adaptation of the step size. According to equation (3.44), the step size will be increased by an amount of γ in the case of a positive estimation error (underestimation), and vice versa. To ensure stability, the step size is only allowed to take on values within the range set by $\mu_{\max}$ and $\mu_{\min}$.

Thus far, we have shown that step size values are adapted in proportion to either the error signal or the input signal samples, or both. In the algorithm known as normalized square VSSLMS (NSVSSLMS) [69], however, the step size is adjusted based on the normalized square Euclidean norm of the smoothed gradient vector ($\tilde{\boldsymbol{g}}(n)$), given by

$$\tilde{\boldsymbol{g}}(n) = \alpha\tilde{\boldsymbol{g}}(n-1) + (1-\alpha)e(n)\boldsymbol{X}(n)$$

$$\mu(n) = \frac{\gamma\left\|\tilde{\boldsymbol{g}}(n)\right\|_2^2}{\left\{N\left[e^2(n) + \sigma_x^2(n)\right]\right\}^2} \tag{3.45}$$

where σ_x^2 is the time varying estimation of the input signal power, $\|\bullet\|_2^2$ denotes the squared Euclidean norm operator, and α and γ are as defined in equation (3.44). The choice of α and γ depends on the input signal conditions.

Since its steady state MSE is independent of noise, this algorithm is robust when operating in a nonstationary environment. However, on the other hand, its performance is highly dependent on the choice of α used which in turns depends on the required error floor. The complexity of the algorithm is much higher than that of the conventional LMS algorithm. For each step size iteration, it requires $N^4 + N^2 + 2N + 7$ multiplications, one division and $2N + 2$ additions, where N is the number of antenna elements.

3.2.2.4 Transform domain algorithms

The convergence rate of the LMS based algorithms considered thus far will degrade when the input samples are highly correlated [46]. To overcome this problem and as an alternative method to achieve higher convergence speed, LMS based algorithms may be implemented in the frequency domain [47] using a unitary orthogonal transformation. There are many transformations discussed in the literature, such as discrete Fourier transform (DFT), discrete cosine transform (DCT) [47], and discrete Walsh transform (DWT) [46]. However, these transform based algorithms are significantly more complex than the conventional LMS algorithm [70]. However, the complexity may be reduced using a scheme of partial coefficient updates [71, 72], which assumes that the output signal samples from the orthogonal transform are statistically independent. On the other hand, in the case of the transform domain LMS (TDLMS) algorithm, these samples could be highly

correlated. In this case, a high order hyperelliptic integral must be used which is going to lead to even greater complexity [73].

An accurate stochastic model is presented in [73] to simplify the high order hyperelliptic integrals for use in the estimation of the inverse of the time-varying power normalization parameter matrix $\hat{\boldsymbol{D}}^{-1}(n)$. Generally, for the TDLMS algorithm, the weight vector is updated according to

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + 2\hat{\boldsymbol{D}}^{-1}(n)\mu e(n)\boldsymbol{X}(n) \tag{3.46}$$

where the error $e(n)$ is given by

$$e(n) = d(n) - \boldsymbol{W}^{H}(n)\boldsymbol{X}(n) + z(n) \tag{3.47}$$

In equation (3.47), $z(n)$ represents the measurement noise which is independent and identically distributed (iid) with zero mean and variance of σ_z^2, and $\hat{\boldsymbol{D}}(n)$ is given by

$$\hat{\boldsymbol{D}}(n) = \begin{bmatrix} \hat{\sigma}_0^2 + \varepsilon_r & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \hat{\sigma}_{N-1}^2 + \varepsilon_r \end{bmatrix} \tag{3.48}$$

where $\hat{\sigma}_i^2$ is the estimated power of the i^{th} output signal of the transformation block and ε_r is a small positive constant (regularization parameter) which is used to avoid possible division by zero during the process of matrix inversion of $\hat{\boldsymbol{D}}(n)$.

It is noted in [73] that the computation of $\hat{\boldsymbol{D}}^{-1}(n)$, the inverse of $\hat{\boldsymbol{D}}(n)$, will require approximately $N^4 + N^2 + 4$ multiplications. Moreover, the choice of the constant ε_r is also becoming difficult when the input signal power is very low. Under such an operating condition, the output signal will not be a reliable estimate of the required signal. Also, from the results presented in [74], the beamwidth of the resultant beam pattern obtained with a transformed domain algorithm is wider than that of a standard LMS algorithm. This in turn will lead

to a decrease in signal to interference ratio (SIR) when the interfering signals are arriving from directions close to that of the desired signal.

3.2.2.5 Variable tap weight length LMS algorithm

Another way of reducing the complexity of VSSLMS algorithms is to vary the number of tap weights. However, an underestimate of the tap length may increase the error floor. It is therefore necessary to find an appropriate tap length which will result in a balanced trade off between complexity and error convergence [63]. In these kinds of algorithms, the length of the tap weights can be adjusted based on different criteria. For example, in the algorithm presented in [75], the tap length is divided into several segments, and the actual number of taps used is adjusted by adding or removing one segment at a time based on the error level. In this case, the step size used for weight update is fixed. Alternatively, for the algorithm presented in [63], both the step size and the tap length are updated simultaneously at each iteration. For this algorithm, the step size value is proportional to the ratio between the excess mean square error and the MSE.

3.2.2.6 Constrained stability LMS algorithm (CSLMS)

Another technique to improve the performance of an LMS algorithm, called the constrained stability LMS algorithm (CSLMS), has been proposed in [9] for enhancing the performance of the LMS algorithm. Similar to the NSVSSLMS, as represented by equation (3.45), CSLMS is also based on the minimization of the squared Euclidean norm of the weight vector under a stability constraint over the posterior estimation errors. This approach introduces a nonlinear relationship between the input data vector and the error sequence, rather than optimizing the step size for updating the weight vector. In other words, the adaptation process is used to arrive at a solution to a constrained optimization problem (to smooth the error sequences), by making use of Lagrangian formulation to minimize the norm of the difference

between two consecutive weight vectors ($\delta \boldsymbol{W}(n)$). The weight adaptation process is based on the following:

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \frac{\mu}{\left\|\delta \boldsymbol{W}(n)\right\|^2 + \varepsilon_{cs}} \delta \boldsymbol{X}(n)\left(\delta e^{[n]}(n)\right)^* \tag{3.49}$$

where ε_{cs} is a small constant introduced to avoid possible division by zero. The variables $\delta \boldsymbol{W}(n)$, $\delta \boldsymbol{X}(n)$, $\delta e^{[n]}(n)$ and $e^{[k]}(n)$ are given by

$$\delta \boldsymbol{W}(n) = \boldsymbol{W}(n) - \boldsymbol{W}(n-1) \tag{3.50}$$

$$\delta \boldsymbol{X}(n) = \boldsymbol{X}(n) - \boldsymbol{X}(n-1) \tag{3.51}$$

$$\delta e^{[n]}(n) = e^{[n]}(n) - e^{[n]}(n-1) \tag{3.52}$$

$$e^{[k]}(n) = d(n) - \boldsymbol{W}^H(k)\boldsymbol{X}(n) \tag{3.53}$$

The adaptation process of the CSLMS algorithm is performed using the knowledge of the estimated error, weight and the input signal. Unlike the VSSLMS algorithms, the CSLMS algorithm does not require many parameters to be specified. On the other hand, however, it does require a large amount of memory for storing past weight and input signal vectors, as well as the past errors.

3.2.2.7 Least mean square-least mean square (LLMS) algorithm

A novel method for improving the convergence rate, as well as reducing the steady state error of a conventional LMS algorithm, is proposed in this thesis. This technique makes use of two LMS algorithm stages connected in series via an array vector. This new algorithm is called the LLMS algorithm. It will be described and analyzed in detail in Chapter 5.

3.3 Family of Recursive Least-Squares (RLS) Algorithms

Unlike the LMS algorithm which makes use of the steepest descent method to obtain the complex weight vector, the recursive least-squares (RLS) algorithm uses the method of least squares to adjust the weight vector [34, 76]. With the method of least squares, the weight vector is chosen based on the recursive minimization of the cost function, which consists of the sum of error squares over a specified window [77]. For an RLS algorithm, the weight vector is obtained by minimizing the following cost function [78]

$$\xi(n) = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left| e(n-i) \right|^2 \tag{3.54}$$

where $e(n)$ is the error signal as defined in equation (3.1), and $0 < \alpha_{\mathrm{RLS}} \leq 1$ is the exponential weighted factor called the forgetting factor, which gives exponentially less influence to the older error samples [78].

Note that the process of minimizing the cost function given in equation (3.54) is equivalent to finding the derivative of equation (3.54) with respect to $\boldsymbol{W}(n)$ and setting the result to zero. The final result is known as the normal equation [67] , which is expressed as

$$\boldsymbol{W}(n) = \boldsymbol{Q}^{-1}(n)\boldsymbol{Z}(n) \tag{3.55}$$

where $\boldsymbol{Q}(n)$ is the approximation at time n of the input signal auto-correlation matrix given by

$$\boldsymbol{Q}(n) = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \boldsymbol{X}(n)\boldsymbol{X}^{H}(n) \tag{3.56}$$

and $\boldsymbol{Z}(n)$ is the approximation at time n of the cross-correlation vector between the input signal and the reference signal $d^{*}(n)$

$$\boldsymbol{Z}(n) = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \boldsymbol{X}(n)d^{*}(n) \tag{3.57}$$

The recursive expression for updating the least square solution for $\boldsymbol{Q}(n)$ and $\boldsymbol{Z}(n)$ can be written as

$$\boldsymbol{Q}(n) = \alpha_{\text{RLS}}\boldsymbol{Q}(n-1) + \boldsymbol{X}(n)\boldsymbol{X}^{H}(n) \tag{3.58}$$

$$\boldsymbol{Z}(n) = \alpha_{\text{RLS}}\boldsymbol{Z}(n-1) + \boldsymbol{X}(n)d^{*}(n) \tag{3.59}$$

Applying the inversion lemma [79, 80] for equation (3.58), we obtain

$$\boldsymbol{Q}^{-1}(n) = \alpha_{\text{RLS}}^{-1}\boldsymbol{Q}^{-1}(n-1) - \frac{\alpha_{\text{RLS}}^{-2}\boldsymbol{Q}^{-1}(n-1)\boldsymbol{X}(n)\boldsymbol{X}^{H}(n)\boldsymbol{Q}^{-1}(n-1)}{1+\alpha_{\text{RLS}}^{-1}\boldsymbol{X}^{H}(n)\boldsymbol{Q}^{-1}(n-1)\boldsymbol{X}(n)} \tag{3.60}$$

where $\boldsymbol{Q}^{-1}$ is initialized by $\delta^{-1}\boldsymbol{I}$, with δ being a small positive constant, and $\boldsymbol{I}$ is an $N \times N$ unity matrix. Substituting equations (3.59) and (3.60) in equation (3.55) yields

$$\begin{aligned}\boldsymbol{W}(n) = &\left(\alpha_{\text{RLS}}^{-1}\boldsymbol{Q}^{-1}(n-1) - \frac{\alpha_{\text{RLS}}^{-2}\boldsymbol{Q}^{-1}(n-1)\boldsymbol{X}(n)\boldsymbol{X}^{H}(n)\boldsymbol{Q}^{-1}(n-1)}{1+\alpha_{\text{RLS}}^{-1}\boldsymbol{X}^{H}(n)\boldsymbol{Q}^{-1}(n-1)\boldsymbol{X}(n)}\right) \\ &\times\left(\alpha_{\text{RLS}}\boldsymbol{Z}(n-1) + \boldsymbol{X}(n)d^{*}(n)\right)\end{aligned} \tag{3.61}$$

Rearranging equation (3.61), yields

$$\begin{aligned}\boldsymbol{W}(n) = &\boldsymbol{Q}^{-1}(n-1)\boldsymbol{Z}(n-1) \\ &+\frac{\alpha_{\text{RLS}}^{-1}\boldsymbol{Q}^{-1}(n-1)\boldsymbol{X}(n)\left(d^{*}(n) - \boldsymbol{X}^{H}(n)\boldsymbol{Q}^{-1}(n-1)\boldsymbol{Z}(n-1)\right)}{1+\alpha_{\text{RLS}}^{-1}\boldsymbol{X}^{H}(n)\boldsymbol{Q}^{-1}(n-1)\boldsymbol{X}(n)}\end{aligned} \tag{3.62}$$

Using equation (3.55) to replace the first term on the right hand side (RHS) of equation (3.62) yields

$$\boldsymbol{W}(n) = \boldsymbol{W}(n-1) + \boldsymbol{K}(n)\boldsymbol{X}(n)e(n) \tag{3.63}$$

where $\boldsymbol{K}(n)$ is the gain matrix given by

$$\boldsymbol{K}(n) = \frac{\alpha_{\text{RLS}}^{-1}\boldsymbol{Q}^{-1}(n-1)}{1+\alpha_{\text{RLS}}^{-1} + \boldsymbol{X}^{H}(n)\boldsymbol{Q}^{-1}(n-1)\boldsymbol{X}(n)} \tag{3.64}$$

and $e(n)$ is the a priori error such that

$$e(n) = d^*(n) - \boldsymbol{W}^H(n-1)\boldsymbol{X}(n) \tag{3.65}$$

Equations (3.63), (3.64) and (3.65) form the updating procedure for the RLS algorithm. An explanation on how to use these equations for simulating the RLS algorithm is given in Appendix A.

As shown in [81], the convergence speed of the RLS algorithm is about 10 times faster than that of the LMS algorithm. However, this is achieved at the expense of much greater complexity for the RLS algorithm, involving $4N^2 + 4N + 2$ complex multiplications where N is the number of antenna elements. Moreover, the superior performance of the RLS algorithm has motivated many researchers to search for ways to reduce the complexity of the RLS algorithm. Some of these techniques are:

- A Fast A posteriori Error Sequential Technique (FAEST) for sequential least-squares (LS) estimation is presented by Carayannis *et al* [82]. In this algorithm, the alternative Kalman gain, $\boldsymbol{K}(n)$, and a posteriori errors are used for updating the weight vector, $\boldsymbol{W}(n)$, instead of the $\boldsymbol{K}^*(n)$ and a priori errors as used in the original RLS algorithm. Such a modification reduces the complexity of the algorithm to only $7N$ multiplications.

- The correlation matrix, $\boldsymbol{Q}(n)$, of the input signal is replaced by a diagonal matrix whose elements are equal to the diagonal of the correlation matrix [81]. As a result, the inverse of the matrix $\boldsymbol{P}(n)$ can be obtained from the inverse of its diagonal elements, i.e., $\boldsymbol{D}_X = diag(\boldsymbol{Q})$, where $diag(\ )$ is the diagonal of the matrix. The modified algorithm is called the fast RLS algorithm (FRLS), which has a complexity comparable to that of the LMS algorithm.

- Splitting the RLS algorithm into several independent RLS sub-algorithms, as in the Hierarchical RLS (HRLS) algorithm proposed by Woo [83]. The idea here is to minimize the MSE of individual RLS sub algorithms. The complexity of the HRLS algorithm is equal to $N\times$

number of sub-groups, where the number of sub-groups is less than N. In [61], Minglu proposes the partial updating RLS algorithm (PURLS), which has even less complexity than the HRLS algorithm. With the PURLS algorithm, the adaptation blocks operate in an alternate fashion with each sub-group being updated partially, while those in the HRLS algorithm are operating simultaneously. As the adaptation process is being operated upon by only one of the sub-groups, the computation complexity of PURLS is reduced even more.

- The use of fast Euclidian direction search (FEDR) and recursive adaptive matching pursuit (RAMP) to arrive at a tradeoff between complexity and performance. For both the FEDR and RAMP algorithms, the updating process is performed for only one element of the weight vector at a time. The difference between these algorithms is in the way the index of the element of $\boldsymbol{W}(n)$ is being selected [84]. The complexity, however, depends on the number of elements used.

FRLS algorithms are known to suffer from an instability problem due to finite numerical precision which may give rise to a sudden divergence of the error signal. Several techniques have been proposed to handle this problem. Kim and Powers [85] examined the sources of the numerical instabilities and statistical properties of the FRLS algorithms. They then proposed a modified FRLS algorithm with soft constrained initialization parameters. These parameters are: initialization of the input signal vector with zero value, and adding a small constant of $\delta \boldsymbol{I}$ to the correlation matrix with $\delta = 0.0001$.

By incorporating the various techniques in an attempt to reduce the complexity of the RLS algorithm, the resultant modified algorithms tend to yield lesser performance and may encounter possible numerical instability [83, 84]. Over the past two decades, various RLS based algorithms have been proposed and some of these algorithms are presented in [85-87].

A drawback of the RLS algorithm is its lack of tracking ability [88]. To overcome this problem, Song and Sung [89] proposed a scheme called the adaptive forgetting factor RLS algorithm (AFFRLS) with which the weight vector is updated based on the gradient of the cost function with respect to the forgetting factor α_{RLS}. The adaptation algorithm is based on equations (3.60) and (3.63) - (3.65) plus the following:

$$\boldsymbol{S}(n)=\left(\frac{1-\boldsymbol{K}^{H}(n)\boldsymbol{X}(n)}{N\alpha_{\mathrm{RLS}}(n-1)}\right)\left(\left(\frac{1-\boldsymbol{K}^{H}(n)\boldsymbol{X}(n)}{N}\right)^{*}\boldsymbol{S}(n-1)-\boldsymbol{P}(n)\right) \tag{3.66}$$

$$\boldsymbol{\psi}(n)=\left(\frac{1-\boldsymbol{K}^{H}(n)\boldsymbol{X}(n)}{N}\right)\boldsymbol{\psi}(n-1)+\boldsymbol{S}(n)\boldsymbol{X}(n)e^{*}(n) \tag{3.67}$$

$$\alpha_{\mathrm{RLS}}(n)=\alpha_{\mathrm{RLS}}(n-1)+\gamma\Re e\left\{\boldsymbol{\psi}^{H}(n-1)\boldsymbol{X}(n)e^{*}(n)\right\}_{\alpha_{\min}}^{\alpha_{\max}} \tag{3.68}$$

where $\boldsymbol{P}(n)=\boldsymbol{Q}^{-1}(n)$ is the inverse of the covariance matrix, $\boldsymbol{S}(n)=\dfrac{\delta\boldsymbol{P}(n)}{\delta\alpha_{\mathrm{RLS}}}$ is a matrix of size ($N\times N$), $\boldsymbol{\psi}(n)=\dfrac{\delta\boldsymbol{W}(n)}{\delta\alpha_{\mathrm{RLS}}}$, and γ is a small positive value used to govern the convergence rate of the forgetting factor. $\alpha_{\max}$ and $\alpha_{\min}$ denote the upper and lower limits of the adaptive forgetting factor. The AFFRLS algorithm shows improved tracking performance over the conventional RLS. This is achieved with an increase in complexity of $2N^{2}+3.5N$ multiplications over the RLS algorithm. However, the AFFRLS algorithm suffers from a gradient noise amplification problem [90].

An improved AFFRLS algorithm has been proposed in [90]. It has been developed using the analogy between the NLMS and AFFRLS algorithms. Hence, it is known as the NAFFRLS algorithm. It makes use of the updating parameters associated with a new algorithm for updating the forgetting factor such that

$$\alpha_{\mathrm{RLS}}(n)=\alpha_{\mathrm{RLS}}(n-1)+\frac{\gamma}{\left|\boldsymbol{\psi}^{H}(n-1)\boldsymbol{X}(n)\right|^{2}}\Re\left\{\boldsymbol{\psi}^{H}(n-1)\boldsymbol{X}(n)e^{*}(n)\right\}_{\alpha_{\min}}^{\alpha_{\max}} \quad (3.69)$$

As it can be seen in equation (3.69), the normalization will affect the estimated error. In addition, the complexity will increase to become $4N^2+6N$.

Table 3-2 Summary of the updating parameters used in the NAFFRLS algorithm

	LMS weight vector update equation	**AFFRLS forgetting factor update equation**
Update term	$\boldsymbol{W}(n)$	$\alpha(n)$
Step size	μ	γ
Gradient term	$\boldsymbol{X}(n)e^{*}(n)$	$\boldsymbol{\psi}^{H}(n-1)\boldsymbol{X}(n)e^{*}(n)$

3.3.1 RLS and LMS combination

As discussed in Section 3.3, the tracking ability of the RLS algorithm has been enhanced by adopting various schemes to make the forgetting factor adaptive. Such improvements have been achieved with increasing complexity, and this makes the RLS based algorithms less attractive for some applications such as array beamforming. In this thesis, a new way of making use of the RLS algorithm in conjunction with the LMS algorithm for beamforming is presented. Details of this new algorithm, called the RLMS algorithm, are described in Chapter 4.

3.4 Blind Algorithms

The algorithms described in Sections 3.2 and 3.3 require the use of a reference signal for their operation. In this section, we briefly consider the

case when the computation of the weight vector does not rely on the availability of a reference signal. In this case, a blind beamformer computes its weight vector based only on the received signal but without prior knowledge of the input signals or the channel [19, 24]. Some of these blind adaptive beamforming algorithms are described in this section.

3.4.1 Constant modulus algorithm (CM)

Some communication signals, such as phase-shift keying (PSK), frequency-shift keying (FSK), and analog frequency modulation (FM) signals, have a constant envelope. After transmitting through a channel, the constant envelope of the signal may become distorted. The constant modulus algorithm (CM) [91] computes its weight factor based on the minimization of the amplitude variation in the received signal. This can be carried out by adopting a positive cost function given by [19, 91]

$$\xi(n) = \mathrm{E}\left[e^2(n) \right] = \mathrm{E}\left[\left(\left| \boldsymbol{W}^H(n)\boldsymbol{X}(n) \right|^2 - 1 \right)^2 \right] \tag{3.70}$$

where

$$e(n) = \left| y(n) \right|^2 - 1 \tag{3.71}$$

Equation (3.70) represents the deviation of the output signal from the unit modulus condition. Minimizing equation (3.70) will yield an optimum weight vector that makes the output signal, $y(n)$, have as constant an envelope as possible. Analytically, equation (3.70), being a fourth-order function, is difficult to compute [19]. However, an iterative approach may be used to search for the minimum ξ in the same way as the LMS algorithm, i.e., by updating the weight vector $\boldsymbol{W}$ in small steps following the negative gradient direction as given in equation (3.11) [19].

Expanding (3.70), we obtain

$$\xi(n) = \mathrm{E}\left[\left(\left|\boldsymbol{X}^H(n)\boldsymbol{W}(n)\right|^2 - 1\right)\left(\left|\boldsymbol{W}^H(n)\boldsymbol{X}(n)\right|^2 - 1\right)\right] \tag{3.72}$$

Then, calculating the gradient $\boldsymbol{\nabla}(\xi(n))$ from (3.72) yields

$$\begin{aligned}\boldsymbol{\nabla}(\xi(n)) = \frac{\partial \xi(n)}{\partial \boldsymbol{W}^H(n)} &= 2\mathrm{E}\left[\left(|y(n)|^2 - 1\right)\frac{\partial}{\partial \boldsymbol{W}^H(n)}\left(\boldsymbol{W}^H(n)\boldsymbol{X}(n)\boldsymbol{X}^H(n)\boldsymbol{W}(n)\right)\right] \\ &= 2\mathrm{E}\left[\left(|y(n)|^2 - 1\right)\left(\boldsymbol{X}(n)\boldsymbol{X}^H(n)\boldsymbol{W}(n)\right)\right] \\ &= 2\mathrm{E}\left[\left(|y(n)|^2 - 1\right)\boldsymbol{X}(n)y^*(n)\right] \end{aligned} \tag{3.73}$$

The instantaneous estimate of equation (3.73) is

$$\boldsymbol{\nabla}(\xi(n)) = 2\left(|y(n)|^2 - 1\right)\boldsymbol{X}(n)y^*(n) \tag{3.74}$$

Substituting equation (3.74) in equation (3.11) yields

$$\begin{aligned}\boldsymbol{W}(n+1) &= \boldsymbol{W}(n) - 2\mu\left(|y(n)|^2 - 1\right)\boldsymbol{X}(n)y^*(n) \\ &= \boldsymbol{W}(n) - 2\mu e^*(n)y^*(n)\boldsymbol{X}(n)\end{aligned} \tag{3.75}$$

Thus the steepest-descent CM (SD-CM) algorithm can be summarized as

$$y(n) = \boldsymbol{W}^H(n)\boldsymbol{X}(n) \tag{3.76}$$

$$e(n) = \left(|y(n)|^2 - 1\right) \tag{3.77}$$

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) - 2\mu\boldsymbol{X}(n)e^*(n) \tag{3.78}$$

From equation (3.77), it is observed that the error signal will be equal to zero under the following conditions [42]:

- The magnitude of the array output is equal to unity.
- The array output is equal to zero, a situation which is not likely to occur in practice due to the presence of noise.

Moreover, the solution of the CM cost function may lead to local minima [19]. This occurs if the beamformer produces a constant envelop output with weight vector being equal to $\kappa \boldsymbol{W}_{opt}$, where κ is a constant. Therefore, more information on the source will need to be provided in order to remove this type of ambiguity. However, such ambiguity does not pose a problem for angle modulated signals [19]. In addition, CM algorithms suffer from the so called selectivity problem involving the presence of multiple signals having similar properties. Under such condition, it is difficult for a CM algorithm to differentiate between the various constant modulus signals. As a result, it is possible for a strong interfering signal instead of the desired signal to be captured [17, 91, 92]. Also, compared with the non-blind algorithms, the convergence rate of the CM algorithm is slow [93], and this has been demonstrated in the practical implementation presented in [94]. Furthermore, CM algorithms tend to encounter a phase rotation problem and larger steady state error [95]. In general, the performance of CM algorithms is poor, particularly when operating at low SNR [96] or with modulated signals having more than two levels.

A modified CM (MCM) algorithm known as blind adaptive beamforming multi-modulus array (MMARY) is proposed in [92] for multi-modulus signals. This algorithm can be used to handle a non-constant envelop high order signal constellation, such as quadrature amplitude modulated (QAM) signals [97]. In this case, the algorithm operates on both the real and imaginary components of the signal separately, thus allowing both the modulus and the phase of the array output to be considered. In addition, a hybrid CM based algorithm has been proposed in [93] for handling QAM signals. This involves circular regions being defined around individual symbol points in a QAM signal constellation as shown in Figure 3-4. In other words, it introduces a radius-adjusted approach to QAM signal constellations, such that

$$R_m(n) = \left|\hat{s}(n) - y(n)\right| \tag{3.79}$$

where R_m is the radius of the specified region, and $\hat{s}$ is the estimated symbol point. The value of R_m is then used to specify the region, i, which is

then used to define the step size and the weighting factor, w_i based on a predefined table.

In this case, the CMA error function given in equation (3.77) is modified to become [93]

$$e(n) = e_R(n) + je_I(n) \tag{3.80}$$

with e_R and e_I being the real and imaginary components of the error signal, defined as

$$e_R(n) = w_i y_R(n)\left[y_R^2(n) - R_m(n) \right] + (1 - w_i)\left[y_R(n) - \hat{s}(n) \right] \tag{3.81}$$

$$e_I(n) = w_i y_I(n)\left[y_I^2(n) - R_m(n) \right] + (1 - w_i)\left[y_I(n) - \hat{s}(n) \right] \tag{3.82}$$

where y_R and y_I are the real and imaginary components of the output y.

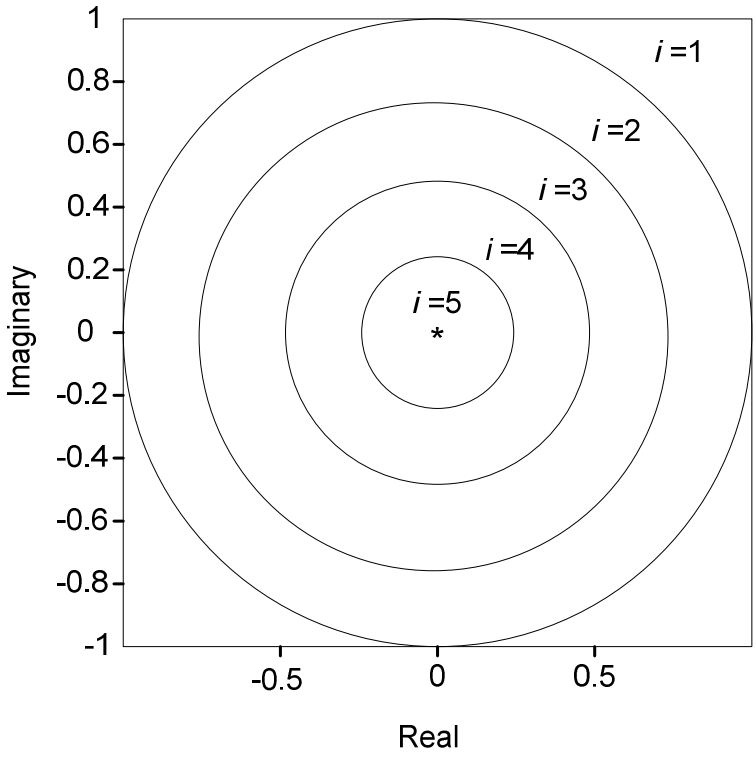


Figure 3-4 Sample decision regions [93]

In an attempt to overcome some of their shortcomings, many blind algorithms have been operated with non-blind algorithms. For example, in [91], the initial weights of a CMA algorithm have been determined using a sample matrix inversion (SMI) algorithm. This initialization with the SMI algorithm could lead to an increase in the convergence rate of the CM algorithm. However, it could also give rise to false solutions particularly when the power of the interfering signal is larger than that of the desired signal. Also, Xu and Tsatsanis [98] combined the constrained minimum variance algorithm (CMV) with the LMS algorithm to form the CMV-LMS algorithm, and with the RLS to form the CMV-RLS algorithm. The CMV-LMS algorithm is simpler, but being an LMS-based algorithm its convergence is also slower than that of the CMV-RLS algorithm. Then Lei Wang and Rodrigo Lamare [98] combined the constrained constant modulus[2] (CCM) and RLS algorithms to produce the CCM-RLS algorithm, which updates the array weights by optimizing a cost function based on the CM criterion. The RLS algorithm, part of the CCM-RLS algorithm, is used to update the inverse of the correlation matrix of the input signal, as represented by equation (3.60). However, this scheme is very sensitive to changes in the operating environments. Also, it suffers from a high complexity requirement, requiring $3N^2+8N+7$ multiplications, which is significantly higher than that for the conventional RLS algorithm [99].

The use of variable step size has also been considered in hybrid blind and non-blind algorithms in order to cater for different channel characteristics, such as time varying multipath channels. In [99], adaptive step size LMS and RLS algorithms are applied to track time-varying direct sequence code division multiple access (DS/CDMA) signals, with the signature waveform and timing being the only prior knowledge of the desired signal. In this paper, the effect of multipath distortion is neglected. However, in [100], the signal distortion caused by multipath transmission is considered for improving the interference cancellation capability in the algorithms

[2] In this algorithm, the cost function is considered as the expected deviation from unity of the squared modulus of the array output under the constraint $\boldsymbol{W}^H(n)A_d = 1$.

proposed in [101]. A variance oriented approach (VOA) algorithm has also been proposed in [101], that uses the LMS algorithm to minimize the cost function for the same purpose. The principle of this algorithm is based on the concept that the cost function is formulated as a ratio between the despread output and the variance of the desired signal. The filtering of the input signal using spatial domain filters is another technique proposed in [102]. An example of such an approach is shown in Figure 3-5 [103]. Note that the input signal vector obtained from the *N* elements is first subjected to fast Fourier transform (FFT). Then, the most significant coefficient (MSC) of the FFT output is selected based on the index of the most significant coefficient, K_{MSC}, given by

$$K_{\mathrm{MSC}} = \left[-\frac{1}{2} \left(\frac{N \left(2\pi N \sin\theta_d - \pi \sin\theta_d + j2 \tanh^{-1} \left(\frac{N - 1/2}{N-1} \right) \right)}{\pi (2N-1)} \right) \right]_{\mathrm{int}} \tag{3.83}$$

where $[x]_{\mathrm{int}}$ represents the integer part of x. θ_d is the AOA of the desired signal.

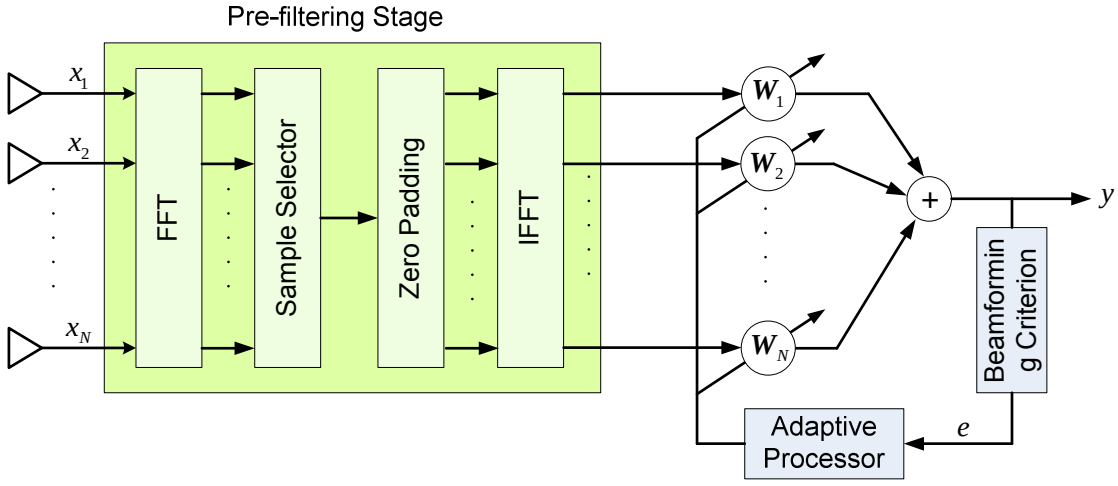


Figure 3-5 Blind beamforming with pre-filtering process [24, 104]

The need for FFT-IFFT manipulations means that the complexity of this type of approach is very high.

In general, all the blind/hybrid blind and non-blind algorithms mentioned above are able to improve both the convergence rate and error floor of the CM algorithm. However, these improvements result in increased complexity [104], for instance, as we mentioned earlier in this section, the hybrid RLS-CM algorithm requires $3N^2+8N+7$ multiplications. More precisely, if the RLS-CM algorithm is used for systems involving orthogonal frequency-division multiplexing (OFDM) signals, the complexity will become $N_{\mathrm{OFDM}}\left(3N^2+6N\right)$ where N_{OFDM} is the number of OFDM subcarriers [24].

3.4.2 Decision directed algorithm

As discussed in Section 3.4.1, the use of blind and non-blind combination algorithms can improve the MSE performance of the CM algorithm at the expense of its complexity. A decision directed (DD) algorithm is an alternative method to obtain initially a sufficiently low MSE, which could be used to further minimize the MSE of a CM algorithm [105].

In a decision directed algorithm, based on hard threshold decision, the received signal is first demodulated to estimate the transmitted signal. Next, the resulting symbol stream is remodulated to generate an approximate reference signal [105], as shown in Figure 3-6. Using the regenerated reference signal, $d(n)$, the error magnitude can then be obtained such that

$$e(n)=d(n)-\boldsymbol{W}^H(n)\boldsymbol{X}(n) \tag{3.84}$$

Once the error is initially estimated, the adaptation process to estimate the weights can be carried out using one of the blind algorithms mentioned earlier, such as the CM algorithm.

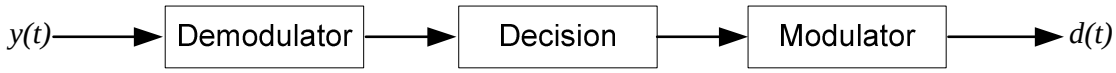


Figure 3-6 Generation of the reference signal in a DD algorithm

The performance of a DD algorithm is highly dependent on the quality of the initial estimation of the error, the noise, and the synchronization degree of the estimated reference signal [105]. Hence, the convergence of a DD algorithm is not guaranteed [106], i.e., the output of a beamformer employing a DD algorithm could be unreliable [107]. For this reason, the use of dedicated training signal to aid the initial reference signal estimation could overcome the reliability concern of a DD algorithm. Such an approach is referred to as soft decision directed (SDD) algorithm [108], which is also known as the semi-blind algorithm. The complexity of the SDD algorithm published in [109] is equivalent to $12N+29$ multiplications, $14N+21$ additions and 4 exponentional functions [110].

3.4.3 Higher-order cumulant algorithm (HoCA)

The analysis presented in [111] shows that when the signal to noise ratio (SNR) is low and the signal sources are closely spaced, the CM based algorithms usually incur considerable performance degradation. It is well known that most current signal processing tasks have been based on the assumption that the signals involved follow second order statistics. This either explicitly or implicitly suggests that the signals are Gaussian [111-113]. Consequently, these signals could only be described by their mean and standard deviation values [114]. However, non-Gaussian processes, which contain more information, are not used in conventional array beamforming algorithms. The use of higher statistics, associated with non-Gaussian signals, could potentially offer the following advantages:

- To separate non-Gaussian from the Gaussian noise/interference sources as well as statistically independent sources [114].

- Provide a better estimation of the signal parameters, such as angle of arrival (AOA) and gain [109].

- Help to reveal phase information of the signals which can not be done using second order statistics [115].

It is noted in [116] that cumulant algorithms are blind to any kind of Gaussian processs. Many researchers have considered the study of these algorithms. For example, in [115] Chiang and Nikias developed a fourth order algorithm based on the eigen structure analysis, and they referred to it as estimating signal parameters via rotational invariance techniques (ESPRIT). Another example is the multiple signal classification (MUSIC)-like algorithm which depends on the eigen decomposition of the fourth-order cumulants [117]. Others, like Martin and Mansour [118-120], apply successfully the higher order statistics (fourth-order) in speech processing for blind separation of up to five different speech signals.

In array beamforming applications, higher-order statistics (fourth-order cumulant) have been used to estimate the steering vector of non-Gaussian desired signals in the presence of Gaussian interference of unknown covariance structure [115]. In a comparison study between second and fourth order MUSIC algorithms, published in [115], it is shown that as the order is increased from 2 to 4, the estimated AOA is more accurate, especially with small number of samples and noisy signals. However, the computation complexity employed by the higher order algorithm is significantly increased. Another drawback of a high order algorithm is its blindness to Gaussian noise, which makes its operation easily corrupted by noise [107]. It is shown in [121] that longer data samples may be used to overcome this problem, but this will result in lower convergence speed. Instead, it is proposed to combine second-order estimation of the signal covariance matrix using Capon's minimum variance distortionless response (MVDR[3]) algorithm [122] in conjunction with a fourth order scheme. The use of the MVDR algorithm can

[3] MVDR beamforming is an alternative approach for signal recovery, such as $y = \left(\beta_1 \boldsymbol{Q}^{-1} A_d(\theta_d)\right)\boldsymbol{X}$, where β_1 is a gain constant.

now allow the Gaussian interference signals to be removed [123]. Such a combined algorithm normally starts with a higher-order statistics algorithm followed by MVDR algorithm, since the latter requires knowledge of the steering vector [123].

To mitigate the performance degradation caused by multipath signals, Martone [124] proposed and implemented a high-order complex zero-mean fourth-order cumulants algorithm for a linear array system for use in a base station. The adaptive algorithm concerned is implemented by means of square-root decomposition of the cumulant matrix to provide the ability of tracking time-varying channels [125].

The main drawback of higher-order cumulant algorithms is the large increase in the complexity [122, 126]. For instance, the higher-order algorithm presented in [127] requires at least $8N^3 + 46N^2 + 30N$ multiplications, $2N$ reciprocals and N square roots, where N is the number of array elements. In addition, higher-order cumulants need an estimation which makes use of a longer data sequence than for second order methods in order to obtain a comparable MSE [128]. In other words, this means that higher order algorithms tend to have lower convergence rates.

3.5 Comments on the Applications of Adaptive Beamformers

As discussed in Section 3.1, smart antenna systems have the ability of forming the main beam in the direction of the desired signal while minimizing the gain in the direction of the interference. This results in better signal-to-interference ratio, which can improve the coverage area and the capacity. These antenna systems can be classified into two categories; switched and adaptive array systems. For the former, the beam is switched between predefined patterns according to the level of the received signal. On the other hand, an adaptive array system is able to track the signal.

The complexity and hence the cost of a full adaptive beamformer is related to the number of antenna elements used [122] , and is higher than

that for a switched array system [123]. In order to reduce the cost, the authors in [123, 129] propose a hybrid adaptive antenna system that combines the advantages of both the adaptive and switched systems. With this hybrid scheme, only those elements having acceptable received signal power are chosen to perform the beamforming process. The system will continue to monitor the received signal level, and if required, change those current elements with those which have higher signal levels. This means that a smaller number of input signals are actually involved in the beamforming process, and this could lead to less complexity and lower implementation cost [123]. However, this hybrid scheme encounters two possible drawbacks. First, the beamwidth obtained is broader due to the use of less array elements, and this in turn suggests poorer suppression of close in interfering signals. Also, the element selection process may give rise to a wrong decision as it is possible that the power level used may come from the interfering rather than the desired signal [123].

As for the actual beamforming, the weights associated with the chosen elements are updated using a hybrid algorithm, such as the CM modified adaptive step size (CM-MASS) or CM time averaging adaptive step size (CM-TAASS) as proposed in [124]. The step sizes of the CM-MASS and CM-TAASS algorithms are made adaptive according to:

- CM- MASS algorithm

$$\mu(n+1) = \alpha\mu(n) + \gamma\left(|y(n)|^2 - 1\right) \tag{3.85}$$

where $y(n)$ is the algorithm output, and α is the forgetting factor given by $0 < \alpha < 1$. The limits of $\mu(n+1)$ are as defined in equation (3.34).

- CM-TAASS algorithm

In order to make an algorithm more robust to noise, the step size in the CM-TAASS algorithm is updated based on a time average

estimate of the correlation between $\left(|y(n)|^2-1\right)$ and $\left(|y(n-1)|^2-1\right)$ such that

$$\mu(n+1)=\alpha\mu(n)+\gamma v^2(n) \tag{3.86}$$

where, $0<\beta<1$, $\gamma>0$, and $v(n)$ is defined as

$$v(n)=\beta v(n-1)+(1-\beta)\left(|y(n)|^2-1\right)\left(|y(n-1)|^2-1\right) \tag{3.87}$$

Recently, Long Term Evolution (LTE) and WiMAX wireless communication technologies have been chosen as candidates for the so called 4G mobile communication systems [130, 131]. To achieve the required data rate and capacity, both WiMAX and LTE systems support multi-antenna technologies including beamforming, using up to 4 antenna elements, for multiple-input and multiple-output (MIMO) enhancement strategy [122, 125, 126]. For instance, a clustered non-uniform linear array is considered in [127], where the antennas in each cluster form a ULA of 2 elements spaced half a carrier wavelength apart. An MMSE algorithm is used for the beamforming to update the weights. This adaptive beamforming is used to mitigate the interfering signals. In the advanced version of the LTE system, called LTE-advanced, array beamforming techniques involving the use of 4 to 8 elements in the base station are also proposed [128] to further enhance the system performance. Since orthogonal frequency division multiple access (OFDMA) is adopted in LTE and WiMAX systems, various OFDMA specific beamforming algorithms have also been proposed [132, 133].

3.6 Summary

In this chapter, several types of adaptive algorithms, which are capable of producing the required vector needed for beam steering, are discussed in some details. These include the LMS family of algorithms, the RLS family of algorithms, blind algorithms and higher order algorithms. Among all these

algorithms, the LMS based algorithms are the simplest. As such, they have been reviewed in greater detail by considering their mathematical models together with their advantages and disadvantages. Over the years, several modifications have been introduced by various researchers in an attempt to improve the low convergence rate of the conventional LMS algorithm. However, the potential for fast convergence of these different modified LMS based algorithms depends on the application environments. These are further governed by the ability to properly choose the parameter values to suit each environment.

On the other hand, the RLS based algorithms are also discussed. It is well known that the RLS algorithm is able to provide a much faster convergence over the LMS based algorithms. However, this is achieved at the expense of a higher computation complexity, stability and tracking ability. It is shown that the AFFRLS algorithm achieves its better tracking ability over the standard RLS algorithm, with an even higher degree of computation complexity.

Blind algorithms have briefly been discussed in this chapter together with hybrid schemes involving the combination of blind and non-blind algorithms. The main problems of the blind algorithms are the selectivity problem and slow convergence rate. The latter may be improved through the use of the hybrid schemes. However, these schemes have even higher complexity.

In addition, higher order blind algorithms are also briefly reviewed. It is found that the performance of these algorithms tends to improve as the order of these algorithms increases. However, such performance enhancements can lead to a large increase in complexity. Finally, the use of some of these algorithms in beamforming for mobile radio communications is discussed.

CHAPTER 4

ADAPTIVE ARRAY BEAMFORMING USING A COMBINED RLS-LMS ALGORITHM

4.1 Introduction

As discussed in Section 3.2.1, the LMS algorithm offers simpler implementation and good tracking capability while the RLS algorithm provides relatively fast convergence [6, 7]. Also, variants of these two algorithms have been investigated aiming at enhancing the convergence, speed and tracking ability in a dynamic environment. For the LMS family of algorithms, there is always a trade off between the speed of convergence and the achievable residual error floor, when a given adaptation step size is used. As discussed in Section 3.2.2, several improvements have been proposed to speed up the convergence of the LMS algorithm. Examples of these modified LMS algorithms include the variable step size LMS algorithm (VSSLMS) [8], constrained-stability LMS (CSLMS) algorithm [9], and modified robust variable LMS (MRVSS) algorithm [10]. All of these algorithms make use of an initial large adaptation step size to speed up the convergence. Upon approaching the steady state, smaller step sizes are then introduced to decrease the level of adjustment, hence maintaining a lower error floor.

On the other hand, some of the improvements made in the tracking ability of the RLS family of algorithms are discussed in Section 3.3.1. These include the adaptive forgetting factor RLS algorithm (AFF-RLS) [89], variable forgetting factor RLS (VFFRLS) [12] and the extended recursive least squares (EX-KRLS) algorithm [13]. For an *N*-element antenna array, the implementation of the VFFRLS, AFF-RLS and EX-KRLS algorithms will incur

$2.5N^2+3N+20$, $9N^2+7N$ and $15N^3+7N^2+2N+4$ complex multiplications, respectively [89]. This compares with $2.5N^2+3N$ complex multiplications for the conventional RLS algorithm [89]. As such, the improvement in tracking ability of the RLS algorithm is achieved at the expense of a large increase in computation complexity.

In this chapter, a novel approach is adopted to achieve the desirable features of high convergence speed and superior tracking without introducing excessive computation complexity. The proposed algorithm is referred to as the RLMS algorithm. Details of this new RLMS algorithm will now be presented.

4.2 RLMS Algorithm

To achieve fast convergence and good tracking ability, the proposed RLMS algorithm combines the use of two algorithms, namely the RLS and LMS algorithms. As shown in Figure 4-1, the input signals picked up by the antenna elements are first processed by an RLS algorithm stage to yield an intermediate output signal, $\mathrm{y}_{\mathrm{RLS}}$. This intermediate output signal is in turn multiplied by the array image vector, $\mathcal{F}_{\mathrm{R}}$, which acts as a "spatial filter" for the desired signal. The resultant signal components are further processed by an LMS algorithm stage to obtain the final estimate of the desired signal, $\mathrm{y}_{\mathrm{RLMS}}$. To enhance the convergence rate and tracking ability of the overall algorithm, the previous error sample, $e_{\mathrm{LMS}}(j-1)$, from the LMS algorithm stage is fed back to combine with the current error sample, $e_{\mathrm{RLS}}(j)$, of the RLS algorithm stage to form the overall error signal, $e_{\mathrm{RLMS}}(j)$, for updating the tap weights of the RLS algorithm stage. In this way, the overall error signal, e_{RLMS}, becomes smoother even though e_{RLS} and e_{LMS} may individually take on large values. This may improve the stability of the RLMS algorithm against sudden changes in the input signals. As shown in Figure 4-1, a common external reference signal is used for both the RLS and LMS algorithm stages, i.e., d_{RLS} and d_{LMS}. This mode of operation will from now on be referred to as the external referencing mode.

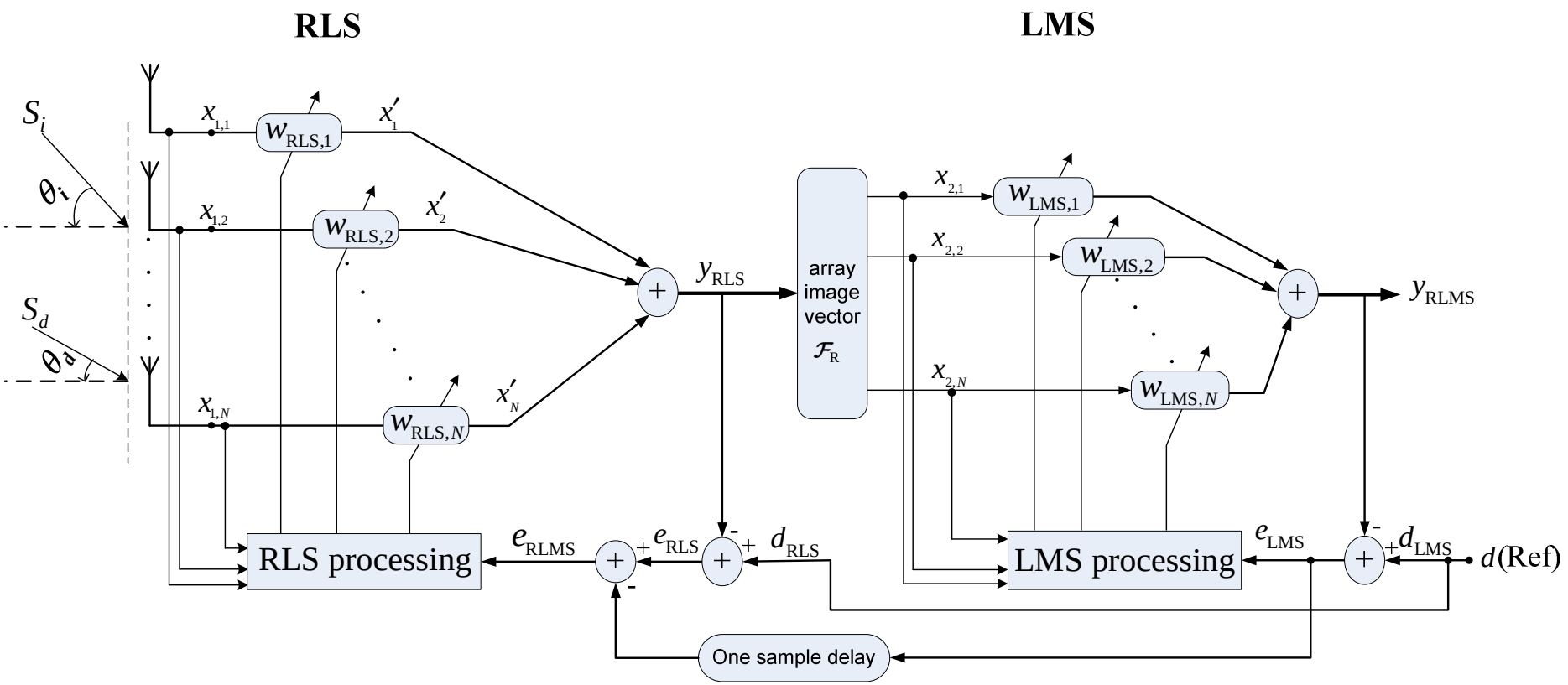


Figure 4-1 The proposed RLMS algorithm with an external reference signal, $d(\mathrm{Ref})$.

Now, let the input signal vector, $\boldsymbol{X}(t)$, of the RLS algorithm stage in Figure 4-1 be represented by

$$\boldsymbol{X}(t) = [x_1(t),\ x_2(t),\ ...,\ x_N(t)]^T = \boldsymbol{A}_d s_d(t) + \boldsymbol{A}_i s_i(t) + \boldsymbol{n}(t) \tag{4.1}$$

where $[\bullet]^T$ represents the matrix transpose; $s_d(t)$, and $s_i(t)$ are the desired and interfering signals, respectively. $\boldsymbol{A}_d$ and $\boldsymbol{A}_i$ are the $[N \times 1]$ complex array vectors for the desired signal and the cochannel interference, respectively, and $\boldsymbol{n}(t)$ is the noise vector. With the first antenna element acting as the reference, then $\boldsymbol{A}_d$ and $\boldsymbol{A}_i$ are given by

$$\boldsymbol{A}_d = [1,\ e^{-j\psi_d},\ e^{-2j\psi_d},\ ...,\ e^{-(N-1)j\psi_d}]^T \tag{4.2}$$

$$\boldsymbol{A}_i = [1,\ e^{-j\psi_i},\ e^{-2j\psi_i},\ ...,\ e^{-(N-1)j\psi_i}]^T \tag{4.3}$$

According to the far-field plane wave model,

$$\psi_d = 2\pi \left[\frac{\mathcal{D} \sin(\theta_d)}{\lambda} \right] \tag{4.4}$$

and

$$\psi_i = 2\pi \left[\frac{\mathcal{D} \sin(\theta_i)}{\lambda} \right] \tag{4.5}$$

where $\mathcal{D}$ is the array inter-element spacing, and λ is the carrier wavelength.

It will be shown in Section 4.3.2 that the proposed RLMS algorithm, which normally operates with a common external reference for both the RLS and LMS algorithm stages, can achieve convergence within a few iterations. As such, the output of the RLS algorithm stage, y_{RLS}, which closely resembles the input desired signal, $s_d(t)$, may be used as the reference signal for the LMS algorithm stage. Also, the output of the LMS algorithm stage, y_{RLMS}, will become the reference for the RLS algorithm stage. This feedforward and feedback arrangement enables the provision of self-referencing in the RLMS

algorithm, and allows the external reference to be discontinued after an initial few iterations. For stability considerations, it is necessary for the step size of the first stage, μ_{RLS}, to be chosen such that the intermediate output, y_{RLS}, is a sufficiently close estimate of the desired signal, $s_d(t)$. In fact, the values of μ_{RLS} and μ_{LMS} do not follow the same relationship in updating the weights. In this case, the weight update process of the LMS algorithm is dictated by μ_{LMS}, whereas μ_{RLS} controls the updating process of the inverse correlation matrix ($\boldsymbol{P}$) [134] as shown in equations (4.10) and (4.11).

For the case of a moving target, it is necessary that the array image vector, $\mathcal{F}_{\mathrm{R}}$**,** is made adaptive in order to follow the angle of arrival (AOA) of the desired signal. This adaptive $\mathcal{F}_{\mathrm{R}}$ version of the RLMS algorithm will from here on be simply known as the RLMS algorithm in order to differentiate it from the scheme that makes use of fixed $\mathcal{F}_{\mathrm{R}}$ with prescribed values for its individual elements**.** The latter will be referred to as the RLMS_1 algorithm for fixed beamforming.

4.3 Convergence Analysis of the Proposed RLMS Algorithm

4.3.1 Analysis with an external reference

As described in Section 4.2, the RLMS algorithm normally operates with a common external reference signal applied to the RLS and LMS algorithm stages. In this section, the convergence of the RLMS algorithm is analyzed based on the mean-square error (MSE) of the overall error signal, e_{RLMS}. For the analysis, the following assumptions are made:

(i) The propagation environment is time invariant.

(ii) The input signal vector $\boldsymbol{X}(n)$ should be independent and identically distributed (iid).

(iii) The individual elements of the input signal vector $\boldsymbol{X}(n)$ are spatially non-correlated.

(iv) All error signals are stationary with zero mean.

From Figure 4-1, the overall error signal for the RLMS algorithm at the n^{th} iteration is given by

$$e_{\mathrm{RLMS}}(n) = e_{\mathrm{RLS}}(n) - e_{\mathrm{LMS}}(n-1) \tag{4.6}$$

with

$$\begin{aligned} e_{\mathrm{RLS}}(n) &= d_{\mathrm{RLS}}(n) - y_{\mathrm{RLS}}(n) \\ &= d_{\mathrm{RLS}}(n) - \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(n) \end{aligned} \tag{4.7}$$

and

$$\begin{aligned} e_{\mathrm{LMS}}(n) &= d_{\mathrm{LMS}}(n) - y_{\mathrm{RLMS}}(n) \\ &= d_{\mathrm{LMS}}(n) - \boldsymbol{W}_{\mathrm{LMS}}^{H}(n)\boldsymbol{X}_{\mathrm{LMS}}(n) \end{aligned} \tag{4.8}$$

where $(\cdot)^{H}$ denotes the Hermitian matrix of $(\cdot)$; and

$$\boldsymbol{X}_{\mathrm{LMS}}(n) = \mathcal{F}_{\mathrm{R}}\, y_{\mathrm{RLS}}(n) = \mathcal{F}_{\mathrm{R}}\, \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(n) \tag{4.9}$$

with $\mathcal{F}_{\mathrm{R}}$ being the image of the array vector. A simple method of estimating $\mathcal{F}_{\mathrm{R}}$ is given in Section 4.5. $\boldsymbol{W}_{\mathrm{RLS}}$ and $\boldsymbol{W}_{\mathrm{LMS}}$ are the weight vectors of the RLS and LMS algorithm stages, respectively, which are updated according to [34],

$$\boldsymbol{W}_{\mathrm{RLS}}(n+1) = \boldsymbol{W}_{\mathrm{RLS}}(n) + \mu_{\mathrm{RLS}} e_{\mathrm{RLS}}(n)\boldsymbol{P}(n+1)\boldsymbol{X}(n) \tag{4.10}$$

$$\boldsymbol{W}_{\mathrm{LMS}}(n+1) = \boldsymbol{W}_{\mathrm{LMS}}(n) + \mu_{\mathrm{LMS}} e_{\mathrm{LMS}}(n)\boldsymbol{X}_{\mathrm{LMS}}(n) \tag{4.11}$$

where μ_{LMS} and μ_{RLS} are the respective step sizes for the LMS and RLS algorithm stages, and $(*)$ represents the complex conjugate. $\boldsymbol{P}(n)$ is a symmetric positive definite matrix given by

$$\boldsymbol{P}(n+1) = \alpha_{\mathrm{RLS}}^{-1}\boldsymbol{P}(n) - \alpha_{\mathrm{RLS}}^{-1}\frac{\alpha_{\mathrm{RLS}}^{-1}\boldsymbol{P}(n)\boldsymbol{X}(n+1)\boldsymbol{X}^{H}(n+1)\boldsymbol{P}(n)}{1+\alpha_{\mathrm{RLS}}^{-1}\boldsymbol{X}^{H}(n+1)\boldsymbol{P}(n)\boldsymbol{X}(n+1)} \tag{4.12}$$

$\boldsymbol{P}(n)$ is initialized by $\delta^{-1}\boldsymbol{I}$, with δ being a small positive constant, and $\boldsymbol{I}$ is an $N \times N$ unity matrix. N is the number of antenna elements, and α_{RLS} is a forgetting factor and is related to μ_{RLS}, such that [80]

$$\mu_{\mathrm{RLS}} = 1 - \alpha_{\mathrm{RLS}} \tag{4.13}$$

Convergence in mean-square error, ξ_{RLMS}, of the RLMS algorithm can be analyzed by observing the expected value of e_{RLMS}^2, so that

$$\begin{aligned}\xi_{\mathrm{RLMS}}(n) &= \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \mathrm{E}\left[\left| d_{\mathrm{RLS}}(i) - \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(i) - e_{\mathrm{LMS}}(i-1)\right|^2\right] \\ &= \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \Big\{ \mathrm{E}\left[\left|D_{\mathrm{R}}(i)\right|^2\right] - \mathrm{E}\Big[D_{\mathrm{R}}(i)\boldsymbol{X}^{H}(i)\boldsymbol{W}_{\mathrm{RLS}}(n) \\ &\qquad + D_{\mathrm{R}}^{*}(i)\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(i)\Big] + \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{Q}_{\mathrm{R}}(n)\boldsymbol{W}_{\mathrm{RLS}}(n)\Big\}\end{aligned} \tag{4.14}$$

where $|x|$ signifies the modulus of x;

$$D_{\mathrm{R}}(n) = d_{\mathrm{RLS}}(n) - e_{\mathrm{LMS}}(n-1), \tag{4.15}$$

and $\boldsymbol{Q}_{\mathrm{R}}(n)$ is the estimation of the input correlation matrix given by

$$\boldsymbol{Q}_{\mathrm{R}}(n) = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \boldsymbol{X}(n)\boldsymbol{X}^{H}(n) \tag{4.16}$$

Now, consider the first and second terms on the right hand side (RHS) of equation (4.14) separately. The first RHS term of equation (4.14) can be expressed as:

$$\begin{aligned}\sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left|D_{\mathrm{R}}(i)\right|^2\right]\right\} &= \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left| d_{\mathrm{RLS}}(i) - e_{\mathrm{LMS}}(i-1)\right|^2\right]\right\} \\ &= \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \Big\{ \mathrm{E}\left[\left| d_{\mathrm{RLS}}(i)\right|^2\right] - \mathrm{E}\Big[d_{\mathrm{RLS}}(i)e_{\mathrm{LMS}}^{*}(i-1) \\ &\qquad + d_{\mathrm{RLS}}^{*}(i)e_{\mathrm{LMS}}(i-1)\Big] + \mathrm{E}\left[\left| e_{\mathrm{LMS}}(i-1)\right|^2\right]\Big\}\end{aligned} \tag{4.17}$$

The second term on the RHS of equation (4.17) is equal to zero, because $d_{\mathrm{RLS}}(i)$ and $e_{\mathrm{LMS}}(i-1)$ are zero mean and uncorrelated based on the assumptions (i), (ii) and (iv). Therefore, equation (4.17) becomes

$$\sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left| D_{\mathrm{R}}(i) \right|^2 \right] \right\} = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left| d_{\mathrm{RLS}}(i) \right|^2 + \left| e_{\mathrm{LMS}}(i-1) \right|^2 \right] \right\} \tag{4.18}$$

Furthermore, by applying equation (4.8) to the last term on the RHS of equation (4.18), we obtain

$$\begin{aligned} \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left| e_{\mathrm{LMS}}(i-1) \right|^2 \right] \right\} = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \Big\{ & \mathrm{E}\left[\left| d_{\mathrm{LMS}}(i-1) \right|^2 \right] \\ & + \mathrm{E}\left[\left| y_{\mathrm{RLMS}}(i-1) \right|^2 \right] - \mathrm{E}\left[d_{\mathrm{LMS}}^{*}(i-1) y_{\mathrm{RLMS}}(i-1) \right. \\ & \left. + d_{\mathrm{LMS}}(i-1) y_{\mathrm{RLMS}}^{*}(i-1) \right] \Big\} \end{aligned} \tag{4.19}$$

The above derivation assumes the reference signals of the RLS and LMS algorithm stages are given by $d_{\mathrm{RLS}}(n)$ and $d_{\mathrm{LMS}}(n)$, respectively. In the case of a common reference signal,

$$d_{\mathrm{RLS}}(n) = d_{\mathrm{LMS}}(n) \equiv d(\mathrm{Ref}) \tag{4.20}$$

Also, y_{RLMS} is given by

$$y_{\mathrm{RLMS}} = \boldsymbol{W}_{\mathrm{RLMS}}^{H} \boldsymbol{X} \tag{4.21}$$

where

$$\boldsymbol{W}_{\mathrm{RLMS}}^{H} = \boldsymbol{W}_{\mathrm{LMS}}^{H} \boldsymbol{\mathcal{F}}_{\mathrm{R}} \boldsymbol{W}_{\mathrm{RLS}}^{H} \tag{4.22}$$

Substituting equations (4.20), (4.21) and (4.22) in equation (4.19) yields

$$\begin{aligned} \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left| e_{\mathrm{LMS}}(i-1) \right|^2 \right] \right\} = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \Big\{ & \mathrm{E}\left[\left| d_{\mathrm{LMS}}(i-1) \right|^2 \right] \Big\} \\ & - \boldsymbol{W}_{\mathrm{RLMS}}^{H}(n-1) \boldsymbol{Z}_{\mathrm{R}}(n-1) - \boldsymbol{Z}_{\mathrm{R}}^{H}(n-1) \boldsymbol{W}_{\mathrm{RLMS}}(n-1) \\ & + \boldsymbol{W}_{\mathrm{RLMS}}^{H}(n-1) \boldsymbol{Q}_{\mathrm{R}}(n-1) \boldsymbol{W}_{\mathrm{RLMS}}(n-1) \end{aligned} \tag{4.23}$$

where $\boldsymbol{Z}_{\mathrm{R}}(n)$ corresponds to the estimation of the input signal cross-correlation vector given by

$$\boldsymbol{Z}_{\mathrm{R}}(n)=\sum_{i=1}^{n}\alpha_{\mathrm{RLS}}^{n-i}\boldsymbol{X}(n)d_{\mathrm{LMS}}^{*}(n) \tag{4.24}$$

Substituting equation (4.23) in equation (4.18), we obtain the first term on the RHS of equation (4.14), such that

$$\begin{aligned}\sum_{i=1}^{n}\alpha_{\mathrm{RLS}}^{n-i}\left\{\mathrm{E}\left[\left|D_{\mathrm{R}}(i)\right|^{2}\right]\right\} &=\sum_{i=1}^{n}\alpha_{\mathrm{RLS}}^{n-i}\left\{\mathrm{E}\left[\left|d_{\mathrm{RLS}}(i)\right|^{2}+\left|d_{\mathrm{LMS}}(i-1)\right|^{2}\right]\right\}\\ &\quad-\boldsymbol{W}_{\mathrm{RLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{R}}(n-1)-\boldsymbol{Z}_{\mathrm{R}}^{H}(n-1)\boldsymbol{W}_{\mathrm{RLMS}}(n-1)\\ &\quad+\boldsymbol{W}_{\mathrm{RLMS}}^{H}(n-1)\boldsymbol{Q}_{\mathrm{R}}(n-1)\boldsymbol{W}_{\mathrm{RLMS}}(n-1)\end{aligned} \tag{4.25}$$

Using the definition of $D_{\mathrm{R}}(j)$ given in (4.15), and applying the assumptions (ii), (iii) and (iv), the second term on the RHS of equation (4.14) can be written as

$$\begin{aligned}&\sum_{i=1}^{n}\alpha_{\mathrm{RLS}}^{n-i}\mathrm{E}\left[D_{\mathrm{R}}(i)\boldsymbol{X}^{H}(i)\boldsymbol{W}_{\mathrm{RLS}}(n)+D_{\mathrm{R}}^{*}(i)\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(i)\right]\\ &\quad=\sum_{i=1}^{n}\alpha_{\mathrm{RLS}}^{n-i}\mathrm{E}\left[\left(d_{\mathrm{RLS}}(i)-e_{\mathrm{LMS}}(i-1)\right)\boldsymbol{X}^{H}(i)\boldsymbol{W}_{\mathrm{RLS}}(n)\right.\\ &\quad\quad\left.+\left(d_{\mathrm{RLS}}^{*}(i)-e_{\mathrm{LMS}}^{*}(i-1)\right)\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(i)\right]\\ &\quad=\boldsymbol{Z}_{\mathrm{R}}^{H}(n)\boldsymbol{W}_{\mathrm{RLS}}(n)+\boldsymbol{W}_{\mathrm{RLS}}^{H}(j)\boldsymbol{Z}_{\mathrm{R}}(n)\end{aligned} \tag{4.26}$$

As a result, the mean square error, ξ_{RLMS}, as specified by equation (4.14) can be rewritten to include the results of equations (4.25) and (4.26). After taking into account the relationship as indicated by equation (4.20), we obtain

$$\xi_{\mathrm{RLMS}} = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \Big\{ \mathrm{E}\Big[\left| d_{\mathrm{RLS}}(i) \right|^2 + \left| d_{\mathrm{LMS}}(i-1) \right|^2 \Big] + \boldsymbol{W}_{\mathrm{RLMS}}^{H}(n-1)\boldsymbol{Q}_{\mathrm{R}}(n-1)\boldsymbol{W}_{\mathrm{RLMS}}(n-1) + \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{Q}_{\mathrm{R}}(n)\boldsymbol{W}_{\mathrm{RLS}}(n) - \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{Z}_{\mathrm{R}}(n) - \boldsymbol{W}_{\mathrm{RLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{R}}(n-1) - \boldsymbol{Z}_{\mathrm{R}}^{H}(n)\boldsymbol{W}_{\mathrm{RLS}}(n) - \boldsymbol{Z}_{\mathrm{R}}^{H}(n-1)\boldsymbol{W}_{\mathrm{RLMS}}(n-1) \Big\} \tag{4.27}$$

Differentiating equation (4.27) with respect to the weight vector $\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)$ then yields the gradient vector $\boldsymbol{\nabla}(\xi_{\mathrm{RLMS}})$, such that

$$\boldsymbol{\nabla}(\xi_{\mathrm{RLMS}}) = -2\boldsymbol{Z}_{\mathrm{R}}(n) + 2\boldsymbol{Q}_{\mathrm{R}}(n)\boldsymbol{W}_{\mathrm{RLS}}(n) \tag{4.28}$$

By equating $\nabla(\xi_{\mathrm{RLMS}})$ to zero, we obtain the optimal weight vector, $\boldsymbol{W}_{opt_{\mathrm{RLS}}}(n)$, given by

$$\boldsymbol{W}_{opt_{\mathrm{RLS}}}(n) = \boldsymbol{Q}_{\mathrm{R}}^{-1}(n)\boldsymbol{Z}_{\mathrm{R}}(n) \tag{4.29}$$

This represents the Wiener-Hopf equation in matrix form. Therefore, the minimum mean square error (MSE) can be obtained from equations (4.29) and (4.27), such that

$$\xi_{\mathrm{RLMS,min}} = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \Big\{ \mathrm{E}\Big[\left| d_{\mathrm{RLS}}(i) \right|^2 + \left| d_{\mathrm{LMS}}(i-1) \right|^2 \Big] - \boldsymbol{Z}_{\mathrm{R}}^{H}(n)\boldsymbol{W}_{opt_{\mathrm{RLS}}}(n) - \boldsymbol{Z}_{\mathrm{R}}^{H}(n-1)\boldsymbol{W}_{\mathrm{RLMS}}(n-1) + \boldsymbol{W}_{\mathrm{RLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{R}}(n-1)\left(\boldsymbol{\mathcal{F}}_{\mathrm{R}}^{H}\boldsymbol{W}_{\mathrm{LMS}}(n-1) - 1 \right) \Big\} \tag{4.30}$$

Based on equations (4.29) and (4.30), and by dropping the index n, equation (4.27) becomes

$$\xi_{\mathrm{RLMS}} = \xi_{\mathrm{RLMS,min}} + \left(\boldsymbol{W}_{\mathrm{RLS}} - \boldsymbol{W}_{opt_{\mathrm{RLS}}} \right)^{H} \boldsymbol{Q}_{\mathrm{R}} \left(\boldsymbol{W}_{\mathrm{RLS}} - \boldsymbol{W}_{opt_{\mathrm{RLS}}} \right) \tag{4.31}$$

Now, let us define the error of the weight vector of the RLS algorithm, $\boldsymbol{V}_{\mathrm{RLS}}$, as

$$\boldsymbol{V}_{\mathrm{RLS}} \triangleq \left(\boldsymbol{W}_{\mathrm{RLS}} - \boldsymbol{W}_{opt_{\mathrm{RLS}}}\right) \tag{4.32}$$

so that equation (4.31) can be written as

$$\xi_{\mathrm{RLMS}} = \xi_{\mathrm{RLMS,min}} + \boldsymbol{V}_{\mathrm{RLS}}^{H} \boldsymbol{Q}_{\mathrm{R}} \boldsymbol{V}_{\mathrm{RLS}} \tag{4.33}$$

Differentiating equation (4.33) with respect to $\boldsymbol{V}_{\mathrm{RLS}}^{H}$ will yield another form for the gradient [32], such that

$$\nabla(\xi_{\mathrm{RLMS}}(n)) = \boldsymbol{Q}_{\mathrm{R}}(n)\boldsymbol{V}_{\mathrm{RLS}}(n) \tag{4.34}$$

Using eigenvalue decomposition (EVD) of $\boldsymbol{Q}_{\mathrm{R}}$ in equation (4.34), we obtain

$$\boldsymbol{Q}_{\mathrm{R}} = \boldsymbol{q}_{\mathrm{R}} \boldsymbol{\Lambda}_{\mathrm{R}} \boldsymbol{q}_{\mathrm{R}}^{-1} = \boldsymbol{q}_{\mathrm{R}} \boldsymbol{\Lambda}_{\mathrm{R}} \boldsymbol{q}_{\mathrm{R}}^{H} \tag{4.35}$$

where $\boldsymbol{q}_{\mathrm{R}}$ and $\boldsymbol{\Lambda}_{\mathrm{R}}$ are the eigenvectors and the diagonal matrices of $\boldsymbol{Q}_{\mathrm{R}}$ respectively, i.e.,

$$\boldsymbol{\Lambda}_{\mathrm{R}} = \mathrm{diag}[E_1, E_2, \ldots\ldots\ldots\ldots\ldots, E_N] \tag{4.36}$$

Let,

$$\boldsymbol{V}'_{\mathrm{RLS}} \triangleq \boldsymbol{q}_{\mathrm{R}}^{-1} \boldsymbol{V}_{\mathrm{RLS}} \tag{4.37}$$

then

$$\boldsymbol{V}_{\mathrm{RLS}} = \boldsymbol{q}_{\mathrm{R}} \boldsymbol{V}'_{\mathrm{RLS}} \tag{4.38}$$

Based on equations (4.35) and (4.38), we can express the MSE of equation (4.33) as

$$\xi_{\mathrm{RLMS}}(n) = \xi_{\mathrm{RLMS,min}}(n) + \boldsymbol{V}'^{H}_{\mathrm{RLS}}(n) \boldsymbol{\Lambda}_{\mathrm{R}} \boldsymbol{V}'_{\mathrm{RLS}}(n) \tag{4.39}$$

For steepest descent, the weight vector is updated according to

$$\boldsymbol{W}_{\mathrm{RLS}}(n+1) = \boldsymbol{W}_{\mathrm{RLS}}(n) - \mu_{\mathrm{RLS}} \nabla(\xi_{\mathrm{RLMS}}(n)) \tag{4.40}$$

where μ_{RLS} is the convergence constant that controls stability and rate of adaptation of the weight vector, and $\nabla(\xi_{\mathrm{RLMS}}(n))$ is the gradient at the n^{th} iteration.

Subtracting $\boldsymbol{W}_{opt_{\mathrm{RLS}}}$ from both sides of equation (4.40) yields

$$\boldsymbol{W}_{\mathrm{RLS}}(n+1)-\boldsymbol{W}_{opt_{\mathrm{RLS}}}(n+1)=\boldsymbol{W}_{\mathrm{RLS}}(n)-\boldsymbol{W}_{opt_{\mathrm{RLS}}}(n)-\mu_{\mathrm{RLS}}\nabla\xi_{\mathrm{RLMS}}(n) \qquad (4.41)$$

Applying equation (4.32) to equation (4.41) gives

$$\boldsymbol{V}_{\mathrm{RLS}}(n+1)=\boldsymbol{V}_{\mathrm{RLS}}(n)-\mu_{\mathrm{RLS}}\nabla(\xi_{\mathrm{RLMS}}(n)) \qquad (4.42)$$

Multiplying both sides of equation (4.42) by $\boldsymbol{q}_{\mathrm{R}}{}^{H}$ yields

$$\boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{V}_{\mathrm{RLS}}(n+1)=\boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{V}_{\mathrm{RLS}}(n)-\mu_{\mathrm{RLS}}\boldsymbol{q}_{\mathrm{R}}{}^{H}\nabla(\xi_{\mathrm{RLMS}}(n)) \qquad (4.43)$$

Now, $\nabla(\xi_{\mathrm{RLMS}}(n))$ in equation (4.43) can be replaced by equation (4.34) to obtain

$$\boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{V}_{\mathrm{RLS}}(n+1)=\boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{V}_{\mathrm{RLS}}(n)-\mu_{\mathrm{RLS}}\boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{Q}_{\mathrm{R}}\boldsymbol{V}_{\mathrm{RLS}}(n) \qquad (4.44)$$

With the correlation matrix, $\boldsymbol{Q}_{\mathrm{R}}$, in equation (4.44) replaced by equation (4.35), we can simplify equation (4.44), such that

$$\boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{V}_{\mathrm{RLS}}(n+1)=\boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{V}_{\mathrm{RLS}}(n)-\mu_{\mathrm{RLS}}\boldsymbol{\Lambda}_{\mathrm{R}}\boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{V}_{\mathrm{RLS}}(n) \qquad (4.45)$$

Using the relationship of equation (4.38), the above equation can be rewritten as

$$\boldsymbol{V}'_{\mathrm{RLS}}(n+1)=\left(\boldsymbol{I}-\mu_{\mathrm{RLS}}\boldsymbol{\Lambda}_{\mathrm{R}}\right)\boldsymbol{V}'_{\mathrm{RLS}}(n) \qquad (4.46)$$

Equation (4.46) can also be expressed as

$$\boldsymbol{V}'_{\mathrm{RLS}}(n)=\left(\boldsymbol{I}-\mu_{\mathrm{RLS}}\boldsymbol{\Lambda}_{\mathrm{R}}\right)^{n}\boldsymbol{V}'_{\mathrm{RLS}}(0) \qquad (4.47)$$

where $\boldsymbol{V}'_{\mathrm{RLS}}(0)$ is the initial value given by

$$\boldsymbol{V}'_{\mathrm{RLS}}(0) = \boldsymbol{W}'_{\mathrm{RLS}}(0) - \boldsymbol{W}'_{opt_{\mathrm{RLS}}} \tag{4.48}$$

Substituting equation (4.47) in equation (4.39), the mean square error becomes

$$\xi_{\mathrm{RLMS}}(n) = \xi_{\mathrm{RLMS,min}} + \boldsymbol{V}^{H}_{\mathrm{RLS}}(0)\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)^{n}\boldsymbol{Q}_{\mathrm{R}}\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)^{n}\boldsymbol{V}_{\mathrm{RLS}}(0) \tag{4.49}$$

where $\boldsymbol{V}_{\mathrm{RLS}}(0) = \boldsymbol{q}_{\mathrm{R}}{}^{-1}\boldsymbol{V}_{\mathrm{RLS}}(0)$.

Following Appendix B, the step size boundaries of the RLS algorithm stage, μ_{RLS}, can be analyzed to obtain

$$0 < \mu_{\mathrm{RLS}} < {}^{2}\!\!\diagup\!\!{}_{E_{\mathrm{RLS}}} \tag{4.50}$$

where E_{RLS} is the maximum eigenvalue of $\boldsymbol{\Lambda}_{\mathrm{R}}$.

In the limit, we have

$$\begin{aligned}
&\lim_{n\to\infty}\left[\xi_{\mathrm{RLMS}}(n)\right] \\
&= \lim_{n\to\infty}\left[\xi_{\mathrm{RLMS,min}} + \boldsymbol{V}^{H}_{\mathrm{RLS}}(0)\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)^{n}\boldsymbol{Q}_{\mathrm{R}}\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)^{n}\boldsymbol{V}_{\mathrm{RLS}}(0)\right] \\
&= \xi_{\mathrm{RLMS,min}} + \boldsymbol{V}^{H}_{\mathrm{RLS}}(0)\left\{\lim_{n\to\infty}\left[\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)^{n}\boldsymbol{Q}_{\mathrm{R}}\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)^{n}\right]\right\}\boldsymbol{V}_{\mathrm{RLS}}(0)
\end{aligned} \tag{4.51}$$

Under the condition that $\left|\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)\right| < 1$, the second RHS term of equation (4.51) vanishes. As a result, the mean square error converges to a minimum value, such that

$$\lim_{n\to\infty}\xi_{\mathrm{RLMS}}(n) = \xi_{\mathrm{RLMS,min}} \tag{4.52}$$

4.3.2 Analysis of the self-referencing scheme

After the RLMS algorithm has converged, usually within a few iterations, the external common reference signal may be replaced by the intermediate

output, y_{RLS}, of the RLS algorithm stage and the output of the LMS algorithm stage, y_{RLMS}, so that

$$d_{\mathrm{RLS}}(n) = y_{\mathrm{RLMS}}(n-1) \tag{4.53}$$

and

$$d_{\mathrm{LMS}}(n) = y_{\mathrm{RLS}}(n) \tag{4.54}$$

This mode of operation of the RLMS algorithm is referred to as self-referencing. With these changes and observing that $e_{\mathrm{LMS}} = d_{\mathrm{LMS}} - y_{\mathrm{RLMS}}$, then we can redefine $D_{\mathrm{R}}(n)$ in equation (4.15) as

$$D_{\mathrm{R}}(n) = 2y_{\mathrm{RLMS}}(n-1) - y_{\mathrm{RLS}}(n-1) \tag{4.55}$$

Based on the above modification, we reanalyze the MSE expression of equation (4.14), as described in Appendix C, to obtain

$$\begin{aligned}\xi_{\mathrm{RLMS}}(n) = {} & \boldsymbol{W}_{\mathrm{RLS}}^{H}(n-1)\boldsymbol{Q}_{\mathrm{R}}(n-1)\boldsymbol{W}_{\mathrm{RLS}}(n-1) - \boldsymbol{Z}_{\mathrm{R}}'^{H}(n)\boldsymbol{W}_{\mathrm{RLS}}(n) \\ & - \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{Z}_{\mathrm{R}}'(n) + \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{Q}_{\mathrm{R}}(n)\boldsymbol{W}_{\mathrm{RLS}}(n)\end{aligned} \tag{4.56}$$

where $\boldsymbol{Z}_{\mathrm{R}}'(n)$ corresponds to the input signal cross-correlation vector given by

$$\boldsymbol{Z}_{\mathrm{R}}'(n) = \mathrm{E}\left[\boldsymbol{X}(n)D_{\mathrm{R}}^{*}(n)\right] \tag{4.57}$$

It is shown in Appendix C that the rest of analysis is similar to that of the previous case involving the use of an external reference. We can conclude that the RLMS algorithm will continue to converge using these internally generated signals as reference signals for the RLS and LMS algorithm stages.

4.4 Mean Weight Vector Convergence

In this section, we determine the values of the step sizes, μ_{RLS} and μ_{LMS},

required for stable operation of the RLMS algorithm. To simplify the analysis, we assume convergence of the RLMS algorithm when both the RLS and LMS algorithm stages have converged. As such, we can treat each of the RLS and LMS algorithm stages separately.

For the RLS algorithm stage, the value of μ_{RLS} for updating the weights in equation (4.11) is specified in (4.50) to be within the range of

$$0 < \mu_{\mathrm{RLS}} < 2 \big/ E_{\mathrm{RLS}} \tag{4.58}$$

For the LMS algorithm stage, let the error signal in equation (4.10) be given by [68, 135]

$$e_{\mathrm{LMS}}(n) = e''_{\mathrm{LMS}}(n) - \boldsymbol{X}^{H}_{\mathrm{LMS}}(n)\boldsymbol{V}_{\mathrm{LMS}}(n) \tag{4.59}$$

where e''_{LMS} is a zero mean measurement noise, independent from the signal, and $\boldsymbol{V}_{\mathrm{LMS}}$ is the weight vector error, such that

$$\boldsymbol{V}_{\mathrm{LMS}}(n) = \boldsymbol{W}_{\mathrm{LMS}}(n) - \boldsymbol{W}_{0\,\mathrm{LMS}}(n) \tag{4.60}$$

Let the time-varying weight vector be modeled by a random walk process [68], so that

$$\boldsymbol{W}_{0\,\mathrm{LMS}}(n+1) = \boldsymbol{W}_{0\,\mathrm{LMS}}(n) + \boldsymbol{r}_{\mathrm{LMS}}(n) \tag{4.61}$$

where $\boldsymbol{W}_{0\,\mathrm{LMS}}(n)$ is the optimal weight vector of the LMS algorithm stage, and $\boldsymbol{r}_{\mathrm{LMS}}(n)$ is a zero mean white sequence vector with diagonal correlation matrix $\sigma^2_{\mathrm{R},r}\boldsymbol{I}$ and $\sigma^2_{\mathrm{R},r}$ being the weight variance.

Now, subtract equation (4.61) from equation (4.10) to yield

$$\boldsymbol{V}_{\mathrm{LMS}}(n+1) = \boldsymbol{V}_{\mathrm{LMS}}(n) + \mu_{\mathrm{LMS}} e_{\mathrm{LMS}}(n)\boldsymbol{X}_{\mathrm{LMS}}(n) - \boldsymbol{r}_{\mathrm{LMS}}(n) \tag{4.62}$$

Substituting equation (4.59) into equation (4.62) and applying the unitary

transformation $\boldsymbol{q}_{\mathrm{R}}^{H}$ in equation (4.35) to both sides of equation (4.62) gives

$$\begin{aligned}\boldsymbol{v}_{\mathrm{LMS}}(n+1) = [\boldsymbol{I} - \mu_{\mathrm{LMS}}\tilde{\boldsymbol{X}}_{\mathrm{LMS}}(n)\tilde{\boldsymbol{X}}_{\mathrm{LMS}}^{H}(n)]\boldsymbol{v}_{\mathrm{LMS}}(n) \\ + \mu_{\mathrm{LMS}}e''_{\mathrm{LMS}}(n)\tilde{\boldsymbol{X}}_{\mathrm{LMS}}(n) - \tilde{\boldsymbol{r}}_{\mathrm{LMS}}(n)\end{aligned} \tag{4.63}$$

where

$$\begin{aligned}\tilde{\boldsymbol{r}}_{\mathrm{LMS}}(n) &= \boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{r}_{\mathrm{LMS}}(n), \\ \tilde{\boldsymbol{X}}_{\mathrm{LMS}}(n) &= \boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{X}_{\mathrm{LMS}}(n), \\ \text{and} \quad \boldsymbol{v}_{\mathrm{LMS}}(n) &= \boldsymbol{q}_{\mathrm{R}}{}^{H}\boldsymbol{V}_{\mathrm{LMS}}(n)\end{aligned} \tag{4.64}$$

To determine the condition for convergence, we consider the expected values on both sides of equation (4.63). This leads to the second and third RHS terms of equation (4.63) vanishing as both $e''_{\mathrm{LMS}}(n)$ and $\tilde{\boldsymbol{X}}_{\mathrm{LMS}}(n)$ are uncorrelated. As a result, we obtain

$$\mathrm{E}\left[\boldsymbol{v}_{\mathrm{LMS}}(n+1)\right] = \left\{\boldsymbol{I} - \mu_{\mathrm{LMS}}\mathrm{E}\left[\tilde{\boldsymbol{X}}_{\mathrm{LMS}}(n)\tilde{\boldsymbol{X}}_{\mathrm{LMS}}^{H}(n)\right]\right\}\mathrm{E}\left[\boldsymbol{v}_{\mathrm{LMS}}(n)\right] \tag{4.65}$$

With $\boldsymbol{R}_{\mathrm{LMS}}(n) = \mathrm{E}\left[\tilde{\boldsymbol{X}}_{\mathrm{LMS}}(n)\tilde{\boldsymbol{X}}_{\mathrm{LMS}}^{H}(n)\right]$, equation (4.65) can be rewritten as

$$\mathrm{E}\left[\boldsymbol{v}_{\mathrm{LMS}}(n+1)\right] = \left\{\boldsymbol{I} - \mu_{\mathrm{LMS}}\boldsymbol{R}_{\mathrm{LMS}}(n)\right\}\mathrm{E}\left[\boldsymbol{v}_{\mathrm{LMS}}(n)\right] \tag{4.66}$$

where the general coefficients of the matrix $\boldsymbol{R}_{\mathrm{LMS}}(n)$ can be expressed as

$$\begin{aligned}\left(r_{r,ka}\right) &= \left(\sum_{l,m}^{N}\mathrm{E}\left[\mathcal{F}_{r,k}w_{\mathrm{RLS},l}^{*}x_l x_m^{*}w_{\mathrm{RLS},m}\mathcal{F}_{r,a}^{*}\right]\right) \\ &= \left(\mathcal{F}_{r,k}\mathcal{F}_{r,a}^{*}\sum_{l}^{N}\mathrm{E}\left[w_{\mathrm{RLS},l}^{*}x_l x_m^{*}w_{\mathrm{RLS},m}\right]\right)\end{aligned} \tag{4.67}$$

Based on the assumption (i), as stated in Section 4.3.1, the elements of the matrix $\left(r_{r,ka}\right)$ in equation (4.67) may be rewritten as

$$r_{r,ka} = \mathcal{F}_{r,k}\mathcal{F}_{r,a}^{*}\sum_{l}^{N}\mathrm{E}\left[y_l y_l^{*}\right] = \mathcal{F}_{r,k}\mathcal{F}_{r,a}^{*}\sigma_{\mathrm{RLS}}^{2} \tag{4.68}$$

where σ^2_{RLS} is the variance of the output of the RLS algorithm stage.

Note that the products $x_l x^*_m$ and $w_{\mathrm{RLS},l} w^*_{\mathrm{RLS},m}$ are non-zero only for $m = l$, according to the Kronecker delta function, i.e., $\delta^m_l = \begin{cases} 0 & m \neq l \\ 1 & m = l \end{cases}$.

In a matrix form, equation (4.68) can be expressed as

$$\boldsymbol{R}_{\mathrm{LMS}}(j) = \sigma^2_{\mathrm{RLS}} \boldsymbol{\mathcal{F}}_{\mathrm{R}} \boldsymbol{\mathcal{F}}^H_{\mathrm{R}} \tag{4.69}$$

where $\boldsymbol{R}_{\mathrm{LMS}}$ is a complex matrix, with a rank equal to one. Substituting this in equation (4.66) and analyzing using eigenvalue decomposition (EVD) [136], we have

$$\begin{aligned} \mathrm{E}\left[\boldsymbol{v}_{\mathrm{LMS}}(n+1)\right] &= \boldsymbol{U}\left(\boldsymbol{I} - \mu_{\mathrm{LMS}} \sigma^2_{\mathrm{RLS}} \boldsymbol{\Lambda}_2\right) \boldsymbol{U}^H \mathrm{E}\left[\boldsymbol{v}_{\mathrm{LMS}}(n)\right] \\ &= \boldsymbol{U}\left(\boldsymbol{I} - \mu_{\mathrm{LMS}} \sigma^2_{\mathrm{RLS}} \boldsymbol{\Lambda}_2\right)^{n+1} \boldsymbol{U}^H \mathrm{E}\left[\boldsymbol{v}_{\mathrm{LMS}}(0)\right] \end{aligned} \tag{4.70}$$

where $\boldsymbol{U}$ is an *N*-by-*N* unitary matrix and $\boldsymbol{\Lambda}_2$ is the diagonal matrix of eigenvalues of the array matrix ($\boldsymbol{\mathcal{F}}_{\mathrm{R}} \boldsymbol{\mathcal{F}}^H_{\mathrm{R}}$). In this case, $\boldsymbol{\Lambda}_2 = \mathrm{diag}[\lambda_{\mathrm{R},2}, 0, 0, \ldots\ldots\ldots, 0]$, provided that this matrix has only one eigenvalue (i.e., it is singular of rank 1). Since $\boldsymbol{\mathcal{F}}_{\mathrm{R}}$ is a normalized vector, then $\left\|\boldsymbol{\mathcal{F}}_{\mathrm{R}}\right\|^2 = N$, so that this eigenvalue is equal to N, that is

$$\lambda_{\mathrm{R},2} = \mathrm{trace}(\boldsymbol{\mathcal{F}}_{\mathrm{R}} \boldsymbol{\mathcal{F}}^H_{\mathrm{R}}) = N \tag{4.71}$$

From equation (4.70), the condition for convergence can be satisfied if $\left|1 - \mu_{\mathrm{LMS}} \sigma^2_{\mathrm{RLS}} \lambda_{\mathrm{R}\,2}\right| < 1$. Therefore, for the LMS algorithm stage to converge, it requires that the step size used for updating the weights is within the range given by

$$0 < \mu_{\mathrm{LMS}} < 2 \big/ N \sigma^2_{\mathrm{RLS}} \tag{4.72}$$

Thus, in order to achieve convergence for the RLMS algorithm, the values of the step sizes, μ_{RLS} and μ_{LMS}, must satisfy both equations (4.58) and (4.72).

4.5 Estimation of the Array Image Vector $\mathcal{F}_{\mathrm{R}}$

For self-adaptive beamforming, it requires that the array image vector, $\mathcal{F}_{\mathrm{R}}$, be adjusted automatically to always track the AOA of the desired signal. A simple method for estimating $\mathcal{F}_{\mathrm{R}}$ is now described.

Rearranging equation (4.1), in element form, gives

$$x_k(t) = A_{d,k}s_d(t) + A_{i,k}s_i(t) + n_k(t) \tag{4.73}$$

where $A_{d,k}$ is the k^{th} element of $\boldsymbol{A}_d$ with $k = 1, 2, \ldots, N$.

The outputs of the individual taps of the RLS algorithm stage, $(w_{\mathrm{RLS},k})$ are given by

$$x'_k(t) = w_{\mathrm{RLS},k}\, x_{1,k}(t) \tag{4.74}$$

When the RLS algorithm stage converges, the output y_{RLS} tends to approach $s_d(t)$ with both the interference $s_i(t)$ and noise $n_k(t)$ being suppressed. Thus, let

$$y_{\mathrm{RLS}}(t) \simeq s_d(t) \tag{4.75}$$

and taking the expectation of both sides of equation (4.73), we have

$$\mathrm{E}[x_{1,k}(t)] = A_{d,k}\mathrm{E}[s_d(t)] \approx A_{d,k}\mathrm{E}[y_{\mathrm{RLS}}(t)] \tag{4.76}$$

Assume that, after convergence, we can approximate

$$\mathrm{E}[y_{\mathrm{RLS}}(t)] \approx y_{\mathrm{RLS}}(t) \tag{4.77}$$

Thus, equation (4.76) can be rewritten as

$$\mathrm{E}[x_{1,k}(t)] \approx A_{d,k}\, y_{\mathrm{RLS}}(t) \tag{4.78}$$

By assuming both the input signal and the RLS algorithm weights are independent, the expectation of equation (4.74) can be written as

$$\mathrm{E}[x'_k(t)] = \mathrm{E}[w_{\mathrm{RLS},k}]\,E[x_{1,k}(t)]. \tag{4.79}$$

From equations (4.78) and (4.79), we can estimate the array vector elements as

$$A_{d,k}(t) \simeq \frac{\mathrm{E}[x'_k(t)]}{\mathrm{E}[w_{\mathrm{RLS},k}]\,y_{\mathrm{RLS}}(t) + \varepsilon_c} \tag{4.80}$$

where ε_c is a small constant introduced to prevent overflow produced by a possible divide by zero condition in equation (4.80). Its value is chosen such that

$$\varepsilon_c \ll \frac{|y_{\mathrm{RLS}}|}{N} \sum_{\mathrm{k}=1}^{\mathrm{N}} |w_{\mathrm{RLS},k}| \tag{4.81}$$

For the computer simulations described in this chapter, ε_c has been set to 0.004[4]. It follows that the instantaneous values of the elements of $\boldsymbol{A}_d$ can be expressed as

$$A_{d,k}(t) \simeq \frac{x'_k(t)}{w_{\mathrm{RLS},k}(t)\,y_{\mathrm{RLS}}(t) + \varepsilon_c} \tag{4.82}$$

Thus, equation (4.82) provides a mean of calculating the array image vector $\mathcal{F}_{\mathrm{R}}$ for use in the RLMS algorithm.

4.6 Fixed Beamforming using the RLMS Algorithm

According to Figure 4.1, the input stage of the RLMS scheme is based on the RLS algorithm with its weight vector given in equation (4.11). Therefore, the output of the RLS algorithm stage at the n^{th} iteration can be expressed as

[4] This constant has been obtained based on the smallest quantization step size associated with a wordlength of 8 bits.

$$y_{\mathrm{RLS}}(n) = \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(n) \tag{4.83}$$

Based on this intermediate output signal, the input signal vector for the LMS algorithm stage can be obtained, such that

$$\boldsymbol{X}_{\mathrm{LMS}} = \boldsymbol{\mathcal{F}}_{\mathrm{R}}\, y_{\mathrm{RLS}} \tag{4.84}$$

Finally, the output of the RLMS beamformer is given by

$$y_{\mathrm{RLMS}} = \boldsymbol{W}_{\mathrm{LMS}}^{H}\boldsymbol{X}_{\mathrm{LMS}} = \boldsymbol{W}_{\mathrm{LMS}}^{H} y_{\mathrm{RLS}} \boldsymbol{\mathcal{F}}_{\mathrm{R}} \tag{4.85}$$

Equation (4.85) indicates the central role played by the array image vector $\boldsymbol{\mathcal{F}}_{\mathrm{R}}$ in beamforming using the RLMS algorithm. Now, by prescribing the individual elements of $\boldsymbol{\mathcal{F}}_{\mathrm{R}}$ with values corresponding to the required AOA, the resulting output will contain only those signal components "selected" by $\boldsymbol{\mathcal{F}}_{\mathrm{R}}$. For example, by setting $\boldsymbol{\mathcal{F}}_{\mathrm{R}} = \boldsymbol{A}_d(\text{at } \theta_d = 30^\circ)$, a fixed beam pointing in the direction of $\theta_d = 30^\circ$ is thus obtained. With this scheme, variations in operating condition and component tolerance are compensated through adaptive adjustments of the tap weights in the RLS and LMS algorithm stages.

In order to differentiate the two modes of operation of the proposed algorithm, the term RLMS_1 algorithm is used to associate with a fixed beamforming using the RLMS algorithm.

4.7 Computer Simulations

4.7.1 Introduction

The performance of the proposed RLMS algorithm, with either external reference or self reference, has been evaluated by means of extensive computer simulations. These simulations study the rate of convergence of

the RLMS algorithm under different input signal-to-noise ratios, SNR[5]. Also, the stability of the algorithm as a function of the step sizes, μ_{RLS} and μ_{LMS}, used is investigated. Furthermore, the simulations are used to verify the tracking performance and robustness to noisy reference of the RLMS algorithm. In addition, the flexibility of the algorithm in realizing fixed beamforming is demonstrated. For comparison purposes, adaptive beamformers using the conventional RLS and LMS algorithms as well as other published algorithms, such as VFFRLS, CSLMS and MRVSS algorithms, which have been reviewed in Chapter 3, are also simulated. These simulations have been carried out with the mathematical functions implemented using the full numerical precision of the computer. However, finite numerical precision is more likely to be used for practical implementation. As such, the influence of quantization and rounding errors introduced by the use of finite wordlength on the operation of a given algorithm has also been considered. The latter will be described in Section 6.2.1.

4.7.2 Simulation setup

For the simulations carried out with full numerical precision, a given algorithm is assumed to operate under the following environment:

- A linear array consisting of 8 isotropic antenna elements spaced half a wavelength apart.
- A desired binary phase shift keying (BPSK) arrives at an angle $\theta_d = 0^\circ$.
- An AWGN channel.
- All weight vectors are initially set to zero.
- Two BPSK interference signals arrive at $\theta_i = -30^\circ$ and $\theta_i = 45^\circ$ have the same amplitude as the desired signal.

---

[5] Signal-to-noise ratio (SNR) is defined as the ratio of the average signal power to AWGN power, determined over the signal bandwidth. The signal power is obtained from averaging over 16 M symbols.

Appendix A gives the procedure used in the simulations of the $RLMS_1$, RLMS, CSLMS, MRVSS, RLS, VFFRLS and LMS algorithms. For the CSLMS algorithm, the parameter ε_{cs} is a small positive constant which has been adjusted to yield the best possible performance. In the case of the MRVSS algorithm, its step size μ is updated using a value within the upper and lower boundary values of $\mu_{\max}$ and $\mu_{\min}$, respectively. Also, α, η, γ and υ are the parameters required by the MRVSS algorithm and these are given in Table 4-1. In addition, the parameters $\tilde{e}_{\max}$ and $\tilde{e}_{\min}$ are the upper and lower bounds of the time averaged error square signal, $\tilde{e}$, of the MRVSS algorithm.

Table 4-1 shows the numerical values of the various constants adopted for the simulations of the seven different adaptive algorithms. The parameter values for the MRVSS algorithm operating in an AWGN channel are those given in [10, 54, 56]. All other values adopted here for the MRVSS and CSLMS algorithms have been chosen for obtaining the best performance out of these algorithms. The step size values, μ_{RLS} and μ_{LMS}, associated with the $RLMS_1$ and RLMS algorithms have been chosen to yield a low error floor.

Often, performance comparison between different adaptive beamforming schemes is made in terms of the convergence errors and resultant beam patterns. Moreover, for a digitally modulated signal, it is also convenient to make use of the error vector magnitude (EVM) as an accurate measure of any distortion introduced by the adaptive beamforming scheme on the received signal at a given signal-to-noise ratio (SNR). It is shown in [137] that EVM is more sensitive to variations in SNR than bit error rate (BER). EVM is defined as [138]

$$EVM_{\text{RMS}} = \sqrt{\frac{1}{k\tilde{P}}\sum_{n=1}^{K}\left|y_{\text{RLMS}}(n)-x(n)\right|^2}\ , \tag{4.86}$$

where k is the number of symbols used, $y_{\text{RLMS}}(n)$ is the n^{th} output of the beamformer, and $x(n)$ is the n^{th} transmit symbol. $\tilde{P}$ is the average power of all the symbols involved for the given modulation.

Table 4-1 Values of the constants used in the simulations.

Algorithm	AWGN Channel	Rayleigh Fading Channel
LMS	$\mu = 0.05$	$\mu = 0.01$
RLS	$\mu_{\text{RLS}} = 0.05$	$\mu_{\text{RLS}} = 0.01$
VFFRLS	$\mu_{\text{VFFRLS}} = 0.05, \lambda_{\max} = 1, \gamma_{\text{VFF}} = 1.5,$ $k_\alpha = 6,\ k_\beta = k_\alpha,\ \varepsilon_{\text{VFF}} = 10^{-8}, \delta = 1$	$\mu_{\text{VFFRLS}} = 0.01, \lambda_{\max} = 1, \gamma_{\text{VFF}} = 1.5,$ $k_\alpha = 6,\ k_\beta = k_\alpha,\ \varepsilon_{\text{VFF}} = 10^{-8}, \delta = 1$
CSLMS	$\varepsilon_{cs} = 0.05,\ \mu = 0.05$	$\varepsilon_{cs} = 1,\ \mu = 0.01$
MRVSS	$\tilde{e}_{\max} = 1,\ \tilde{e}_{\min} = 0,\ \upsilon = 5\times10^{-4}$ $\mu_{\max} = 0.2,\ \mu_{\min} = 10^{-4}$ Initial $\mu = \mu_{\max},\ \alpha = 0.97$ $\gamma = 4.8\times10^{-4},\ \eta = 0.97$	Same as column 2 except for: Initial $\mu = 0.1,\ \mu_{\max} = 0.05$ $\mu_{\min} = 10^{-4}$
RLMS_1	$\mu_{\text{RLS}} = 0.03,\ \ \mu_{\text{LMS}} = 0.02$	$\mu_{\text{RLS}} = 0.1,\ \ \mu_{\text{LMS}} = 0.01$
RLMS	$\mu_{\text{RLS}} = 0.05,\ \ \mu_{\text{LMS}} = 0.05$	$\mu_{\text{RLS}} = 0.1,\ \ \mu_{\text{LMS}} = 0.01$

4.7.3 Simulation results

Computer simulations based on Matlab have been carried out to evaluate and compare the performances of the RLMS and RLMS_1 algorithms with the other five algorithms, namely the CSLMS, MRVSS, RLS, VFFRLS and LMS algorithms. The performance indicators adopted are error convergence obtained with either an ideal or noisy reference signal, signal tracking ability, EVM and scatter plot.

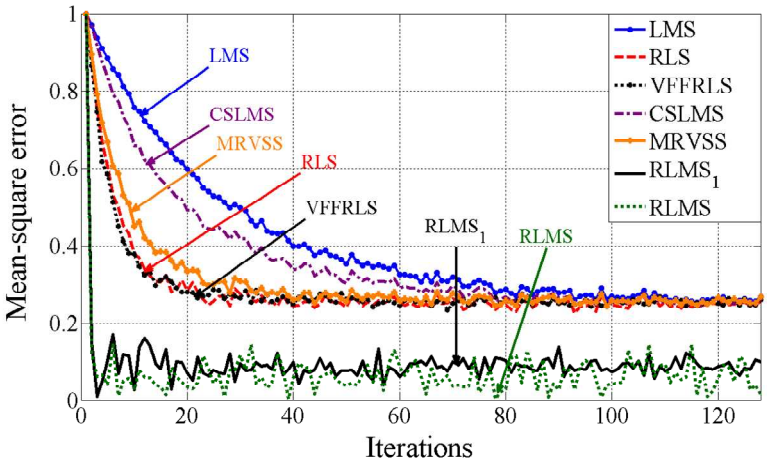
4.7.3.1 Error convergence with an ideal external reference

First, the convergence performance of a beamformer implemented using one of the seven algorithms given is evaluated in the presence of an ideal external reference signal. In each case, the ensemble average squared error

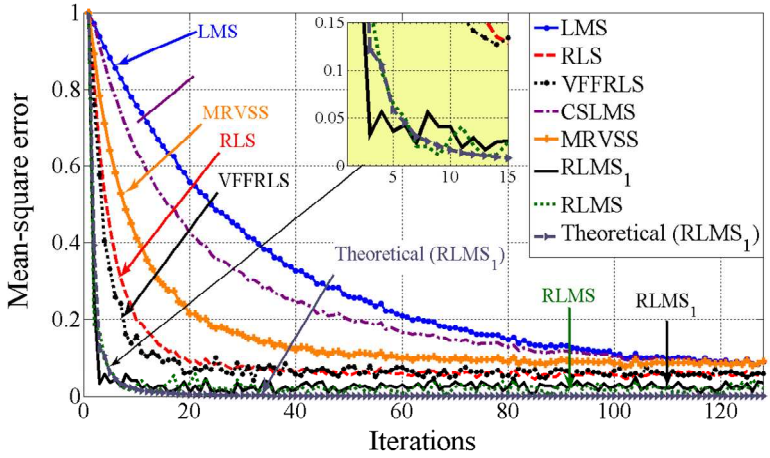
$\left(\tilde{e}^2\right)$ is obtained from 100 individual simulation runs with each run consisting of 128 iterations. This has been carried out for three different input signal-to-noise ratios.

Figures 4-2(a)–(c) show the convergence behaviours of the seven adaptive schemes for SNR values of 5, 10, and 15 dB, respectively. For the proposed $RLMS_1$ algorithm, which makes use of the prescribed values for the individual elements of the image array vector $\mathcal{F}_R$, the theoretical convergence error calculated using equation (4.27) for an SNR of 10 dB is also shown in Figure 4-2b. It is observed that under the given conditions, the two variants of the proposed RLMS algorithm converge much faster than the other five schemes. Furthermore, their error floors are less sensitive to variations in the input SNR, even for an input SNR as small as 5 dB. Also, as shown in Figure 4-2b, there is a close agreement between the simulated and theoretical error curves for the proposed RLMS and $RLMS_1$ algorithms. This validates the method for estimating $\mathcal{F}_R$ for the RLMS algorithm as described in Section 4.4. Among the other five algorithms, the RLS and VFFRLS algorithms outperform the MRVSS, CSLMS, and LMS algorithms for all the three SNR values considered. As expected, the conventional LMS algorithm is the slowest among the seven algorithms. Moreover, both the $RLMS_1$ and RLMS algorithms have almost identical convergence performance for all the three SNR values considered.

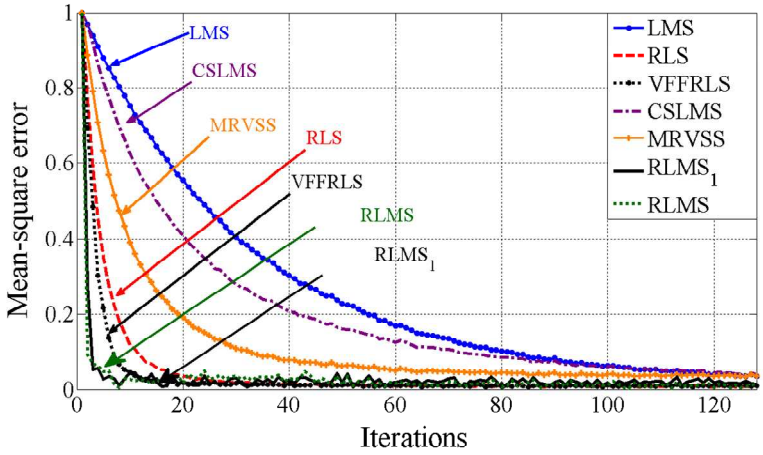
Next, the sensitivity of the $RLMS_1$ and RLMS algorithms to the step sizes used is considered. In this case, the channel is assumed to be free from noise and interference. Also, the step size of the RLS algorithm stage, μ_{RLS}, is set to three different values (0, 0.07, 0.1). Figure 4-3 shows the mean square value of the overall error signal, e_{RLMS}, measured after 128 iterations as a function of the step size, μ_{LMS}, used in the LMS algorithm stage. From Figure 4-3, it is observed that the use of any one of the three different values of μ_{RLS} has very little or no effect on the stability of the $RLMS_1$ and RLMS algorithms.



(a) SNR=5dB



(b) SNR=10 dB



(c) SNR=15 dB

Figure 4-2 The convergence of RLMS, $RLMS_1$, CSLMS, MRVSS, RLS, VFFRLS and LMS algorithms with the parameters given in the 2nd column of Table 4-1, for three different values of input SNR.

Moreover, the operation of the $RLMS_1$ and RLMS algorithms will remain stable provided μ_{LMS} is less than 1.7 and 1.9, respectively. It is also noticed from Figure 4-3, that for very small values of μ_{LMS}, the MSE values are high. This is due to the fact that under such conditions, the contribution to overall error from the LMS algorithm stage becomes negligible.

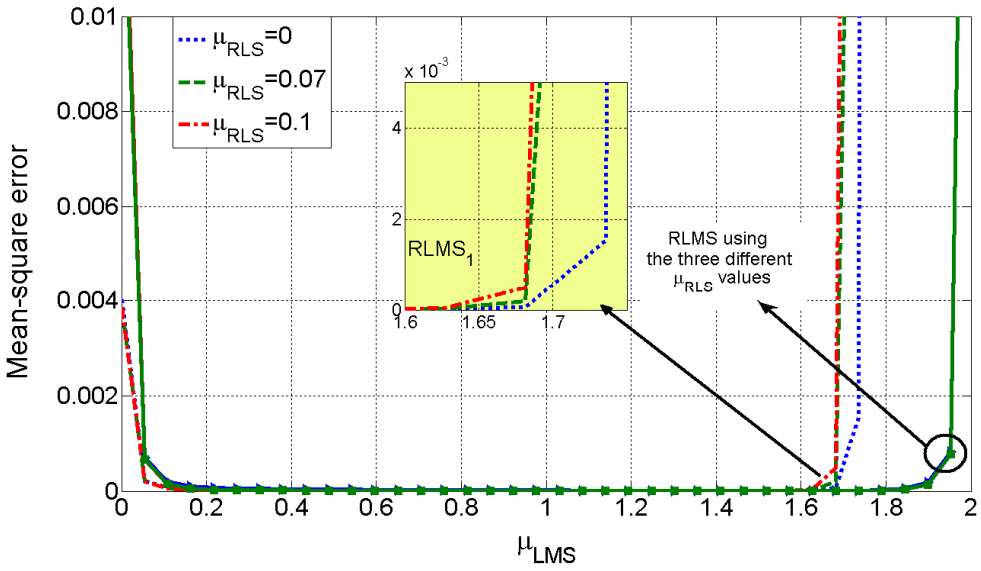


Figure 4-3 Mean square value of the overall error signal as a function of the step size, μ_{LMS}, achieved with the $RLMS_1$ and RLMS algorithms for 3 different μ_{RLS} values (0, 0.07 and 0.1).

As shown in Figure 4-3, the upper bounds of μ_{LMS} for the $RLMS_1$ and RLMS algorithms are approximately 1.7 and 1.8, respectively. These observations have been verified by plotting, in Figure 4.4, the theoretical step size boundaries of μ_{LMS} given in equation (4.72) for $\mu_{RLS} = 0.07$. In this case, the upper limit given in equation (4.72) is plotted against the number of iterations. It is observed that, the upper limit value of μ_{LMS} for both the RLMS and $RLMS_1$ algorithms is approximately 1.73, which is in very close agreement with the maximum value allowed for μ_{LMS}, as indicated in Figure 4-3.

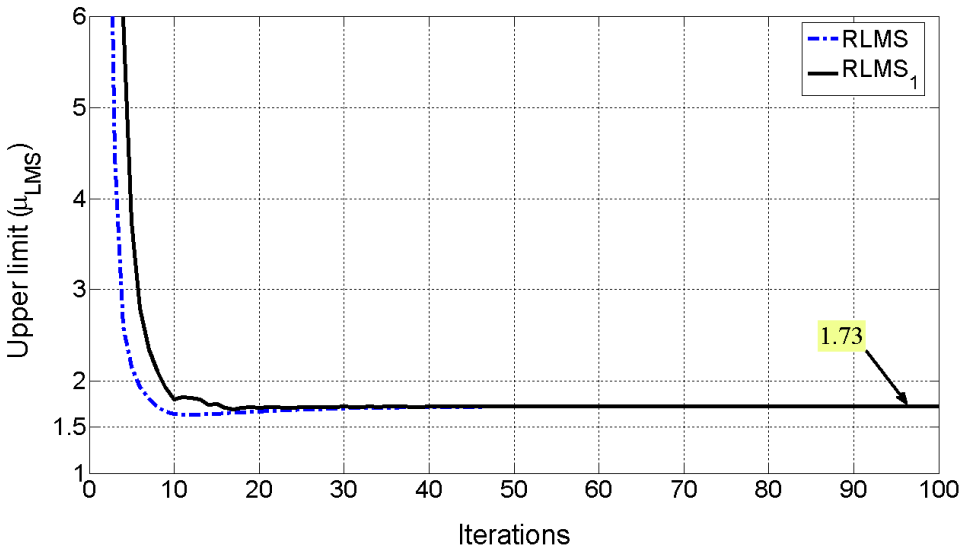


Figure 4-4 The theoretical upper limit of μ_{LMS} used in the RLMS$_1$ and RLMS algorithms versus the number of iterations with μ_{RLS} set at 0.07.

4.7.3.2 Performance with self-referencing

As shown in Figure 4-2, both the RLMS$_1$ and RLMS algorithms are able to converge rapidly in less than ten iterations. This suggests that upon convergence, the output of the RLS algorithm stage, y_{RLS}, will closely resemble the desired input signal $s_d(t)$. As such, this output can be used as the reference signal for the next iteration of the LMS algorithm stage in the RLMS$_1$ and RLMS algorithms. As the LMS algorithm stage converges, its output, y_{RLMS}, becomes the estimated $s_d(t)$, and may be used as the reference for the RLS algorithm stage. This feedforward and feedback arrangement enables the provision of self-referencing in the RLMS$_1$ and RLMS algorithms, and allows the external reference signal to be discontinued after an initial few iterations.

For proper operation with self-referencing, the step size, μ_{RLS}, has to be chosen such that this stage converges quicker than the LMS algorithm stage. This yields a sufficiently accurate output to be used as a reference signal for

the following LMS algorithm stage. For this reason, the values of the step sizes as listed in the second column of Table 4-1 for the $RLMS_1$ and RLMS algorithms have been modified to $\mu_{RLS} = 0.02$ and $\mu_{LMS} = 0.01$ for use in this simulation. The ability of the $RLMS_1$ and RLMS algorithms to maintain operation with the internally generated reference signals is demonstrated in Figure 4-5 for the case with an input SNR of 10 dB. In this experiment, the external reference is switched off after an initial 5 iterations. From there on, both the $RLMS_1$ and RLMS algorithms continue to operate with the internally generated reference signals. On the other hand, the LMS, RLS, VFFRLS, MRVSS and CSLMS algorithms will not converge without the use of the correct reference signal.

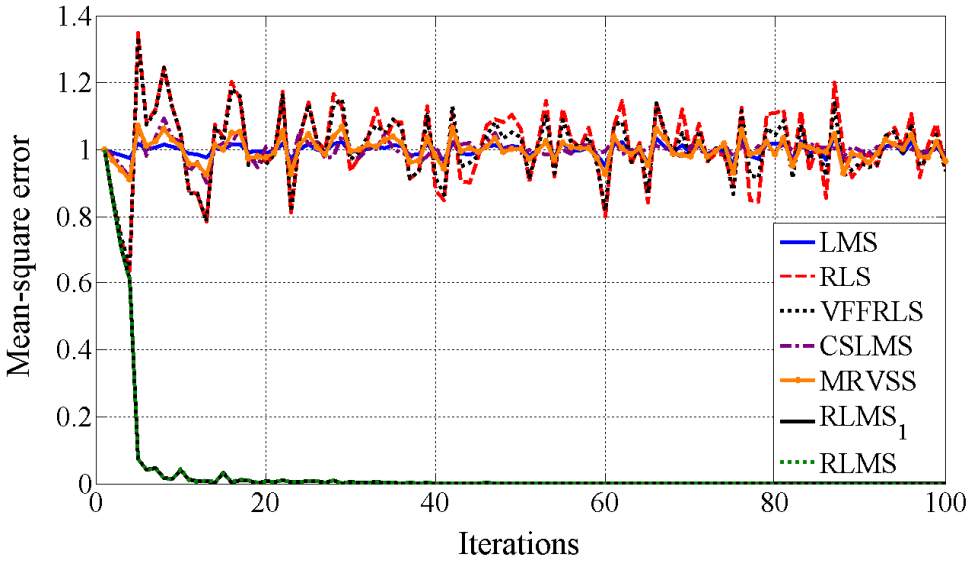


Figure 4-5 The convergence of the $RLMS_1$ and RLMS algorithm with self-referencing at an input SNR of 10 dB. For comparison, the other four algorithms fail to converge when the reference signal is switched off.

4.7.3.3 Performance with a noisy reference signal

The operations of the $RLMS_1$, RLMS, CSLMS, MRVSS, RLS, VFFRLS and LMS algorithms have also been investigated when the reference signal

used is corrupted by additive white Gaussian noise (AWGN). This is done by observing the resultant mean square error, ξ_{RLMS}, of the overall error signal, e_{RLMS}, when the noise level in the reference signal is changed. Figure 4-6 shows the ensemble average of the mean square error, $\overline{\xi}_{\mathrm{RLMS}}$, obtained from 100 individual simulation runs, as a function of the ratio of the rms noise level, σ_n, to the amplitude of the reference signal.

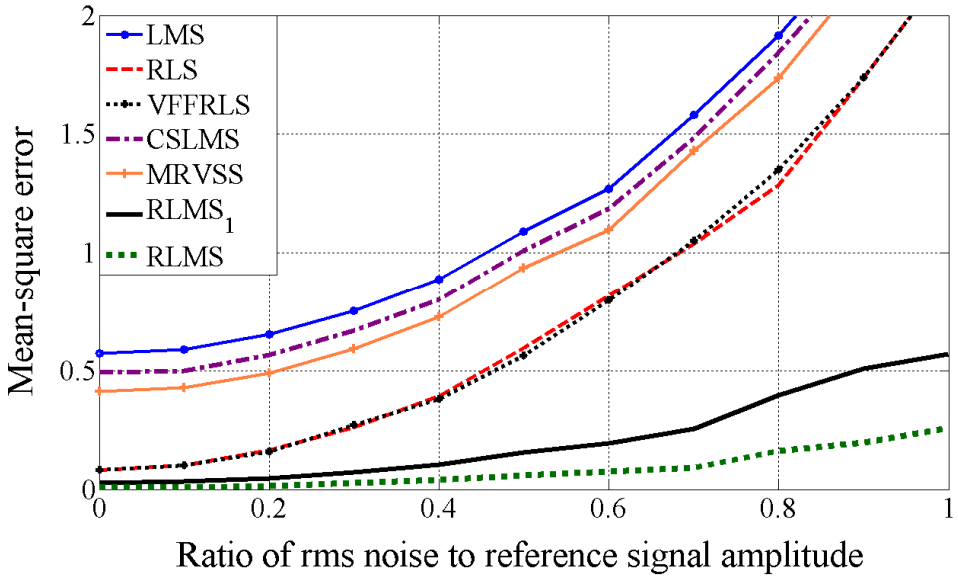


Figure 4-6 The influence of noise in the reference signal on the mean square error, $\overline{\xi}_{\mathrm{RLMS}}$, when operating with the parameters given in the second column of Table 4-1 for an input SNR of 10 dB.

It is interesting to note that the LMS, CSLMS and MRVSS algorithms are quite sensitive to the presence of noise in the reference signal. The RLS and VFFRLS algorithms on their own can still tolerate the presence of low noise level. However, when the RLS and LMS algorithms are incorporated to form the RLMS algorithm, the resulting scheme becomes very tolerant to noisy reference signals. As shown in Figure 4-6, the values of $\overline{\xi}_{\mathrm{RLMS}}$ associated with the RLMS_1 and RLMS algorithms remain very small even when the rms noise level becomes as large as the reference signal.

4.7.3.4 Tracking performance

The ability of the RLMS, $RLMS_1$, CSLMS, MRVSS, RLS, VFFRLS and LMS algorithms to track sudden interruptions in the input signal is investigated by examining the behaviour of their respective error signal. For this investigation, the input signal is interrupted periodically for 25 out of 100 iterations. In the mean time, the weight vector updating process continues to operate without any interruption. Figure 4-7 shows that in the case of the $RLMS_1$ and RLMS algorithms, the mean square error, ξ_{RLMS}, of the overall error, e_{RLMS}, changes very rapidly each time the input signal is switched on or off. This verifies the fast response of these algorithms to sudden changes in the input signal. Unlike the responses for the other five algorithms, as shown in Figure 4-7 for comparison purposes, the values of the mean square error, ξ_{RLMS}, associated with the $RLMS_1$ and RLMS algorithms remain low despite the interruption in the input signal.

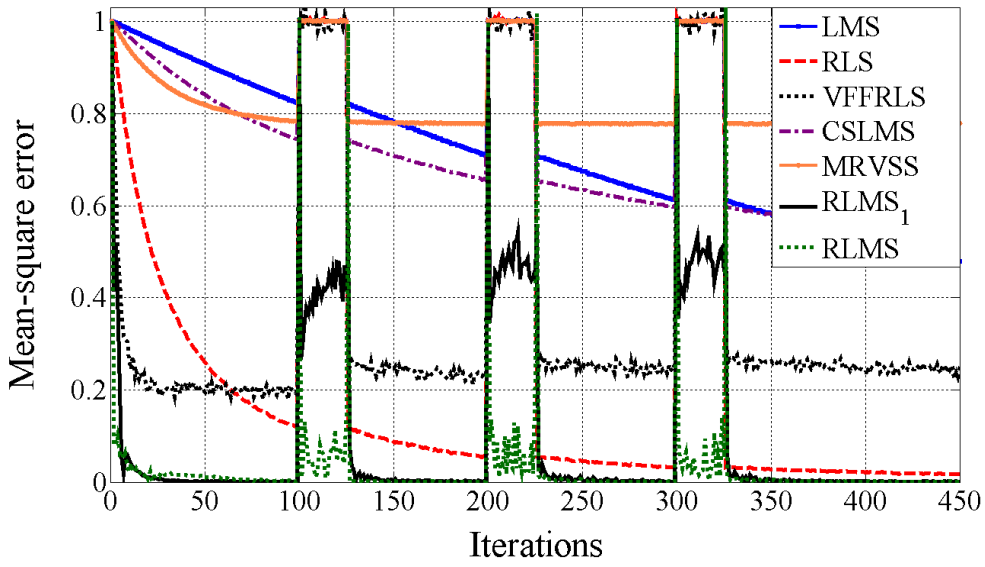


Figure 4-7 Tracking performance of the $RLMS_1$ and RLMS algorithms compared with the RLS, VFFRLS, LMS, CSLMS, MRVSS algorithms implemented using the parameters given in the second column of Table 4-1 for an input SNR of 10 dB.

4.7.3.5 Performance in the presence of multiple interfering signals

Consider a desired signal that arrives at $\theta_d = 10^\circ$ and is corrupted by both cochannel interference and AWGN. In this case, we assume a signal-to-noise power ratio (SNR) of 10 dB, in the presence of four cochannel interfering signals. These four interference signals arrive at angles of -10°, -30°, -50° and 45°, and each has an amplitude equal to that of the desired signal. The resultant beam patterns obtained with the $RLMS_1$ and RLMS algorithms in the presence of these extraneous signals are shown in Figure 4-8. It is observed that the beam patterns obtained with the $RLMS_1$ and RLMS algorithms have almost identical gain at the angle of arrival (AOA) of the desired signal. On the other hand, each of the interfering signals is being suppressed by a minimum of close to 50 dB.

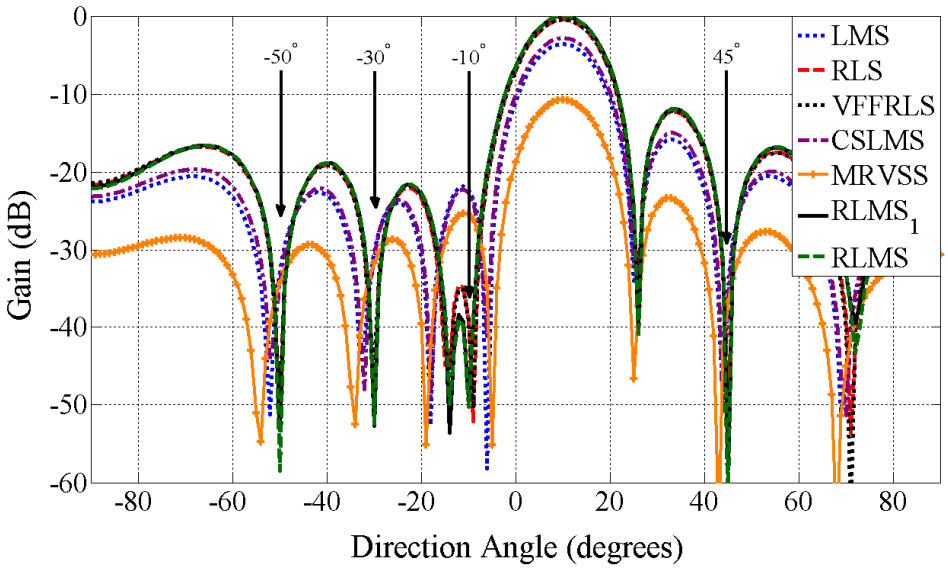


Figure 4-8 The beam patterns obtained with the RLMS, $RLMS_1$, CSLMS, MRVSS, RLS, VFFRLS and LMS algorithms for an SNR of 10 dB in the presence of four equal-amplitude interfering signals arriving at $\theta_{i1} = -50^\circ$, $\theta_{i2} = -30^\circ$, $\theta_{i3} = -10^\circ$ and $\theta_{i4} = 45^\circ$.

For comparison purposes, the beam patterns obtained using the LMS, RLS, VFFRLS, CSLMS and MRVSS algorithms, are also plotted in Figure 4-

8. When compared with the RLMS and $RLMS_1$ algorithms, all these five algorithms suffer from a loss in gain of up to 10 dB at the AOA of the desired signal. Also, they achieve a lower suppression at the four angles of arrival of the interfering signals. Table 4-2 shows the suppression values for all algorithms under test. It shows that the RLMS and $RLMS_1$ algorithms have the largest interference attenuation values among all the algorithms considered. Also, Table 4-2 shows the interference attenuations for the LMS, RLS, VFFRLS, CSLMS and MRVSS algorithms, which are less than those for the RLMS and $RLMS_1$ algorithms. Note that the RLS and VFFRLS algorithms achieved larger interference suppression compared to the LMS, CSLMS and MRVSS algorithms.

Table 4-2 Suppression of the interfering signals with respect to the desired signal expressed in dB.

	AOA of the four interference signals			
Algorithm	**-10^o**	**-30^o**	**-50^o**	**45^o**
LMS	22	30	33	34
RLS	39	49	50	51.5
VFFRLS	39.2	45.9	50.5	57
CSLMS	23	31	34	36
MRVSS	25.5	32	34.5	37
$RLMS_1$	49.3	53	58.5	58.5
RLMS	50.3	52.5	53	64

4.7.3.6 Fixed beamforming

To demonstrate that the $RLMS_1$ algorithm is capable of realizing accurate fixed beamforming, the individual elements of the array vector, $\mathcal{F}_{\mathrm{R}}$, are assigned values pre-calculated using equation (4.2) for the desired direction. In this simulation, we consider an input signal with an SNR of 10 dB is arriving at an angle θ_d corresponding to -20^o, 0^o, 20^o, 40^o or 60^o. The

resultant fixed beam patterns are plotted in Figure 4-9, which shows that the main lobe in each case is accurately located at the specified desired direction. Except for the case of θ_d of 60^o, which suffers from the grating lobe problem[6], the beam patterns are almost identical, with a worst case side lobe suppression of -13 dB. Although not shown, the same observation can be made when a desired signal is arriving from an angle of 20^o, 0^o, -20^o, -40^o or -60^o.

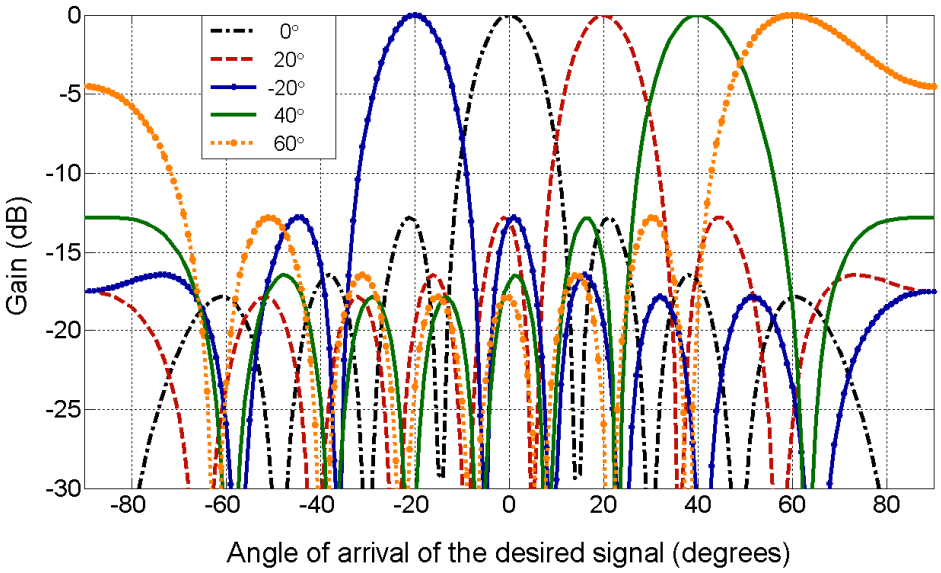


Figure 4-9 The beam patterns achieved with the RLMS$_1$ algorithm for five relatively large angles of θ_d (-20^o, 0^o, 20^o, 40^o and 60^o) at an input SNR=10 dB using the parameters given in the second column of Table 4-1.

Furthermore, the beam resolution that could be achieved with this fixed beamforming scheme is also investigated. The resulting beam patterns achieved for the desired direction set at either -2^o, 0^o, 1^o or 5^o are shown in Figure 4-10. These results indicate that it is possible to differentiate very small differences in beam direction. The same beam resolution can also be achieved when the desired signal arrives from a larger angle of around 30^o, as shown in Figure 4-11.

[6] The grating lobe problem is discussed in Section 2.3

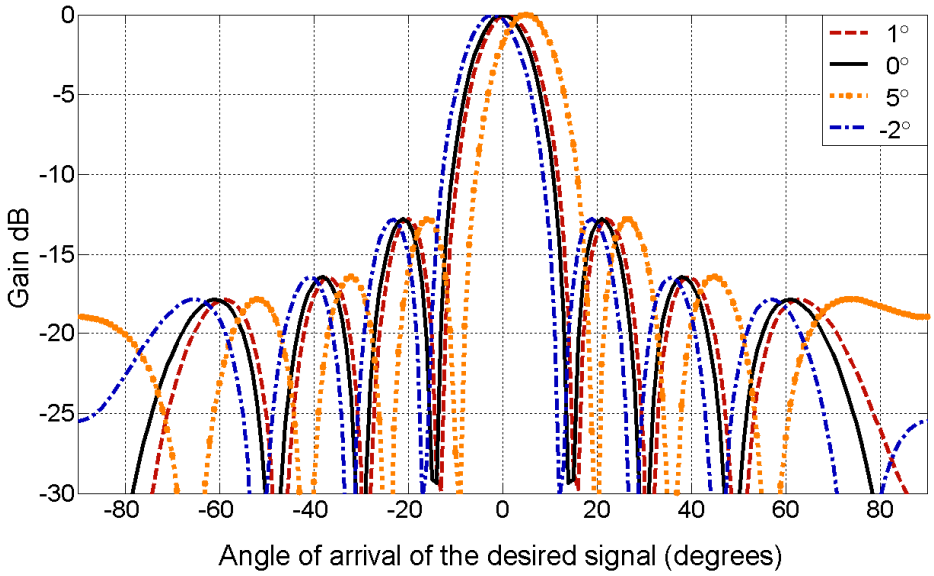


Figure 4-10 The beam patterns achieved with the RLMS$_1$ algorithm for four small angles of θ_d (-2^o, 0^o, 1^o and 5^o) at an input SNR of 10 dB using the parameters given in the second column of Table 4-1.

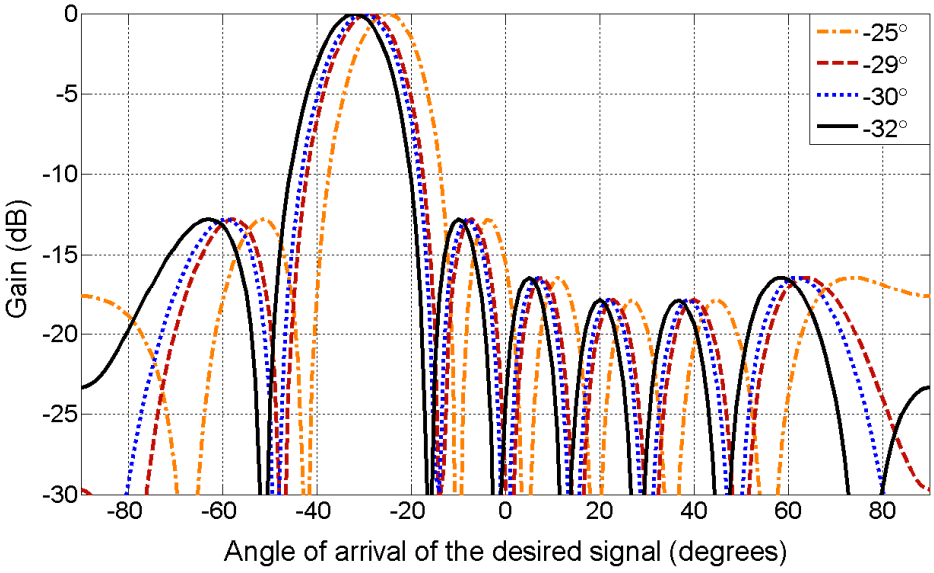


Figure 4-11 The beam patterns achieved with the RLMS$_1$ algorithm for four closely spaced angles of θ_d (-32^o, -30^o, -29^o and -25^o) at an input SNR of 10 dB using the parameters given in the second column of Table 4-1.

4.7.3.7 EVM and scatter plot

The performances of the seven algorithms, namely RLMS, $RLMS_1$, CSLMS, MRVSS, RLS, VFFRLS and LMS, based on the root mean square (rms) value of the error vector magnitude (EVM) computed using equation (4.86), for values of input SNR ranging from 0–30 dB in steps of 5 dB are shown in Figure 4-12. These EVM values have been calculated after each adaptive algorithm has converged. It is observed that the proposed RLMS algorithm achieves the lowest EVM values with those obtained from the $RLMS_1$ algorithm being slightly larger. This suggests that the RLMS algorithm may be able to better readjust itself to the operating environment due to the use of an adaptive array image vector. On the other hand, all the other five algorithms suffer from much larger EVM values, particularly for SNR smaller than 10 dB. This further confirms the observation made from Figure 4-2 showing that the operations of the $RLMS_1$ and RLMS algorithms are very insensitive to changes in input SNR.

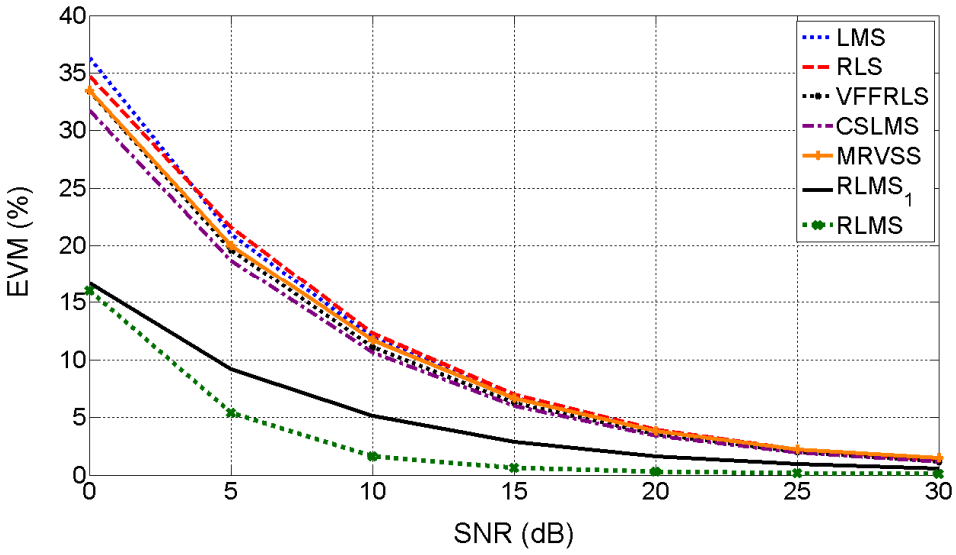


Figure 4-12 The EVM values obtained with the RLMS, $RLMS_1$, CSLMS, MRVSS, RLS, VFFRLS and LMS algorithms at different values of input SNR.

Next, the scatter plots of the BPSK signal recovered using the adaptive beamformer, based on the $RLMS_1$, RLMS, CSLMS, MRVSS, RLS, VFFRLS

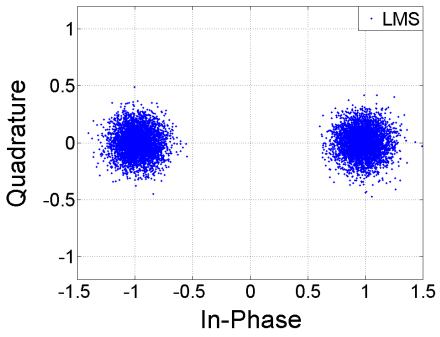
and LMS algorithms are shown in Figures 4-13(a)–(f), respectively. The scatter plots, obtained from 8192 signal samples, are based on the algorithm operating at an input SNR of 10dB and signal to interference ratio (SIR) of -6dB. Again, the scatter plots of the $RLMS_1$ and RLMS algorithms show the least spreading among the seven algorithms, indicating their ability to retain the signal fidelity.

4.7.3.8 Operation in a flat Rayleigh fading channel

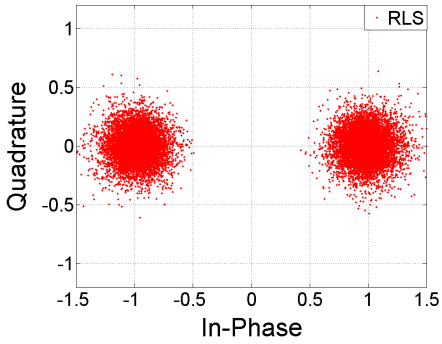
The ability of an adaptive beamformer to operate in a fast changing signal environment is examined by subjecting the input signal to undergo flat Rayleigh fading. In this case, the rms EVM is again used as the performance metric for comparison between the different adaptive beamforming algorithms. The following conditions are considered in the performance evaluation:

- The signals arriving at each antenna element, for both the desired and interfering signals, undergo independent flat Rayleigh fading. A typical Rayleigh flat fading envelope observed on the first antenna element is shown in Figure 4-14.
- Two interfering signals, each with the same amplitude as the desired signal, are emanating from -30^o and 45^o.
- The parameters as tabulated in column 3 of Table 4-1 are adopted for the different algorithms. Note that the parameter values have been adjusted somewhat to suit the new channel environments when compared with the values used in the case of AWGN.
- Each simulation involves a run of 16 Mbits.
- A Doppler frequency f_d of 60 Hz, corresponding to a mobility of 72 km/h at 900 MHz, is used in the simulation.

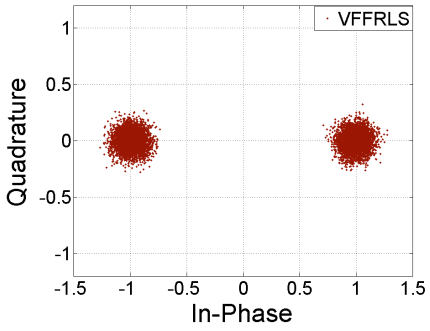
The resultant EVM values achieved at different input SNR for the cases of with and without co-channel interference are plotted in Figure 4-15a and Figure 4-15b, respectively. From these figures, the following observations are made:



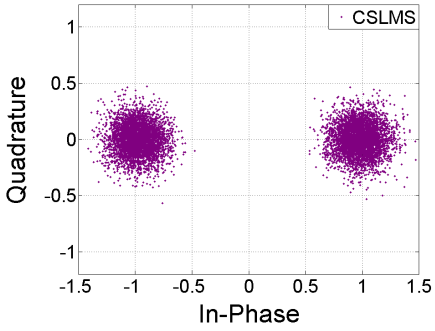
(a) LMS algorithm



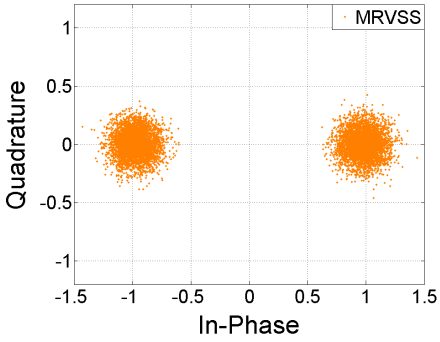
(b) RLS algorithm



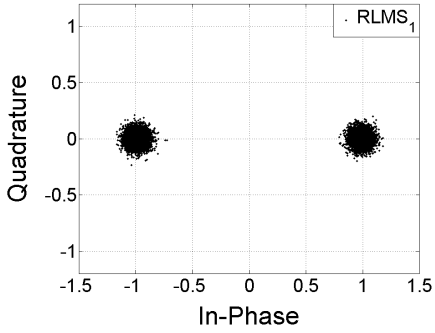
(c) VFFRLS algorithm



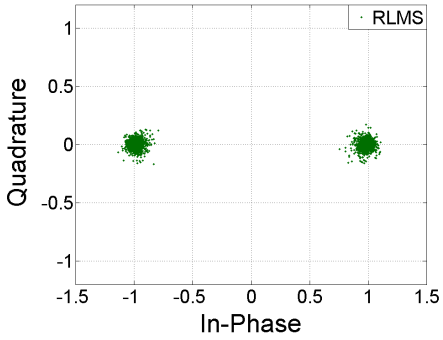
(d) CSLMS algorithm



(e) MRVSS algorithm



(f) $RLMS_1$ algorithm



(f) RLMS algorithm

Figure 4-13 The scatter plots of the recovered BPSK signal obtained with (a) LMS, (b) RLS, (c) VFFRLS, (d) CSLMS, (e) MRVSS, (f) $RLMS_1$, and (g) RLMS algorithms for input SNR=10 dB and SIR= -6 dB.

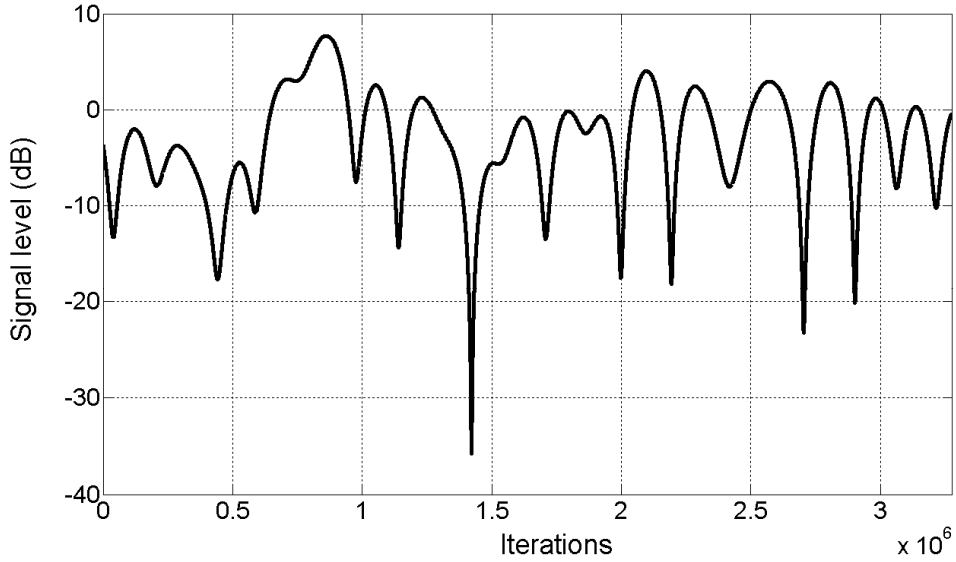
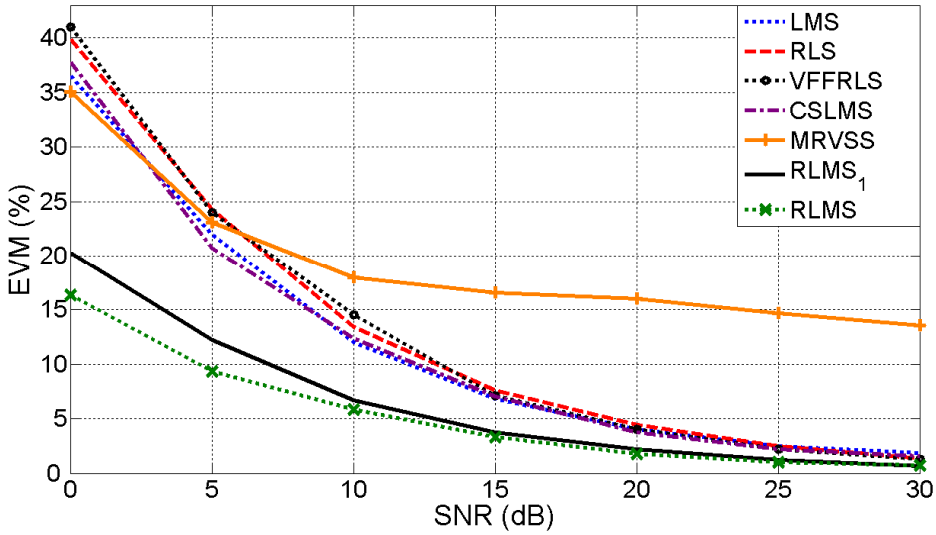
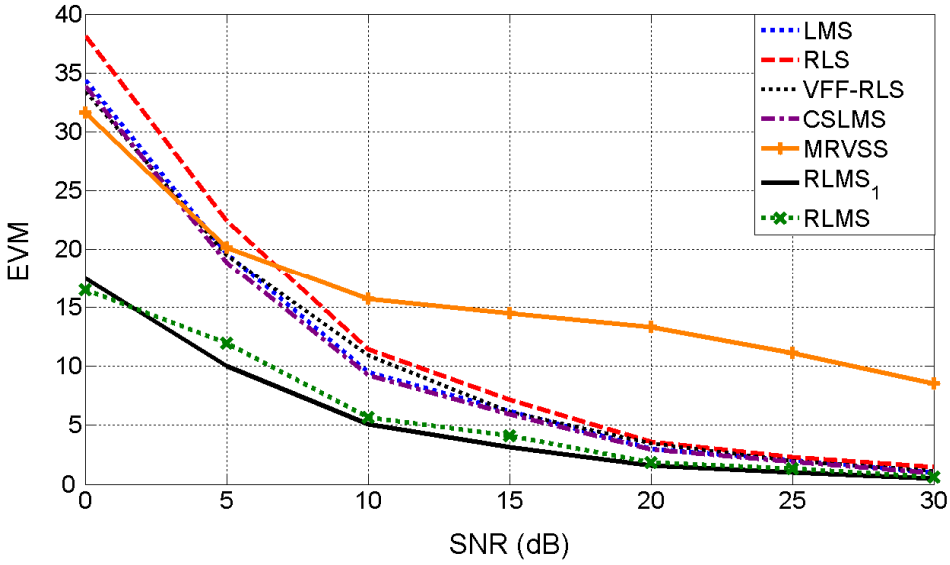


Figure 4-14 The Rayleigh flat fading envelope observed at the first antenna element of the array.

- With the exception of the MRVSS algorithm, all the other algorithms considered seem to be able to operate in the presence of Rayleigh fading. As the EVM values are calculated after each algorithm has achieved convergence, the interfering signals emanating from the unwanted directions would have been suppressed. Consequently, the resultant EVM values, as shown in Figure 4-15a and Figure 4-15b, are similar for the case with and without the interfering signals, respectively.
- Irrespective of whether interfering signals are present or not, the RLMS and $RLMS_1$ algorithms outperform the other four algorithms.
- The RLMS algorithm is the least affected by interference.



(a) In the presence of fading and co-channel interference



(b) In the presence of fading but without interference

Figure 4-15 The EVM values obtained with the RLMS, $RLMS_1$, CSLMS, MRVSS, RLS, VFFRLS and LMS algorithms for different values of input SNR in the presence of Rayleigh fading: (a) signal-to-interference ratio (SIR) of -6 dB, and (b) without co-channel interference.

4.8 Summary

A new array beamforming algorithm, called the RLMS algorithm, is presented in this chapter. It incorporates an RLS algorithm stage connected in series with an LMS algorithm stage via an array image vector, $\mathcal{F}_{\mathrm{R}}$. This algorithm adopts a different approach compared with earlier algorithms, such as the CSLMS and MRVSS algorithms, which make use of step size adaptation to enhance their performance. For proper operation, those LMS algorithms, modified to make use of variable step size to enhance convergence speed, often require many input signal dependent parameters to be specified. As noted in [139], it is difficult in practice to obtain the exact values simultaneously for all these parameters. On the other hand, modifications introduced to improve the tracking ability of an RLS algorithm tend to increase significantly the computation complexity. Examples of these modified RLS algorithms are the AFF-RLS [89], VFFRLS [12] and EX-KRLS algorithms [13].

As discussed in Section 4.2, with the proposed RLMS algorithm, it is possible to make use of the array image vector to provide a flexible means of achieving either fixed or adaptive array beamforming. The former mode of operation is referred to, in this chapter, as the RLMS_1 algorithm, and it can provide an accurate fixed beam by prior setting the elements of $\mathcal{F}_{\mathrm{R}}$ with the prescribed values for the required direction. Alternatively, $\mathcal{F}_{\mathrm{R}}$ may be made adaptive to automatically track the target signal. This adaptive version of the algorithm is described simply as the RLMS algorithm. A simple and effective method has been proposed in Section 4.5 for calculating the element values of $\mathcal{F}_{\mathrm{R}}$ adaptively. This involves the use of the signal at the output of the RLS algorithm stage in conjunction with its tap weights.

The convergence of the RLMS algorithm, operating in either the fixed or adaptive mode, has been analyzed assuming the use of an external reference signal. The analysis is then extended to cover the case that makes use of self-referencing. The boundary values for the step sizes, μ_{RLS} and μ_{LMS}, used in the respective RLS algorithm and LMS algorithm stages have

been derived analytically. It is shown that a stable operation of the RLMS algorithm can be achieved using a broad range of values for μ_{RLS} and μ_{LMS}.

As discussed in Section 4.8.1, both the RLMS_1 and RLMS algorithms are shown to have rapid convergence, typically within a few iterations, as well as good signal tracking ability. Also, the resulting steady state MSE is quite insensitive to changes in input SNR. Furthermore, unlike the conventional LMS, RLS, VFFRLS, CSLMS and MRVSS algorithms, the proposed RLMS_1 and RLMS algorithms are able to operate with noisy reference signals. Once initial convergence is achieved, usually within a few iterations, both the RLMS_1 and RLMS algorithms can maintain their operation through self-referencing. Moreover, the resultant EVM values and scatter plots, obtained for operation in an AWGN channel or fast changing Rayleigh fading environment, further demonstrate the superior performance of the RLMS_1 and RLMS algorithms over the other five published algorithms considered in this chapter.

It is to be noted that the superior performance of the proposed RLMS algorithm is achieved with a complexity only slightly larger than the conventional RLS algorithm scheme, i.e., equivalent to $2.5N^2+5N+1$ complex multiplications for an N-element array. Moreover, its complexity is significantly lower than some of the RLS based algorithms, such as VFFRLS, AFF-RLS and EX-KRLS algorithms, which have been proposed for improving the tracking performance of the RLS algorithm.

CHAPTER 5

ADAPTIVE ARRAY BEAMFORMING USING A COMBINED LMS-LMS ALGORITHM

5.1 Introduction

It is well known that the LMS algorithm is simple to implement and robust in operation. For this reason, the LMS algorithm has become one of the most popular adaptive signal processing techniques adopted for many applications including antenna array beamforming. Moreover, as reviewed in Section 3.2.2, there is always a tradeoff between the speed of convergence and achievable residual error floor when a given adaptation step size is used with an LMS algorithm. This observation has led to several improvements being proposed over the last three decades to speed up the convergence of an LMS algorithm. Some of these modified LMS algorithms, such as the constrained-stability LMS (CSLMS) algorithm [9] and modified robust variable step size LMS (MRVSS) algorithm [10] have been reviewed in Chapter 3.

In Chapter 4, a new approach to adaptive array beamforming using a combined RLS-LMS algorithm has been proposed. The fast convergence and robust operation of this new RLMS algorithm have been verified through a detailed analytical study and extensive computer simulations, as described in Chapter 4. However, the complexity of the RLS algorithm, which is used in the first stage of the RLMS algorithm, remains quite high when compared with a conventional LMS algorithm. This observation provides an incentive to search for a simpler replacement for the RLS algorithm while still being able to maintain the superior performance of the RLMS algorithm.

In this chapter, it is proposed that an LMS algorithm stage is used to replace the RLS algorithm stage in the RLMS algorithm. The resultant scheme is referred to as the LLMS algorithm, which maintains the low

complexity generally associated with an LMS algorithm. It can be shown that an *N*-element antenna array employing the LLMS algorithm involves 4*N*+1 complex multiplications and 3*N* complex additions, i.e., slightly doubling the computational requirements of a conventional LMS algorithm scheme.

In the following section, we will analyze and study the performance of the proposed LLMS algorithm.

5.2 LLMS Algorithm

With the proposed LLMS algorithm, as shown in Figure 5-1, the intermediate output, y_{LMS1}, yielded from the first LMS algorithm or LMS_1 stage, is multiplied by the array image vector, $\boldsymbol{\mathcal{F}}_{\mathrm{L}}$**,** of the desired signal. The resultant "filtered" signal is further processed by the second LMS algorithm or LMS_2 stage. For the adaptation process, the error signal of the LMS_2 algorithm stage, e_2, is fed back to combine with that of the LMS_1 algorithm stage, to form the overall error signal, e_{LLMS}, for updating the tap weights of the LMS_1 algorithm stage. As shown in Figure 5.1, a common external reference signal $d(n)$ is used for both the LMS_1 and LMS_2 algorithm sections, i.e., d_1 and d_2. Moreover, this external reference signal may be replaced after a few initial iterations by y_{LMS1} in place of d_2, and y_{LLMS} for d_1 to produce a self-referenced version of the LLMS algorithm scheme. This will be discussed in Section 5.3.2.

As in the case of the RLMS algorithm, described in Chapter 4, it is possible for the array image vector, $\boldsymbol{\mathcal{F}}_{\mathrm{L}}$**,** used in the LLMS algorithm to be made adaptive in order to follow the angle of arrival (AOA) of the wanted signal. This adaptive $\boldsymbol{\mathcal{F}}_{\mathrm{L}}$ version will from here on be simply known as the LLMS algorithm, in order to differentiate it from the scheme that makes use of a prescribed $\boldsymbol{\mathcal{F}}_{\mathrm{L}}$**.** The latter will be referred to as the LLMS_1 algorithm.

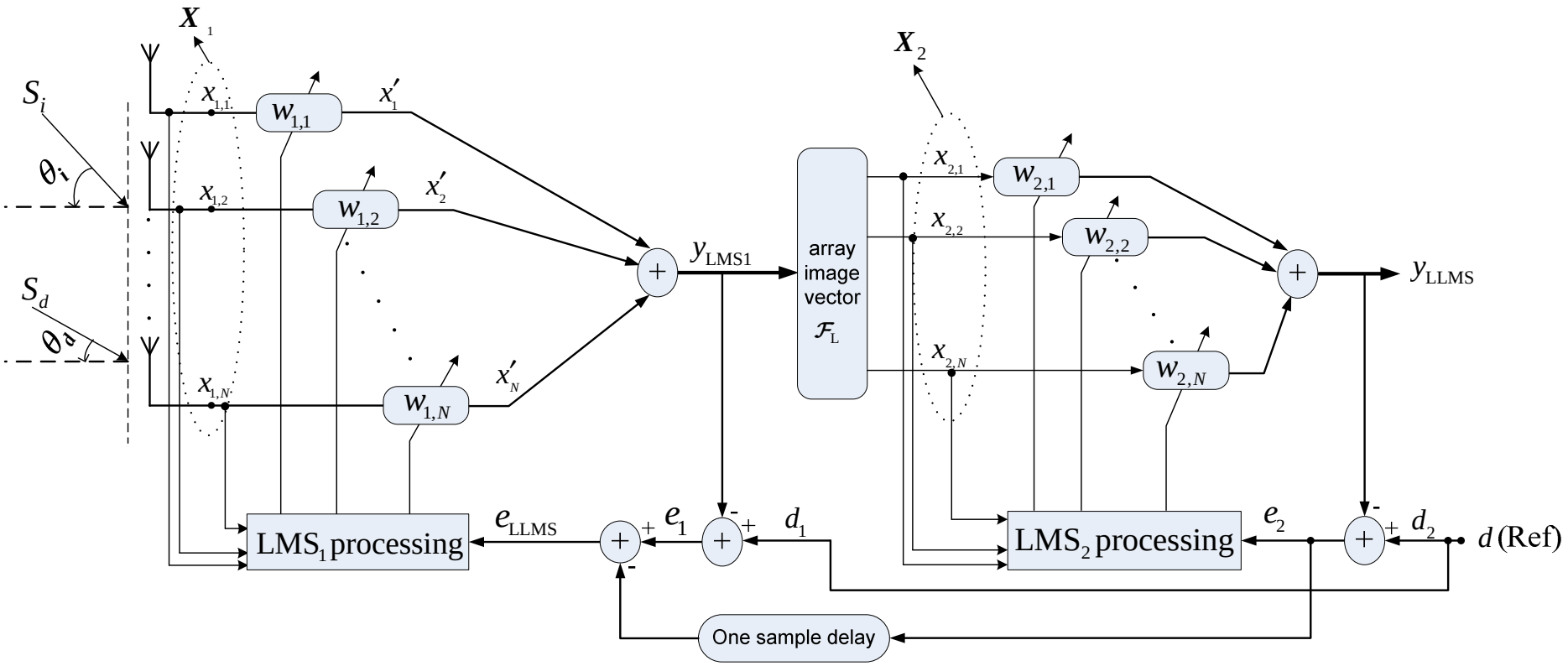


Figure 5-1 The proposed LLMS algorithm with an external reference signal.

A detailed analysis of the proposed LLMS algorithm, operating with either an external reference or self reference, will be presented in the next section. The analysis follows the same approach as described in Section 4.3 for the RLMS algorithm. Again, the boundary values of the step sizes used in the LMS_1 and LMS_2 algorithm stages are derived in Section 5.3.4.2.

5.3 Convergence of the Proposed LLMS Algorithm

5.3.1 Analysis for operation with an external reference

The convergence of the proposed $LLMS_1$ algorithm, which employs a prescribed array image vector, $\mathcal{F}_{\mathrm{L}}$, is analyzed with the same assumptions that are used in Section 4.3.1

First, we consider the case when an external reference signal is used. From Figure 5-1, the overall error signal for updating the LLMS algorithm at the n^{th} iteration is given by

$$e_{\mathrm{LLMS}}(n) = e_1(n) - e_2(n-1) \tag{5.1}$$

with the individual error signals

$$e_i(n) = d_i(n) - \boldsymbol{W}_i^H(n)\boldsymbol{X}_i(n) \tag{5.2}$$

where the subscript *i* takes on the value of 1 and 2 for the LMS_1 and the LMS_2 algorithm stages respectively; $\boldsymbol{X}_i(\cdot)$ and $\boldsymbol{W}_i(\cdot)$ represent the input signal and weight vectors respectively, and $(\bullet)^H$ denotes the Hermitian matrix of $(\bullet)$.

The input signal of the LMS_2 algorithm stage is derived from the LMS_1 algorithm, such that

$$\boldsymbol{X}_2(n) = \mathcal{F}_{\mathrm{L}}\, y_{\mathrm{LMS1}}(n) = \mathcal{F}_{\mathrm{L}}\, \boldsymbol{W}_1^H(n)\boldsymbol{X}_1(n) \tag{5.3}$$

where $\mathcal{F}_{\mathrm{L}}$ is the image of the array vector of the desired signal as defined in equation (4.9) and is assumed fixed for this analysis. The weight vector $\boldsymbol{W}_i(\cdot)$ for the i^{th} LMS algorithm section is updated according to [34],

$$\boldsymbol{W}_i(n+1) = \boldsymbol{W}_i(n) + \mu_i e_i(n)\boldsymbol{X}_i(n) \qquad 0 < \mu_i < \mu_l \tag{5.4}$$

where μ_i is the step size with the subscript *i* as defined in equation (5.2), and μ_l is a positive value that is inversely proportional to the input signal power.

Convergence performance of mean-square error, ξ_{LLMS}, for the LLMS algorithm can be analyzed in terms of the expected value of e^2_{LLMS}, such that

$$\begin{aligned}\xi_{\mathrm{LLMS}} &\triangleq \mathrm{E}\left[\left|e_{\mathrm{LLMS}}(n)\right|^2\right] = \mathrm{E}\left[\left|e_1(n) - e_2(n-1)\right|^2\right] \\ &= \mathrm{E}\left[\left|D_{\mathrm{L}}(n)\right|^2\right] + \boldsymbol{W}_1^{H}(n)\boldsymbol{Q}_1(n)\boldsymbol{W}_1(n) \\ &\quad - \mathrm{E}\left[D_{\mathrm{L}}(n)\boldsymbol{X}_1^{H}(n)\boldsymbol{W}_1(n) + D_{\mathrm{L}}^{*}(n)\boldsymbol{W}_1^{H}(n)\boldsymbol{X}_1(n)\right]\end{aligned} \tag{5.5}$$

where $\mathrm{E}[\cdot]$ denotes expectation; $|\bullet|$ signifies modulus; * stands for conjugate operator;

$$D_{\mathrm{L}}(n) = d_1(n) - e_2(n-1) \tag{5.6}$$

and $\boldsymbol{Q}_1$ is the correlation matrix of the input signals given by

$$\boldsymbol{Q}_1 = \mathrm{E}\left[\boldsymbol{X}_1(n)\boldsymbol{X}_1^{H}(n)\right] \tag{5.7}$$

From Appendix D, the final form of equation (5.5) is derived as

$$\begin{aligned}\xi_{\mathrm{LLMS}} &= \mathrm{E}\left[\left|d_1(n)\right|^2\right] + \mathrm{E}\left[\left|d_2(n-1)\right|^2\right] \\ &\quad + \boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Q}_1(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1) - \boldsymbol{Z}_{\mathrm{L}}^{H}(n)\boldsymbol{W}_1(n) \\ &\quad - \boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{L}}(n-1) - \boldsymbol{Z}_{\mathrm{L}}^{H}(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1) \\ &\quad - \boldsymbol{W}_1^{H}(n)\boldsymbol{Z}_{\mathrm{L}}(n) + \boldsymbol{W}_1^{H}(n)\boldsymbol{Q}_1(n)\boldsymbol{W}_1(n)\end{aligned} \tag{5.8}$$

It follows from Appendix D that the minimum mean square error (MSE) of equation (5.5) becomes

$$\begin{aligned}\xi_{\text{LLMS,min}} &= \mathrm{E}\left[\left|d_1(n)\right|^2\right] + \mathrm{E}\left[\left|d_2(n-1)\right|^2\right] \\ &- \boldsymbol{Z}_{\mathrm{L}}^{H}(n)\boldsymbol{W}_{opt1}(n) - \boldsymbol{Z}_{\mathrm{L}}^{H}(n-1)\boldsymbol{W}_{\text{LLMS}}(n-1) \\ &+ \boldsymbol{W}_{\text{LLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{L}}(n-1)\left\{\boldsymbol{\mathcal{F}}_{\mathrm{L}}^{H}\boldsymbol{W}_2(n-1) - 1\right\}\end{aligned} \tag{5.9}$$

where $\boldsymbol{Z}_{\mathrm{L}}$ corresponds to the input signal cross-correlation vector given by

$$\boldsymbol{Z}_{\mathrm{L}} = \mathrm{E}\left[\boldsymbol{X}_1(n)d_2^{*}(n)\right] \tag{5.10}$$

Based on the same analysis carried in equations 4.31 to 4.39, the weight vector is updated according to

$$\boldsymbol{W}_1(n+1) = \boldsymbol{W}_1(n) - \mu_1\boldsymbol{\nabla}\xi_{\text{LLMS}}(n) \tag{5.11}$$

where μ_1 is the convergence constant that controls the stability and the rate of adaptation of the weight vector, and $\boldsymbol{\nabla}\xi_{\text{LLMS}}(n)$ is the gradient at the n^{th} iteration.

We may rewrite equation (5.11) in the form of a linear homogeneous vector difference equation following the same procedure given in equations equation (4.42) to equation (4.45), to yield

$$\boldsymbol{V}_1(n+1) = \boldsymbol{V}_1(n) - \mu_1\boldsymbol{Q}_1\boldsymbol{V}_1(n) \tag{5.12}$$

Alternatively, equation (5.12) can be written as

$$\begin{aligned}\boldsymbol{V}_1(n) &= \left(\boldsymbol{q}_1\boldsymbol{q}_1^{H} - \mu_1\boldsymbol{q}_1\boldsymbol{\Lambda}_1\boldsymbol{q}_1^{H}\right)\boldsymbol{V}_1(n-1) \\ &= \boldsymbol{q}_1\left(\boldsymbol{I} - \mu_1\boldsymbol{\Lambda}_1\right)^{n}\boldsymbol{q}_1^{H}\boldsymbol{V}_1(0)\end{aligned} \tag{5.13}$$

By using similar steps in (4.49), the MSE at the n^{th} iteration is given by

$$\xi_{\mathrm{LLMS}} = \xi_{\mathrm{LLMS,min}} + \boldsymbol{V}_1^H(0)\boldsymbol{q}_1\left(\left(\boldsymbol{I}-\mu_1\boldsymbol{\Lambda}_1\right)^n\right)^H \boldsymbol{q}_1^H\boldsymbol{q}_1\boldsymbol{\Lambda}_1\boldsymbol{q}_1^H\boldsymbol{q}_1\left(\boldsymbol{I}-\mu_1\boldsymbol{\Lambda}_1\right)^n\boldsymbol{q}_1^H\boldsymbol{V}_1(0) \tag{5.14}$$

Rearranging equation (5.14) yields

$$\xi_{\mathrm{LLMS}} = \xi_{\mathrm{LLMS,min}} + \boldsymbol{V}_1^H(0)\boldsymbol{q}_1\left(\left(\boldsymbol{I}-\mu_1\boldsymbol{\Lambda}_1\right)^n\right)^H \boldsymbol{\Lambda}_1\left(\boldsymbol{I}-\mu_1\boldsymbol{\Lambda}_1\right)^n\boldsymbol{q}_1^H\boldsymbol{V}_1(0) \tag{5.15}$$

From equation (5.15), the asymptotic value of ξ_{LLMS} becomes zero since $\lim_{n\to\infty}\left(\boldsymbol{I}-\mu_1\boldsymbol{\Lambda}_1\right)^n = 0$. With the term $\left(\boldsymbol{I}-\mu_1\boldsymbol{\Lambda}_1\right)$ converging, (as will be discussed in Section 5.3.3), the mean square error will finally approach its minimum value, such that

$$\lim_{n\to\infty}\xi_{\mathrm{LLMS}} = \xi_{\mathrm{LLMS,min}} \tag{5.16}$$

5.3.2 Analysis of the self-referencing scheme

After the convergence of the LLMS algorithm, usually within few iterations, the external reference of the LMS_1 algorithm stage can be replaced by the internally generated output signal, y_{LLMS}, and the reference signal for the LMS_2 algorithm stage may be replaced by the output y_{LMS1}, so that

$$d_1(n) = y_{\mathrm{LLMS}}(n-1) \tag{5.17}$$

and

$$d_2(n) = y_{\mathrm{LMS1}}(n) \tag{5.18}$$

This mode of operation of the LLMS algorithm is referred to as self-referencing. With these changes and observing that

$$e_2(n) = d_2(n) - y_{\mathrm{LLMS}}(n) \tag{5.19}$$

then we can redefine $D_{\mathrm{L}}(n)$ in equation (5.6) as

$$D_{\mathrm{L}}(n) = 2y_{\mathrm{LLMS}}(n-1) - y_{\mathrm{LMS1}}(n-1) \tag{5.20}$$

Based on the definition of equation (5.20), and following the same procedure steps in Appendix D, we reanalyze the MSE as defined in equation (5.5) to yield

$$\begin{aligned} \xi_{\mathrm{LLMS}} = \mathrm{E}\left[\left|D_{\mathrm{L}}(n)\right|^2\right] - \boldsymbol{Z}_{\mathrm{L}}'^{H}\boldsymbol{W}_1(n) \\ -\boldsymbol{W}_1^{H}(n)\boldsymbol{Z}_{\mathrm{L}}' + \boldsymbol{W}_1^{H}(n)\boldsymbol{Q}_1\boldsymbol{W}_1(n), \end{aligned} \tag{5.21}$$

where $\boldsymbol{Z}_{\mathrm{L}}'(n)$ corresponds to the input signal cross-correlation vector given by

$$\boldsymbol{Z}_{\mathrm{L}}' = \mathrm{E}\left[D_{\mathrm{L}}^{*}(n)\boldsymbol{X}_1(n)\right] \tag{5.22}$$

The error values obtained from equation (5.21) are plotted as the theoretical curve in Figure 5-4.

The minimum MSE ($\xi_{\mathrm{LLMS,min}}$) of equation (5.21) can be obtained following the same analyzing steps in Appendix C to obtain a similar formula to that given in equation (5.9). This is followed by finding the weight error vector similar to that in equation (5.12). Then using this weight error, the convergence of MSE in equation (5.21) to that obtained in equation (5.16) can be verified. Therefore, based on these steps, it can be shown that the proposed LLMS algorithm will converge under the condition of self-referencing.

5.3.3 Mean weight vector convergence

This section derives the values of the step sizes, μ_1 and μ_2 required to ensure stable operation of the LLMS algorithm. To simplify the analysis, we use the concept that once the two individual LMS algorithm sections that make up the LLMS algorithm are converging, the LLMS algorithm as a whole is also converging. This enables the range of allowed step size values to be

separately determined for the LMS_1 and LMS_2 algorithms. Those values that overlap these two ranges of step sizes are then considered valid for use in the LLMS algorithm to ensure its convergence.

In equation (5.4), we define the error e_i as [135, 140]

$$e_i(n) = e_i''(n) - \boldsymbol{X}_i^H(n)\boldsymbol{V}_i(n), \tag{5.23}$$

where $e_i''(n)$ is a zero mean measurement noise and $\boldsymbol{V}_i$ is the weight vector error. Then, let the time-varying weight vector be modeled by a random walk process [68], such that

$$\boldsymbol{W}_{0i}(n+1) = \boldsymbol{W}_{0i}(n) + \boldsymbol{r}_i(n), \tag{5.24}$$

where $\boldsymbol{W}_{0i}$ is the optimal weight vector of the i^{th} LMS algorithm section, $\boldsymbol{r}(n)$ is a zero mean white sequence vector with diagonal correlation matrix $\sigma_{\mathrm{L},r}^2\boldsymbol{I}$, and $\sigma_{\mathrm{L},r}^2$ is the weight variance. Also, let the weight error vector be

$$\boldsymbol{V}_i(n) = \boldsymbol{W}_i(n) - \boldsymbol{W}_{0i}(n) \tag{5.25}$$

From equations (5.4), (5.24) and (5.25), we obtain

$$\boldsymbol{V}_i(n+1) = \boldsymbol{V}_i(n) + \mu_i e_i(n)\boldsymbol{X}_i(n) - \boldsymbol{r}_i(n) \tag{5.26}$$

Substituting equation (5.23) into equation (5.26) and multiplying both sides of equation (5.26) by $\boldsymbol{q}_i^H$, which defines the eigenvector matrix for the i^{th} LMS algorithm stage, gives

$$\begin{aligned}\boldsymbol{v}_i(n+1) &= \boldsymbol{v}_i(n) + \mu_i e_i \boldsymbol{X}_i(n) - \tilde{\boldsymbol{r}}_i(n) \\ &= [\boldsymbol{I} - \mu_i \tilde{\boldsymbol{X}}_i(n)\tilde{\boldsymbol{X}}_i^H(n)]\boldsymbol{v}_i(n) + \mu_i e_i''\tilde{\boldsymbol{X}}_i(n) - \tilde{\boldsymbol{r}}_i(n)\end{aligned} \tag{5.27}$$

where

$$\begin{aligned}\tilde{\boldsymbol{r}}_i(n) &= \boldsymbol{q}_i^H \boldsymbol{r}_i(n), \\ \tilde{\boldsymbol{X}}_i(n) &= \boldsymbol{q}_i^H \boldsymbol{X}_i(n) \\ \boldsymbol{v}_i(n) &= \boldsymbol{q}_i^H \boldsymbol{V}_i(n)\end{aligned} \tag{5.28}$$

To find the condition for convergence, we take the expected value of equation (5.27). This leads the second and third right hand side (RHS) terms of equation (5.27) to vanish as both e_i'' and $\tilde{\boldsymbol{X}}_i(n)$ are uncorrelated. As a result, we obtain

$$\mathrm{E}\left[\boldsymbol{v}_i(n+1)\right] = \left\{\boldsymbol{I} - \mu_i \mathrm{E}\left[\tilde{\boldsymbol{X}}_i(n)\tilde{\boldsymbol{X}}_i^H(n)\right]\right\}\mathrm{E}\left[\boldsymbol{v}_i(n)\right] \tag{5.29}$$

For the LMS_1 algorithm stage, using the eigenvalue decomposition of $\boldsymbol{Q}_1$, and using similar definition in equation (4.35), equation (5.29) can be rewritten as

$$\begin{aligned}
\mathrm{E}\left[\boldsymbol{v}_1(n+1)\right] &= \left(\boldsymbol{I} - \mu_1 \mathrm{E}\left[\boldsymbol{q}_1{}^H \boldsymbol{X}_1(n)\boldsymbol{X}_1^H(n)\boldsymbol{q}_1\right]\right)\mathrm{E}\left[\boldsymbol{v}_1(n)\right] \\
&= \left(\boldsymbol{I} - \mu_1 \boldsymbol{q}_1{}^H \boldsymbol{Q}_1 \boldsymbol{q}_1\right)\mathrm{E}\left[\boldsymbol{v}_1(n)\right] \\
&= \left(\boldsymbol{I} - \mu_1 \boldsymbol{q}_1{}^H \boldsymbol{q}_1 \boldsymbol{\Lambda}_1 \boldsymbol{q}_1{}^H \boldsymbol{q}_1\right)\mathrm{E}\left[\boldsymbol{v}_1(n)\right] \\
&= \left(\boldsymbol{I} - \mu_1 \boldsymbol{\Lambda}_1\right)^n \mathrm{E}\left[\boldsymbol{v}_1(0)\right]
\end{aligned} \tag{5.30}$$

Using the definition given in equation (4.70) for the first stage of the LLMS algorithm, i.e., the LMS_1 algorithm stage, convergence can be satisfied if $\left|1 - \mu_1 E_{\max}\right| < 1$. This gives

$$0 < \mu_1 < 2 \big/ E_{\max} \tag{5.31}$$

where $E_{\max}$ is the largest eigenvalue of $\boldsymbol{Q}_1$.

For the LMS_2 algorithm stage, equation (5.29) is rewritten as

$$\mathrm{E}\left[\boldsymbol{v}_2(n+1)\right] = \left\{\boldsymbol{I} - \mu_2 \boldsymbol{R}_2(n)\right\}\mathrm{E}\left[\boldsymbol{v}_2(n)\right] \tag{5.32}$$

where $\boldsymbol{R}_2(n) = \mathrm{E}\left[\boldsymbol{X}_2(n)\boldsymbol{X}_2^H(n)\right]$ is the cross-correlation matrix of $\boldsymbol{X}_2$ and its general coefficient can be expressed as

$$r_{l,ka} = \mathcal{F}_{l,k}\, \mathcal{F}_{l,a}^* \sum_{l,m}^{N} \mathrm{E}\left[w_{1,l}^* x_l x_m^* w_{1,m}\right] \tag{5.33}$$

where $\mathcal{F}_{l,k}$ and $\mathcal{F}_{l,a}$ are the k^{th} and a^{th} elements of $\boldsymbol{\mathcal{F}}_{\mathrm{L}}$.

According to the assumptions (ii) and (iii) as stated in Section 4.3.1, we conclude that, equation (5.33) is nonzero only when $l = m$. Therefore, equation (5.33) can be rewritten as

$$r_{l,ka} = \mathcal{F}_{l,k}\, \mathcal{F}_{l,a}^{*} \mathrm{E}\left[y_l y_l^{*} \right] = \mathcal{F}_{l,k}\, \mathcal{F}_{l,a}^{*} \sigma_1^2 \tag{5.34}$$

where σ_1^2 is the variance of the output of the LMS_1 algorithm stage. In matrix form, equation (5.34) can be expressed as

$$\boldsymbol{R}_2(n) = \sigma_1^2 \boldsymbol{\mathcal{F}}_{\mathrm{L}}\, \boldsymbol{\mathcal{F}}_{\mathrm{L}}^{H} \tag{5.35}$$

where $\boldsymbol{R}_2$ is a complex matrix having a rank of one, that can be analyzed according to equation (5.32) using EVD [136], such that

$$\begin{aligned} \mathrm{E}\left[\boldsymbol{v}_2(n+1)\right] &= \boldsymbol{U}\left(\boldsymbol{I} - \mu_2 \sigma_1^2 \boldsymbol{\Lambda}_2\right) \boldsymbol{U}^{H} \mathrm{E}\left[\boldsymbol{v}_2(n)\right] \\ &= \boldsymbol{U}\left(\boldsymbol{I} - \mu_2 \sigma_1^2 \boldsymbol{\Lambda}_2\right)^{n+1} \boldsymbol{U}^{H} \mathrm{E}\left[\boldsymbol{v}_2(0)\right] \end{aligned} \tag{5.36}$$

where $\boldsymbol{U}$ is an *N*-by-*N* unitary matrix, and $\boldsymbol{\Lambda}_2 = \mathrm{diag}[\lambda_{\mathrm{L},2}, 0, 0, ..., 0]$ is the diagonal matrix of eigenvalues of the array matrix ($\boldsymbol{\mathcal{F}}_{\mathrm{L}}\, \boldsymbol{\mathcal{F}}_{\mathrm{L}}^{H}$).

Since this matrix is singular of rank 1, it has only one eigenvalue. With $\left\| \boldsymbol{\mathcal{F}}_{\mathrm{L}} \right\|^2 = N$, this eigenvalue is equal to N so that $\lambda_{\mathrm{L},2} = \mathrm{trace}(\boldsymbol{\mathcal{F}}_{\mathrm{L}} \cdot \boldsymbol{\mathcal{F}}_{\mathrm{L}}^{H}) = N$. From equation (5.36), the convergence of the LMS_2 algorithm can be satisfied if $\left| \boldsymbol{I} - \mu_2 \sigma_1^2 \boldsymbol{\Lambda}_2 \right| < 1$. This gives

$$0 < \mu_2 < {}^{2}\!\!\diagup\!\!{}_{N\sigma_1^2} \tag{5.37}$$

Thus, to ensure convergence of the LLMS algorithm, the step size values of μ_1 and μ_2 must satisfy equations (5.31) and (5.37), respectively.

5.3.4 LLMS algorithm with adaptive array image vector $\boldsymbol{\mathcal{F}}_{\mathrm{L}}$

It will be shown that the LLMS algorithm can be used for self-adaptive

beamforming in the same way as the RLMS algorithm, as described in Section 4.5. Also, the boundary values for the step size, μ_2, required for stable operation of the LLMS algorithm in this adaptive mode will also be derived.

5.3.4.1 Estimation of the array image vector $\mathcal{F}_{\mathrm{L}}$

Self-adaptive beamforming requires that the array image vector $\mathcal{F}_{\mathrm{L}}$ be adjusted automatically to always tracking the AOA of the desired signal. This can be achieved following the same procedure as that described for the RLMS algorithm in Section 4.5. In this case, the array image vector, $\mathcal{F}_{\mathrm{L}}$, is estimated based on the weights, $\boldsymbol{W}_1$, and the output, y_{LMS1}, of the LMS_1 algorithm stage.

Recalling equation (4.73), the output of the k^{th} antenna element is given by

$$x_{1,k}(t) = A_{d,k}s_d(t) + A_{i,k}s_i(t) + n_k(t) \tag{5.38}$$

where $A_{d,k}$, and $A_{i,k}$ are the k^{th} element of the complex array vectors for the desired signal and the cochannel interference, respectively. n_k is the additive white Gaussian noise associated with the k^{th} array element with $k = 1, 2, ..., N$.

The outputs of the individual LMS_1 algorithm tap weights, $w_{1,k}$, can be expressed as

$$x'_{1,k}(t) = w_{1,k}x_{1,k}(t) \tag{5.39}$$

Following the reasoning presented in Section 4.5 for the RLMS algorithm, the array vector elements are estimated as

$$A_{d,k}(t) \approx \frac{\mathrm{E}[x'_{1,k}(t)]}{\mathrm{E}[w_{1,k}]y_{\mathrm{LMS1}}(t) + \varepsilon_c} \tag{5.40}$$

where $y_{\mathrm{LMS1}}(t)$ is the output of the LMS$_1$ algorithm stage, and is assumed to approach $s_d(t)$ when the LMS$_1$ algorithm stage converges.

The value of the constant, ε_c, introduced to avoid a possible division-by-zero condition, is chosen such that

$$\varepsilon_c \ll \frac{|y_{\mathrm{LMS1}}|}{N} \sum_{\mathrm{k}=1}^{\mathrm{N}} |w_{1,k}| \tag{5.41}$$

For the computer simulations described in this chapter, ε_c has been set to the same value as in Section 4.7. It follows that the instantaneous values of the estimated array image vector for the k^{th} element can be calculated from

$$A_{d,k}(t) \approx \frac{x'_{1,k}(t)}{w_{1,k}(t)y_{\mathrm{LMS1}}(t) + \varepsilon_c} \tag{5.42}$$

5.3.4.2 Range of step size, μ_2 values for the LLMS algorithm

For the adaptive LLMS algorithm scheme, the convergence of the LMS$_1$ follows the same condition given by equation (5.31). However, the allowable step size values, μ_2, for the LMS$_2$ algorithm stage differ somewhat from those of equation (5.37). This is because the array image vector $\mathcal{F}_{\mathrm{L}}$ calculated using equation (5.40) is highly correlated with the difference $\boldsymbol{V}_1(n)$, between the estimated and actual tap weights, for the LMS$_1$ algorithm stage. The correlation coefficient between $x_{1,k}(t)$ in equation (5.38) and the error of LMS$_1$ algorithm, $\boldsymbol{V}_1(n)$, in equation (5.26) is defined as

$$\rho_{x_{1,k},\boldsymbol{V}_1}(n)=\frac{\mathrm{E}\left[x_{1,k}(n)\boldsymbol{V}_1(n)\right]-\mathrm{E}\left[x_{1,k}(n)\right]\mathrm{E}\left[\boldsymbol{V}_1(n)\right]}{\sigma_{x_{1,k}}\sigma_{\boldsymbol{V}_1}} \tag{5.43}$$

where $\sigma_{x_{1,k}}$ and $\sigma_{\boldsymbol{V}_1}$ are the standard deviations of the input signal, $x_{1,k}(n)$, and $\boldsymbol{V}_1(n)$, respectively.

As $x_{1,k}(n)$ and $\boldsymbol{V}_1(n)$ are highly correlated and almost zero mean, the first RHS of equation (5.43) becomes

$$\mathrm{E}\left[x_{1,k}(n)\boldsymbol{V}_1(n)\right]\approx\sigma_{x_{1,k}}\sigma_{\boldsymbol{V}_1} \tag{5.44}$$

so that equation (5.43) is approximated as

$$\rho_{x_{1,k},\boldsymbol{V}_1}(n)\approx 1-\frac{\mathrm{E}\left[x_{1,k}(n)\right]\mathrm{E}\left[\boldsymbol{V}_1(n)\right]}{\sigma_{x_{1,k}}\sigma_{\boldsymbol{V}_1}} \tag{5.45}$$

Using the similar estimation in equation (4.78) of the RLS algorithm stage of the RLMS algorithm, equation (5.45) can be rewritten as

$$\rho_{x_{1,k},\boldsymbol{V}_1}(n)=1-\frac{A_{d,k}(n)\,y_{\mathrm{LMS1}}(n)\mathrm{E}\left[\boldsymbol{V}_1(n)\right]}{\sigma_{x_{1,k}}\sigma_{\boldsymbol{V}_1}} \tag{5.46}$$

Rewriting equation (5.46) to obtain $A_{d,k}(n)$, such that

$$A_{d,k}(n)=\frac{\sigma_{x_{1,k}}\sigma_{\boldsymbol{V}_1}\left(1-\rho_{x_{1,k},\boldsymbol{V}_1}(n)\right)}{y_{\mathrm{LMS1}}(n)\mathrm{E}\left[\boldsymbol{V}_1(n)\right]} \tag{5.47}$$

Since $x_{2,k}=A_{d,k}\,y_{\mathrm{LMS1}}$, the element form of $\boldsymbol{R}_2$ in equation (5.32) can be written as

$$\begin{aligned} R_{2,k}(n)&=\mathrm{E}\left[x_{2,k}(n)x_{2,k}^{*}(n)\right]\\ &=\mathrm{E}\left[A_{d,k}\,y_{\mathrm{LMS1}}\,y_{\mathrm{LMS1}}^{*}A_{d,k}^{*}\right]\\ &=\mathrm{E}\left[\left|y_{\mathrm{LMS1}}\right|^2\right]\mathrm{E}\left[\left|A_{d,k}\right|^2\right] \end{aligned} \tag{5.48}$$

Substituting equation (5.47) in the above equation gives

$$R_{2,k}(n) = \mathrm{E}\left[\left|y_{\mathrm{LMS1}}\right|^2\right] \frac{\sigma_{x_{1,k}}^2 \sigma_{V_1}^2 \left|1-\rho_{x_{1,k},V_1}(n)\right|^2}{\mathrm{E}\left[\left|y_{\mathrm{LMS1}}\right|^2\right] \mathrm{E}\left[\left|V_1(n)\right|^2\right]}$$

$$= \frac{\sigma_{x_{1,k}}^2 \sigma_{V_1}^2 \left|1-\rho_{x_{1,k},V_1}(n)\right|^2}{\mathrm{E}\left[\left|V_1(n)\right|^2\right]} \tag{5.49}$$

Now, multiply both sides of equation (5.30) by $\boldsymbol{q}_i^{-1}$ and introducing the maximum eigenvalue, $E_{\max}$, yields

$$\mathrm{E}\left[V_1(n+1)\right] = [1-\mu_1 E_{\max}]\,\mathrm{E}\left[V_1(n)\right] \tag{5.50}$$

Substituting equation (5.50) in equation (5.49), gives

$$R_{2,k}(n) = \frac{\sigma_{x_{1,k}}^2 \sigma_{V_1}^2 \left|1-\rho_{x_{1,k},V_1}(n)\right|^2}{[1-\mu_1 E_{\max}]^2\,\mathrm{E}\left[\left|V_1(n)\right|^2\right]} \tag{5.51}$$

Finally, from equation (5.32), the condition for convergence of the LMS_2 algorithm stage is given by

$$\left|1-\mu_2 \frac{\sigma_{x_{1,k}}^2 \sigma_{V_1}^2 \left|1-\rho_{x_{1,k},V_1}(n)\right|^2}{[1-\mu_1 E_{\max}]^2\,\mathrm{E}\left[\left|V_1(n)\right|^2\right]}\right|^2 < 1 \tag{5.52}$$

This leads to

$$0 < \mu_2 < 2 \frac{[1-\mu_1 E_{\max}]^2\,\mathrm{E}\left[\left|V_1(n)\right|^2\right]}{\sigma_{x_{1,k}}^2 \sigma_{V_1}^2 \left|1-\rho_{x_{1,k},V_1}(n)\right|^2} \tag{5.53}$$

where $\sigma_{x_{1,k}}$ and σ_{V_1} are as defined in equation (5.43).

5.4 Computer Simulation

5.4.1 Introduction

The convergence of the proposed LLMS algorithm, operating with either an external reference or self referencing, has been established analytically in the previous section. In this section, the performance of the LLMS algorithm will be further evaluated by means of computer simulations. Included in the evaluations are the two versions of the algorithm, namely the LLMS_1 algorithm which makes use of a fixed image array vector, $\mathcal{F}_{\text{L}}$, and the LLMS algorithm with an adaptive $\mathcal{F}_{\text{L}}$. Also, we consider three other LMS based algorithms, including the conventional LMS, CSLMS and MRVSS algorithms, for performance comparison.

For the simulations, carried out using full numerical precision, the antenna array beamformer is assumed to operate under similar conditions to those adopted in Chapter 4 for the RLMS algorithm. For clarity, these conditions are restated as follows:

- A linear array of 8 isotropic antenna elements, spaced half carrier wavelength apart.
- A desired binary phase shift keying (BPSK) arrives at an angle of 0°, or if specified at either -20° or 10°.
- An AWGN channel.
- All weight vectors are initially set to zero.
- A BPSK interference signal with the same amplitude as the desired signal arrives at $\theta_i = 45^\circ$.
- For operation in a Rayleigh fading channel, the maximum Doppler frequency is equal to 60 Hz. In this case, both the desired and interference signals that arrive at each antenna element undergo independent Rayleigh fading.

The values of the parameters required for the five algorithms are tabulated in Table 5-1. These values have been chosen in order to obtain as good a performance out of these algorithms as possible.

Table 5-1 Values of the constants used in the simulations

Algorithm	AWGN Channel	Rayleigh Fading Channel
LMS	$\mu = 0.05$	$\mu = 0.01$
CSLMS	$\varepsilon = 0.05,\ \mu = 0.05$	$\varepsilon = 1,\ \mu = 0.05$
MRVSS	$\beta_{\max} = 1,\ \beta_{\min} = 0,\ \upsilon = 5\times10^{-4}$ $\mu_{\max} = 0.2,\ \mu_{\min} = 10^{-4}$ Initial $\mu = \mu_{\max},\ \alpha = 0.97$ $\gamma = 4.8\times10^{-4},\ \eta = 0.97$	Same as column 2 except for: Initial $\mu = 0.1,\ \mu_{\max} = 0.05$ $\mu_{\min} = 10^{-4}$
$LLMS_1$	$\mu_1 = \mu_2 = 0.05$	$\mu_1 = \mu_2 = 0.01$
LLMS	$\mu_1 = 0.08,\ \mu_2 = 0.05$	

5.4.2 Simulation results

Computer simulated results, obtained using Matlab, for the $LLMS_1$, LLMS, CSLMS, MRVSS and LMS algorithms are presented. As adopted in the previous chapter for the RLMS algorithm, the performance metrics used here are MSE, beam pattern, EVM, scatter plot and SINR. Wherever possible, the simulated results are compared with the theoretical values derived from the analyses presented earlier in Section 5.3.1.

5.4.2.1 Error convergence with an ideal external reference

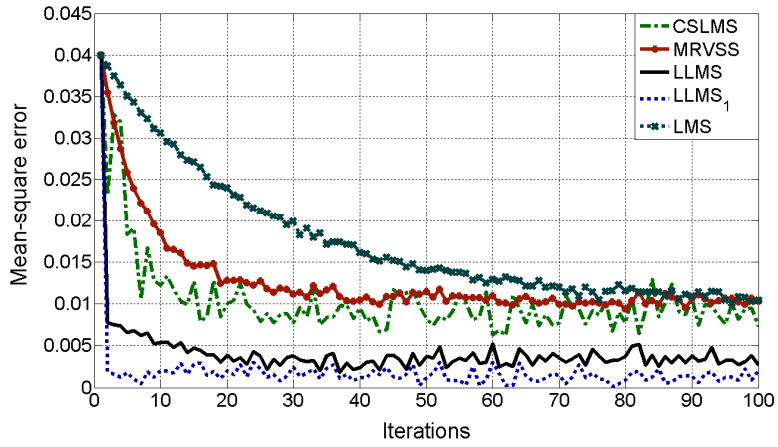
First, the performance of a beamformer implemented using one of the five given algorithms is evaluated in the presence of an ideal external reference signal. The convergence behaviour of each of these algorithms is studied based on the ensemble average squared error, $\tilde{e}^2$, of the overall error signal obtained from 100 individual simulation runs. The effects of AWGN on the convergence have been considered for the different values of step sizes, μ_1 and μ_2, used.

Figures 5-2(a)–(c) show the convergence behaviours of the five adaptive schemes for SNR values of 5, 10, and 15 dB, respectively. For the proposed $LLMS_1$ algorithm scheme, the theoretical convergence error calculated using

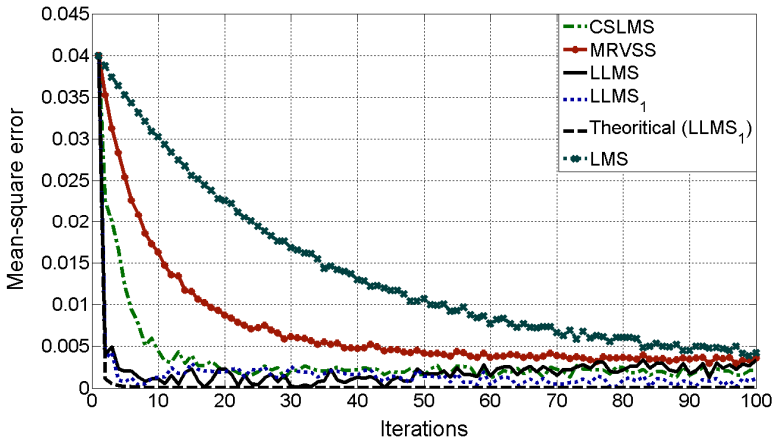
equation (5.5) for SNR of 10 dB is also shown in Figure 5-2b. It is observed that under the given conditions, the two variants of the proposed LLMS algorithm converge much faster than the other three schemes. Furthermore, their error floors are less sensitive to variations in the input SNR, even for an input SNR as small as 5 dB. Also, as shown in Figure 5-2b, there is a close agreement between the simulated and theoretical error curves for the proposed LLMS and LLMS_1 algorithms. This validates the method used for estimating $\mathcal{F}_{\text{L}}$ for the LLMS algorithm as described in Section 5.3.4.1. As for the CSLMS and MRVSS algorithms, they share the same performance for all the three SNR values considered. As expected, the conventional LMS algorithm achieves the slowest convergence among the five algorithms considered. It is observed that both of the two modes of operation of the proposed LLMS algorithm achieve very similar performance compared with their counterparts based on the RLMS algorithms described in Section 4.5.

Next, we consider the influence of the step sizes, μ_1 and μ_2, on the stability of the proposed LLMS_1 and LLMS algorithms, operating with either a fixed $\mathcal{F}_{\text{L}}$ or adaptive $\mathcal{F}_{\text{L}}$. It is verified that to ensure convergence, the values of the step size used have to fall within the bounds given in equations (5.31) and (5.37) for the LLMS_1 algorithm, and equations (5.31) and (5.53) for the LLMS algorithm.

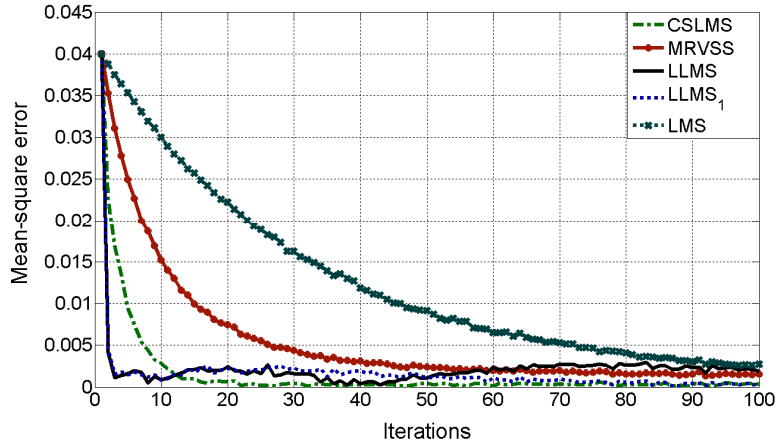
For an 8-element array operating with an input SNR of 10 dB, the required bounds are $0 < \mu_1 < 0.8$ and $0 < \mu_2 < 0.7$ for the LLMS_1 algorithm, and $0 < \mu_1 < 0.8$ and $0 < \mu_2 < 0.26$ for the LLMS algorithm. When the step sizes are chosen to be well within their limits, such as for the values used in Figure 5-2, both versions of the LLMS algorithm are able to converge within a few iterations to a low error floor. However, the LLMS_1 algorithm shows some sign of instability when operating with step sizes close to their upper limits. Such instability in convergence behaviour of the LLMS_1 algorithm is shown in Figure 5-3 for two cases that make use of $\mu_1 = 0.008$ and $\mu_2 = 0.6$, and $\mu_1 = 0.7$ and $\mu_2 = 0.08$.



(a) SNR=5 dB



(b) SNR=10 dB



(c) SNR=15 dB

Figure 5-2 The convergence of the LLMS, $LLMS_1$, CSLMS, MRVSS and LMS algorithms implemented using the parameters as tabulated in the 2nd column of Table 5-1, for three different values of input SNR.

As discussed in Sections 5.3.3 and 5.3.4.2, the limits of the step size, μ_2, for the LLMS algorithm are different from those for the LLMS_1 algorithm. This is due to the fact that the calculation of the array image vector in the LLMS algorithm is correlated with the output of its first LMS algorithm stage. The allowable upper limits of μ_1 and μ_2 for the LLMS algorithm are demonstrated in Figure 5-3, for the two cases involving the use of $\mu_1 = 0.7$ and $\mu_2 = 0.005$, and $\mu_1 = 0.08$ and $\mu_2 = 0.23$, respectively.

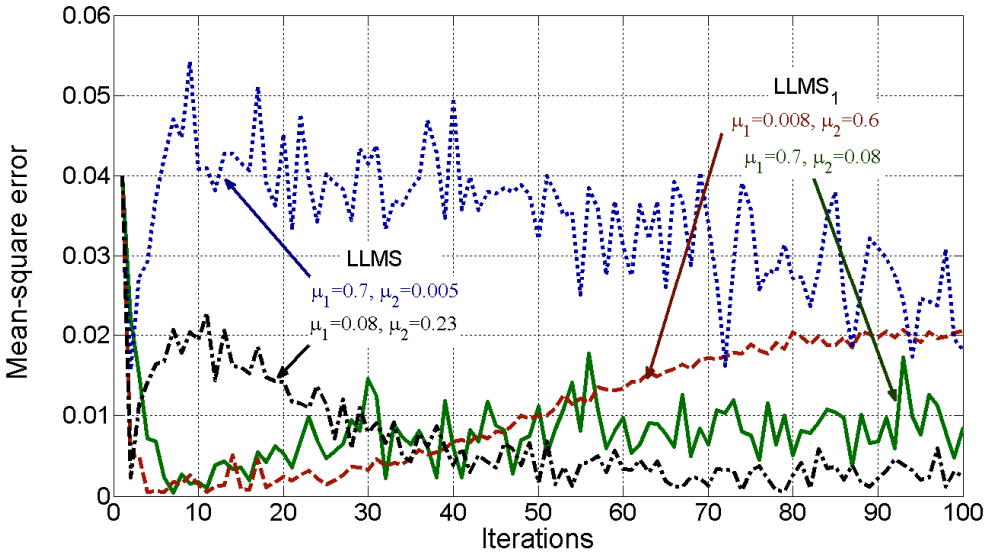


Figure 5-3 The convergence behaviours of the LLMS_1 and LLMS algorithms at SNR=10 dB for step size values set at their upper limits.

For the LLMS algorithm, the array image vector, $\mathcal{F}_{\text{L}}$, is being determined adaptively, which in turn has an effect on its convergence behaviour. As shown in Figure 5-3, the resulting error floors achieved when the LLMS algorithm is operating with one of its two step sizes, (either μ_1 or μ_2), close to the upper limit, tend to first diverge before finally converging. Consequently, this results in a longer convergence time.

5.4.2.2 Performance with self-referencing

As shown in Figure 5-2 and Figure 5-3, both the $LLMS_1$ and LLMS algorithms are able to converge within ten iterations. Once this occurs, the intermediate output, y_{LMS1}, tends to resemble the desired signal $s_d(t)$, and may then be used in place of the external reference, d_2, for the current iteration of the LMS_2 algorithm stage. As the LMS_2 algorithm section converges, its output, y_{LLMS}, becomes the estimated $s_d(t)$. As a result, y_{LLMS} may be used to replace d_1 as the reference for the LMS_1 algorithm stage. This feedforward and feedback arrangement enables the provision of self-referencing for the proposed LLMS algorithm. If needed, this allows the external reference signal to be discontinued after an initial four iterations. The ability of the LLMS algorithms to maintain operation with the internally generated reference signals is demonstrated in Figure 5-4. Furthermore, it clearly shows that the traditional LMS, CSLMS, MRVSS algorithms are unable to converge without the use of an external reference signal. For comparison, the theoretical convergence errors calculated from equation (5.21) are also plotted in Figure 5-4.

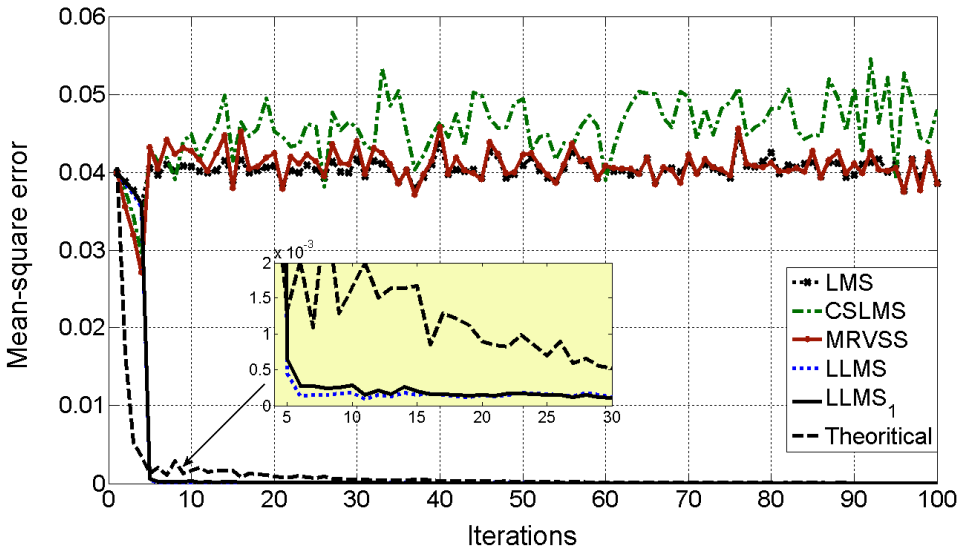


Figure 5-4 The convergence of the LLMS and $LLMS_1$ algorithms with self-referencing when operating with the parameters as tabulated in column 2 of Table 5-1 for SNR=10 dB. An external reference is used for the initial four iterations before switching to self-referencing.

5.4.2.3 Performance with a noisy reference signal

The effects of the use of a reference signal corrupted by AWGN on the operation of the LLMS, $LLMS_1$, CSLMS, MRVSS and LMS algorithms have also been investigated. This is done by examining the resultant mean square error ξ of the overall error signal, e_{LLMS}, when the noise level in the reference signal is varied. Figure 5-5 shows the ensemble average of the mean square error, $\overline{\xi}$, obtained from 100 individual simulation runs, as a function of the ratio of the rms noise σ_n to the amplitude of the reference signal.

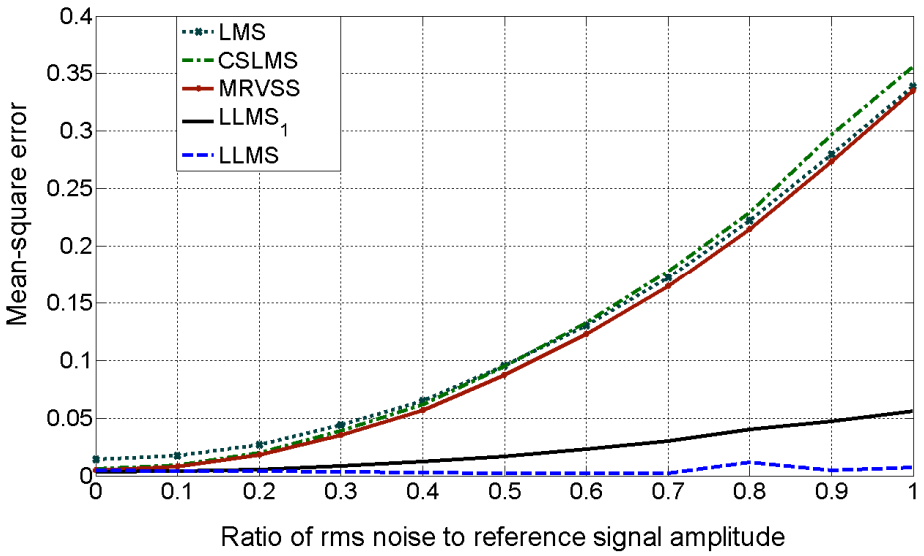


Figure 5-5 The influence of noise in the reference signal on the mean square error $\overline{\xi}$ when operating using the parameters given in column 2 of Table 5-1.

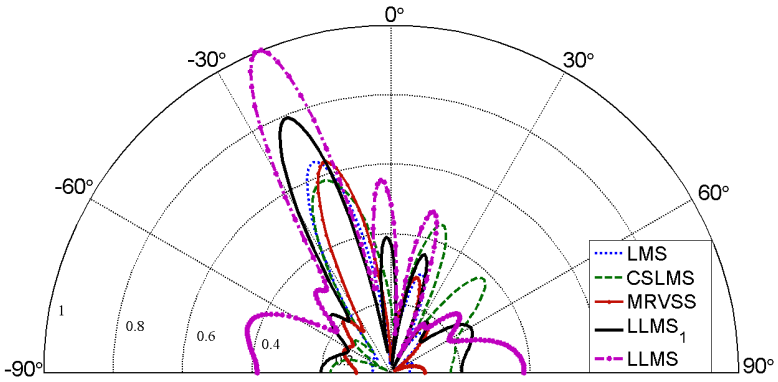
It is interesting to note that the conventional LMS, CSLMS and MRVSS algorithms are quite sensitive to the presence of noise in the reference signal. On the other hand, both the $LLMS_1$ and LLMS algorithms are very tolerant to noisy reference signals. As shown in Figure 5-5, the values of $\overline{\xi}$ associated with the LLMS and $LLMS_1$ algorithms remain very small even when the rms noise becomes as large as the reference signal. The effect of noise is even less pronounced on the LLMS algorithm due to the fact that its

array image vector $\mathcal{F}_{\mathrm{L}}$ is being continuously updated rather than being fixed at the prescribed values.

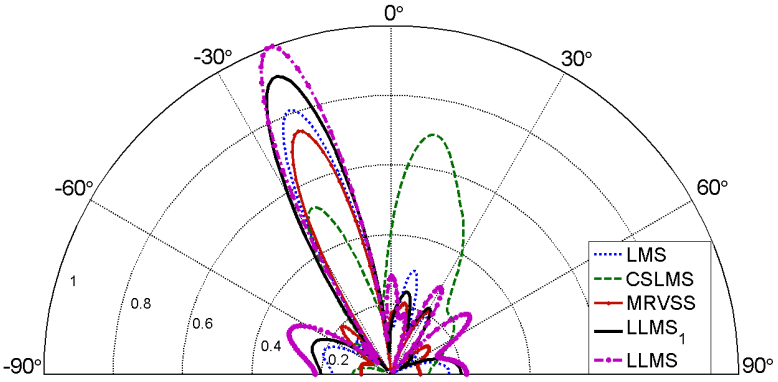
Next, Figure 5-6 shows the resulting beam patterns obtained with the five given algorithms when the reference signal used is corrupted by AWGN. In this case, the desired signal is arriving at an angle, θ_d, of -20$^{\mathrm{o}}$. From Figure 5-3a and Figure 5-3b, it is observed that both the LLMS$_1$ and LLMS algorithms are able to maintain their correct beam patterns, when the ratios of the rms noise to the reference signal amplitude are -3dB and -9dB, respectively. However, in both cases, the LLMS$_1$ algorithm with a fixed prescribed $\mathcal{F}_{\mathrm{L}}$ suffers from a drop in gain. When the reference used is free from AWGN, then all the five algorithms are able to achieve similar beamforming performance, as shown in Figure 5-6c. Moreover, as noise in the reference signal increases, the beam pattern of the LLMS algorithm tends to deviate slightly from its designated direction as a result of the use of the estimated values for the array image vector $\mathcal{F}_{\mathrm{L}}$.

5.4.2.4 Tracking performance of the LLMS algorithm

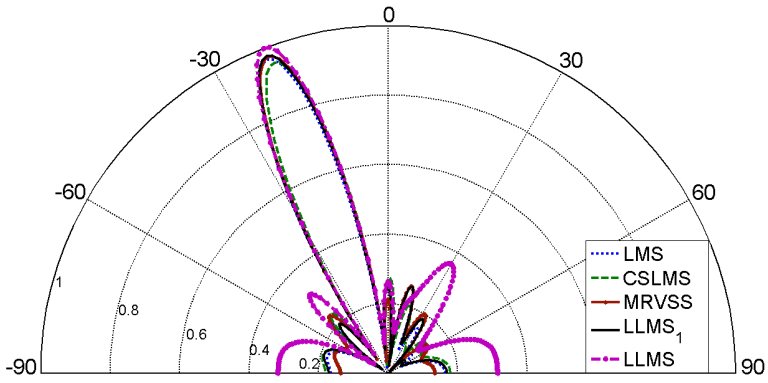
The ability of the LLMS and LLMS$_1$ algorithms in tracking sudden interruptions in the input signal is investigated by examining the behaviour of the overall error signal, e_{LLMS}. For this study, the input signal is assumed to be periodically interrupted for 25 out of 100 iterations. The resulting tracking performances of the LLMS and LLMS$_1$ algorithms are shown in Figure 5-7, which shows their respective values of mean square errors ξ are increasing very rapidly when the input signal is switched on or off. This indicates the fast response of the LLMS and LLMS$_1$ algorithms to sudden interruptions in the input signal. Both the LLMS$_1$ and LLMS algorithms behave in similar manner. Unlike the responses for the LMS, CSLMS and MRVSS algorithms, which are also included in Figure 5-7 for comparison purposes, the mean square errors ξ associated with both the LLMS$_1$ and LLMS algorithms remain low when the input signal is interrupted.



(a) $(\sigma_N / S_R)_{\mathrm{dB}} = -3\,\mathrm{dB}$



(b) $(\sigma_N / S_R)_{\mathrm{dB}} = -9\,\mathrm{dB}$



(c) Without noise

Figure 5-6 The beams patterns obtained with the LLMS, LLMS_1, CSLMS, MRVSS and LMS algorithms when the reference signal is contaminated by AWGN. The desired signal arrives at $\theta_d = -20^{\mathrm{o}}$.

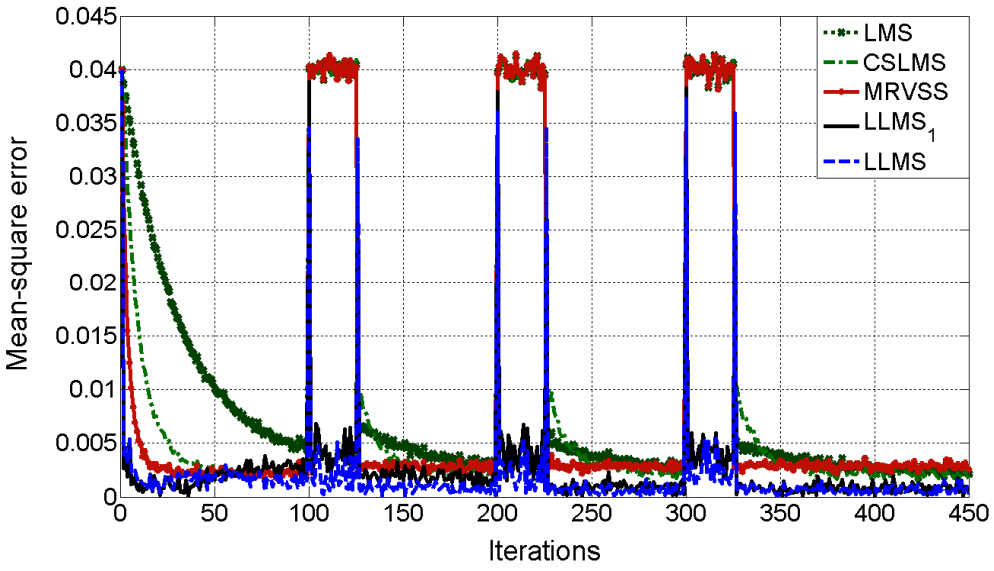


Figure 5-7 Tracking performance comparison of the LLMS, $LLMS_1$, CSLMS, MRVSS and LMS algorithms operating with $\mu = \mu_1 = \mu_2 = 0.5$ and an input SNR of 10 dB.

5.4.2.5 EVM and scatter plot

The performances of the five algorithms, based on the rms EVM computed using equation (4.86), for values of input SNR ranging from 0–30 dB in steps of 5 dB, are shown in Figure 5-8. These EVM values have been computed after each of the adaptive algorithms has converged. It is observed that the two lowest EVM values are being achieved with the proposed LLMS and $LLMS_1$ algorithms. In fact, through the use of adaptive $\mathcal{F}_L$, the LLMS algorithm is able to achieve slightly smaller EVM than its counterpart, the $LLMS_1$ algorithm at lower input SNR. The superior performance of the LLMS algorithm among the five algorithms considered becomes even more pronounced at lower values of input SNR. This further confirms the observation made from Figure 5-2 showing that the operations of the $LLMS_1$ and LLMS algorithms are very insensitive to changes in input SNR.

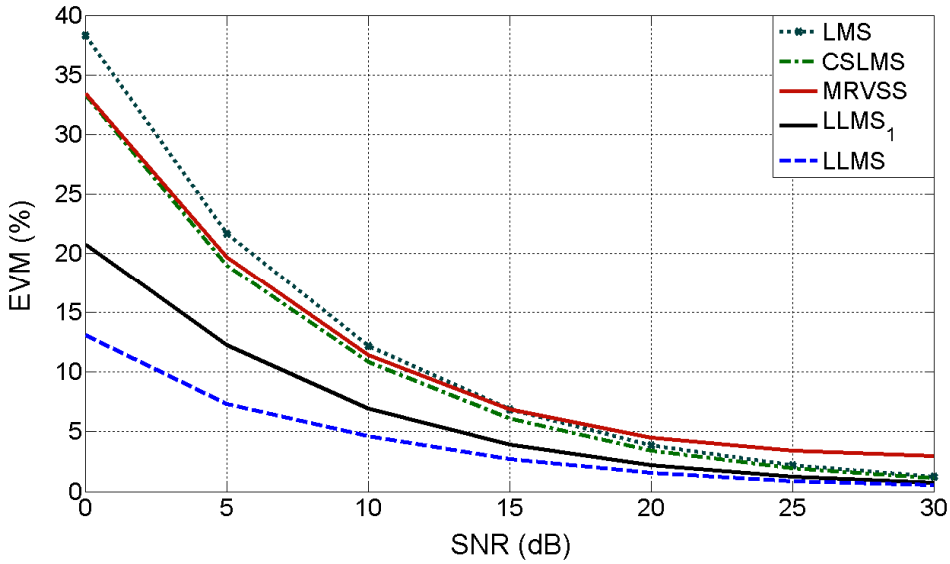
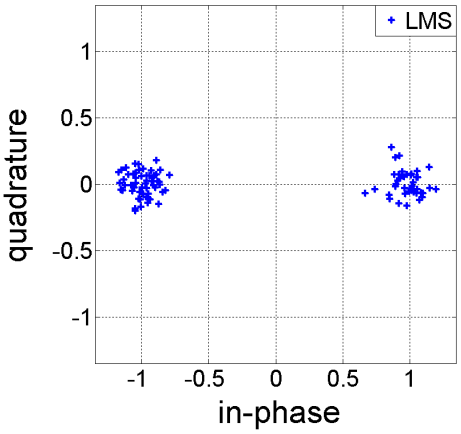
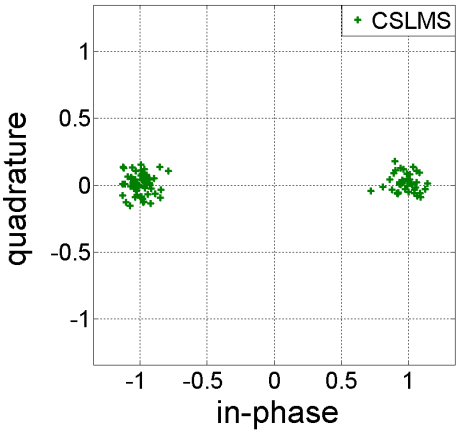


Figure 5-8 The EVM values obtained with the LLMS, $LLMS_1$, CSLMS, MRVSS and LMS algorithms at different input SNR.

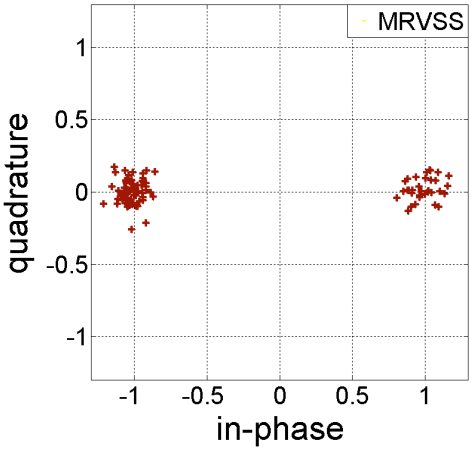
Next, the scatter plots of the BPSK signal recovered through the use of an adaptive beamformer, based on the LMS, CSLMS, MRVSS, $LLMS_1$ and LLMS algorithms, are shown in Figures 5-9(a)–(e), respectively. These scatter plots have been obtained from 100 signal samples after each of the adaptive algorithms has converged. In each case, the input SNR is 10 dB with two cochannel interfering signals, arriving at angles of -30^o and 45^0, having similar amplitude to the desired signal. Again, the scatter plots with the $LLMS_1$ and LLMS algorithms show the least spreading, indicating their ability to retain the signal fidelity, and suppress interference arriving from outside the main beam width of the array.



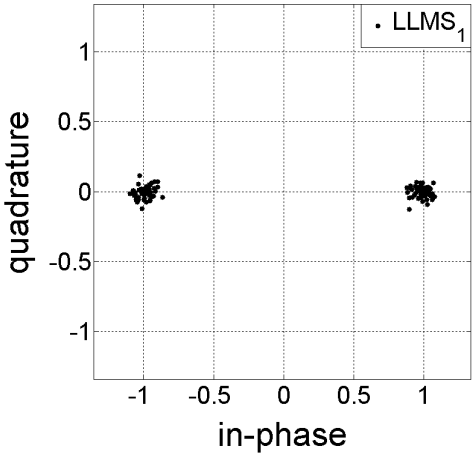
(a) LMS algorithm



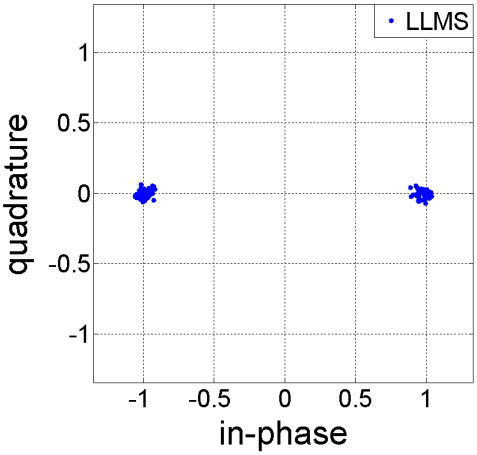
(b) CSLMS algorithm



(c) MRVSS algorithm



(d) $LLMS_1$ algorithm



(e) LLMS algorithm

Figure 5-9 The scatter plots of the recovered BPSK signal obtained with (a) LMS, (b) CSLMS, (c) MRVSS, (d) $LLMS_1$, and (e) LLMS algorithms obtained with an input SNR of 10 dB and an SIR of -6 dB.

5.4.2.6 Operation in flat Rayleigh fading channel

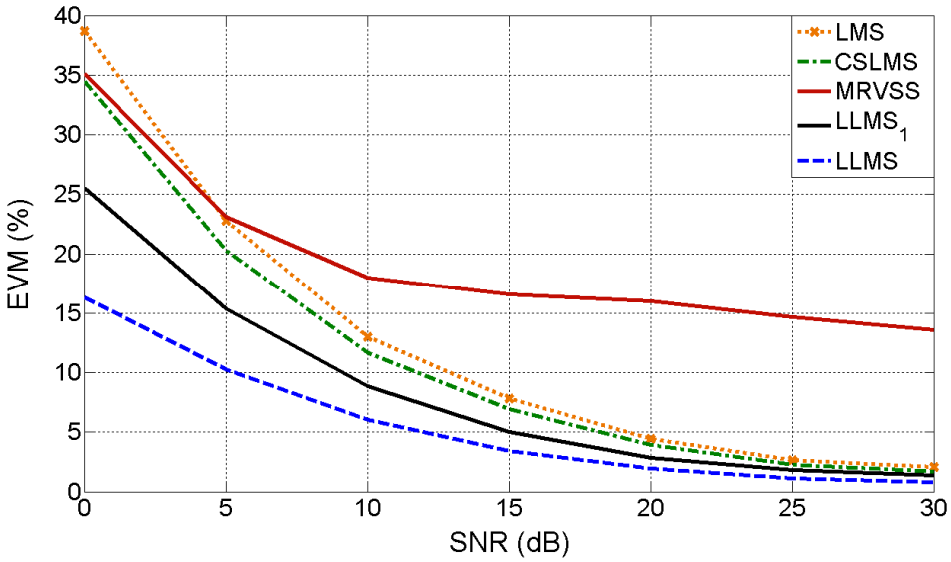
The ability of an adaptive beamformer to operate in a fast changing signal environment is examined by subjecting the input signals to flat Rayleigh fading. In this case, the rms EVM is again used as the performance metric for comparison between the different adaptive beamforming algorithms. The same operating conditions, as stated in Section 4.7.3.8 for the RLMS algorithm, are adopted here for the performance evaluation of the LLMS algorithm by computer simulation:

Figure 5-10a and Figure 5-10b show the resultant EVM values as a function of the input SNR achieved for the cases with and without interference, respectively. From these figures, the following observations are made:

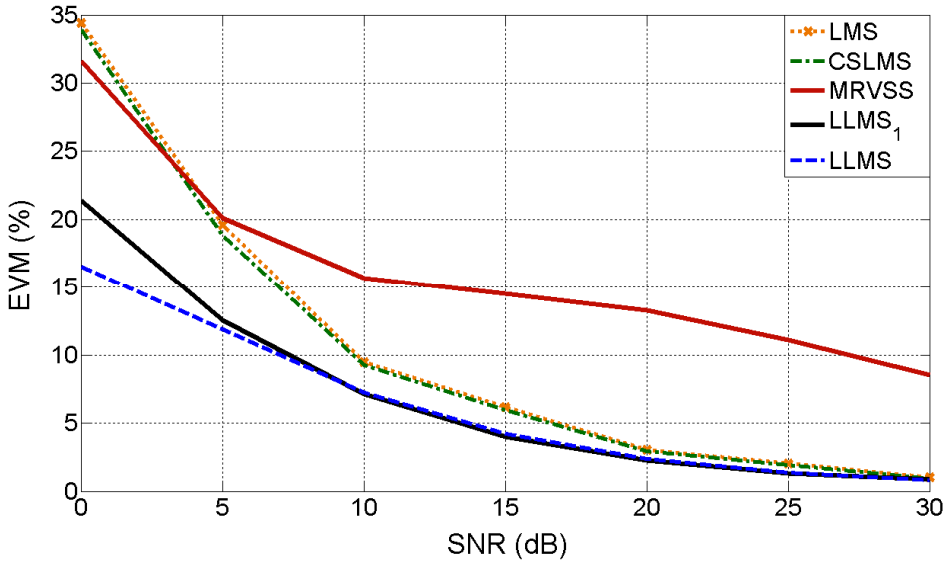
- With the exception of the MRVSS algorithm, the other four algorithms are able to operate in the presence of Rayleigh fading.
- Irrespective of whether interfering signals are present or not, both the LLMS and $LLMS_1$ algorithms perform best among the algorithms considered.
- The LLMS algorithm, operating with an adaptive array image vector, is the least affected by interference.

5.4.2.7 Influence of the AOA, θ_i, of the interference

The effect of the AOA of an interfering signal on the desired signal recovered at the output of an adaptive beamformer is also investigated. In this study, the desired signal is arriving at either θ_d of 0^o, (bore-side), or θ_d of 90^o, (end-fire), with three different input SNR values, i.e., 5 dB, 10 dB or 15 dB. The interference has the same power as that of the desired signal, i.e., SIR=0 dB, and is arriving over a range of AOA from -90^o to 90^o. The performance measure is the output signal-to-noise plus interference ratio, SINR_o, obtained with a given algorithm after convergence has been achieved. In each case, the output signal-to-noise plus interference ratio, at



(a) In the presence of fading and cochannel interference



(b) In the presence of fading but without interference

Figure 5-10 The EVM values obtained with the LLMS, $LLMS_1$, CSLMS, MRVSS and LMS algorithms for different values of input SNR in the presence of Rayleigh fading: (a) signal-to-interference ratio (SIR) of -6 dB, and (b) without cochannel interference.

the n^{th} iteration, is computed according to

$$\mathrm{SINR}_{\mathrm{o}}(n) = \frac{P_d(n)}{P_i(n) + P_n(n)} \tag{5.54}$$

where $\mathrm{SINR}_{\mathrm{o}}$ is the ensemble average output SINR obtained from 30 simulation runs,

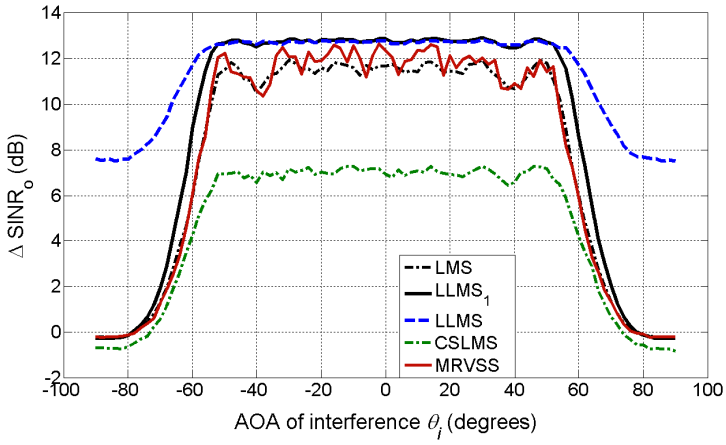
$$P_d(n) = \left|\mathbf{W}^H(n)\mathbf{A}_d\right|^2 V_S^2/2 \tag{5.55}$$

$$P_i(n) = \left|\mathbf{W}^H(n)\mathbf{A}_i\right|^2 V_i^2/2 \tag{5.56}$$

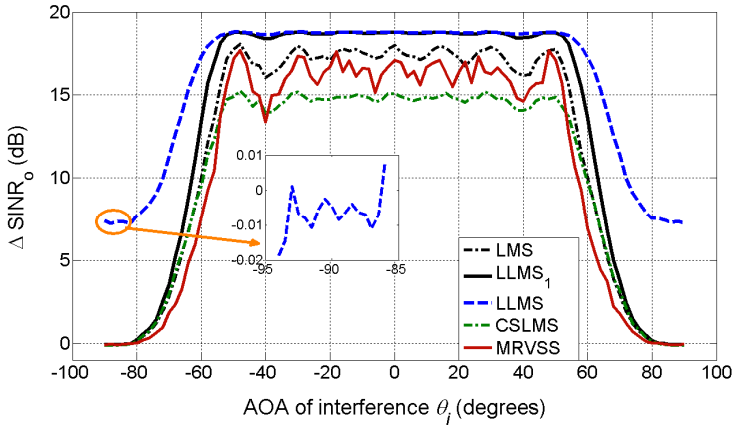
and $$P_n(n) = \sigma_n^2 \left|\mathbf{W}^H(n)\right|^2 \tag{5.57}$$

are the average output powers, at the n^{th} iteration, of the desired signal, the interference signal and the AWGN, respectively. The input amplitudes of the desired and interfering BPSK signals are V_s and V_i, respectively, while σ_n is the rms noise voltage. The weights, at the n^{th} iteration, associated with a given algorithm are represented by $\mathbf{W}^H(n)$.

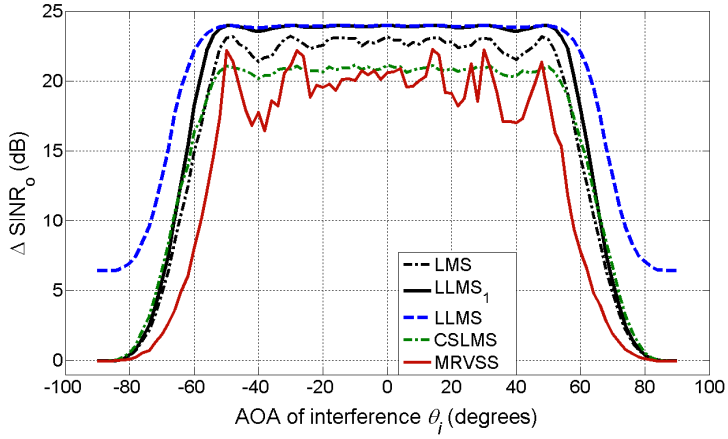
Figure 5-11 and Figure 5-12 show how the angle of arrival of an interfering signal is affecting the $\mathrm{SINR}_{\mathrm{o}}$ of the desired signal under three different values of input SNR, i.e., 5, 10 and 15 dB. As expected, the interference has little effect on the $\mathrm{SINR}_{\mathrm{o}}$ when its AOA is outside the main beam of the array. This is true for both the bore-side and end-fire cases. Among the cases considered, the beamformers based on the LLMS and LLMS_1 algorithms achieve higher values of $\mathrm{SINR}_{\mathrm{o}}$, and show less fluctuations in $\mathrm{SINR}_{\mathrm{o}}$ when the interference is arriving outside the beamwidth. The latter is the result of the lower side lobes associated with the LLMS and LLMS_1 algorithms.



(a) SNR=5 dB

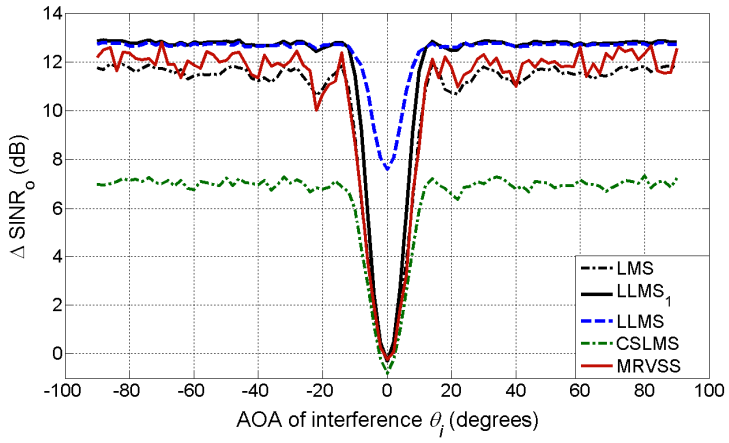


(b) SNR = 10 dB

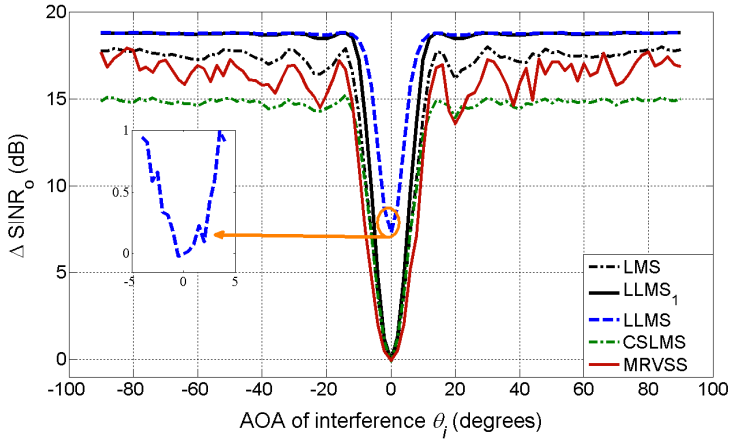


(c) SNR = 15 dB

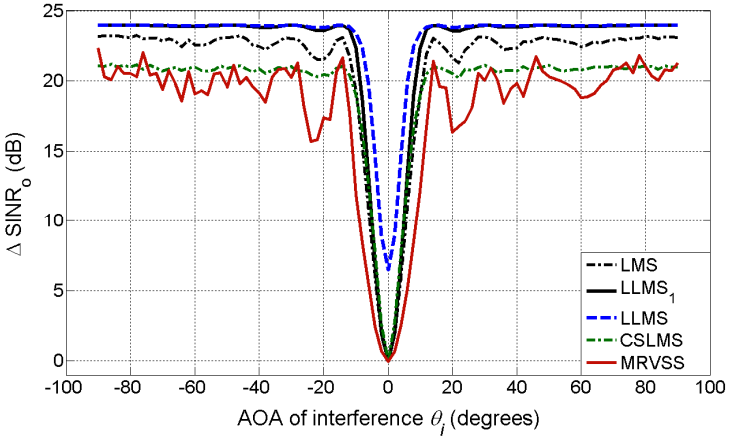
Figure 5-11 Influence of AOA of the interference on the output SINR_o for three different values of input SNR. The desired signal arrives at $\theta_d = 90^o$ (end-fire).



(b) SNR = 5 dB



(d) SNR = 10 dB



(f) SNR = 15 dB

Figure 5-12 Influence of AOA of the interference on the output SINR_o for three different values of input SNR. The desired signal arrives at $\theta_d = 0^o$ (bore-side).

5.5 Comparison Between RLMS and LLMS Algorithms

In this section, the proposed RLMS and LLMS beamforming algorithms are compared on the basis of their resulting mean square errors and beam patterns. The operating parameters of the RLMS and LLMS algorithms are as given in Table 4-1 and Table 5-1, respectively. It is assumed that the signal of interest is arriving at $\theta_d = 10^o$, together with four co-channel interfering signals having angles of arrival of $\theta_i = -50^\circ,\ -30^\circ,\ -10^\circ,$ and 45°. All the four interfering signals have the same amplitude as the desired signal, with SNR of 10 dB and θ_d of 10^o.

Figure 5-13 and Figure 5-14 show the error convergence and the beam patterns obtained based on an SNR of 10 dB. From these figures, the following observations are made:

- The two variants of the RLMS algorithm converge slightly faster than the two LLMS algorithm schemes, while they share similar error floor.
- The resultant beam patterns obtained with both variants of the RLMS and LLMS algorithms have almost identical gain at the angle of arrival of the desired signal. Moreover, the two variants of the RLMS algorithms are able to achieve a greater suppression of all the four interfering signals.

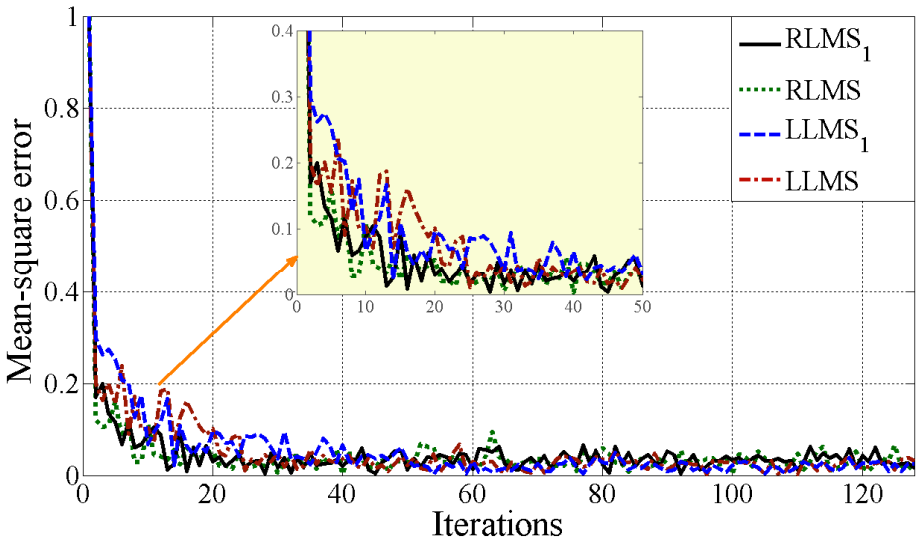


Figure 5-13 The rate of convergence of the RLMS, $RLMS_1$, LLMS and $LLMS_1$ algorithms operating with an SNR of 10 dB in the presence of four-equall-amplitude interfering signals arriving at $\theta_{i1} = -50^\circ, \theta_{i2} = -30^\circ, \theta_{i3} = -10^\circ$ and $\theta_{i4} = 45^\circ$.

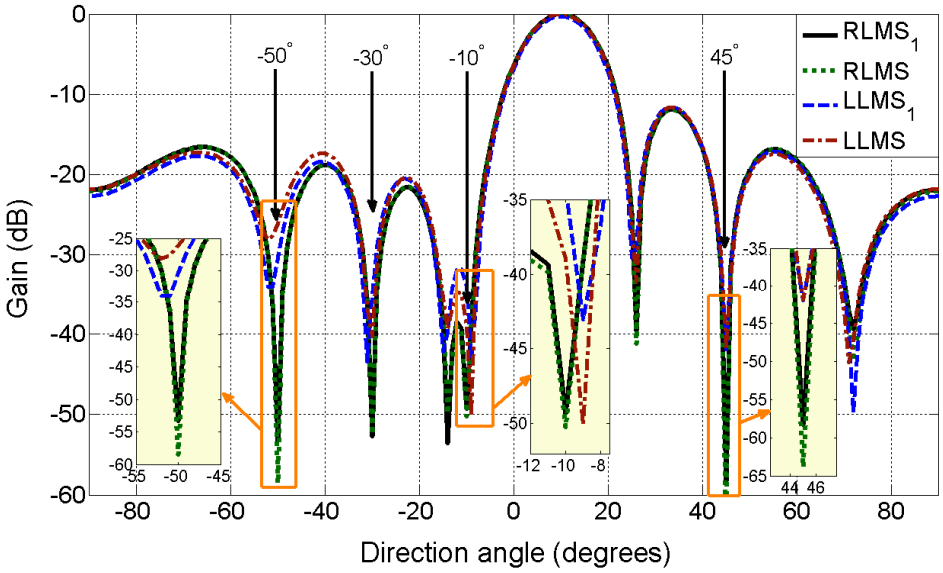


Figure 5-14 The beams patterns obtained with the RLMS, $RLMS_1$, LLMS and $LLMS_1$ for an SNR of 10 dB in the presence of four-equal-amplitude interfering signals arriving at $\theta_{i1} = -50^{\circ}$, $\theta_{i2} = -30^{\circ}$, $\theta_{i3} = -10^{\circ}$ and $\theta_{i4} = 45^{\circ}$. The desired signal arrives at $\theta_d = 10^{\circ}$.

5.6 Summary

In this chapter, a new algorithm, called the LLMS algorithm, is proposed to reduce the complexity of the RLMS algorithm discussed in Chapter 4. In this case, the RLS algorithm stage of the RLMS algorithm is being replaced by another LMS algorithm stage. Consequently, the LLMS algorithm makes use of two concatenated LMS algorithm stages separated by an array image vector, $\mathcal{F}_{\mathrm{L}}$. With this modification, a linear *N*-element adaptive beamformer implemented with the LLMS algorithm will require only $4N+1$ complex multiplications as opposed to $2.5N^2+5N+1$ complex multiplications for the RLMS algorithm. Unlike the previously published LMS based algorithms, such as the CSLMS and MRVSS algorithms, which require many input signal dependent parameters to be specified, the proposed LLMS algorithm depends for its operation only on two step size values, one for each of the two LMS algorithm stages. This makes the LLMS algorithm attractive for

practical applications.

As discussed in Section 5.2, the proposed algorithm makes use of the array image vector $\mathcal{F}_L$ to interface between the two LMS algorithm stages. In this way, an accurate fixed beam can be obtained by prior setting of the elements of $\mathcal{F}_L$ with the prescribed values for the required direction. Alternatively, $\mathcal{F}_L$ may be made adaptive to automatically track the target signal. The same simple and effective method as described in Section 4.5 can also be adopted for calculating the element values of $\mathcal{F}_L$, based on the estimated output signal of the first LMS algorithm section and its tap weights.

The convergence of the two versions of the LLMS algorithm, i.e., the $LLMS_1$ and LLMS algorithms, has been analyzed assuming the use of an external reference signal. This is then extended to cover the case that makes use of self-referencing. The convergence behaviours of these two LLMS algorithms with different step size combinations of μ_1 and μ_2 have been demonstrated by means of computer simulations under different input SNR conditions. Also, the boundary values of μ_1 and μ_2 for stable operation have been derived analytically in Sections 5.3.3 and Section 5.3.4.2. Operations with these values have been verified by computer simulations.

It is shown that the simpler LLMS algorithm performs very similar to the RLMS algorithm. For example, the two variants of the LLMS algorithm are able to have rapid convergence, typically within a few iterations. Also, the resulting steady state MSE is quite insensitive to input SNR. Furthermore, they are able to operate with noisy reference signals. Once initial convergence is achieved, usually within a few iterations, both the $LLMS_1$ and LLMS algorithms can maintain their operation through self-referencing. Furthermore, the resultant EVM values and scatter plots, obtained for operation in the presence of Rayleigh fading further demonstrate the superior performance of LLMS and $LLMS_1$ algorithms over the other three LMS-based algorithm schemes in a fast changing signal environment. From the study of the influence of the AOA of the interference on the desired signal at the

output of an adaptive beamformer, it is observed that the proposed LLMS algorithm schemes outperform the other three algorithms.

Based on the comparison, presented in Section 5.5, between the LLMS and RLMS algorithms, it is observed that the latter performs only slightly better in terms of convergence as well as suppression of the co-channel interference. Moreover, the LLMS algorithm is much less complex than the RLMS algorithm.

CHAPTER 6

PRACTICAL CONSIDERATIONS

6.1 Introduction

The proposed RLMS and LLMS algorithms have been described and analyzed in Chapter 4 and Chapter 5, respectively. Also, their performances when used in digital array beamforming have been investigated by means of extensive computer simulations with the various mathematical functions represented in full numerical precision. However, it is envisaged that finite numerical arithmetic will be used for real-time implementations of these algorithms. In addition, any practical implementation of an array beamformer is likely to encounter random error sources caused by tolerances in inter-element spacing and element gain. These error sources could affect the array gain, beam pattern and null depth at the interference direction in an unpredictable manner.

In this chapter, we begin by considering the effect on the operation of the RLMS and LLMS algorithms due to the use of finite precision arithmetic. This leads to the determination of the minimum word length, in terms of the number of binary bits, required to achieve an acceptable degradation in performance when compared with a full precision implementation. Following this, we will examine tolerances in inter-element spacing and element gain on the resulting beam pattern. Finally, the performance of a linear antenna array with two or four elements is investigated.

6.2 Finite Precision Arithmetic

In this section, the convergence behaviours of the RLMS and LLMS algorithms due to the use of finite numerical precision in their implementation are analyzed. Analytical expressions for the estimation of the quantization

error associated with the overall error signals, e_{RLMS} and e_{LLMS} for the RLMS and LLMS algorithms, respectively, have been derived.

In the following analysis, the same assumptions applied in Chapters 4 and 5 are also adopted here. These are:

- The propagation environment is time invariant.
- The components of the signal vector $\boldsymbol{X}(n)$ are independent and identically distributed (iid).
- All signals are zero mean and statistically stationary at least to the second order.

6.2.1 Error convergence of the RLMS algorithm

The RLMS algorithm is depicted in Figure 4-1. It shows that the overall error signal, e_{RLMS}, at the n^{th} iteration is given by equation (4.6), which is repeated here as

$$e_{\mathrm{RLMS}}(n) = e_{\mathrm{RLS}}(n) - e_{\mathrm{LMS}}(n-1) \tag{6.1}$$

where

$$e_{\mathrm{RLS}}(n) = d(n) - \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(n), \tag{6.2}$$

is the error signal of the RLS algorithm stage, while that of the LMS algorithm stage is

$$e_{\mathrm{LMS}}(n) = d(n) - \boldsymbol{W}_{\mathrm{LMS}}^{H}(n)\boldsymbol{X}_{\mathrm{LMS}}(n). \tag{6.3}$$

$\boldsymbol{X}(n)$ and $\boldsymbol{X}_{\mathrm{LMS}}(n)$ are the input signal vectors of the RLS and LMS algorithm stages, respectively, and $d(n)$ is the external reference signal. Substituting equations (6.2) and (6.3) into equation (6.1) gives

$$e_{\mathrm{RLMS}}(n) = d(n) - d(n-1) - \boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(n) + \boldsymbol{W}_{\mathrm{LMS}}^{H}(n-1)\boldsymbol{X}_{\mathrm{LMS}}(n-1) \tag{6.4}$$

Now, with the RLMS algorithm being implemented using finite numerical precision, the results of the various mathematical calculations will be affected by round-off and truncation errors. The influence of these errors on the operation of the RLMS algorithm is analyzed as follows.

First, the signal terms expressed in finite numerical precision are represented by primed symbols to differentiate them from their corresponding counterparts in full numerical precision. For example, the input signal and weight vectors in finite numerical precision can be expressed as

$$\boldsymbol{X}'(n) = \boldsymbol{X}(n) + \boldsymbol{\alpha}(n) \tag{6.5}$$

$$\boldsymbol{W}'(n) = \boldsymbol{W}(n) + \boldsymbol{\rho}(n) \tag{6.6}$$

where $\boldsymbol{\alpha}(n)$ and $\boldsymbol{\rho}(n)$ are the corresponding quantization error vectors. The elements of $\boldsymbol{\alpha}(n)$ and $\boldsymbol{\rho}(n)$ are assumed to be independent of $\boldsymbol{X}(n)$ and $\boldsymbol{W}(n)$ respectively, and both are white sequences with zero mean and variance of σ_q^2. For a signal of $\pm 1\ \mathrm{V}$ amplitude range represented by an N_b–bit wordlength, the resulting variance of the quantization error is given by (see Appendix F) [141]

$$\sigma_q^2 = 2^{2(1-N_b)}/12 \tag{6.7}$$

Substituting equations (6.5) and (6.6) in equation (6.1), we obtain the overall error signal in finite numerical precision, e'_{RLMS}, as

$$\begin{aligned} e'_{\mathrm{RLMS}}(n) = {} & d(n) - d(n-1) - \boldsymbol{W}_{\mathrm{RLS}}^H(n)\boldsymbol{X}(n) + \boldsymbol{W}_{\mathrm{LMS}}^H(n-1)\boldsymbol{X}_{\mathrm{LMS}}(n-1) \\ & - \boldsymbol{\rho}_{\mathrm{RLS}}^H(n)\boldsymbol{X}(n) + \boldsymbol{\rho}_{\mathrm{LMS}}^H(n-1)\boldsymbol{X}_{\mathrm{LMS}}(n-1) - \boldsymbol{W}_{\mathrm{RLS}}^H(n)\boldsymbol{\alpha}_{\mathrm{RLS}}(n) \\ & + \boldsymbol{W}_{\mathrm{LMS}}^H(n-1)\boldsymbol{\alpha}_{\mathrm{LMS}}(n-1) - \eta_{\mathrm{RLS}}(n) + \eta_{\mathrm{LMS}}(n-1) \end{aligned} \tag{6.8}$$

where $\eta_{\mathrm{RLS}}(n)$ and $\eta_{\mathrm{LMS}}(n-1)$ are the truncation and round-off errors associated with the RLS and LMS algorithm stages, respectively. Both $\eta_{\mathrm{RLS}}(n)$ and $\eta_{\mathrm{LMS}}(n-1)$ have approximately the same variance, i.e.,

$$\sigma_\eta^2 = c_q \sigma_q^2, \tag{6.9}$$

where the constant c_q depends on how the inner product of a vector manipulation is implemented. In our case, the inner product is performed with both the signal and weights vectors quantized. In this case, $c_q = N$, where N is the number of array elements [141].

Since the first four terms on the right hand side of equation (6.8) are the same as those for e_{RLMS} given in equation (6.4), we can rewrite equation (6.8) as

$$\begin{aligned} e'_{\mathrm{RLMS}}(n) = e_{\mathrm{RLMS}}(n) &- \boldsymbol{\rho}^H_{\mathrm{RLS}}(n)\boldsymbol{X}(n) + \boldsymbol{\rho}^H_{\mathrm{LMS}}(n-1)\boldsymbol{X}_{\mathrm{LMS}}(n-1) \\ &- \boldsymbol{W}^H_{\mathrm{RLS}}(n)\boldsymbol{\alpha}_{\mathrm{RLS}}(n) + \boldsymbol{W}^H_{\mathrm{LMS}}(n-1)\boldsymbol{\alpha}_{\mathrm{LMS}}(n-1) \\ &- \eta_{\mathrm{RLS}}(n) + \eta_{\mathrm{LMS}}(n-1) \end{aligned} \tag{6.10}$$

Rearranging equation (6.10) and performing the expectation on both sides yields

$$\begin{aligned} \mathrm{E}\left[e'_{\mathrm{RLMS}}(n) - e_{\mathrm{RLMS}}(n)\right] = \mathrm{E}\Big[&\boldsymbol{\rho}^H_{\mathrm{LMS}}(n-1)\boldsymbol{X}_{\mathrm{LMS}}(n-1) - \boldsymbol{\rho}^H_{\mathrm{RLS}}(n)\boldsymbol{X}(n) \\ &- \boldsymbol{W}^H_{\mathrm{RLS}}(n)\boldsymbol{\alpha}_{\mathrm{RLS}}(n) + \boldsymbol{W}^H_{\mathrm{LMS}}(n-1)\boldsymbol{\alpha}_{\mathrm{LMS}}(n-1) \\ &- \eta_{\mathrm{RLS}}(n) + \eta_{\mathrm{LMS}}(n-1)\Big] \end{aligned} \tag{6.11}$$

Based on the assumptions used in (6.5) and equations (6.6), $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$ are independent of $\boldsymbol{X}$ and $\boldsymbol{W}$, and they are white sequences with zero mean. As such, equation (6.11) can be simplified to become

$$\begin{aligned} \mathrm{E}\left[e'_{\mathrm{RLMS}}(n) - e_{\mathrm{RLMS}}(n)\right] &= \mathrm{E}\left[\eta_{\mathrm{LMS}}(n-1) - \eta_{\mathrm{RLS}}(n)\right] \\ &= \eta_{\mathrm{LMS}}(n-1) - \eta_{\mathrm{RLS}}(n) \end{aligned} \tag{6.12}$$

From equation (6.12), it is noted that the overall error, e'_{RLMS}, converges to a value larger than e_{RLMS} by an amount given by the difference between the truncation and round-off error of the two algorithm stages. Also, η_{RLS} and η_{LMS} are approximately the same. Under these conditions, the expectation of

the difference between e'_{RLMS} and e_{RLMS} given in equation (6.12) will tend to be zero. This observation has been verified by the simulation results as shown in Figure 6-3 for a wordlength greater than 8 bits. However, for lower numerical precision, the quantizing step becomes larger and this gives rise to a larger difference between e'_{RLMS} and e_{RLMS}.

6.2.2 Error convergence of the LLMS algorithm

In the case of the LLMS algorithm, as depicted in Figure 5.1, the overall error signal, e_{LLMS}, is given by equation (5.1), which is repeated here as

$$e_{\mathrm{LLMS}}(n) = e_1(n) - e_2(n-1) \tag{6.13}$$

with

$$e_i(n) = d(n) - \boldsymbol{W}_i^H(n)\boldsymbol{X}_i(n) \tag{6.14}$$

where $\boldsymbol{X}_i(\cdot)$ and $\boldsymbol{W}_i(\cdot)$ represent the input signal and weight vectors of the i^{th} LMS algorithm stage, respectively. The terms associated with the first and second LMS algorithm stages are denoted with the subscripts $i=1$ and $i=2$, respectively. After obtaining $e_1(n)$ and $e_2(n-1)$ from equation (6.14), we can substitute them into equation (6.13) to yield

$$e_{\mathrm{LLMS}}(n) = d(n) - d(n-1) - \boldsymbol{W}_1^H(n)\boldsymbol{X}_1(n) + \boldsymbol{W}_2^H(n-1)\boldsymbol{X}_2(n-1) \tag{6.15}$$

Now, the same notations for the input signal and weight vectors in finite numerical precision, as expressed in equations (6.5) and (6.6) respectively, are adopted here. By substituting equations (6.5) and (6.6) in equation (6.13), we can obtain the overall error signal in finite numerical precision, $e'_{\mathrm{LLMS}}(n)$, for the LLMS algorithm as given by

$$\begin{aligned} e'_{\mathrm{LLMS}}(n) = {} & d(n) - d(n-1) - \boldsymbol{W}_1^H(n)\boldsymbol{X}_1(n) + \boldsymbol{W}_2^H(n-1)\boldsymbol{X}_2(n-1) \\ & - \boldsymbol{\rho}_1^H(n)\boldsymbol{X}_1(n) + \boldsymbol{\rho}_2^H(n-1)\boldsymbol{X}_2(n-1) - \boldsymbol{W}_1^H(n)\boldsymbol{\alpha}_1(n) \\ & + \boldsymbol{W}_2^H(n-1)\boldsymbol{\alpha}_2(n-1) - \eta_1(n) + \eta_2(n-1) \end{aligned} \tag{6.16}$$

where $\boldsymbol{\rho}_1(n)$, $\boldsymbol{\rho}_2(n)$, $\boldsymbol{\alpha}_1(n)$, $\boldsymbol{\alpha}_2(n-1)$, $\eta_1(n)$ and $\eta_2(n-1)$ are as previously defined in equations (6.5), (6.6) and (6.8).

As the first four terms on the right hand side of equation (6.16) are the same as those for e_{LLMS} given in equation (6.15), we can rewrite equation (6.16) as

$$\begin{aligned} e'_{\mathrm{LLMS}}(n) = e_{\mathrm{LLMS}}(n) + \boldsymbol{\rho}_2^H(n-1)\boldsymbol{X}_2(n-1) - \boldsymbol{\rho}_1^H(n)\boldsymbol{X}_1(n) \\ -\boldsymbol{W}_1^H(n)\boldsymbol{\alpha}_1(n) + \boldsymbol{W}_2^H(n-1)\boldsymbol{\alpha}_2(n-1) - \eta_1(n) + \eta_2(n-1) \end{aligned} \tag{6.17}$$

After rearranging equation (6.17) and performing the expectation on both sides of the equation, we obtain

$$\begin{aligned} \mathrm{E}\left[e'_{\mathrm{LLMS}}(n) - e_{\mathrm{LLMS}}(n)\right] = \mathrm{E}\Big[\boldsymbol{\rho}_2^H(n-1)\boldsymbol{X}_2(n-1) - \boldsymbol{\rho}_1^H(n)\boldsymbol{X}_1(n) \\ -\boldsymbol{W}_1^H(n)\boldsymbol{\alpha}_1(n) + \boldsymbol{W}_2^H(n-1)\boldsymbol{\alpha}_2(n-1) - \eta_1(n) + \eta_2(n-1)\Big] \end{aligned} \tag{6.18}$$

Based on the same considerations in the assumptions used in (6.5) and (6.6) discussed in Section 6.2.1, the expectation value for the RHS of equation (6.18) will also approach to zero when the LLMS algorithm has converged. Again, this observation is verified by the simulation results plotted in Figure 6-3 for a numerical precision equivalent to equal or better than 8 bits. When a smaller wordlength is used, the expectation of the difference between e'_{RLMS} and e_{RLMS} follows the same trend as that of the RLMS algorithm as shown in Figure 6-3.

6.2.3 Performance evaluation by computer simulation

In Chapters 4 and 5, the operations of the RLMS and LLMS beamforming algorithms have been evaluated under various channel conditions, including in the presence of additive white Gaussian noise and cochannel interference. These earlier results are based on the algorithms being implemented with full numerical precision. In this section, we examine the influence on the

performance of the RLMS and LLMS algorithms when they are implemented using finite precision arithmetic. The performance evaluation includes the two modes of operation of the algorithm, namely operating with an adaptive array vector $\mathcal{F}$ (referred to as the RLMS, or LLMS algorithm), as well as with a fixed prescribed $\mathcal{F}$ (referred to as the $RLMS_1$ or $LLMS_1$ algorithm).

An adaptive uniform linear array consisting of eight isotropic antenna elements spaced half a wavelength apart is simulated. To emulate the adaptation process of the algorithm operating with finite numerical precision, the quantization models of Figure 6-1 and Figure 6-2 have been adopted for the RLMS and LLMS algorithms, respectively. In each case, the input signal vector is first normalized with the largest component taking on an amplitude range of ± 1 Volt in order to make full use of a given word length. Furthermore, the inputs to every arithmetic function, such as multiplication and addition, are expressed with the same specified numerical precision. The result of each arithmetical operation is then rounded to the specified wordlength.

6.2.3.1 MSE performance obtained with a finite wordlength

In an attempt to determine an acceptable numerical precision required for implementing the RLMS and LLMS algorithms, we consider how their convergence behaviours are affected through the use of a different wordlength in a noise free condition. First, the values of MSE of the overall error signals, (i.e., e'_{RLMS}, obtained with the RLMS or $RLMS_1$ algorithm, and e'_{LLMS} associated with the LLMS and $LLMS_1$ algorithms), for a given wordlength are obtained from computer simulation. These MSE values are obtained after 1024 iterations to ensure complete convergence of the algorithm. The values of the step sizes used for simulating the RLMS, $RLMS_1$, $LLMS_1$ and LLMS algorithms are tabulated in Table 6-1 for operation with wordlength ranging from 7 to 12 bits. Moreover, when operating with a wordlength of 6 bits, lower values of MSE could be achieved by adjusting somewhat the step size values for the RLMS, $RLMS_1$, $LLMS_1$ and LLMS algorithms as tabulated in Table 6-2.

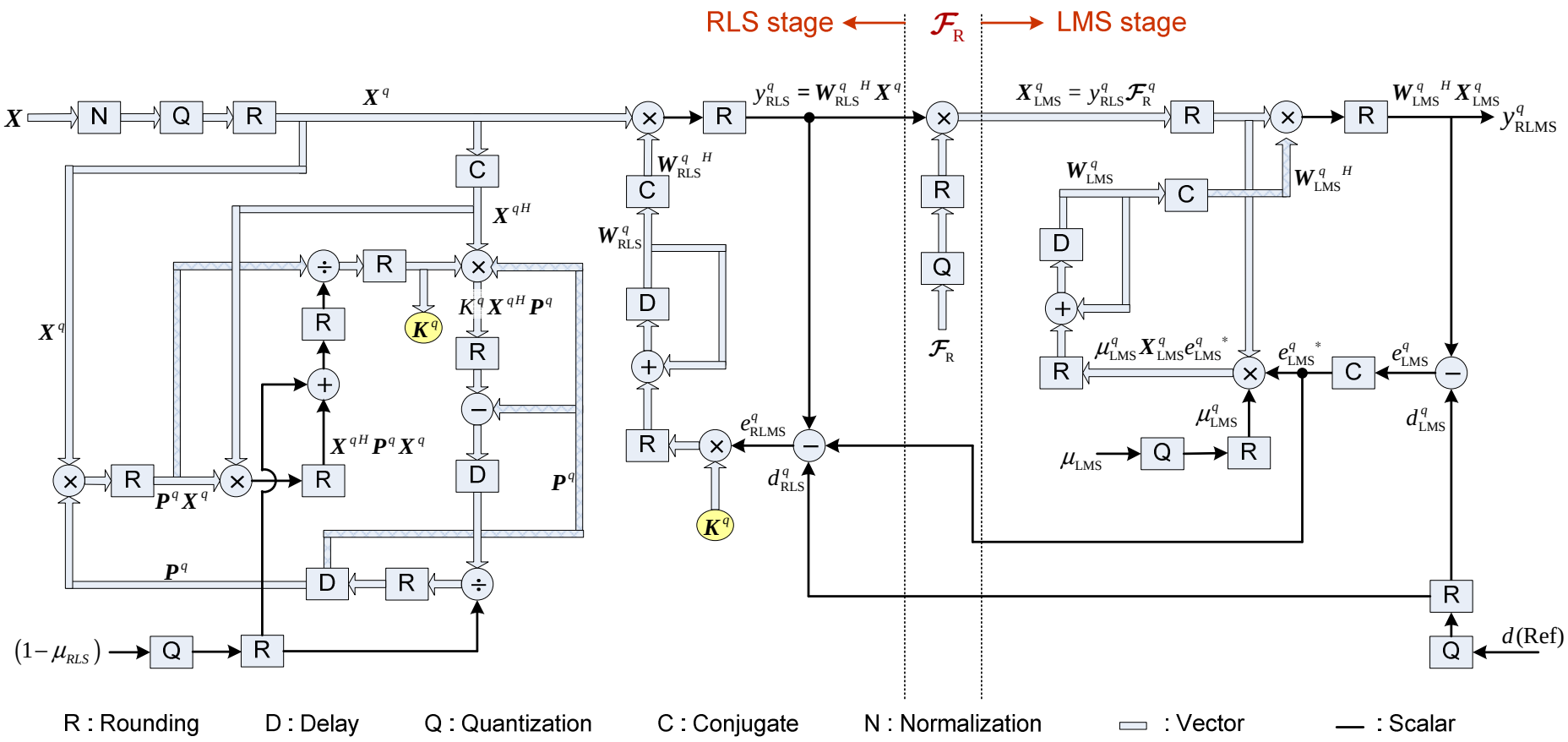


Figure 6-1 Quantization model used for evaluating the RLMS algorithm with finite numerical precision.

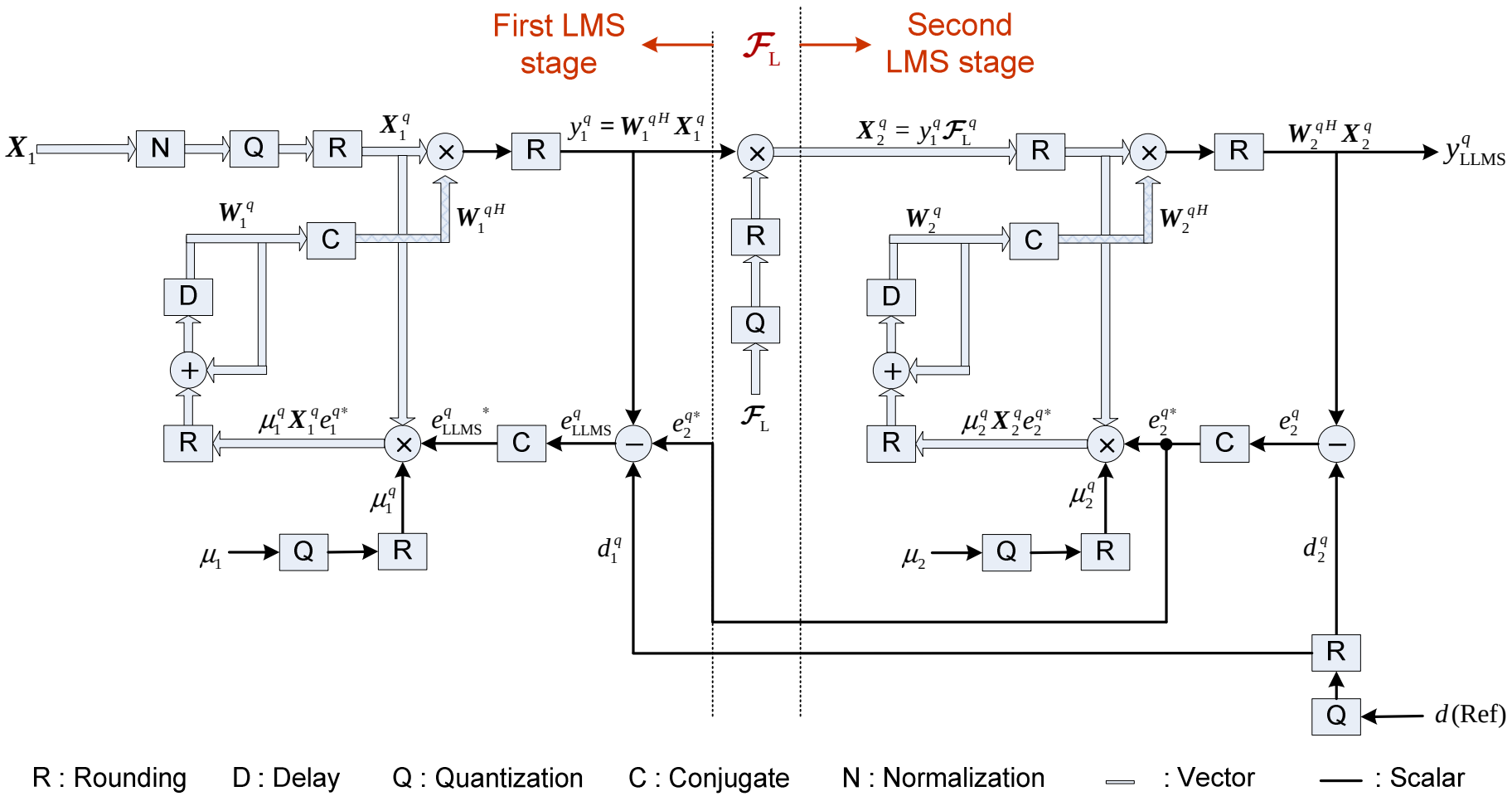


Figure 6-2 Quantization model used for evaluating the LLMS algorithm with finite numerical precision.

Table 6-1 Values of the constants adopted for operation with the wordlength, N_B, in the range of 7 to 12 bits.

Algorithm	Parameters for operation in a noise free channel
$RLMS_1$	$\mu_{RLS} = 0.01,\ \mu_{LMS} = 0.08$
RLMS	$\mu_{RLS} = 0.01,\ \mu_{LMS} = 0.15$
$LLMS_1$	$\mu_1 = 0.2,\ \mu_2 = 0.1$
LLMS	$\mu_1 = 0.2,\ \mu_2 = 0.3$

The MSE values obtained when the RLMS, $RLMS_1$, $LLMS_1$ and LLMS algorithms are operating with wordlengths N_b ranging from 6 to 12 bits are plotted in Figure 6-3. From the results, it is observed that a minimum numerical precision equivalent to a wordlength of 8 bits is considered adequate to yield an acceptable MSE performance for the RLMS, $RLMS_1$, $LLMS_1$ and LLMS algorithms.

Table 6-2 Values of the constants adopted for operation with a 6-bit wordlength.

Algorithm	Noise Free Channel
$RLMS_1$	$\mu_{RLS} = 0.01,\ \mu_{LMS} = 0.12$
RLMS	$\mu_{RLS} = 0.01,\ \mu_{LMS} = 0.15$
$LLMS_1$	$\mu_1 = 0.2,\ \mu_2 = 0.05$
LLMS	$\mu_1 = 0.2,\ \mu_2 = 0.1$

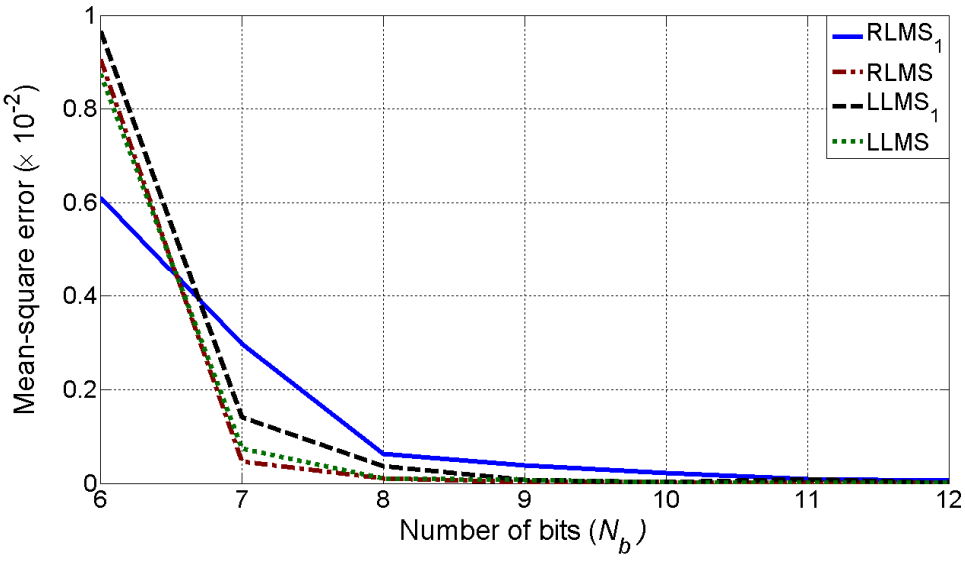


Figure 6-3 Residual MSE as a function of the wordlength used to implement the RLMS, $RLMS_1$, LLMS and $LLMS_1$ algorithms in a noise free channel.

The theoretical overall error signals, e'_{RLMS}, for the RLMS and $RLMS_1$ algorithms have been computed from equation (6.8) for a wordlength of 8 bits. These are plotted in Figure 6-4. For comparison, the overall error signals, e_{RLMS}, computed from equation (6.4) with full numerical precision are also plotted in Figure 6-4 for the two versions of the RLMS algorithm. It is observed that the values of e'_{RLMS} are only slightly larger than the corresponding e_{RLMS} values during their transition to convergence. The difference becomes insignificant beyond 30 iterations. This confirms the asymptotic behaviour of the RLMS algorithm operating with finite numerical precision, as predicted by equation (6.8). Similar observations can be made for the LLMS and $LLMS_1$ algorithms by referring to Figure 6-5. In the case of the LLMS algorithm, the theoretical values for e'_{LLMS} based on 8-bit numerical precision, and e_{LLMS} are computed using equation (6.16) and (6.15), respectively. Figure 6-4 and Figure 6-5 show that the convergence speeds of the RLMS and LLMS algorithms achieved with an 8-bit precision are only

marginally slower than the version implemented with full precision. Also, the RLMS algorithm convergences slightly faster than the of LLMS algorithm.

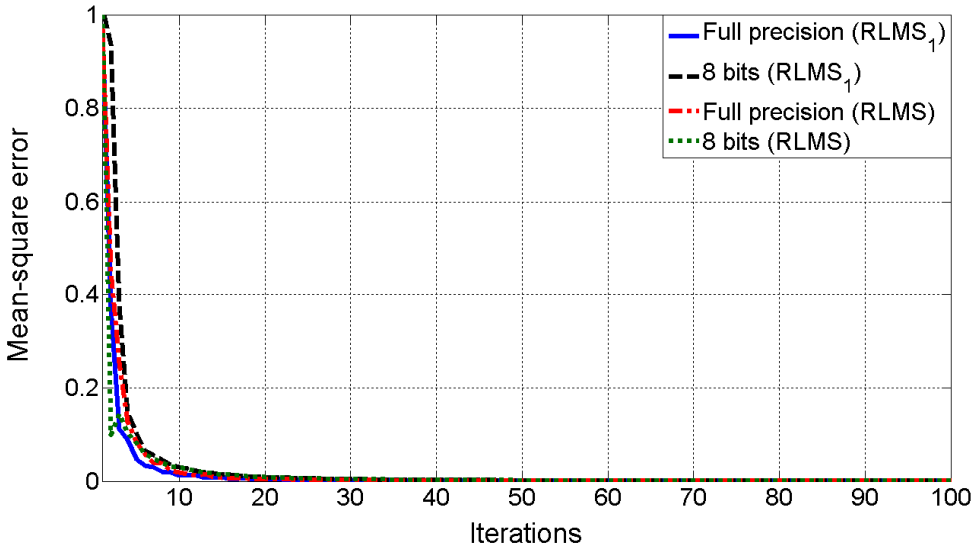


Figure 6-4 The theoretical values of MSE of the RLMS and RLMS$_1$ algorithms obtained with full numerical precision and 8-bit precision.

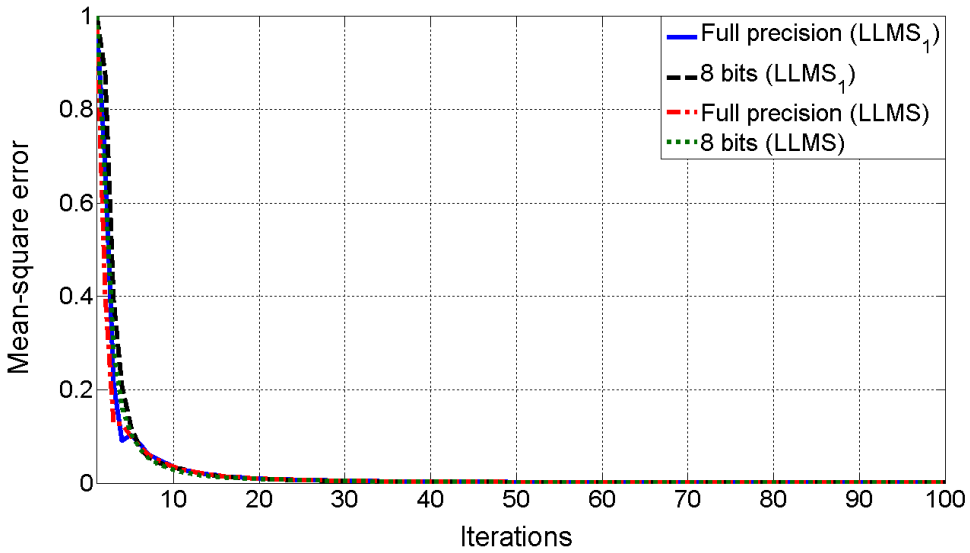


Figure 6-5 The theoretical values of MSE of the LLMS and LLMS$_1$ algorithms obtained with full numerical precision and 8-bit precision.

Next, the rates of convergence of the RLMS, $RLMS_1$, LLMS and $LLMS_1$ algorithms have been simulated using an 8-bit wordlength. The resulting curves are plotted as shown in Figure 6-6. The simulated results compare well with the theoretical curve presented in Figure 6-4 for the RLMS and $RLMS_1$ algorithms, and in Figure 6-5 for the LLMS and $LLMS_1$ algorithms. It may be observed from Figure 6-6 that the RLMS, $RLMS_1$, $LLMS_1$ and LLMS algorithms converge much quicker than the other four algorithms, which have also been implemented with the same numerical precision.

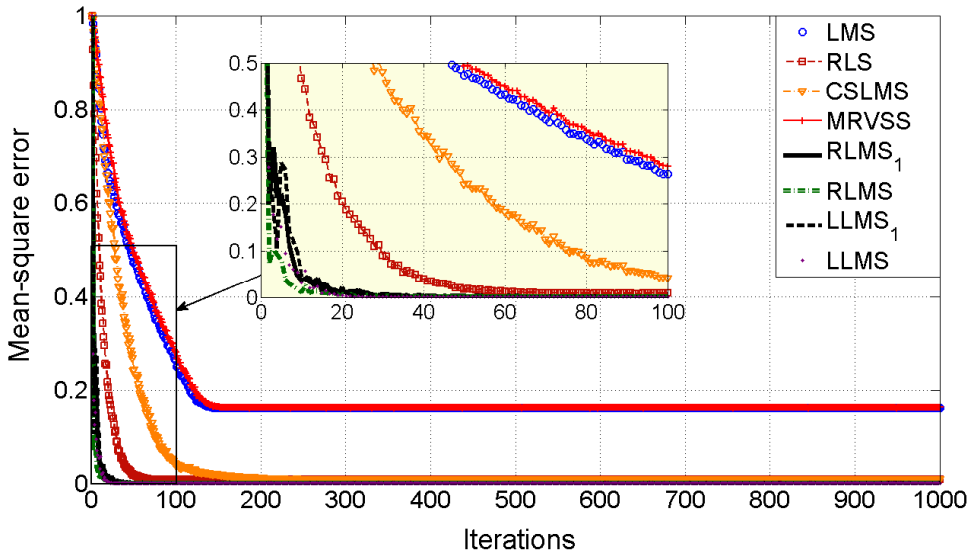


Figure 6-6 The rates of convergence of the RLMS, $RLMS_1$, LLMS, $LLMS_1$, CSLMS, MRVSS, RLS and LMS algorithms based on 8-bit precision.

6.2.3.2 EVM and scatter plot

The influence of finite numerical precision on the fidelity of the received signal is investigated based on the EVM as expressed in equation (4.86). The EVM values are calculated after 1024 iterations to make sure that final convergence is achieved for a given algorithm. This has been carried out for the $RLMS_1$, RLMS, $LLMS_1$, LLMS, LMS, RLS, MRVSS and CSLMS algorithms with a different precision ranging from 6 to 12 bits. The values of

the parameters used for simulating the RLS, LMS, CSLMS and MRVSS algorithms are tabulated in Table 6-3. The results are plotted in Figure 6-7, which clearly shows that the proposed RLMS, $RLMS_1$, $LLMS_1$ and LLMS algorithms can cope better with the use of finite precision among the eight schemes considered.

Table 6-3 Values of the constants adopted for operation with the wordlength, N_B for RLS, LMS, CSLMS and MRVSS algorithms.

Algorithm	**Parameters for operation in a noise free channel**
RLS	$\mu = 0.01$
LMS	$\mu = 0.02$
CSLMS	$\varepsilon_{CS} = 0.02,\ \mu = 0.02$
MRVSS	$\beta_{\max} = 1,\ \beta_{\min} = 0,\ \upsilon = 5\times10^{-4},\ \mu_{\max} = 0.2,\ \mu_{\min} = 10^{-4}$ Initial $\mu = \mu_{\max},\ \alpha = 0.97,\ \gamma = 4.8\times10^{-4},\ \eta = 0.97$

To demonstrate how well the signal fidelity is retained, the scattered plots of the recovered BPSK signal obtained for the eight algorithms, based on an 8-bit implementation, are shown in Figure 6-8. Each of these scatter plots is obtained from 5012 signal samples after the convergence of a given algorithm. Ideally, a BPSK signal has only two states, namely, -1 and +1. It is observed that the use of finite precision is causing spreading of these two states. Among the eight algorithms considered, the scattered plots of the RLMS, $RLMS_1$, LLMS and $LLMS_1$ algorithms show the least spreading. This observation is verified by the low values of EVM achieved with these four algorithms.

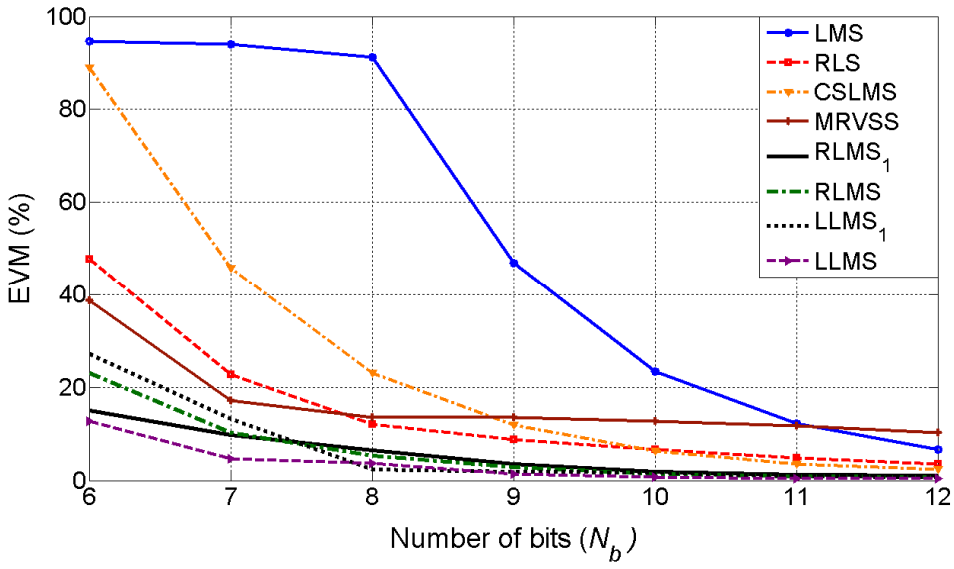
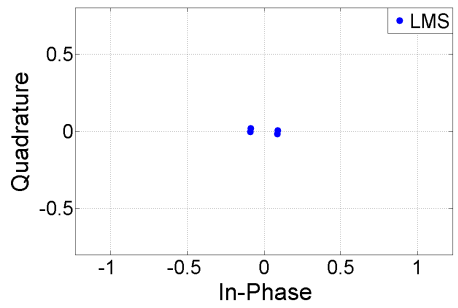


Figure 6-7 The EVM values of the RLMS$_1$, RLMS, LLMS$_1$, LLMS , LMS, CSLMS and MRVSS algorithms implemented with different wordlengths.

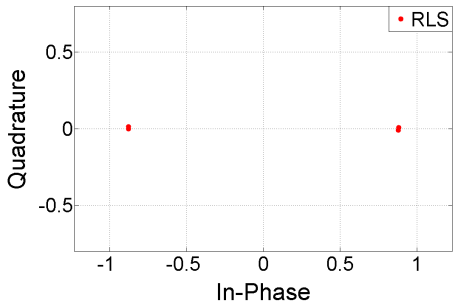
6.2.3.3 Beam pattern performance

Figure 6-9 and Figure 6-10 show the beam patterns obtained through the use of the RLMS$_1$, RLMS, LLMS$_1$, LLMS, LMS, RLS, CSLMS and MRVSS algorithms implemented with 8-bit and 9-bit accuracy, respectively. For this simulation, the desired signal arrives at an angle of 10°, while the two cochannel interfering signals of equal amplitude as the desired signal are coming from -30° and 45°.

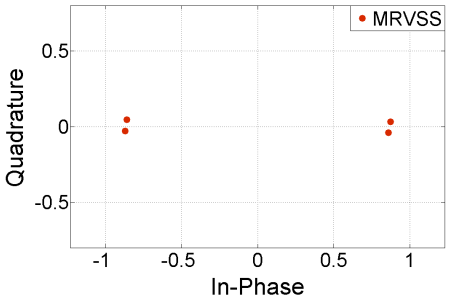
It is observed that all the eight algorithms achieve almost the same gain in the direction of the desired signal when implemented with 9-bit precision. The gain difference is only barely noticeable when the algorithms are implemented with 8-bit precision. Moreover, the RLMS, RLMS$_1$, LLMS$_1$ and LLMS algorithms provide greater rejection to the interfering signals arriving at an angle (θ_i) of -30° and 45°. This suggests that the use of 8-bit precision is sufficient to maintain the effectiveness of these four proposed algorithms in rejecting interfering signals emanating from outside their main lobes.



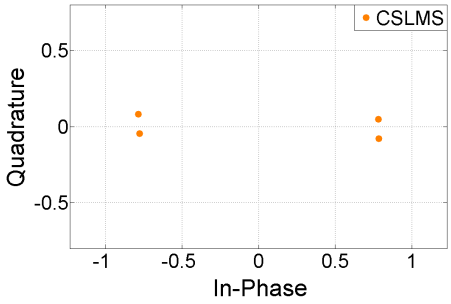
(a) LMS algorithm



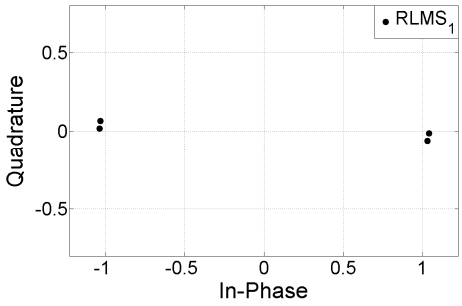
(b) RLS algorithm



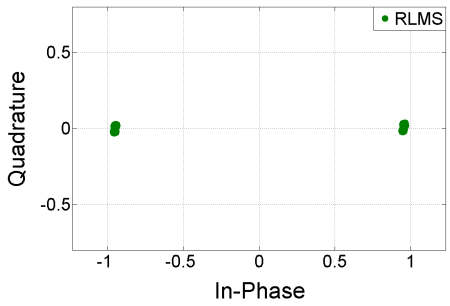
(c) MRVSS algorithm



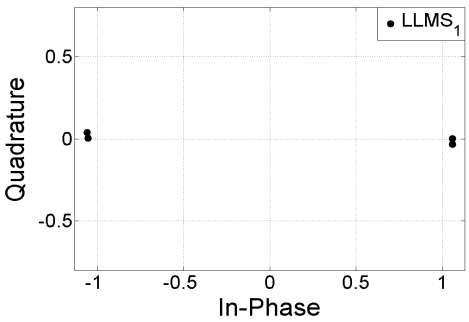
(d) CSLMS algorithm



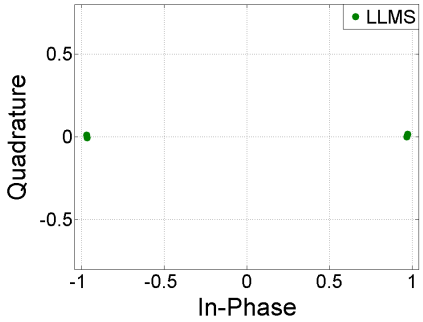
(e) $RLMS_1$ algorithm



(f) RLMS algorithm



(g) $LLMS_1$ algorithm



(h) LLMS algorithm

Figure 6-8 The scatter plots of the recovered BPSK signal obtained with all the eight algorithms being implemented in 8-bit precision.

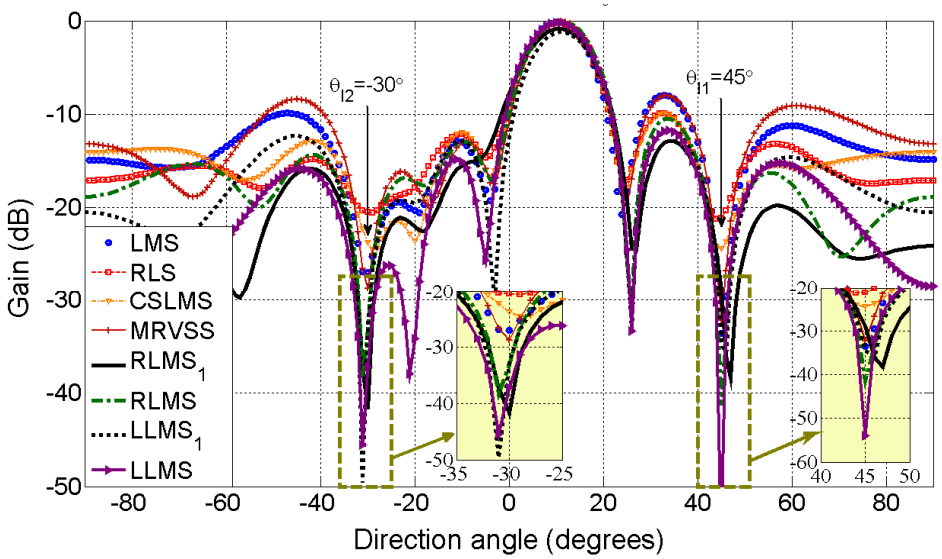


Figure 6-9 The beam patterns obtained with the LMS, RLS, CSLMS, MRVSS, $RLMS_1$, RLMS, $LLMS_1$ and LLMS algorithms using an 8-bit wordlength.

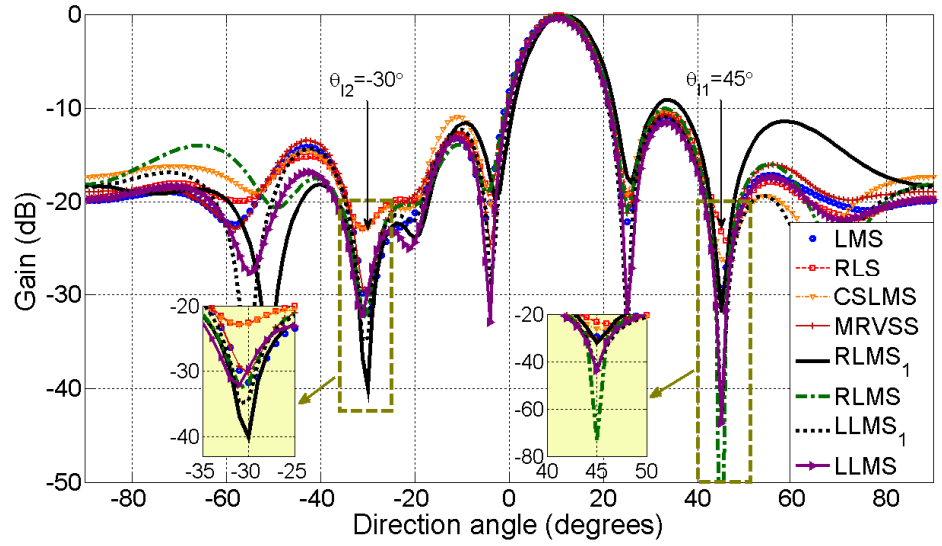


Figure 6-10 The beam patterns obtained with the LMS, RLS, CSLMS, MRVSS, $RLMS_1$, RLMS, $LLMS_1$ and LLMS algorithms using a 9-bit wordlength.

6.3 Tolerances in Array Element Spacing and Gain

As discussed in Section 4.2, the antenna array vector of size N is expressed as

$$\boldsymbol{A}_d = G[1, e^{-j\psi_d}, e^{-2j\psi_d}, ..., e^{-(N-1)j\psi_d}]^T \tag{6.19}$$

where

$$\psi_d = 2\pi\left[\frac{\mathcal{D}\sin(\theta_d)}{\lambda}\right] \tag{6.20}$$

with $\mathcal{D}$ being the array inter-element spacing, λ is the carrier wavelength and G is the array element gain.

Ideally, both the inter-element spacing, $\mathcal{D}$, and gain, G, are assumed to be uniform among all the individual array elements. However, variations in $\mathcal{D}$ and G between elements are likely to be introduced due to implementation tolerances. The effects of these variations on the performance of a beamforming algorithm are investigated in this section.

Let the maximum tolerance in the inter-element spacing of the array be restricted to $\pm\Upsilon_{\max}$. Now, with the first element acting as the reference, we represent the element spacing and gain as shown in Figure 6-11. In this case, the values of the element spacing can be expressed as

$$\mathcal{D}(i) = i\mathcal{D} + \Upsilon \tag{6.21}$$

where $i = 1, 2, ..., N-1$, and Υ takes on a random value in the range of $-\Upsilon_{\max} \le \Upsilon \le \Upsilon_{\max}$, with $\Upsilon_{\max}$ being the maximum tolerance.

In a similar manner, the gains of the individual elements can be expressed as

$$G(i) = G \times (1 + g_r) \tag{6.22}$$

where $i = 1, 2, ..., N$, and g_r takes on a random value in the range of $-g_{\max} \le g_r \le g_{\max}$, with $g_{\max}$ being the maximum deviation in gain with respect to the nominal gain G.

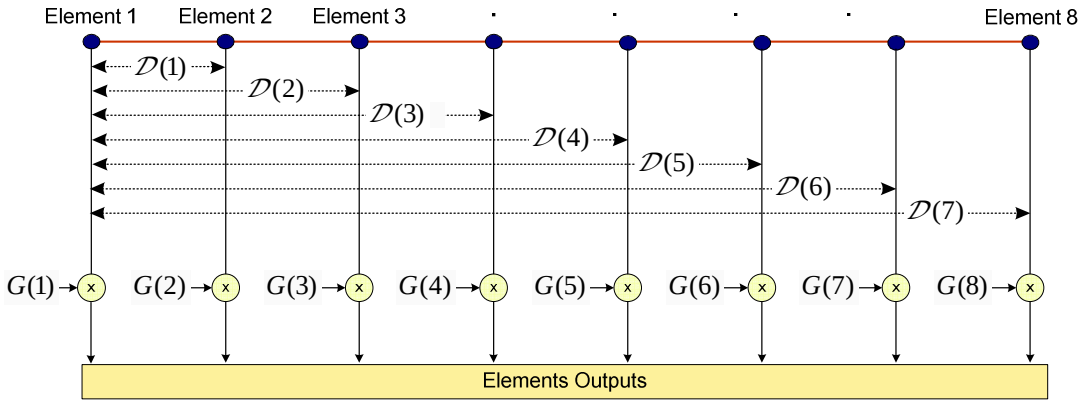


Figure 6-11 Modelling tolerances in inter-element spacing and element gain for an 8-element array.

After accounting for tolerances in the element spacing and gain, the antenna array vector of size N is given by

$$\boldsymbol{A}_d = \left[G(1),\, G(2)e^{-j2\pi\left[\mathcal{D}(1)\sin(\theta_d)/\lambda\right]},\, G(3)e^{-j2\pi\left[\mathcal{D}(2)\sin(\theta_d)/\lambda\right]}, \right. \\ \left. ..... ,\, G(N)e^{-j2\pi\left[\mathcal{D}(N-1)\sin(\theta_d)/\lambda\right]} \right]^T \tag{6.23}$$

6.3.1 Effects of tolerance in the array inter-element spacing

The influence of tolerance in the inter-element spacing on the resulting EVM and beam pattern of an adaptive beamformer is investigated. In this experiment, the EVM values and beam pattern are obtained in a noise free channel after 1200 iterations to ensure complete convergence of a given

beamforming algorithm. The maximum tolerance, $\Upsilon_{\max}$, for each element is assumed to be ±10% of the nominal inter-element spacing, $\mathcal{D}$, which is taken to be half a carrier wavelength. The investigation is carried out for the proposed RLMS$_1$, RLMS, LLMS$_1$, LLMS algorithms. For comparison purposes, we also consider the RLS and CSLMS algorithms. In each case, the desired signal is assumed to arrive at an angle, θ_d, of 10°, and a cochannel interfering signal of equal amplitude as the desired signal is coming from an angle, θ_i, of 45°. Table 6-4 shows the values of the various constants adopted for the computer simulations involving five different scenarios of tolerance in the element spacing as tabulated in Table 6-5.

Table 6-4 Values of the constants adopted for operation with array spacing and gain tolerances.

Algorithm	**Noise Free Channel**
RLS	$\mu = 0.01$
CSLMS	$\varepsilon_{cs} = 0.02,\ \mu = 0.05$
RLMS$_1$	$\mu_{\text{RLS}} = 0.01,\ \mu_{\text{LMS}} = 0.05$
RLMS	$\mu_{\text{RLS}} = 0.01,\ \mu_{\text{LMS}} = 0.25$
LLMS$_1$	$\mu_1 = 0.5,\ \mu_2 = 0.05$
LLMS	$\mu_1 = 0.5,\ \mu_2 = 0.25$

The array vector for each case of tolerance in the inter-element spacing is calculated using equation (6.23). Case 1 corresponds to the scenario where each element is precisely spaced half a wavelength apart, and is used here as the reference for performance comparison. The second and third cases represent random spacing deviation within the specified range encountered by each individual element. The last two cases may be considered as the worst case scenario with the adjacent elements, each experiencing

maximum tolerance in the opposite direction. The resulting EVM values and beam patterns obtained for each of the five cases of tolerance in the element spacing are plotted in Figure 6-12(a-e) and Figure 6-13(a-e), respectively.

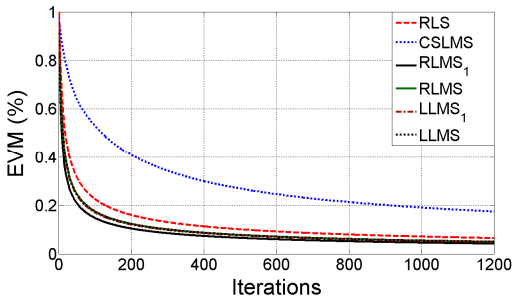
Table 6-5 Five different scenarios of tolerances in the inter-element spacing.

Cases	**Element Spacing**						
	$\mathcal{D}(1)$	$\mathcal{D}(2)$	$\mathcal{D}(3)$	$\mathcal{D}(4)$	$\mathcal{D}(5)$	$\mathcal{D}(6)$	$\mathcal{D}(7)$
1	$\mathcal{D}$	$2\mathcal{D}$	$3\mathcal{D}$	$4\mathcal{D}$	$5\mathcal{D}$	$6\mathcal{D}$	$7\mathcal{D}$
2	$0.93\mathcal{D}$	$1.92\mathcal{D}$	$3\mathcal{D}$	$4.1\mathcal{D}$	$5.03\mathcal{D}$	$6.02\mathcal{D}$	$6.94\mathcal{D}$
3	$1.1\mathcal{D}$	$2.03\mathcal{D}$	$3.02\mathcal{D}$	$4\mathcal{D}$	$4.94\mathcal{D}$	$5.93\mathcal{D}$	$6.9\mathcal{D}$
4	$0.9\mathcal{D}$	$2.1\mathcal{D}$	$2.9\mathcal{D}$	$4.1\mathcal{D}$	$4.9\mathcal{D}$	$6.1\mathcal{D}$	$6.9\mathcal{D}$
5	$1.1\mathcal{D}$	$1.9\mathcal{D}$	$3.1\mathcal{D}$	$3.9\mathcal{D}$	$5.1\mathcal{D}$	$5.9\mathcal{D}$	$7.1\mathcal{D}$

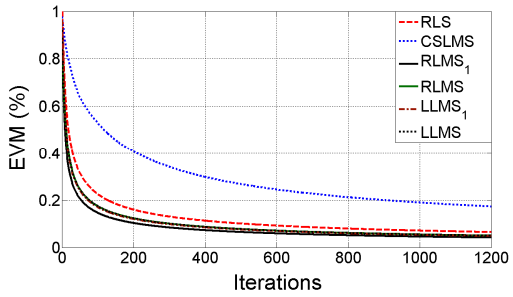
From Figure 6-12, it is observed that the EVM performance of each of the six beamforming algorithms considered is hardly affected by the presence of a maximum ±10% tolerance in the element spacing. This is due to the fact that the only source of the cochannel interference, which in this case is arriving from an angle, $\theta_i = 45^{\circ}$, has been greatly suppressed by the null response correctly produced by each of the four beamformers, as shown in Figure 6-13.

Also, according to the beam patterns as shown in Figure 6-13, the array is still able to direct its main beam towards the correct direction of the desired signal with a null response at the direction of the interfering signal. However, the sidelobes of the array are affected somewhat by the displacements in the locations of the various elements. As a result, the sidelobe suppression is reduced especially for responses at the two ends of the array. The same observations hold true for all the six algorithms considered.

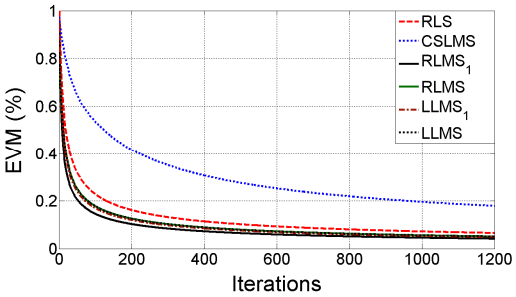
(a) Case 1 – Precise inter-element spacing



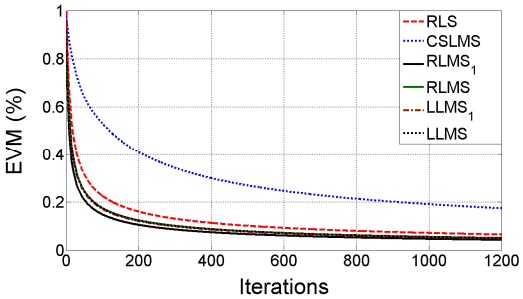
(b) Case 2 – Random spacing deviation (i)



(c) Case 3 – Random spacing deviation (ii)



(d) Case 4 –Worst case spacing deviation (i)



(e) Case 5 – worst case spacing deviation (ii)

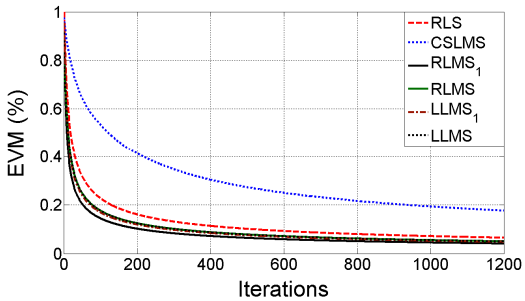
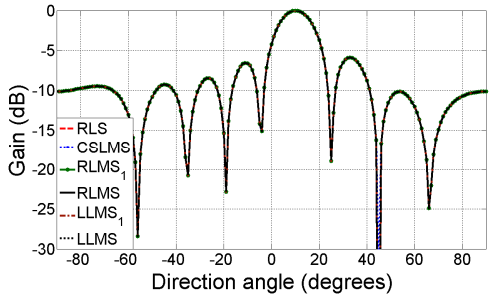
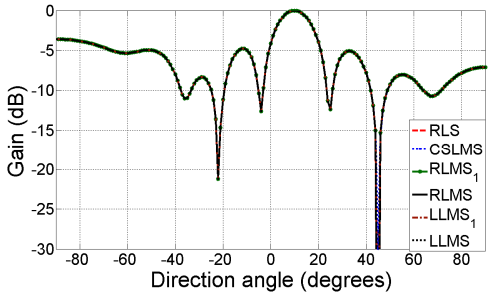


Figure 6-12 The EVM values obtained with the RLS, CSLMS, $RLMS_1$, RLMS, $LLMS_1$ and LLMS algorithms under five different scenarios of tolerance in the inter-element spacing. The maximum allowable tolerance, $\Upsilon_{\max}$, is ±10% of the nominal inter-element spacing, $\mathcal{D}$.

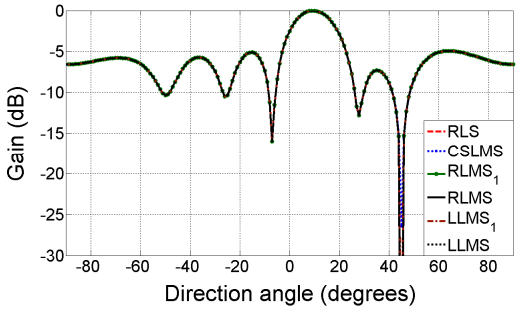
(a) Case 1 – Precise inter-element spacing



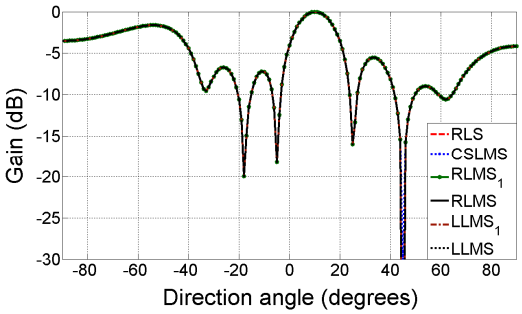
(b) Case 2 – Random spacing deviation (i)



(c) Case 3 – Random spacing deviation (ii)



(d) Case 4 –Worst case spacing deviation (i)



(e) Case 5 – worst case spacing deviation (ii)

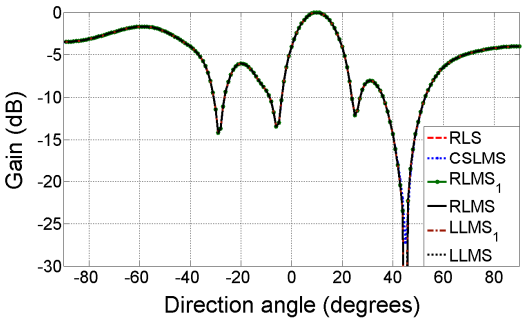


Figure 6-13 The beam pattern obtained with the RLS, CSLMS, $RLMS_1$, RLMS, $LLMS_1$ and LLMS algorithms under five different scenarios of tolerance in the inter-element spacing. The maximum allowable tolerance, Υ_{max}, is ±10% of the nominal inter-element spacing, $\mathcal{D}$.

6.3.2 Effects of tolerance in the array element gain

Next, we investigate how variations in the gain of individual elements are likely to affect the EVM performance and beam pattern of a linear array. For this simulation, we follow the same procedures as adopted in Section 6.3.1 but assume that the individual elements are correctly placed at their respective nominal locations. However, the gains of the individual elements are deviating from the nominal gain, G, according to equation (6.23). In this simulation, the maximum allowable gain deviation, $g_{\max}$ is assumed to be equal to ±10% of the nominal gain. Again, we consider four different cases of gain variations in the elements, as tabulated in Table 6-6. The first two cases correspond to random gain variations of up to ±10% of the nominal gain, while the last two cases correspond to the worst-case conditions.

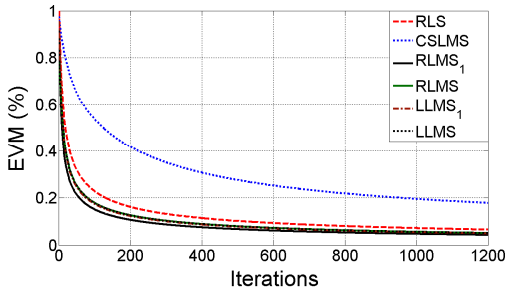
Table 6-6 Five different scenarios of tolerances in the element gain.

Cases	**Element Spacing**							
	$G(1)$	$G(2)$	$G(3)$	$G(4)$	$G(5)$	$G(6)$	$G(7)$	$G(8)$
1	G	$0.93G$	$0.92G$	G	$1.1G$	$1.03G$	$1.02G$	$0.94G$
2	$1.1G$	$1.03G$	$1.02G$	G	G	$0.94G$	$0.93G$	$0.92G$
3	$1.1G$	$0.9G$	$1.1G$	$0.9G$	$1.1G$	$0.9G$	$1.1G$	$0.9G$
4	$0.9G$	$1.1G$	$0.9G$	$1.1G$	$0.9G$	$1.1G$	$0.9G$	$1.1G$

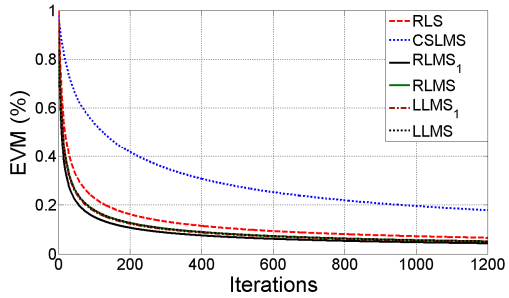
Figure 6-14 show the EVM performance obtained with these four cases of element gain variations when an 8-element linear array is implemented using the RLMS, $RLMS_1$, LLMS, $LLMS_1$, RLS and CSLMS algorithm. When compared with the ideal case represented by the EVM values of Figure 6-12a, it is observed that gain variations of the order of ±10% of the nominal gain will not significantly affect the EVM performance of each of the six algorithms considered. This is because under the given operating conditions, the array can still maintain almost the same gain in the desired direction

while providing a null in the direction of the cochannel interference, as shown in Figure 6-15. Also, it is observed that the beam patterns of Figure 6-15 are less affected by element gain variations than displacements in the element locations.

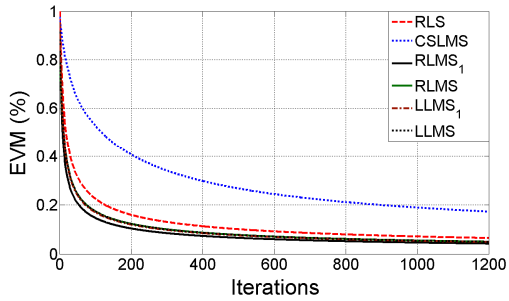
(a) Case 1 – Random gain deviation (i)



(b) Case 2 – Random gain deviation (ii)



(c) Case 3 –Worst case gain deviation (i)



(d) Case 4 – worst case gain deviation (ii)

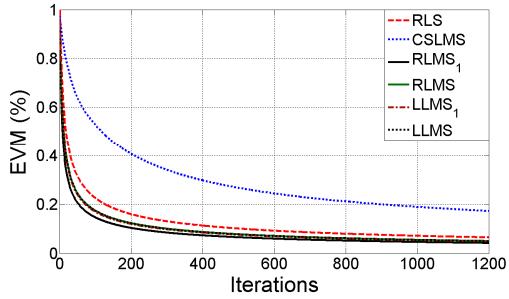
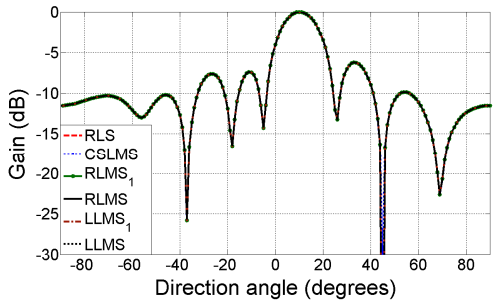
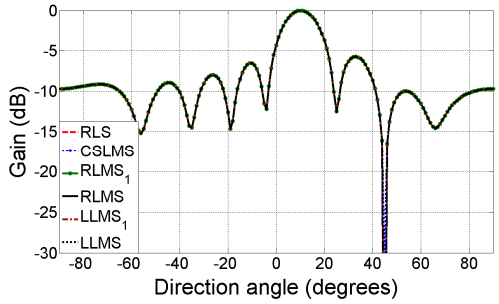


Figure 6-14 The EVM values obtained with the RLS, CSLMS, $RLMS_1$, RLMS, $LLMS_1$ and LLMS algorithms for different random variations in element gain. The maximum allowable gain variation is ±10% of the nominal gain.

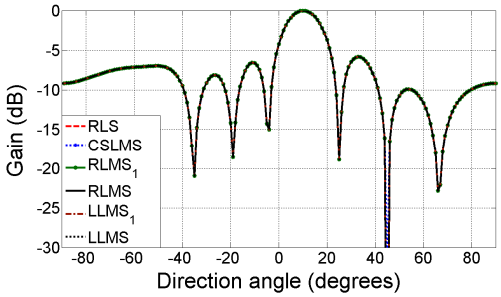
(a) Case 1 – Random gain deviation (i)



(b) Case 2 – Random gain deviation (ii)



(c) Case 3 –Worst case gain deviation (i)



(d) Case 4 – worst case gain deviation (ii)

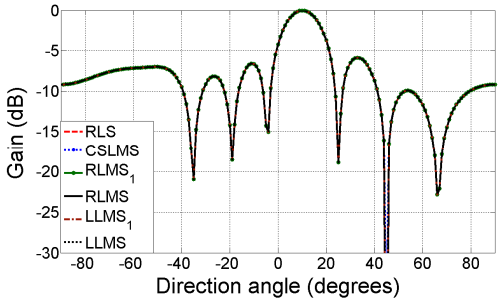


Figure 6-15 The beam pattern obtained with the RLS, CSLMS, $RLMS_1$, RLMS, $LLMS_1$ and LLMS algorithms for different random variations in element gain. The maximum allowable gain variation is ±10% of the nominal gain.

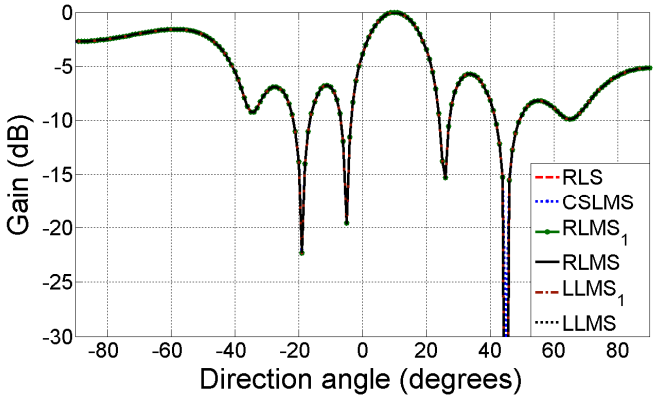
6.3.3 Combined variations in element spacing and element gain

As described in Sections 6.3.1, and 6.3.2, when random variations in element gain are kept within ±10% of the nominal gain, they have little effect

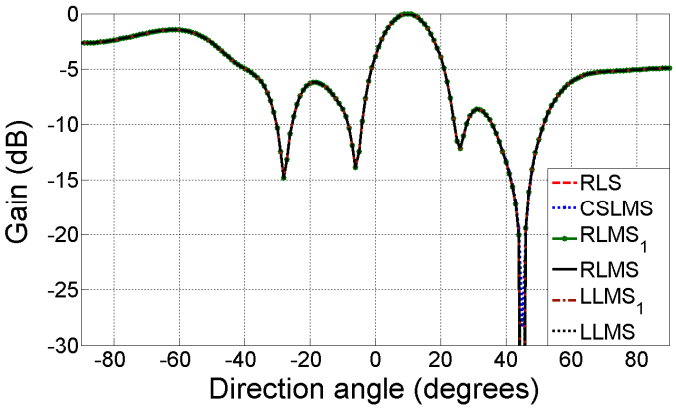
on the beam patterns. On the other hand, the presence of a maximum ±10% tolerance in inter-element spacing will result in a reduced level of sidelobe suppression. Moreover, these two forms of practical imperfections, when occurring separately, do not affect the ability of the array to correctly direct its main lobe to the desired signal, while maintaining a null response in the direction of the interfering signal. As a result, the EVM performance is shown to be barely affected by the occurrence of these imperfections. These observations are equally applicable to all the six algorithms considered.

Next, we investigate the influence on the beam patterns when variations are present in both the element spacing and element gain. For this study, we consider two rather severe cases of combined variations in element spacing and element gain. The first set of imperfections is made up of case (4) of Table 6-5 combined with case (3) of Table 6-6, while the second set is a combination of case (5) of Table 6-5 and case (4) of Table 6-6. The resultant beam patterns are shown in Figure 6-16 (a) and (b), which demonstrate similar behaviours to those observed in Sections 6.3.1 and 6.3.2.

As expected, the maximum gain of the array remains pointing towards the direction of the desired signal. Also, a null is still occurring at the angle of arrival of the interfering signal. When compared with the beam patterns of the previous two sections, the side lobe levels have increased, particularly at angles towards the two ends of the array. From the last two sections, no clear effect has been noticed on the EVM values. However, the beam patterns have been more affected by element-gain tolerances, resulting in their higher side lobes. Moreover, the gain as well as the null response in the direction of the interfering signal are not noticeably affected. This confirms that variations of ±10% in both the inter-element spacing and element-gain tolerances have little effect on the received signal.



(a) Case (4) of Table 6-5 combined with Case (3) of Table 6-6



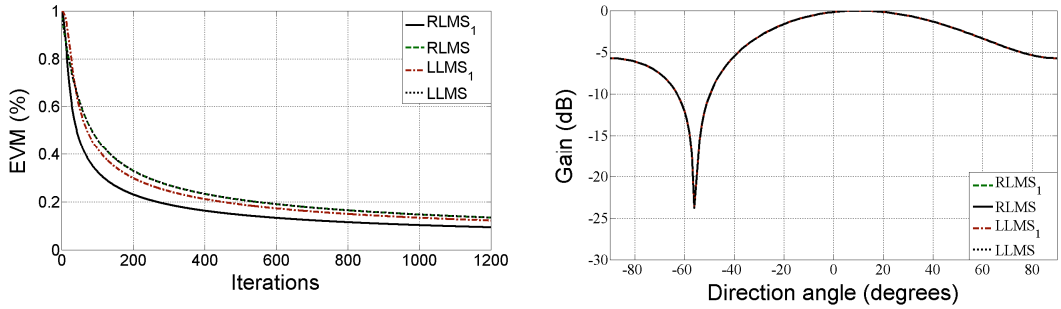
(b) Case (5) of Table 6-5 combined with Case (4) of Table 6-6.

Figure 6-16 The beam patterns obtained with RLS, CSLMS, $RLMS_1$, RLMS, $LLMS_1$ and LLMS algorithms for two different sets of combined variations in inter-element spacing and element gain.

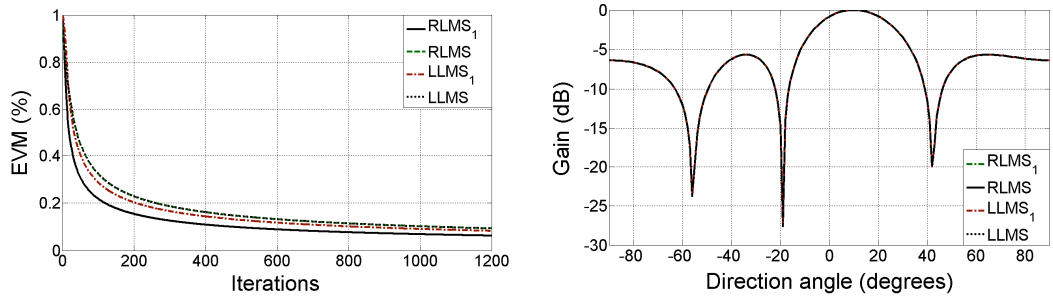
6.4 Arrays with Two and Four Elements

Up to now, we have only considered a linear array with 8 elements. It may be beneficial to study the influence of the number of elements on the array performance for the proposed RLMS and LLMS algorithms. The use of a smaller number of array elements will also lead to lesser tap weights being required for a given algorithm. This will then result in a simpler implementation. For this reason, we will consider the performance, in terms of EVM and beam patterns, of arrays with two and four elements.

First, we consider applying the same parameter values of Table 6-4 which were previously used for the 8-element array, in an array with two and four elements. The results obtained for the RLMS, $RLMS_1$, LLMS and $LLMS_1$ algorithms in an interference free channel are shown in Figure 6-17. In this case, the desired signal is arriving from an angle of 10^o. It is observed from Figure 6-17 and Figure 6-13a for an 8-element array that the number of side lobes is, as expected increased with the number of array elements used. Also, the EVM values converge faster and to a lower floor when a larger number of array elements are used.



(a) 2-element array

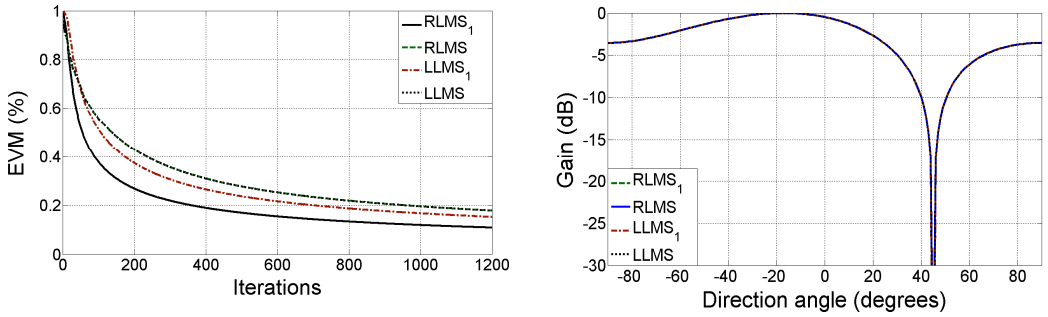


(b) 4-element array

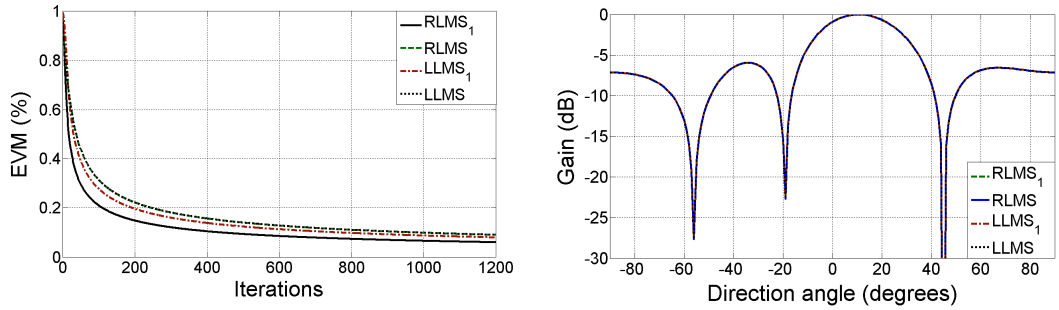
Figure 6-17 The EVM values and beam patterns obtained for a (a) 2-tap, and (b) 4-tap array in an interference free channel.

Next, cochannel interference emanating from an angle of 45^o is introduced into the channel. The resultant EVM values and beam patterns for an array with two and four elements are shown in Figure 6-18. As in the case

of an 8-element array (see Figure 6-13a), the array is able to direct correctly the main lobe to the desired signal, while the channel interference is suppressed with a null response in the direction of the interfering signal. However, the 2-element array suffers an approximately 1.3 dB loss in gain in the direction of the desired signal. As a result, its EVM performance is degraded slightly when compared with Figure 6-17a for the case with no interference.



(a) 2-element array



(b) 4-element array

Figure 6-18 The EVM values and beam patterns obtained for a (a) 2-tap, and (b) 4-tap array in the presence of cochannel interference emanating from an angle of 45^{o}.

Unlike other algorithms, the proposed RLMS and LLMS algorithms make use of two sets of tap weights in cascade. This offers a unique opportunity to examine the possibility of adopting a smaller number of tap weights in the second algorithm stage as an attempt to further reduce the implementation

complexity. For example, in the case of a 4-element array implemented using the RLMS and LLMS algorithms, one may halve the number of tap weights used in the second algorithm stage. The results of this modification are shown in Figure 6-19. When compared with Figure 6-18b, it is observed that a reduction in the number of tap weights used in the second algorithm stage of an RLMS and LLMS algorithms has little effect on the beam pattern. However, with this modification, the EVM performance of Figure 6-19 shows a significant reduction in the convergence speed for the RLMS and LLMS algorithms. This can be corrected by adjusting the step sizes of the RLMS and LLMS algorithms. For example, Figure 6-20 shows the EVM values and beam patterns achieved through the use of step sizes tabulated in Table 6-7. It shows there is no visible change in the beam patterns but the EVM convergence speed has improved markedly. This further suggests that when the step size values are chosen correctly, it is possible to maintain the performance of the proposed RLMS and LLMS algorithms, while lessening the computational complexity with a reduced number of tap weights in the second algorithm stage.

Table 6-7 Values of the constants adopted for operation in a 4-element array with only 2 tap weights in the second algorithm stage.

Algorithm	**Noise Free Channel**
$RLMS_1$	$\mu_{RLS} = 0.01,\ \mu_{LMS} = 0.2$
RLMS	$\mu_{RLS} = 0.01,\ \mu_{LMS} = 2$
$LLMS_1$	$\mu_1 = 0.8,\ \mu_2 = 0.25$
LLMS	$\mu_1 = 0.8,\ \mu_2 = 2$

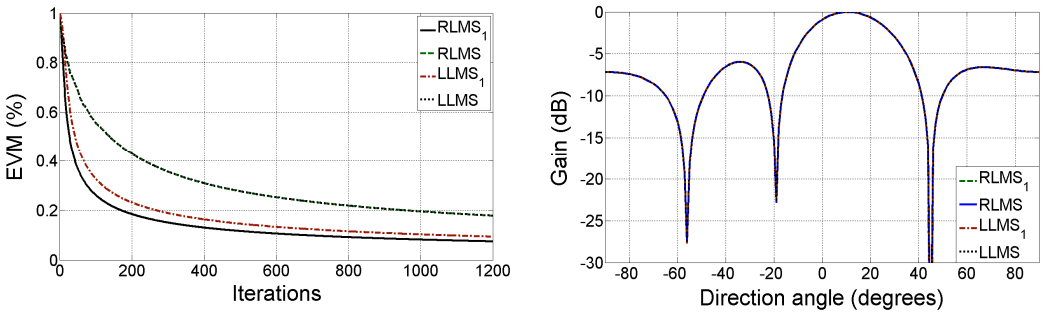


Figure 6-19 The EVM values and beam patterns obtained with $RLMS_1$, RLMS, $LLMS_1$ and LLMS algorithms for a 4-element array with only two tap weights in the second algorithm stage.

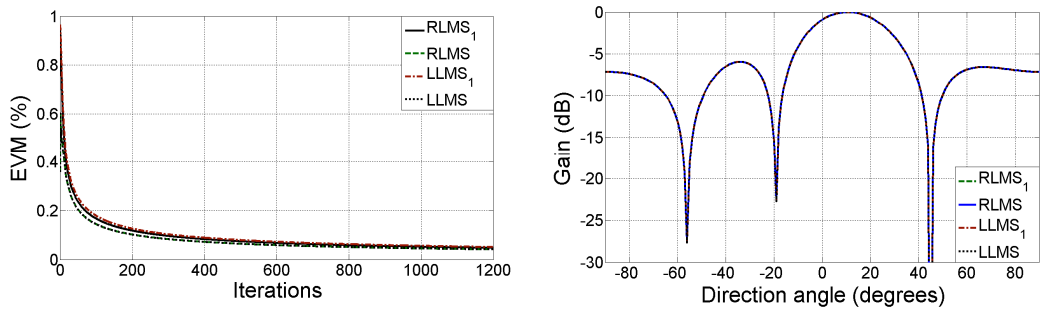


Figure 6-20 The EVM values and beam patterns obtained with $RLMS_1$, RLMS, $LLMS_1$ and LLMS algorithms for a modified 4-element array using the parameters given in Table 6-7.

6.5 Summary

In this chapter, the effects of some practical constraints, which are likely to be encountered in the implementation of the proposed RLMS and LLMS algorithms have been examined. These include the use of finite wordlength, tolerances in array element spacing and element gain, and the array size. First, the convergence behaviours of the RLMS and LLMS algorithms, based on the minimum mean square error, have been analyzed for operation with

finite numerical precision. It is shown that the implementation of an eight element uniform linear array using the RLMS and LLMS algorithms with a wordlength of eight bits is sufficient to achieve performance close to that provided by full numerical precision. Comparisons based on various performance measures, such as residual MSE, rate of convergence, error vector magnitude, and beam pattern, show that the RLMS, $RLMS_1$, LLMS and $LLMS_1$ algorithms outperform four other previously published algorithms, namely, least mean square (LMS), recursive least square (RLS), modified robust variable step size (MRVSS) and constrained stability LMS (CSLMS).

Furthermore, tolerances in the element spacing and element gain tend to raise the sidelobe level but show no visible effect on the resulting EVM values. The latter may be explained by the fact that such practical imperfections do not seem to influence the ability of the array in directing correctly towards the desired signal, while at the same time producing a null in the direction of an interfering signal. It is shown that the proposed RLMS, $RLMS_1$, LLMS and $LLMS_1$ algorithms outperform both the RLS and CSLMS algorithms when tolerances are present in the element spacing and gain.

As expected, the beam pattern of a linear array implemented using the proposed RLMS and LLMS algorithm contains a smaller number of side lobes when the number of elements used is reduced. It is observed that a linear array with a smaller number of elements tend to converge a bit slower and to a slightly higher floor value.

Furthermore, it is shown that the RLMS and LLMS algorithms can maintain the same performance when the number of tap weights used in the second stage is reduced. This observation is likely to simplify the implementation of an adaptive antenna array which makes use of the proposed RLMS and LLMS algorithms.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The main contribution of this research has been the development of a novel approach in combining the use of two well known algorithms, namely the recursive least square (RLS) and least mean square (LMS) algorithms, in a new adaptive architecture for antenna beamforming. As a result, two new algorithms, which can simultaneously achieve rapid convergence as well as low steady state error, have been proposed and analyzed. The first algorithm, called the RLMS algorithm incorporates a RLS algorithm stage and a LMS algorithm stage connected in series via an array image vector, $\mathcal{F}$. In this case, the array image vector, which could be made either fixed[7] or adaptive, is used as a "spatial filter" for the desired signal. This results in the convergence rate being increased without the need to use step size adaptation, while an improved tracking ability is achieved with only a slight increase in computation complexity. Such an arrangement maintains a complexity generally associated with a conventional RLS algorithm.

Initially, it appears logical to make use of the RLS algorithm, which generally converges faster than a LMS algorithm, in the first stage of the RLMS algorithm. Moreover, the complexity of the RLS algorithm may still be considered too high for some applications. This observation motivates the search for a simpler alternative to replace the RLS algorithm while still keeping the superior performance of the RLMS algorithm. The proposed flexible architecture of the RLMS algorithm offers a rather obvious possibility to replace the more complicated RLS algorithm with a simpler LMS algorithm. The resultant scheme is referred to as the LLMS algorithm, which retains the low complexity generally associated with an LMS algorithm.

[7] When a fixed array is used, the resultant algorithm is referred to as the $RLMS_1$ algorithm.

With the proposed RLMS and LLMS algorithms, the intermediate output, estimated using the first algorithm stage, is multiplied by the array image vector of the desired signal. The resultant signal is further processed by the second algorithm stage. To enhance the convergence rate of the overall algorithm, the error of the second algorithm stage is then fed back to combine with the error of the first stage to form the overall error signal for updating the tap weights of the first algorithm stage.

The proposed RLMS and LLMS algorithms have been described and analyzed in detail in Chapter 4 and Chapter 5, respectively. Also, their performance when used in digital array beamforming has been investigated by means of extensive computer simulations with the various mathematical functions represented in full numerical precision. In Chapter 6, some practical considerations, such as finite wordlength, and tolerances in inter-element spacing and element gain have also been studied.

The followings are the main conclusions of this research:

- Chapter 2 provides an introduction to the fundamentals of antenna array beamforming. The array configurations discussed include linear, circular and planar arrays. This is followed by a literature review, in Chapter 3, on digital signal processing algorithms applicable for array beamforming. Emphasis has been directed to the simpler non-blind algorithms, particularly the LMS and RLS based families of algorithms.

- The development of the RLMS algorithm is presented in Chapter 4. The RLMS algorithm follows a different approach to achieve simultaneously fast convergence and good tracking. Mathematical derivations have been made, in Section 4.3.1, for the mean square error convergence employing an external reference. Simulation results given in Section 4.7.3.1 have shown that both the $RLMS_1$ and RLMS algorithms can converge rapidly in 7 iterations. Also, the

resulting steady state MSE is quite insensitive to variations in input SNR. On the other hand, the same array beamforming realized with the conventional RLS algorithm converges after 20 iterations, and with a steady state MSE which is 2.5 times larger than that of the proposed RLMS algorithm.

The superior performance of the RLMS algorithm has also been verified through the use of error vector magnitude (EVM) and scatter plots, as shown in Section 4.7.3.7.

- The boundary values for the step sizes, μ_{RLS} and μ_{LMS}, used in the respective RLS stage and LMS stage, have been derived in Sections 4.3.1 and 4.4, for operation with either fixed or adaptive array image vector. It is shown in Section 4.7.3.1 that stable operation of the RLMS algorithm can be achieved with a broad range of values for μ_{RLS} and μ_{LMS}.

- It is shown in Section 4.3.2, that once initial convergence is achieved, usually within a few iterations, both the RLMS_1 and RLMS algorithms can maintain their operation through self-referencing. This mode of operation has been confirmed by the simulation results presented in Section 4.7.3.2.

- A simple and effective method for calculating adaptively the element values of the array image vector, $\mathcal{F}_{\mathrm{R}}$, has been proposed in Section 4.5. This is based on the output signal of the RLS algorithm stage in conjunction with its tap weights. This adaptive version of the algorithm is described simply as the RLMS algorithm, which is able to automatically track the target signal.

  For fixed array beamforming, the elements of the array image vector can be prescribed with values for the required direction. For this mode of operation, the algorithm is referred to as the RLMS_1 algorithm, in order to differentiate it from the adaptive version of the

RLMS algorithm. It has been shown in Section 4.7.3.6 that the $RLMS_1$ algorithm can provide an accurate fixed beam pointing towards the prescribed direction.

- Unlike the conventional LMS, RLS, and more recently published VFFRLS, CSLMS and MRVSS algorithms, the proposed RLMS and $RLMS_1$ algorithms are able to operate with a noisy reference signal. It is shown in Section 4.7.3.3 that the residual MSE associated with the two versions of the proposed algorithm remain very small, even when the rms noise level becomes as large as the reference signal.

- The original RLS algorithm is known to suffer from slow tracking ability. The results in Section 4.7.3.4 show that both the $RLMS_1$ and RLMS algorithms are capable of fast response to sudden changes in the input signal. Unlike the RLS, VFFRLS, LMS, CSLMS and MRVSS algorithms, the MSE values associated with the RLMS and $RLMS_1$ algorithms remain low when the input signal is interrupted.

- According to the beam patterns presented in Section 4.7.3.5, both the $RLMS_1$ and RLMS algorithms show superior suppression of co-channel interference when compared with beamformers realized using the RLS, VFFRLS, LMS, CSLMS and MRVSS algorithms. For the 8-element uniform linear array under consideration in this study, the proposed algorithms are able to achieve a signal-to-interference ratio (SIR) of better than 10 dB over the best of the other four algorithms.

- In Section 4.7.3.8, the EVM values and scatter plots obtained for operation in either an AWGN channel or fast changing Rayleigh fading environment, further demonstrate the superior performance of the $RLMS_1$ and RLMS algorithms when compared to the other four published algorithms, namely, the LMS, RLS, VFFRLS, CSLMS and MRVSS algorithms. Both the EVM and scatter plots are

commonly used to measure signal fidelity, for example, a low EVM value is generally associated with a low bit error rate.

- The superior performance of the proposed RLMS algorithm is achieved with a complexity only slightly larger than that of the original RLS algorithm. Moreover, this complexity is significantly lower than some RLS based algorithms, such as the VFFRLS, AFF-RLS and EX-KRLS algorithms, which have been proposed for improving the tracking performance of the RLS algorithm. For example, an N-element uniform linear array will require computation complexity equivalent to $2.5N^2+5N+1$ multiplications for the proposed RLMS algorithm, and $2.5N^2+3N+20$, $9N^2+7N$ and $15N^3+7N^2+2N+4$ multiplications for the VFFRLS, AFF-RLS and EX-KRLS algorithms, respectively.

- The complexity of the RLS stage in the RLMS algorithm remains an issue for some applications. This leads to another proposed algorithm being presented in Chapter 5, called the LLMS algorithm, in which the RLS algorithm stage in the RLMS algorithm is replaced with another LMS algorithm stage.

- Since the two algorithms share the same architecture, both the LLMS and RLMS algorithms can operate in similar manner, i.e., both of them can operate with either a fixed or adaptive array image vector, and with either an external reference or self referencing. Again, the LLMS_1 algorithm is referred to the version of the LLMS algorithm which makes used of a fixed array image vector.

  Following similar procedures as used for the analysis of the RLMS algorithm, the convergence of the LLMS algorithm has been established in Section 5.3.1 assuming the use of an external reference signal. This is then extended in Section 5.3.2 to cover the case that makes use of self-referencing. Mathematical derivations of the boundary values for the step sizes used in the two LMS

algorithm stages are presented in Sections 5.3.3 and 5.3.4.2, respectively.

- Simulation results in Sections 5.4.2.1, 5.4.2.2, and 5.4.2.3 show that the LLMS algorithm performs similar to the RLMS algorithm, in terms of rapid convergence, steady state MSE, and robustness to noisy reference signal. Under these measures, the RLMS algorithm, which is significantly more complex, performs only slightly better than the LLMS algorithm. The LLMS algorithm with its ability to adapt to the operating conditions through the use of an adaptive array image vector shows a little better performance than the $LLMS_1$ algorithm.

- In addition, it is shown in Sections 5.4.2.5 and 5.4.2.6 that the LLMS algorithm can retain the fidelity of the signal in the presence of Rayleigh fading, as indicated by the low EVM values.

- The computation complexity of the LLMS algorithm is equivalent to only $4N+1$ complex multiplications as opposed to $2.5N^2+5N+1$ complex multiplications for the RLMS algorithm, where *N* is the number of array elements.

- It is shown in Section 5.4.2.1 that the $LLMS_1$ and LLMS algorithms can operate with a wide range of step sizes, which are not too sensitive to changes in input SNR and noisy reference signal. Moreover, the convergence speed of the LLMS algorithm can be adjusted by varying the values of the two step sizes used.

- The convergence behaviours of the RLMS and LLMS algorithms when implemented with finite numerical precision have been analyzed based on the MSE in Sections 6.2.1 and 6.2.2, respectively. For an eight element uniform linear array, it has been confirmed through simulations that an 8-bit numerical precision is

sufficient for either the RLMS or LLMS algorithm to achieve a performance close to that obtained with full numerical precision.

- Simulation results presented in Section 6.3 show that deviations in inter-element spacing and gain tend to raise the sidelobe level by 5 dB. However, these practical imperfections have little visible effect on the resulting EVM values. It is further observed that such practical imperfections do not seem to influence the ability of the array in directing its main beam correctly towards the desired signal, while at the same time producing a null in the direction of an interfering signal.
- In an attempt to further reduce the complexity of the RLMS and LLMS algorithms, simulations have been carried out in Section 6.4 to investigate the possibility of operating with less tap weights in the second LMS algorithm stage. It is observed that by readjusting the step size values, it is possible to achieve similar performance using a lesser number of tap weights in the second algorithm stage.

- Generally, the performance achieved with the LLMS algorithm under the various parameters considered in this study is very close to that obtained with the RLMS algorithm. This suggests that the simpler LLMS algorithm is more attractive than the RLMS algorithm for applications in adaptive beamforming.

7.2 Recommendations for Future Work

The proposed LLMS and RLMS algorithms appear to be attractive candidates for use in adaptive beamforming. As such, it is recommended that the following topics be studied to further establish their potential for future adaptive signal processing.

- Currently the two proposed algorithms have only been simulated using MATLAB code. It will be useful to verify the actual

computation complexity by implementing the algorithm using either a digital signal processor (DSP) chip or field programmable gate arrays (FPGA).

- The geometry of the array that has been used in this study is limited to a uniform linear array (ULA). It remains interesting and worthwhile to examine how the proposed algorithms will perform when operating with different array geometry, such as a planar or circular array.

- In this thesis, the beamforming algorithms are only used for the receiving mode. It will be a challenge to use, in the transmission mode, the same weight vector obtained in the receiving mode.

- The modulation scheme used in this research study is simple binary phase shift keying (BPSK). However, higher order modulation schemes, such as OFDM and 64 QAM are used in modern cellular communications systems, such as LTE and WiMax systems. For the proposed algorithms to be applied in these systems, it is necessary to examine how they perform with high order modulated signals.

- Both the LLMS and RLMS algorithms require a reference signal for their operation. This calls for an investigation of what is a good and effective way to provide such a signal.

- This study has only considered a single-beam array. However, multiple beams are required for future cellular mobile communication systems. This should provide the necessary motivation in investigating how the proposed algorithms may be used to synthesize a multiple-beam pattern.

- If necessary, the RLS algorithm stage in the proposed RLMS algorithm may be simplified through the use of a fast RLS algorithm [82, 142], which has been proposed to lower the complexity of the RLS algorithm.

APPENDIX A

SUMMARY OF ALGORITHMS USED IN THE THESIS

A.1 Introduction

This appendix provides a summary of the proposed RLMS and LLMS algorithms together with those algorithms used for performance comparison in Chapters 4 and 5. These algorithms are, the LMS, RLS, VFFRLS, CSLMS and MRVSS algorithms, as given in [32], [13], [9], [12], [10], [143] and [144].

A.2 LMS Algorithm

The LMS algorithm updates the beamformer tap weights so that the error, $e(n)$, is minimized in the mean-square sense. When the input vector data, $\boldsymbol{X}(n)$, and the reference signal, d, are jointly stationary, this algorithm converges to a set of tap-weights, which are on average, equivalent to the Wiener-Hopf solution. The LMS algorithm is a practical scheme for realizing Wiener filters using the steepest descent method, without explicitly solving the Wiener-Hopf equation. It was first proposed by Widrow *et. al*. in [145].The LMS algorithm can be summarized as

Initialize $\boldsymbol{W}(0) = 0$

Iterate for $n \geq 1$

$$e(n) = d(n) - \boldsymbol{W}^H(n)\boldsymbol{X}(n)$$

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \mu \boldsymbol{X}(n)e^*(n)$$

Output:

$$y_{\mathrm{LMS}}(n) = \boldsymbol{W}^H(n)\boldsymbol{X}(n)$$

Definitions:

$\boldsymbol{W}$ ≡ Weight vector

$\boldsymbol{X}$ ≡ Input signal vector

e ≡ Error signal

μ ≡ Step size

d ≡ Reference signal

$(\bullet)^H$ ≡ Hermitian operator

$(\bullet)^*$ ≡ Conjugate operator

A.3 RLS Algorithm

Contrary to the LMS algorithm, which uses the steepest descent method to obtain the weight vector, the RLS algorithm uses the method of least squares. In this case, the weight vector, $\boldsymbol{W}$ is updated by minimizing an exponentially weighted cost function. The standard RLS algorithm performs the following operations to update the weights of an adaptive beamformer:

Initialize $\boldsymbol{W}(0) = 0$, $\boldsymbol{P}(0) = \delta^{-1}\boldsymbol{I}$

Iterate for $n \geq 1$

$$\boldsymbol{K}(n) = \frac{\boldsymbol{P}(n-1)\boldsymbol{X}(n)}{1-\mu+\boldsymbol{X}^H(n)\boldsymbol{P}(n-1)\boldsymbol{X}(n)}$$

$$\boldsymbol{P}(n) = \frac{1}{1-\mu}\left[\boldsymbol{P}(n-1) - \boldsymbol{K}(n)\boldsymbol{X}^H(n)\boldsymbol{P}(n-1)\right]$$

$$e(n) = d(n) - \boldsymbol{W}^H(n)\boldsymbol{X}(n)$$

$$\boldsymbol{W}(n) = \boldsymbol{W}(n-1) + \boldsymbol{K}(n)e(n)$$

Output:

$$y_{\mathrm{RLS}}(n) = \boldsymbol{W}^H(n)\boldsymbol{X}(n)$$

Definitions:

$\boldsymbol{P}$ ≡ Inverse of the input correlation matrix

$\boldsymbol{K}$ ≡ Kalman gain vector

$\boldsymbol{W}$, $\boldsymbol{X}$, e, d, μ and $(\bullet)^H$ are as defined in Appendix A.2 for this algorithm.

A.4 VFFRLS Algorithm

The variable forgetting factor RLS algorithm is proposed in [12] for system identification. This algorithm updates the forgetting factor, λ, of the RLS algorithm. The adaptation of the forgetting factor is carried as follows:

Initialize $\lambda_{\max}$, $\beta_{\text{VFF}} = 1 - \frac{1}{k_\beta N}$, $\alpha_{\text{VFF}} = 1 - \frac{1}{k_\alpha N}$

Iterate for $n \geq 1$

$$q(n) = \boldsymbol{X}^H(n)\boldsymbol{P}(n-1)\boldsymbol{X}(n)$$

$$\hat{\sigma}_v^2(n) = \beta_{\text{VFF}}\hat{\sigma}_v^2(n-1) + (1-\beta_{\text{VFF}})e^2(n)$$

$$\hat{\sigma}_e^2(n) = \alpha_{\text{VFF}}\hat{\sigma}_e^2(n-1) + (1-\alpha_{\text{VFF}})e^2(n)$$

$$\hat{\sigma}_q^2(n) = \alpha_{\text{VFF}}\hat{\sigma}_q^2(n-1) + (1-\alpha_{\text{VFF}})q^2(n)$$

$$\lambda(n) = \begin{cases} \lambda_{\max}\ ; & \hat{\sigma}_e(n) \leq \gamma_{\text{VFF}}\hat{\sigma}_v(n) \\ \min\left\{ \dfrac{\hat{\sigma}_q(n)\hat{\sigma}_v(n)}{\varepsilon_{\text{VFF}} + \left|\hat{\sigma}_e(n) - \hat{\sigma}_v(n)\right|}, \lambda_{\max} \right\}; & \text{otherwise} \end{cases}$$

Definitions:

$\hat{\sigma}_v^2 \equiv$ Power of system noise.

$\hat{\sigma}_e^2 \equiv$ Power of the a priori error signal.

$\lambda \equiv$ Forgetting factor.

$\alpha_{\text{VFF}} \equiv$ Weighting factor with $k_\alpha \geq 2$.

$\beta_{\text{VFF}} \equiv$ A constant given as $\beta_{\text{VFF}} = 1 - 1/k_\beta N$, and $k_\beta > k_\alpha$.

$\lambda_{\max} \equiv$ Upper limit of the forgetting factor.

$\gamma_{\text{VFF}} \equiv$ A constant $(1, 2)$.

$\varepsilon_{\text{VFF}} \equiv$ A small positive constant for avoiding division by zero.

$\boldsymbol{X}$ and $\boldsymbol{P}$ are as defined in Appendix A.3 for this algorithm.

A.5 CSLMS Algorithm

Górriz *et. al*. [9] proposed the CSLMS algorithm for filtering speech sounds. The CSLMS algorithm is based on the minimization of the squared

Euclidean norm of the weight vector under a stability constraint over the a posteriori estimation error. This algorithm employs the following operations in order to update the weight vector:

Initialize $\boldsymbol{W}(0) = 0$, $e(-1) = 0$

Iterate for $n \geq 1$

$$\delta \boldsymbol{W}(n) = \boldsymbol{W}(n) - \boldsymbol{W}(n-1)$$

$$\delta \boldsymbol{X}(n) = \boldsymbol{X}(n) - \boldsymbol{X}(n-1)$$

$$e^{[k]}(n) = d(n) - \boldsymbol{W}^{H}(k)\boldsymbol{X}(n)$$

$$\delta e^{[n]}(n) = e^{[n]}(n) - e^{[n]}(n-1)$$

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \frac{\mu}{\left\|\delta \boldsymbol{W}(n)\right\|^2 + \varepsilon_{cs}} \delta \boldsymbol{X}(n)\left(\delta e^{[n]}(n)\right)^*$$

Output:

$$y_{\text{CSLMS}}(n) = \boldsymbol{W}^{H}(n)\boldsymbol{X}(n)$$

Definitions:

δe ≡ Error signal difference.

$e^{[k]}(n)$ ≡ Error signal at the n^{th} iteration derived from the weight vector at k^{th} iteration; with $n > k$.

ε_{cs} ≡ A small positive constant introduced to prevent division by zero.

$\boldsymbol{W}$ and $\boldsymbol{X}$ are as defined in Appendix A.2 for this algorithm.

A.6 MRVSS Algorithm

Zou Kun and Zhao Xiubing [10] proposed the MRVSS algorithm for improving the performance of the VSSLMS algorithms [54], and the RVSS algorithm [56]. The MRVSS algorithm makes use of the following steps:

Initialize $\boldsymbol{W}(0) = 0$, $\mu_{\max}$, $\mu_{\min}$, $R_e(0) = 0$, $\tilde{e}(0) = 0$, $e(0) = 0$

Iterate for $n \geq 1$

$$e(n) = d(n) - \boldsymbol{W}^H(n)\boldsymbol{X}(n)$$

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \mu(n)\,\boldsymbol{X}(n)\,e(n)$$

$$\mu(n+1) = \begin{cases} \mu_{\max} \text{ ; if } \mu(n+1) > \mu_{\max} \\ \mu_{\min} \text{ ; if } \mu(n+1) < \mu_{\min} \\ \alpha\mu(n) + \gamma R_e^2(n) \end{cases}$$

$$R_e(n+1) = (1-\tilde{e}(n))R_e(n) + \tilde{e}(n)e(j)e(n-1)$$

$$\tilde{e}(n+1) = \begin{cases} \tilde{e}_{\max} \text{ ; } & \text{if } \tilde{e}(n+1) > \tilde{e}_{\max} \\ \tilde{e}_{\min} \text{ ; } & \text{if } \tilde{e}(n+1) < \tilde{e}_{\min} \\ \eta_e\,\tilde{e}(n) + \upsilon e^2(n) \end{cases}$$

Output:

$$y_{\text{MRVSS}}(n) = \boldsymbol{W}^H(n)\boldsymbol{X}(n)$$

Definitions:

$\mu_{\max}$ ≡ Upper bound of the step size.

$\mu_{\min}$ ≡ Lower bound of the step size.

α ≡ A constant with $\alpha > 0$.

γ ≡ A constant with $\gamma > 0$.

R_e ≡ Time averaged error correlation over two consecutive values.

$\tilde{e}$ ≡ Time averaged error square signal.

$\tilde{e}_{\max}$ ≡ Upper bounds of $\tilde{e}$.

$\tilde{e}_{\min}$ ≡ Lower bounds of $\tilde{e}$.

η_e ≡ A constant with $\eta_e > 1$.

υ ≡ A constant with $\upsilon > 0$.

$\boldsymbol{W}$, $\boldsymbol{X}$, e and $(\bullet)^H$ are as defined in Appendix A.2 for this algorithm.

A.7 RLMS Algorithm

RLMS algorithm is a new algorithm proposed in this thesis. It combines the use of a RLS algorithm stage followed by a LMS algorithm stage. In this case, the intermediate output, y_{RLS}, estimated using the RLS algorithm is fed to an LMS section after it has been multiplied by the array image vector, $\boldsymbol{\mathcal{F}}_{\mathrm{R}}$. The error signal, $e_{\mathrm{LMS}}(n)$, produced by the LMS algorithm stage, is fed back to combine with $e_{\mathrm{RLS}}(n)$ to form the overall error signal, $e_{\mathrm{RLMS}}(n)$, for updating the RLS weights, $\boldsymbol{W}_{\mathrm{RLS}}(n)$. The tap weights of the RLS and LMS algorithm stages are updated according to the following::

Initialize $\boldsymbol{W}_{\mathrm{RLS}}(0)=0$, $\boldsymbol{W}_{\mathrm{LMS}}(0)=0$, $\boldsymbol{P}(0)=\delta^{-1}\boldsymbol{I}$, $e_{\mathrm{LMS}}(0)=0$

Iterate for $n\geq 1$

Input RLS stage:

$$\boldsymbol{K}(n)=\frac{\boldsymbol{P}(n-1)\boldsymbol{X}(n)}{1-\mu_{\mathrm{RLS}}+\boldsymbol{X}^{H}(n)\boldsymbol{P}(n-1)\boldsymbol{X}(n)}$$

$$\boldsymbol{P}(n)=\frac{1}{1-\mu_{\mathrm{RLS}}}\left[\boldsymbol{P}(n-1)-\boldsymbol{K}(n)\boldsymbol{X}^{H}(j)\boldsymbol{P}(n-1)\right]$$

$$e_{\mathrm{RLMS}}(n)=d(n)-\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(n)-e_{\mathrm{LMS}}(n-1)$$

$$\boldsymbol{W}_{\mathrm{RLS}}(n+1)=\boldsymbol{W}_{\mathrm{RLS}}(n)+\boldsymbol{K}(n)\,e_{\mathrm{RLMS}}(n)$$

Second stage input:

$$\boldsymbol{X}_{\mathrm{LMS}}(n)=\boldsymbol{\mathcal{F}}_{\mathrm{R}}\,\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{X}(n)$$

Output LMS stage:

$$e_{\mathrm{LMS}}(n)=d(n)-\boldsymbol{W}_{\mathrm{LMS}}^{H}(n)\boldsymbol{X}_{\mathrm{LMS}}(n)$$

$$\boldsymbol{W}_{\mathrm{LMS}}(n+1)=\boldsymbol{W}_{\mathrm{LMS}}(n)+\mu_{\mathrm{LMS}}\boldsymbol{X}_{\mathrm{LMS}}(n)e_{\mathrm{LMS}}^{*}(n)$$

Output:

$$y_{\mathrm{RLMS}}(n)=\boldsymbol{W}_{\mathrm{RLMS}}^{H}(n)\boldsymbol{X}(n)$$

Definitions:

e_{RLMS} ≡ Overall error signal

μ_{RLS} ≡ Step size of the RLS algorithm stage

μ_{LMS} ≡ Step size of the LMS algorithm stage

e_{LMS} ≡ Error signal of the LMS algorithm stage

$\boldsymbol{W}_{\mathrm{RLS}}$ ≡ Weight vector of the RLS algorithm stage

$\boldsymbol{W}_{\mathrm{LMS}}$ ≡ Weight vector of the LMS algorithm stage

$\boldsymbol{X}_{\mathrm{LMS}}$ ≡ Input signal vector of the LMS algorithm stage

$\mathcal{F}_{\mathrm{R}}$ ≡ Array image vector

$\boldsymbol{X}$, $\boldsymbol{K}$ and $\boldsymbol{P}$ are as defined in Appendix A.3 for this algorithm.

A.8 LLMS Algorithm

The LLMS algorithm is another new algorithm proposed in this thesis. It shares the same architecture as the RLMS algorithm but employs the LMS algorithm for both the two stages. The two weight $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ vectors of the LLMS algorithm are updated according to the following procedure:

Initialize $\boldsymbol{W}_1(0) = 0$, $\boldsymbol{W}_2(0) = 0$, $e_2(0) = 0$

Iterate for $n \geq 1$

First LMS algorithm stage:

$$e_{\mathrm{LLMS}}(n) = d(n) - \boldsymbol{W}_1^H(n)\boldsymbol{X}(n) - e_2(n-1)$$

$$\boldsymbol{W}_1(n+1) = \boldsymbol{W}_1(n) + \mu_1 \boldsymbol{X}(n) e_{\mathrm{LLMS}}^*(n)$$

Second stage input:

$$\boldsymbol{X}_2(n) = \mathcal{F}_{\mathrm{L}}\, \boldsymbol{W}_1^H(n)\boldsymbol{X}(n)$$

Second LMS algorithm stage:

$$e_2(n) = d(n) - \boldsymbol{W}_2^H(n)\boldsymbol{X}_2(n)$$

$$\boldsymbol{W}_2(n+1) = \boldsymbol{W}_2(n) + \mu_2 \boldsymbol{X}_2(n) e_2^*(n)$$

Output:

$$y_{\mathrm{LLMS}}(n) = \boldsymbol{W}_{\mathrm{LLMS}}^{H}(n)\boldsymbol{X}(n)$$

Definitions:

e_{LLMS} ≡ Overall error signal

μ_1 ≡ Step size of the first LMS algorithm stage

μ_2 ≡ Step size of the second LMS algorithm stage

e_2 ≡ Error signal of the second LMS algorithm stage

$\boldsymbol{W}_1$ ≡ Weight vector of the first LMS algorithm stage

$\boldsymbol{W}_2$ ≡ Weight vector of the second LMS algorithm stage

$\boldsymbol{X}_2$ ≡ Input signal vector of the second LMS algorithm stage

$\mathcal{F}_{\mathrm{L}}$ ≡ Array image vector

APPENDIX B

STEP SIZE BOUNDARY VALUES OF THE RLS ALGORITHM STAGE

To ensure stable operation of the RLMS algorithm, the RLS algorithm stage is required to operate with a proper step size. This appendix provide an analysis for determining the boundary values for the step size based on the mean-square error (MSE) of the overall error signal, ξ_{RLMS}, of the RLMS algorithm.

Recalling equation (4.49)

$$\xi_{\mathrm{RLMS}}(n) = \xi_{\mathrm{RLMS,min}} + \boldsymbol{V}_{\mathrm{RLS}}^{H}(0)\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}\right)^{n}\boldsymbol{Q}\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}\right)^{n}\boldsymbol{V}_{\mathrm{RLS}}(0) \qquad \text{(B.1)}$$

From equations (B.1) and (4.39), we obtain the asymptotic value of $\xi_{\mathrm{RLMS}}(n)$ as

$$\lim_{n\to\infty}\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}\right)^{n} = \lim_{n\to\infty}\boldsymbol{q}_1\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{\Lambda}_1\right)^{n}\boldsymbol{q}_1^{-1} \qquad \text{(B.2)}$$

Using equation (4.36), the RHS term of equation (B.2) becomes

$$\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{\Lambda}_1\right)^{n} = \mathrm{diag}[\left(1-\mu_{\mathrm{RLS}}E_1\right)^{n}, \left(1-\mu_{\mathrm{RLS}}E_2\right)^{n}, ..., \left(1-\mu_{\mathrm{RLS}}E_{\mathrm{N}}\right)^{n}] \qquad \text{(B.3)}$$

Now, if $\left|1-\mu_{\mathrm{RLS}}\max(E_i)\right| < 1$, where $i = 1, 2, ..., N$, then

$$\begin{aligned} -1 < 1 - \mu_{\mathrm{RLS}}\max(E_i) < 1 \\ \text{or} \quad 0 < \mu_{\mathrm{RLS}}\max(E_i) < 2 \end{aligned} \qquad \text{(B.4)}$$

where E_{RLS} is the maximum eigenvalue of $\boldsymbol{\Lambda}_1$.

From equation (B.4), the step size of the RLS algorithm stage, μ_{RLS}, required for stable operation is given by

$$0 < \mu_{\mathrm{RLS}} < {}^{2}\!\diagup\!_{E_{\mathrm{RLS}}} \tag{B.5}$$

APPENDIX C

DERIVATION AND PROOF OF CONVERGENCE OF EQUATION (4.56)

In this appendix, the convergence of the RLMS algorithm is analyzed based on the mean-square error (MSE) of the overall error signal of the RLMS algorithm, ξ_{RLMS}, when the RLMS algorithm is operating in the self-referencing mode.

Recalling equation (4.56), we have

$$\xi_{\mathrm{RLMS}}(n)=\sum_{i=1}^{n}\alpha_{\mathrm{RLS}}^{n-i}\left\{\mathrm{E}\left[\left|D_{\mathrm{R}}(i)\right|^{2}\right]-\boldsymbol{Z}_{\mathrm{R}}'^{H}(n)\boldsymbol{W}_{\mathrm{RLS}}(n)\right.$$
$$\left.-\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{Z}_{\mathrm{R}}'(n)+\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{Q}_{\mathrm{R}}(n)\boldsymbol{W}_{\mathrm{RLS}}(n)\right\} \tag{C.1}$$

In the case of self-referencing, the reference signals for the RLS and LMS algorithm stages are given as

$$d_{\mathrm{RLS}}(n)=y_{\mathrm{RLMS}}(n-1) \tag{C.2}$$

and

$$d_{\mathrm{LMS}}(n)=y_{\mathrm{RLS}}(n) \tag{C.3}$$

Based on equations (C.1) and (C.3), we can redefine $D_{\mathrm{R}}(n)$ in equation (4.15) as

$$D(n)=2y_{\mathrm{RLMS}}(n-1)-y_{\mathrm{RLS}}(n-1) \tag{C.4}$$

where

$$y_{\mathrm{RLMS}} = \boldsymbol{W}_{\mathrm{LMS}}^{H} \boldsymbol{X}_{\mathrm{LMS}} \tag{C.5}$$

By considering the asymptotic behaviour of the RLMS algorithm, we can approximate, after reaching the final convergence, that y_{RLMS} is approximately equals to y_{RLS}. In this case, equation (C.5) can be rewritten as

$$y_{\mathrm{RLMS}} = y_{\mathrm{RLS}} \tag{C.6}$$

Based on equation (C.6), equation (C.4) becomes

$$D_{\mathrm{R}}(n) = y_{\mathrm{RLS}}(n-1) \tag{C.7}$$

The only unknown term in equation (C.1) is the first term on the RHS. Using equation (C.7), this term can be rewritten as

$$\sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left| D_{\mathrm{R}}(i) \right|^2 \right] = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left| y_{\mathrm{RLS}}(i-1) \right|^2 \right] \right\} \right. \tag{C.8}$$

Then, solving equation (C.8) yields

$$\sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left| D_{\mathrm{R}}(i) \right|^2 \right] = \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[y_{\mathrm{RLS}}(i-1) y_{\mathrm{RLS}}^{*}(i-1) \right] \right\} \right. \tag{C.9}$$

Using equation (C.6), equation (C.9) can be analyzed to become

$$\begin{aligned} \sum_{i=1}^{n} \alpha_{\mathrm{RLS}}^{n-i} \left\{ \mathrm{E}\left[\left| D_{\mathrm{R}}(i) \right|^2 \right] &= \mathrm{E}\left[\boldsymbol{W}_{\mathrm{RLS}}^{H}(n-1) \boldsymbol{X}(i-1) \boldsymbol{X}^{H}(i-1) \boldsymbol{W}_{\mathrm{RLS}}(n-1) \right] \right. \\ &= \boldsymbol{W}_{\mathrm{RLS}}^{H}(n-1) \mathrm{E}\left[\boldsymbol{X}(i-1) \boldsymbol{X}^{H}(i-1) \right] \boldsymbol{W}_{\mathrm{RLS}}(n-1) \\ &= \boldsymbol{W}_{\mathrm{RLS}}^{H}(n-1) \boldsymbol{Q}_{\mathrm{R}}(n-1) \boldsymbol{W}_{\mathrm{RLS}}(n-1) \end{aligned} \tag{C.10}$$

where $\boldsymbol{Q}_{\mathrm{R}}$ is as defined in equation (4.16).

Now substituting equation (C.10) in equation (C.1), we obtain the overall MSE, such that

$$\xi_{\mathrm{RLMS}}(n)=\sum_{i=1}^{n}\alpha_{\mathrm{RLS}}^{n-i}\left\{\boldsymbol{W}_{\mathrm{RLS}}^{H}(n-1)\boldsymbol{Q}_{\mathrm{R}}(n-1)\boldsymbol{W}_{\mathrm{RLS}}(n-1)-\boldsymbol{Z}_{\mathrm{R}}'^{H}(n)\boldsymbol{W}_{\mathrm{RLS}}(n)\right.$$
$$\left.-\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{Z}_{\mathrm{R}}'(n)+\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)\boldsymbol{Q}_{\mathrm{R}}(n)\boldsymbol{W}_{\mathrm{RLS}}(n)\right\} \tag{C.11}$$

Differentiating equation (C.11) with respect to the weight vector $\boldsymbol{W}_{\mathrm{RLS}}^{H}(n)$ then yields the gradient vector $\nabla(\xi_{\mathrm{RLMS}})$, given by

$$\nabla(\xi_{\mathrm{RLMS}})=-\boldsymbol{Z}_{\mathrm{R}}'(n)+\boldsymbol{Q}_{\mathrm{R}}(n)\boldsymbol{W}_{\mathrm{RLS}}(n) \tag{C.12}$$

By equating $\nabla(\xi_{\mathrm{RLMS}})$ to zero, we obtain a similar optimal weight vector to that given in equation (4.29). This is given by

$$\boldsymbol{W}_{opt_{\mathrm{RLS}}}(n)=\boldsymbol{Q}_{\mathrm{R}}^{-1}(n)\boldsymbol{Z}_{\mathrm{R}}'(n) \tag{C.13}$$

Now, substituting equation (C.13) into equation (C.11), we can express the MSE as

$$\xi_{\mathrm{RLMS,min}}(n)=\sum_{i=1}^{n}\alpha_{\mathrm{RLS}}^{n-i}\left\{\boldsymbol{W}_{\mathrm{RLS}}^{H}(n-1)\boldsymbol{Q}_{\mathrm{R}}(n-1)\boldsymbol{W}_{\mathrm{RLS}}(n-1)-\boldsymbol{Z}_{\mathrm{R}}'^{H}(n)\boldsymbol{W}_{\mathrm{RLS}}(n)\right\} \tag{C.14}$$

Substituting (C.14) in equation (C.11), and by dropping the index n, equation (C.11) becomes

$$\xi_{\mathrm{RLMS}}=\xi_{\mathrm{RLMS,min}}+\left(\boldsymbol{W}_{\mathrm{RLS}}-\boldsymbol{W}_{opt_{\mathrm{RLS}}}\right)^{H}\boldsymbol{Q}_{\mathrm{R}}\left(\boldsymbol{W}_{\mathrm{RLS}}-\boldsymbol{W}_{opt_{\mathrm{RLS}}}\right) \tag{C.15}$$

Now, we repeat the steps from equation (4.32) to equation (4.52) as follows:

Defining the error of the weight vector, $\boldsymbol{V}_{\mathrm{RLS}}$, as

$$\boldsymbol{V}_{\mathrm{RLS}}\triangleq\left(\boldsymbol{W}_{\mathrm{RLS}}-\boldsymbol{W}_{opt_{\mathrm{RLS}}}\right) \tag{C.16}$$

so that equation (C.15) can be written as

$$\xi_{\mathrm{RLMS}} = \xi_{\mathrm{RLMS,min}} + \boldsymbol{V}_{\mathrm{RLS}}^{H} \boldsymbol{Q}_{\mathrm{R}} \boldsymbol{V}_{\mathrm{RLS}} \tag{C.17}$$

Differentiating equation (C.17) with respect to $\boldsymbol{V}_{\mathrm{RLS}}^{H}$ yields

$$\boldsymbol{\nabla}(\xi_{\mathrm{RLMS}}) = \boldsymbol{Q}_{\mathrm{R}} \boldsymbol{V}_{\mathrm{RLS}} \tag{C.18}$$

where the EVD of $\boldsymbol{Q}_{\mathrm{R}}$ in equation (C.18) is given as

$$\boldsymbol{Q}_{\mathrm{R}} = \boldsymbol{q}_{\mathrm{R}} \boldsymbol{\Lambda}_{\mathrm{R}} \boldsymbol{q}_{\mathrm{R}}^{-1} = \boldsymbol{q}_{\mathrm{R}} \boldsymbol{\Lambda}_{\mathrm{R}} \boldsymbol{q}_{\mathrm{R}}^{H} \tag{C.19}$$

Also, defining

$$\boldsymbol{V}'_{\mathrm{RLS}} \triangleq \boldsymbol{q}_{\mathrm{R}}^{-1} \boldsymbol{V}_{\mathrm{RLS}} \tag{C.20}$$

Based on equations (C.19) and (C.20), we can express the MSE of equation (C.17) as

$$\xi_{\mathrm{RLMS}}(n) = \xi_{\mathrm{RLMS,min}}(n) + \boldsymbol{V}'^{H}_{\mathrm{RLS}}(n) \boldsymbol{\Lambda}_{\mathrm{R}} \boldsymbol{V}'_{\mathrm{RLS}}(n) \tag{C.21}$$

For steepest descent, the weight vector is updated according to

$$\boldsymbol{W}_{\mathrm{RLS}}(n+1) = \boldsymbol{W}_{\mathrm{RLS}}(n) - \mu_{\mathrm{RLS}} \boldsymbol{\nabla}(\xi_{\mathrm{RLMS}}(n)) \tag{C.22}$$

where μ_{RLS} is the convergence constant that controls stability and rate of adaptation of the weight vector, and $\boldsymbol{\nabla}(\xi_{\mathrm{RLMS}}(n))$ is the gradient at the n^{th} iteration.

Subtracting $\boldsymbol{W}_{opt_{\mathrm{RLS}}}$ from both sides of equation (C.22) yields

$$\boldsymbol{W}_{\mathrm{RLS}}(n+1) - \boldsymbol{W}_{opt_{\mathrm{RLS}}}(n+1) = \boldsymbol{W}_{\mathrm{RLS}}(n) - \boldsymbol{W}_{opt_{\mathrm{RLS}}}(n) - \mu_{\mathrm{RLS}} \boldsymbol{\nabla} \xi_{\mathrm{RLMS}}(n) \tag{C.23}$$

Multiplying equation (C.23) by $\boldsymbol{q}_{\mathrm{R}}^{H}$, we may rewrite equation (C.23) in the form of a linear homogeneous vector difference equation using the relationships of equations (C.16), (C.18), (C.19), and (C.20) to give

$$\boldsymbol{V}'_{\mathrm{RLS}}(n+1) = \left(\boldsymbol{I} - \mu_{\mathrm{RLS}} \boldsymbol{\Lambda}_{\mathrm{R}}\right) \boldsymbol{V}'_{\mathrm{RLS}}(n) \tag{C.24}$$

In another form, equation (C.24) can be written as

$$\boldsymbol{V}'_{\mathrm{RLS}}(n) = \left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{\Lambda}_{\mathrm{R}}\right)^{n} \boldsymbol{V}'_{\mathrm{RLS}}(0) \tag{C.25}$$

where $\boldsymbol{V}'_{\mathrm{RLS}}(0)$ is the initial value given by

$$\boldsymbol{V}'_{\mathrm{RLS}}(0) = \boldsymbol{W}'_{\mathrm{RLS}}(0) - \boldsymbol{W}'_{opt_{\mathrm{RLS}}} \tag{C.26}$$

Substituting equation (C.25) into equation (C.21) yields

$$\xi_{\mathrm{RLMS}}(n) = \xi_{\mathrm{RLMS,min}} + \boldsymbol{V}^{H}_{\mathrm{RLS}}(0)\left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)^{n} \boldsymbol{Q}_{\mathrm{R}} \left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)^{n} \boldsymbol{V}_{\mathrm{RLS}}(0) \tag{C.27}$$

The asymptotic value $\xi_{\mathrm{RLMS}}(n)$ of equation (C.27) becomes

$$\lim_{n\to\infty} \left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{Q}_{\mathrm{R}}\right)^{n} = \lim_{n\to\infty} \boldsymbol{q}_{\mathrm{R}} \left(\boldsymbol{I} - \mu_{\mathrm{RLS}}\boldsymbol{\Lambda}_{\mathrm{R}}\right)^{n} \boldsymbol{q}_{\mathrm{R}}{}^{-1} \tag{C.28}$$

Following the same analyzing steps for equation (4.51), it is able to show that the MSE of the overall error signal can converge to a minimum value, such that

$$\lim_{n\to\infty} \xi_{\mathrm{RLMS}}(n) = \xi_{\mathrm{RLMS,min}} \tag{C.29}$$

APPENDIX D

DERIVATION OF EQUATION (5.8)

In this appendix, the derivation of equation (5.5) in Chapter 5 is provided. In the analysis, we make use of the same assumptions to those given in Chapter 5, and these are

(i) The propagation environment is time invariant.

(ii) The components of the signal vector $\boldsymbol{X}_1(n)$ should be independent and identically distributed (iid).

(iii) All signals are zero mean and statistically stationary at least to the second order.

Recalling equation (5.1), the overall error signal of the LLMS algorithm is expressed as

$$e_{\mathrm{LLMS}}(n) = e_1(n) - e_2(n-1) \tag{D.1}$$

As given in (5.5), the expected values of e^2_{LLMS} is

$$\begin{aligned}
\xi_{\mathrm{LLMS}}(n) &\triangleq \mathrm{E}\left[\left|e_{\mathrm{LLMS}}(n)\right|^2\right] = \mathrm{E}\left[\left|e_1(n) - e_2(n-1)\right|^2\right] \\
&= E\left[\left|d_1(n) - \boldsymbol{W}_1^H(n)\boldsymbol{X}_1(n) - e_2(n-1)\right|^2\right] \\
&= \mathrm{E}\left[\left|D_{\mathrm{L}}(n)\right|^2\right] + \boldsymbol{W}_1^H(n)\boldsymbol{Q}_{\mathrm{L}}(n)\boldsymbol{W}_1(n) \\
&\quad - \mathrm{E}\left[D_{\mathrm{L}}(n)\boldsymbol{X}_1^H(n)\boldsymbol{W}_1(n) + D_{\mathrm{L}}^*(n)\boldsymbol{W}_1^H(n)\boldsymbol{X}_1(n)\right],
\end{aligned} \tag{D.2}$$

Referring to Figure 5-1, the error $e_i(n)$ is given by $e_i(n) = d_i(n) - \boldsymbol{W}_i^H(n)\boldsymbol{X}_i(n)$ where $i = 1$ for the LMS_1 algorithm, and 2 for the LMS_2 algorithm; $\boldsymbol{X}_i(\cdot)$, $\boldsymbol{W}_i(\cdot)$ and $d_i(n)$ represent the input signal, weight

vectors and reference signal associated with the n^{th} LMS algorithm section respectively, $\mathrm{E}[\cdot]$ denotes expectation, $|\bullet|$ signifies modulus, * stands for conjugate operator, $(\bullet)^H$ denotes the Hermitian matrix of $(\bullet)$, $D_{\mathrm{L}}(n) = d_1(n) - e_2(n-1)$, and $\boldsymbol{Q}_{\mathrm{L}}(n)$ is the correlation matrix of the input signals given by $\boldsymbol{Q}_{\mathrm{L}}(n) = \mathrm{E}\left[\boldsymbol{X}_1(n)\boldsymbol{X}_1{}^H(n)\right]$.

Consider the first term on the right hand side (RHS) of (D2). It can be expressed as

$$\begin{aligned}\mathrm{E}\left[|D_{\mathrm{L}}(n)|^2\right] &= \mathrm{E}\left[|d_1(n) - e_2(n-1)|^2\right] \\ &= \mathrm{E}\left[|d_1(n)|^2\right] + \mathrm{E}\left[|e_2(n-1)|^2\right] \\ &\quad - \mathrm{E}\left[d_1(n)e_2^*(n-1) + d_1^*(n)e_2(n-1)\right]\end{aligned} \tag{D.3}$$

where * stands for the complex conjugate operator.

With $d_1(n)$ and $e_2(n-1)$ being zero mean and uncorrelated based on the assumptions (ii) and (iii) of Chapter 5, the last RHS term of (D3) is therefore equal to zero. This gives

$$\mathrm{E}\left[|D_{\mathrm{L}}(n)|^2\right] = \mathrm{E}\left[|d_1(n)|^2\right] + \mathrm{E}\left[|e_2(n-1)|^2\right] \tag{D.4}$$

Using $e_2(n) = d_2(n) - \boldsymbol{W}_2^H(n)\boldsymbol{X}_2(n) = d_2(n) - y_{\mathrm{LLMS}}(n)$, the last RHS term of (D4) becomes

$$\begin{aligned}\mathrm{E}\left[|e_2(n-1)|^2\right] &= \mathrm{E}\left[|d_2(n-1)|^2\right] + \mathrm{E}\left[|y_{\mathrm{LLMS}}(n-1)|^2\right] \\ &\quad - \mathrm{E}\left[d_2^*(n-1)y_{\mathrm{LLMS}}(n-1) + d_2(n-1)y_{\mathrm{LLMS}}^*(n-1)\right]\end{aligned} \tag{D.5}$$

Assume $d_2(n) = d_1(n)$, where d_2 is stationary so that $d_2(n-1) = d_2(n)$, and let $y_{\mathrm{LLMS}} = \boldsymbol{W}_{\mathrm{LLMS}}^H\boldsymbol{X}_1$, where $\boldsymbol{W}_{\mathrm{LLMS}}^H = \boldsymbol{W}_2^H\boldsymbol{\mathcal{F}}_{\mathrm{L}}\boldsymbol{W}_1^H$, (D5) can be rewritten as

$$\mathrm{E}\left[\left|e_2(n-1)\right|^2\right]=\mathrm{E}\left[\left|d_2(n)\right|^2\right]-\boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{L}}(n-1)$$
$$-\boldsymbol{Z}_{\mathrm{L}}^{H}(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1)+\boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Q}_{\mathrm{L}}(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1) \quad \text{(D.6)}$$

where $\boldsymbol{Z}_{\mathrm{L}}(n)=\mathrm{E}\left[\boldsymbol{X}_1(n)d_2^{*}(n)\right]$ corresponds to the input signal cross-correlation vector.

Substituting (D6) in (D4), the first term on the RHS of (D2) becomes

$$\mathrm{E}\left[\left|D_{\mathrm{L}}(n)\right|^2\right]=2\mathrm{E}\left[\left|d_1(n)\right|^2\right]-\boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{L}}(n-1)$$
$$-\boldsymbol{Z}_{\mathrm{L}}^{H}(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1)+\boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Q}_{\mathrm{L}}(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1) \quad \text{(D.7)}$$

Since $d_2(n)=d_1(n),$ the last RHS term of (D2) may be written as

$$\mathrm{E}\left[D_{\mathrm{L}}(n)\boldsymbol{X}_1^{H}(n)\boldsymbol{W}_1(n)+D_{\mathrm{L}}^{*}(n)\boldsymbol{W}_1^{H}(n)\boldsymbol{X}_1(n)\right]=$$
$$\boldsymbol{Z}_{\mathrm{L}}^{H}(n)\boldsymbol{W}_1(n)+\boldsymbol{W}_1^{H}(n)\boldsymbol{Z}_{\mathrm{L}}(n)-\mathrm{E}\left[e_2(n-1)\boldsymbol{X}_1^{H}(n)\right]\boldsymbol{W}_1(n) \quad \text{(D.8)}$$
$$-\boldsymbol{W}_1^{H}(n)\mathrm{E}\left[e_2^{*}(n-1)\boldsymbol{X}_1(n)\right]$$

Again, applying the assumptions (ii) and (iii) given in Chapter 5, the last two terms of (D8) are equal to zero, therefore (D8) becomes

$$\mathrm{E}\left[D_{\mathrm{L}}(n)\boldsymbol{X}_1^{H}(n)\boldsymbol{W}_1(n)+D_{\mathrm{L}}^{*}(n)\boldsymbol{W}_1^{H}(n)\boldsymbol{X}_1(n)\right]=\boldsymbol{Z}_{\mathrm{L}}^{H}(n)\boldsymbol{W}_1(n)+\boldsymbol{W}_1^{H}(n)\boldsymbol{Z}_{\mathrm{L}}(n) \quad \text{(D.9)}$$

As a result, the mean square error $\xi_{\mathrm{LLMS}}(n)$ as specified by (D2) can be rewritten to include the results of (D7) and (D9) to become

$$\xi_{\mathrm{LLMS}}(n)=\mathrm{E}\left[\left|d_1(n)\right|^2\right]+\mathrm{E}\left[\left|d_2(n-1)\right|^2\right]$$
$$+\boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Q}_{\mathrm{L}}(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1)-\boldsymbol{Z}_{\mathrm{L}}^{H}(n)\boldsymbol{W}_1(n)$$
$$-\boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{L}}(n-1)-\boldsymbol{Z}_{\mathrm{L}}^{H}(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1) \quad \text{(D.10)}$$
$$-\boldsymbol{W}_1^{H}(n)\boldsymbol{Z}_{\mathrm{L}}(n)+\boldsymbol{W}_1^{H}(n)\boldsymbol{Q}_{\mathrm{L}}(n)\boldsymbol{W}_1(n)$$

Differentiating (D10) with respect to the weight vector $\boldsymbol{W}_1^{H}(n),$ and by

equating the results to zero, we obtain the optimal weight vector as

$$\boldsymbol{W}_{opt1}(n) = \boldsymbol{Q}_{\mathrm{L}}^{-1}(n)\boldsymbol{Z}_{\mathrm{L}}(n) \tag{D.11}$$

where $\boldsymbol{X}_1$ is well excited and $\boldsymbol{Q}_{\mathrm{L}}$ could be considered as a full rank matrix.

Substituting equation (D.11) in equation (D.10) and using $\boldsymbol{W}_{\mathrm{LLMS}} = \boldsymbol{W}_1\boldsymbol{\mathcal{F}}_{\mathrm{L}}^{H}\boldsymbol{W}_2$, gives

$$\begin{aligned}\xi_{\mathrm{LLMS,min}}(n) = {} & \mathrm{E}\left[\left|d_1(n)\right|^2\right] + \mathrm{E}\left[\left|d_2(n-1)\right|^2\right] \\ & + \boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Q}_{\mathrm{L}}(n-1)\boldsymbol{W}_1(n-1)\boldsymbol{\mathcal{F}}_{\mathrm{L}}^{H}\boldsymbol{W}_2(n-1) - \boldsymbol{Z}_{\mathrm{L}}^{H}(n)\boldsymbol{W}_{opt1}(n) \\ & - \boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{L}}(n-1) - \boldsymbol{Z}_{\mathrm{L}}^{H}(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1) \\ & - \boldsymbol{W}_{opt1}^{H}(n)\,\boldsymbol{Z}_{\mathrm{L}}(n) + \boldsymbol{W}_{opt1}^{H}(n)\boldsymbol{Z}_{\mathrm{L}}(n)\end{aligned} \tag{D.12}$$

Simplification of equation (D.12) yields the minimum mean square error (MSE) such that

$$\begin{aligned}\xi_{\mathrm{LLMS,min}} = {} & \mathrm{E}\left[\left|d_1(n)\right|^2\right] + \mathrm{E}\left[\left|d_2(n-1)\right|^2\right] \\ & - \boldsymbol{Z}_{\mathrm{L}}^{H}(n)\boldsymbol{W}_{opt1}(n) - \boldsymbol{Z}_{\mathrm{L}}^{H}(n-1)\boldsymbol{W}_{\mathrm{LLMS}}(n-1) \\ & + \boldsymbol{W}_{\mathrm{LLMS}}^{H}(n-1)\boldsymbol{Z}_{\mathrm{L}}(n-1)\left\{\boldsymbol{\mathcal{F}}^{H}\boldsymbol{W}_2(n-1) - 1\right\}\end{aligned} \tag{D.13}$$

APPENDIX E

DERIVATION OF THE VARIANCE OF THE QUANTIZATION ERROR

This appendix analyses the variance of the quantization error associated with linear amplitude quantization. For a uniformly distributed signal, $v(n)$, $v \in \left[\frac{-q}{2}, \frac{q}{2}\right]$, where q is the quantization step size. In this case, $v(n)$ has a mean of zero, i.e., $\mathrm{E}[v] = 0$, and the variance σ_q^2 of $v(n)$ is given by

$$\sigma_q^2 = \mathrm{E}\left[v^2(n)\right] - \left(\mathrm{E}[v(n)]\right)^2 = \mathrm{E}\left[v^2(n)\right] \tag{E.1}$$

Since $v(n)$ is uniformly distributed, its probability density function $P_V(v)$ is given as

$$P_V(v) = \begin{cases} 1/q \ , & |v| \le q/2 \\ 0 \ , & |v| \ge q/2 \end{cases} \tag{E.2}$$

Therefore, the variance σ_q^2 in (E.1) can be obtained as

$$\sigma_q^2 = \int_{-q/2}^{q/2} v^2 \, P_V(v) \, dv = \frac{1}{q} \int_{-q/2}^{q/2} v^2 \, dv = \frac{1}{q} \left. \frac{v^3}{3} \right|_{-q/2}^{q/2} = \frac{q^2}{12} \tag{E.3}$$

For a signal of $\pm 1\ \mathrm{V}$ amplitude range, its quantization error using an N_b–bits wordlength is given by

$$q = \frac{2}{2^{N_b} - 1} \approx 2^{1-N_b} \tag{E.4}$$

By substituting (E.4) in (E.3), the variance of the quantization error, σ_q^2, can

be obtained as

$$\sigma_q^2 = \frac{2^{2(1-N_b)}}{12} \tag{E.5}$$

REFERENCES

[1] N. A. Mohamed and J. G. Dunham, "Adaptive beamforming for DS-CDMA using conjugate gradient algorithm in a multipath fading channel," in *Proc. IEEE Emerging Technologies Symposium on Wireless Communications and Systems*, Richardson, USA, pp. 1-5, April 1999.

[2] S. Chandran, "Performance of adaptive antenna arrays in the presence of varying noise power in WiMAX applications," in *Proc. IET International Conference on Wireless, Mobile and Multimedia Networks*, Mumbai, India, pp. 3-5, Jan. 2008.

[3] L. Qinghua, L. Guangjie, L. Wookbong, L. Moon-il, D. Mazzarese, B. Clerckx, and L. Zexian, "MIMO techniques in WiMAX and LTE: a feature overview," *IEEE Communications Magazine*, vol. 48, pp. 86-92, May 2010.

[4] G. Amitava, R. Rapeepat, M. Bishwarup, M. Nitin, and T. Tim, "LTE-advanced: next-generation wireless broadband technology," *IEEE Wireless Communications*, vol. 17, pp. 10-22.

[5] C. Ball, T. Hindelang, I. Kambourov, and S. Eder, "Spectral efficiency assessment and radio performance comparison between LTE and WiMAX," in *Proc. 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Cannes, France, pp. 1-6, Sept. 2008.

[6] F.-B. Ueng, J.-D. Chen, and S.-H. Cheng, "Smart antenna for multiuser DS/CDMA communication in multipath fading channels," *IEICE Trans. on Communications*, vol. E88, pp. 2944-2954, Jul. 2005.

[7] C. Schuldt, F. Lindstrom, and I. Claesson, "A low-complexity delayless selective subband adaptive filtering algorithm," *IEEE Trans. on Signal Processing*, vol. 56, pp. 5840-5850, 2008.

[8] A. Gupta and S. Joshi, "Variable Step-Size LMS Algorithm for Fractal Signals," *IEEE Trans. on Signal Processing*, vol. 56, pp. 1411-1420, 2008.

[9] J. M. Górriz, J. Ramírez, S. Cruces-Alvarez, D. Erdogmus, C. G. Puntonet, and E. W. Lang, "Speech enhancement in discontinuous transmission systems using the constrained-stability least-mean-squares algorithm," *Jornal of Acoustical Society of America*, vol. 124(6), pp. 3669-3683, 2008.

[10] k. Zou and X. Zhao, "A new modified robust variable step size LMS algorithm," in *Proc. 4th IEEE Conference on Industrial Electronics and Applications* Xian, China, pp. 2699-2703, May 2009.

[11] S. Song and K. M. Sung, "Reduced Complexity Self-Tuning Adaptive Algorithms in Application to Channel Estimation," *IEEE Trans. on Communications*, vol. 55, pp. 1448-1452, 2007.

[12] C. Paleologu, J. Benesty, and S. Ciochina, "A robust variable forgetting factor recursive least-squares algorithm for system identification," *IEEE Signal Processing Letters*, vol. 15, pp. 597-600, 2008.

[13] L. Weifeng, P. Il, W. Yiwen, and J. C. Principe, "Extended kernel recursive least squares algorithm," *IEEE Trans. on Signal Processing*, vol. 57, pp. 3801-3814, 2009.

[14] Z. Shengkui, "Performance analysis and enhancements of adaptive algorithms and their applications," PhD, School of computer engineering, Nanyang technological university, 2009

[15] Z. Rong, "Simulation of adaptive array algorithms for CDMA systems," M.Sc., Electrical Engineering, State University, Blacksburg, Virginia, 1996

[16] C. Balanis, *Antenna theory analysis and design*, 3rd ed. New York: Hoboken, N.J. : Wiley-Interscience, 2005.

[17] P. Petrus, "Novel adaptive array algorithms and their impact on cellular system capacity," Ph.D., Electrical Engineering, Faculty of the Virginia Polytechnic Institute and State University, Virginia, USA, 1997

[18] F. Alam, "Space time processing for third generation CDMA systems," Ph.D., Electrical Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, 2002

[19] A. Der and G. Leus, *Signal processing for communications*. The Netherlands Delft university of technology, 2005.

[20] R. A. Monzingo and T. W. Miller, *Introduction to adaptive arrays*. New York: SciTech Publishing, Inc., 2004.

[21] J. Sahalos, *Orthogonal methods for array synthesis: Theory and the ORAMA computer tool*. Greece: John Wiley & Sons Ltd, 2006.

[22] M. Skolnik, "Resolution of angular ambiguities in radar array antennas with widely-spaced elements and grating lobes," *IRE Trans. on Antennas and Propagation*, vol. 10, pp. 351-352, 1962.

[23] S. Orfanidis, *Electromagnetic waves and antennas*: Rutgers University, 2008.

[24] J. Litva and T. K.-Y. Lo., *Digital beamforming in wireless communications*. London: Artech House, 1996.

[25] B. Allen and M. Ghavami, *Adaptive array systems fundamentals and applications*. London, UK: John Wiley & Sons, 2005.

[26] P. Ioannides and C. A. Balanis, "Uniform circular arrays for smart antennas," *IEEE Antennas and Propagation Magazine*, vol. 47, pp. 192-206, 2005.

[27] T. B. Seow, "Uniform circular antenna array applications in coded DS-CDMA mobile communication systems," M.Sc., Electrical Engineering, Nanyang Technological University, Monterey, California, 2003

[28] S. Rani, P. V. Subbaiah, and K. C. Reddy, "Music and LMS algorithms for a smart antenna system," in *Proc. International Conference on Information and Communication Technology in Electrical Sciences*, Chennai, India, pp. 965-969, Dec. 2007.

[29] L. Zhenwei, "Antenna array pattern null steering - A theoretical introduction," *Journal of Electronics (China)*, vol. 1, pp. 217-224, 1984.

[30] H. L. V. Trees, *Optimum Array Processing*, vol. Part IV: A John Wiley & Sons, 2002

[31] G. Tsoulos, J. McGeehan, and M. Beach, "Space division multiple access (SDMA) field trials. I. Tracking and BER performance," *IEE Proceedings of Radar, Sonar and Navigation*, vol. 145, pp. 73-78, 1998.

[32] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS

adaptive filter," *Proceedings of the IEEE*, vol. 64, pp. 1151-1162, 1976.

[33] S. Anderson, M. Millnert, M. Viberg, and B. Wahlberg, "An adaptive array for mobile communication systems," *IEEE Trans. on Vehicular Technology*, vol. 40, pp. 230-236, 1991.

[34] E. Eweda, "Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels," *IEEE Trans. on Signal Processing*, vol. 42, no.11, pp. 2937-2944, 1994.

[35] A. O. Boukalov and S. G. Haggman, "System aspects of smart-antenna technology in cellular wireless communications-an overview," *IEEE Trans. on Microwave Theory and Techniques*, vol. 48, pp. 919-929, 2000.

[36] I. Motorola, "Driving 4G: WiMAX & LTE". vol. 2010, 2007.

[37] S. Bellofiore, J. Foutz, C. A. Balanis, and A. S. Spanias, "Smart-antenna system for mobile communication networks .Part 2. Beamforming and network throughput," *IEEE Antennas and Propagation Magazine*, vol. 44, pp. 106-114, 2002.

[38] M. Joho and H. Mathis, "Performance comparison of combined blind/non-blind source separation algorithms," in *Proc. 1st international workshop on independent component analysis and blind signal separation*, Aussois, France, pp. 139-142, Jan. 1999.

[39] S. Talwar, M. Viberg, and A. Paulraj, "Blind separation of synchronous co-channel digital signals using an antenna array. I. Algorithms," *IEEE Trans. on Signal Processing*, vol. 44, pp. 1184-1197, 1996.

[40] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, Eugene Dong, Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, pp. 1692-1716, 1975.

[41] K. Joonwan and A. D. Poularikas, "Performance analysis of the adjusted step size NLMS algorithm," in *Proc. 36th Southeastern Symposium on System Theory*, Atlanta, Georgia, pp. 467-471, 2004.

[42] J. Connor, "A study of despread-respread multitarget adaptive algorithms in AWGN channel," M.Sc., Electrical and Computer Engineering, The Florida State university, Florida, 2005

[43] B. Widrow, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems," *IEEE Proceedings*, vol. 55, pp. 2143-2159, 1970.

[44] S. Koike, "A class of adaptive step-size control algorithms for adaptive filters," *IEEE Trans. on Signal Processing*, vol. 50, pp. 1315-1326, 2002.

[45] B. Widrow and J. McCool, "A comparison of adaptive algorithms based on the methods of steepest descent and random search," *IEEE Trans. on Antennas and Propagation*, vol. 24, pp. 615-637, 1976.

[46] S. Attallah, "The wavelet transform-domain LMS adaptive filter with partial subband-coefficient updating," *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 53, pp. 8-12, 2006.

[47] S. Narayan, A. Peterson, and M. Narasimha, "Transform domain LMS algorithm," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 31, pp. 609-615, 1983.

[48] P. A. C. Lopes, G. Tavares, and J. B. Gerald, "A new type of normalized LMS algorithm based on the Kalman filter," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Hawaii, U.S.A., pp. III-1345-III-1348, May 2007.

[49] V. H. Nascimento, "The normalized LMS algorithm with dependent noise," in *Proc. Anais do 19° Simpòsio Brasileiro de Telecomunicações*, Fortaleza, Brazil, 2001.

[50] D. T. M. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms," *IEEE Trans. on Signal Processing*, vol. 41, pp. 2811-2825, 1993.

[51] F. Casco, R. C. Medina-Ramirez, M. Lopez-Guerrero, and C. Jalpa-Villanueva, "VS-SC: a variable step size NLMS algorithm," in *Proc. Canadian Conference on Electrical and Computer Engineering*, Vancouver, Canada, pp. 896-899, Apr. 2007.

[52] N. J. Bershad, J. C. M. Bermudez, and J. Y. Tourneret, "An affine combination of two LMS adaptive filters - transient mean-square analysis," *IEEE Trans. on Signal Processing*, vol. 56, pp. 1853-1864, 2008.

[53] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. on Signal Processing*, vol. 54, pp. 1078-1090, 2006.

[54] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Trans. on Signal Processing*, vol. 40, pp. 1633-1642, 1992.

[55] W. Mikhael, F. Wu, L. Kazovsky, G. Kang, and L. Fransen, "Adaptive filters with individual adaptation of parameters," *IEEE Trans. on Circuits and Systems*, vol. 33, pp. 677-686, 1986.

[56] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-type algorithm: analysis and simulations," *IEEE Trans. on Signal Processing*, vol. 45, pp. 631-639, 1997.

[57] J. G. F. Zipf, O. J. Tobias, and R. Seara, "A VSSLMS algorithm based on error autocorrelation", in *16th European Signal Processing Conference* Lausanne, Switzerland, Aug. 2008.

[58] I. H. Tarek, "A simple variable step size LMS adaptive algorithm," *International Journal of Circuit Theory and Applications*, vol. 32, pp. 523-536, 2004.

[59] S. Karni and G. Zeng, "A new convergence factor for adaptive filters," *IEEE Trans. on Circuits and Systems*, vol. 36, pp. 1011-1012, 1989.

[60] T. J. Shan and T. Kailath, "Adaptive algorithms with an automatic gain control feature," *IEEE Trans. on Circuits and Systems*, vol. 35, pp. 122-127, 1988.

[61] V. J. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. on Signal Processing*, vol. 41, pp. 2075-2087, 1993.

[62] A. Wee-Peng and B. Farhang-Boroujeny, "A new class of gradient adaptive step-size LMS algorithms," *IEEE Trans. on Signal Processing*, vol. 49, pp. 805-810, 2001.

[63] L. Ning, Z. Yonggang, and H. Yanling, "A new variable tap-length LMS algorithm with variable step size," in *Proc. IEEE International Conference on Mechatronics and Automation*, Kagawa, Japan pp. 525-529, Aug. 2008.

[64] S. Gazor and K. Shahtalebi, "A new NLMS algorithm for slow noise magnitude variation," *IEEE Signal Processing Letters*, vol. 9, pp. 348-351, 2002.

[65] M. Ferrer, M. d. Diego, A. Gonzalez, and G. Pi˜nero, "Convex combination of affine projection algorithms," in *Proc. 17th European Signal Processing Conference*, Glasgow, Scotland, pp. 431-435, Aug. 2009.

[66] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 33, pp. 222-230, 1985.

[67] S. Zhao, "Performance analysis and enhancements of adaptive algorithms and their applications," PhD, School of computer engineering, Nanyang technological university, 2009

[68] S. Zhao, Z. Man, and S. Khoo, "A fast variable step-size LMS algorithm with system identification," in *Proc. 2nd IEEE Conference on Industrial Electronics and Applications*, Harbin, China, pp. 2340-2345, May 2007.

[69] Z. Yonggang, J. A. Chambers, W. Wenwu, P. Kendrick, and T. J. Cox, "A new variable step-size LMS algorithm with robustness to nonstationary noise," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Hawai, USA, pp. 1349-1352, May 2007.

[70] K. Dogançay, "Complexity considerations for transform-domain adaptive filters," *Signal Processing*, vol. 83, pp. 1177-1192, 2003.

[71] S. Attallah and S. W. Liaw, "Analysis of DCTLMS algorithm with a selective coefficient updating," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, pp. 628-632, 2001.

[72] S. Attallah, "The wavelet transform-domain LMS algorithm: a more practical approach," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, pp. 209-213, 2000.

[73] E. M. Lobato, O. J. Tobias, and R. Seara, "Stochastic modeling of the transform-domain εLMS algorithm for correlated Gaussian data," *IEEE Trans. on Signal Processing*, vol. 56, pp. 1840-1852, 2008.

[74] M. Atinati, A. Rastegarnia, and T. Y. Rezaii, "A transform domain based least mean mixed-norm algorithm to improve adaptive beamforming," in *Proc. 2nd International Conference on Electrical Engineering*, Hong Kong, pp. 1-4, Mar. 2008.

[75] F. Riera-Palou, J. M. Noras, and D. G. M. Cruickshank, "Linear equalisers, with dynamic and automatic length selection," *Electronics Letters*, vol. 37, pp. 1553-1554, 2001.

[76] S. Haykin, A. Sayed, J. R. Zeidler, P. Yee, and P. C. Wei, "Adaptive tracking of linear time-variant systems by extended RLS algorithms," *IEEE trans. on Signal Processing*, vol. 45, pp. 1118-1128, 1997.

[77] C. A. Balanis and P. I. Ioannides, *Introduction to smart antennas (synthesis lectures on antennas)*: Morgan & Claypool Publishers, 2007.

[78] W. Junfeng, "A variable forgetting factor RLS adaptive filtering algorithm," in *Proc. 3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, Beijing, China, pp. 1127-1130, Oct. 2009.

[79] W. G. Najm, "Constrained least squares in adaptive, imperfect arrays," *IEEE Trans. on Antennas and Propagation*, vol. 38, pp. 1874-1878, 1990.

[80] S. Haykin, *Adaptive filter theory*, Fourth ed. N.J.: Prentic Hall, 2002.

[81] M. Chansarkar and B. Desai, "A fast approximate RLS algorithm," in *Proc. IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering*, Beijnn, China, pp. 532-536 vol.3, Oct. 1993.

[82] G. Carayannis, D. Manolakis, and N. Kalouptsidis, "A fast sequential algorithm for least-squares filtering and prediction," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 31, pp. 1394-1402, 1983.

[83] T.-K. Woo, "HRLS: a more efficient RLS algorithm for adaptive FIR filtering," *IEE Communication Letters*, vol. 5, pp. 81-84, Mar. 2001.

[84] J. H. Husoy and M. S. E. Abadi, "A comparative study of some simplified RLS-type algorithms," in *Proc. 1st international symposium*

on control, communications and signal processing, Hammamet, Tunisia, pp. 705-708, Mar. 2004.

[85] K. H. Kim and E. J. Powers, "Analysis of initialization and numerical instability of fast RLS algorithms," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ontario, pp. 1861-1864, May 1991.

[86] M. Chansarkar, B. Desai, and V. Rao, "A comparative study of the approximate RLS with LMS and RLS algorithms," in *Proc. 4th IEEE Region 10 International Conference*, Bombay, India, pp. 255-258, Nov. 1989.

[87] G. Kubin, "Stabilization of the RLS algorithm in the absence of persistent excitation," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, USA, pp. 1369-1372, Apr. 1988.

[88] T.-s. Li, K. Tian, and W.-x. Li, "Method for improving RLS algorithms," *Journal of Marine Science and Application*, vol. 6, pp. 68-70, 2007.

[89] S. Song and K.-M. Sung, "Reduced complexity self-tuning adaptive algorithms in application to channel estimation," *IEEE Trans. on Communications*, vol. 55, pp. 1448-1452, 2007.

[90] S. Lee, J.-s. Lim, and K.-M. Sung, "A low-complexity AFF-RLS algorithm using a normalization technique," *IEICE Electronic Express*, vol. 6, pp. 1774-1780, 2009.

[91] R. Yonezawa, K. Hirata, T. Kirimoto, and I. Chiba, "A combination of two adaptive algorithms SMI and CMA," in *Proc. IEEE Global Telecommunications Conference*, Sydney, Australia, pp. 3181-3186, Nov. 1998.

[92] S. E. El-Khamy and A. M. Gaballa, "Smart MSINR-MMA arrays for high data rate DS-CDMA systems under carrier frequency offset conditions," in *Proc. National Radio Science Conference*, Tanta, Egypt, pp. 1-8, Mar. 2008.

[93] K. Banovic, E. Abdel-Raheem, and M. A. S. Khalid, "Hybrid methods for blind adaptive equalization: new results and comparisons," in *Proc. 10th IEEE Symposium on Computers and Communications*, Cartagena, Spain, pp. 275-280, Jun. 2005.

[94] M. Martone, "Cumulant-based adaptive multichannel filtering for wireless communication systems with multipath RF propagation using antenna arrays," *IEEE Trans. on Vehicular Technology*, vol. 47, pp. 377-391, 1998.

[95] L. Zhensu, H. Mingqiu, Z. Tongfeng, D. Juan, and H. Shi, "Improved concurrent constant modulus algorithm and soft decision directed algorithm based on fractionally spaced equalization," in *Proc. International Conference on Wireless Communications, Networking and Mobile Computing*, Wuhan, China, pp. 619-622, Sept. 2005.

[96] X. Ning, Z. Yuanping, X. Minghua, and T. Wenming, "Fast blind adaptive beamforming algorithm with interference suppression," *IEEE Trans. on Vehicular Technology*, vol. 57, pp. 1985-1988, 2008.

[97] A. Ikhlef and D. Le Guennec, "A simplified constant modulus algorithm for blind recovery of MIMO QAM and PSK signals: A criterion with convergence analysis," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, p. 13, 2007.

[98] W. Lei and R. C. de Lamare, "Constrained constant modulus RLS-based blind adaptive beamforming algorithm for smart antennas," in *Proc. 4th International Symposium on Wireless Communication Systems*, Trondheim, Norway, pp. 657-661, Oct. 2007.

[99] V. Krishnamurthy, G. Yin, and S. Singh, "Adaptive step-size algorithms for blind interference suppression in DS/CDMA systems," *IEEE Trans. on Signal Processing*, vol. 49, pp. 190-201, 2001.

[100] U. Fang-Biau, C. Jun-Da, and C. Po-Yu, "Blind adaptive DS/CDMA receivers in multipath fading channels," in *Proc. IEEE 60th Vehicular Technology Conference*, Los Angeles, USA, pp. 3768-3772, Sept. 2004.

[101] O. A. Abu-Ella and B. A. El-Jabu, "Performance improvement of blind adaptive beamforming algorithms using pre-filtering technique," in *Proc. IEEE Aerospace Conference*, Big Sky, USA, pp. 1-4, Mar. 2007.

[102] B. Gao, C.-C. Tu, and B. Champagne, "Computationally efficient approaches for blind adaptive beamforming in SIMO-OFDM systems," in *Proc. IEEE Pacific Rim Conference on Communications,*

Computers and Signal Processing, Victoria, Canada, pp. 245-250, Aug. 2009.

[103] S. Chen, A. Wolfgang, and L. Hanzo, "Constant modulus algorithm aided soft decision directed scheme for blind space-time equalisation of SIMO channels," *IEEE Trans. on Signal Processing*, vol. 87, pp. 2587-2599, 2007.

[104] A. Swindlehurst, S. Daas, and J.Yang, "Analysis of a decision directed beamformer," *IEEE trans. of signal processing*, vol. 43, pp. 2920-2927, 1995.

[105] S. Chen, W. Yao, and L. Hanzo, "CMA and soft decision-directed scheme for semi-blind beamforming of QAM systems," in *Proc. IEEE 68th Vehicular Technology Conference*, pp. 1-5, Sept. 2008.

[106] S. Chen, A. Wolfgang, and L. Hanzo, "Constant modulus algorithm aided soft decision-directed blind space-time equalization for SIMO channels," in *Proc. IEEE 60th Vehicular Technology Conference*, Los Angeles, USA, pp. 1718-1722, Sept. 2004.

[107] Y. Kehu, T. Ohira, Z. Yimin, and C. Chong-Yung, "Super-exponential blind adaptive beamforming," *IEEE Trans. on Signal Processing*, vol. 52, pp. 1549-1563, 2004.

[108] A. Martin and A. Mansour, "Comparative study of high order statistics estimators," in *Proc. International Conference on Software, Telecommunications and Computer Networks*, Venice, Italy, pp. 511-515, Oct. 2004.

[109] B. Porat and B. Friedlander, "Direction finding algorithms based on higher-order statistics," *IEEE Trans. of Signal Processing*, vol. 39, pp. 2016-2024, 1991.

[110] L. Xinbo, S. Yaowu, and Z. Han, "RARE-cumulant algorithm for direction of arrival estimation and array calibration", in *International Conference on Advanced Infocomm Technology* Shenzhen, China: ACM, Jul. 2008.

[111] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications," *Proceedings of the IEEE*, vol. 79, pp. 278-305, 1991.

[112] O. Arikan, A. Enis Cetin, and E. Erzin, "Adaptive filtering for non-Gaussian stable processes," *IEEE Signal Processing Letters*, vol. 1, pp. 163-165, 1994.

[113] J. V. Candy, *Bayesian signal processing: classical, modern, and particle filtering methods*. California: John Wiley & Sons, Inc, 2009.

[114] C. Hsing-Hsing and C. L. Nikias, "The esprit algorithm with higher-order statistics," in *Proc. Workshop on Higher-Order Spectral Analysis*, Colorado, USA, pp. 163-168, June,1989.

[115] C. Dogan and M. Mendel, "Cumulant-based blind optimum beamforming," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 30, pp. 722-741, 1994.

[116] M. Bouzaien and A. Mansour, "HOS criteria & ICA algorithms applied to radar detection," in *Proc. 4th international Symposium on Independent Component Analysis and Blind Signal Separation*, Nara, Japan, pp. 433-438, Apr. 2003.

[117] E. Gonen and J. M. Mendel, "Optimum cumulant-based blind beamforming for coherent signals and interferences," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, Michigan, USA, pp. 1908-1911, May, 1995.

[118] A. Martin and A. Mansour, "High order statistic estimators for speech processing," in *Proc. 5th International Conference on ITS Telecommunications*, Brest, France, Jun. 2005.

[119] A. Mansour and C. Jutten, "Fourth-order criteria for blind sources separation," *IEEE Trans. on Signal Processing*, vol. 43, pp. 2022-2025, 1995.

[120] A. Mansour and N. Ohnishi, "Multichannel blind separation of sources algorithm based on cross-cumulant and the Levenberg-Marquardt method," *IEEE Trans. on Signal Processing*, vol. 47, pp. 3172-3175, 1999.

[121] A. K. Nandi, *Blind estimation using higher-order statistics*. The Netherlands: Kluwer Academic Publishers, 1999.

[122] N. S. Networks, "Advanced antenna systems for WiMAX". vol. 2009 Karaportti, Finland: www.nokiasiemensnetworks.com, 2007.

[123] M. Rezk, W. Kim, Y. Zhengqing, and F. Iskander, "Performance comparison of a novel hybrid smart antenna system versus the fully adaptive and switched beam antenna arrays," in *Proc. International Conference on Wireless Networks, Communications and Mobile Computing*, Montreal, Canada, pp. 874-878 vol.2, 2005.

[124] W. Lei, C. Yunlong, and R. C. de Lamare, "Low-complexity adaptive step size constrained constant modulus SG-based algorithms for blind adaptive beamforming," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Las Vagas, pp. 2593-2596, May 2008.

[125] S. Parkvall, E. Dahlman, A. Furuskar, Y. Jading, M. Olsson, S. Wanstedt, and K. Zangi, "LTE-advanced - evolving LTE towards IMT-advanced," in *Proc. IEEE 68th Vehicular Technology Conference*, Alberta, Canada, pp. 1-5, Sept. 2008.

[126] P. Sergeant, "Advanced antenna system (AAS) techniques for WiMAX". vol. 2009: WiMAX360, Jan. 2009.

[127] A. Osseiran, K. Zangi, D. Hui, and L. Krasny, "Interference mitigation for MIMO systems employing user-specific, linear precoding," in *Proc. IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, Cannes, France, pp. 1-6, Sept. 2008.

[128] P. E. Mogensen, T. Koivisto, K. I. Pedersen, I. Z. Kovacs, B. Raaf, K. Pajukoski, and M. J. Rinne, "LTE-Advanced: the path towards gigabit/s in wireless mobile communications," in *Proc. 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, Aalborg, Denmark, pp. 147-151, May 2009.

[129] H. Xiaojing, Y. J. Guo, and J. Bunton, "Adaptive AoA estimation and beamforming with hybrid antenna arrays," in *Proc. IEEE 70th Vehicular Technology Conference*, Alaska, USA, pp. 1-5, Sept. 2009.

[130] D. Martin-Sacristan, J. F. Monserrat, J. Cabrejas-Penuelas, D. Calabuig, S. Garrigas, and N. Cardona, "On the way toward forth-generation mobile: 3GPP LTE-advanced," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, Jun. 2009.

[131] H. Yikun, "WiMAX dynamnic beamforming antenna," *IEEE Aerospace and Electronic Systems Magazine*, vol. 23, pp. 26-31, 2008.

[132] L. Ye and N. R. Sollenberger, "Adaptive antenna arrays for OFDM systems with cochannel interference," *IEEE Trans. on Communications*, vol. 47, pp. 217-229, 1999.

[133] C. Yung-Fang and L. Chih-Peng, "Adaptive beamforming schemes for interference cancellation in OFDM communication systems," in *Proc. IEEE 59th Vehicular Technology Conference*, Milan, Italy, pp. 103-107, May 2004.

[134] J. L. Ming, "Performance investigantion of adaptive filter algorithms and their implementation for MIMO systems," M.Sc., Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand, 2005

[135] S. B. Gelfand, W. Yongbin, and J. V. Krogmeier, "The stability of variable step-size LMS algorithms," *IEEE Trans. on Signal Processing* vol. 47, pp. 3277-3288, 1999.

[136] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. Maryland, USA: The Johns Hopkins University Press, 1996.

[137] Q. Zhang, Q. Xu, and W. Zhu, "A new EVM calculation method for broadband modulated signals and simulation," in *Proc. 8th International Conference on Electronic Measurement and Instruments*, Beijing, China, pp. 2-661-2-665, July 2007.

[138] H. Arslan and H. Mahmoud, "Error vector magnitude to SNR conversion for nondata-aided receivers," *IEEE Trans. on Wireless Communications*, vol. 8, pp. 2694-2704, 2009.

[139] Y. Li and X. Wang, "A modified VS LMS algorithm," in *Proc. The 9th International Conference on Advanced Communication Technology*, Seoul, Korea, pp. 615-618, Feb. 2007.

[140] S. Zhao, Z. Man, S. Khoo, and H. R. Wu, "Variable step-size LMS algorithm with a quotient form," *IEEE Trans. on Signal Processing*, vol. 89, pp. 67-76, 2009.

[141] P. S. Chang and A. N. Willson, Jr., "A roundoff error analysis of the normalized LMS algorithm," in *Proc. 29th Asilomar Conference on*

Signals, Systems and Computers, California, USA, pp. 1337-1341, Oct. 1995.

[142] D.-Z. Feng, X.-D. Zhang, D.-X. Chang, and W. X. Zheng, "A fast recursive total least squares algorithm for adaptive FIR filtering," *IEEE Trans. on Signal Processing*, vol. 52, pp. 2729-2737, 2004.

[143] J. A. Srar and K. S. Chung, "Adaptive RLMS algorithm for antenna array beamforming," in *Proc. IEEE Region 10 Conference*, Singapore, pp. 1-6, Nov. 2009.

[144] J. Srar, K. Chung, and A. Mansour, "Adaptive array beamforming using a combined LMS-LMS algorithm," in *Proc. 2010 IEEE Aerospace Conference*, pp. 1-10, Mar. 2010.

[145] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proceedings of the IEEE*, vol. 55, pp. 2143-2159, 1967.