

**TWO-DIMENSIONAL ARMA PARAMETER
IDENTIFICATION USING TWO-CHANNEL AR
LATTICE APPROACH**

**Ph.D. Thesis by
Nursen YILDIZ SARI, M.Sc.**

Department : **Electronics and Communication
Engineering**

Programme: **Electronics and Communication
Engineering**

DECEMBER 2005

**TWO-DIMENSIONAL ARMA PARAMETER
IDENTIFICATION USING TWO-CHANNEL AR
LATTICE APPROACH**

**Ph.D. Thesis by
Nursen YILDIZ SARI, M.Sc.
494.D.029**

**Date of submission : 20 January 2004
Date of defence examination: 30 December 2005**

**Supervisor (Chairman): Prof. Dr. Ahmet Hamdi KAYRAN
Members of the Examining Committee Prof.Dr. Erdal PANAYIRCI (Bilkent Un.)
Prof.Dr. Mehmet Tahir ÖZDEN (Deniz
Harp Okulu)
Prof.Dr. Ümit AYGÖLÜ (İTÜ)
Prof.Dr. Serhat SEKER (İTÜ)**

DECEMBER 2005

**İKİ-KANALLI AR KAFES YAKLASIMI
KULLANARAK İKİ-BOYUTLU ARMA PARAMETRE
TANILAMA**

**DOKTORA TEZİ
Y. Müh. Nursen YILDIZ SARI
494.D.029**

Tezin Enstitüye Verildiği Tarih : 20 Ocak 2004

Tezin Savunulduğu Tarih : 30 Aralık 2005

Tez Danismanı: Prof. Dr. Ahmet Hamdi KAYRAN
Diğer Jüri Üyeleri Prof.Dr. Erdal PANAYIRCI (Bilkent Ün.)
Prof.Dr. Mehmet Tahir ÖZDEN (Naval
War Academy)
Prof.Dr. Ümit AYGÖLÜ (İTÜ)
Prof.Dr. Serhat SEKER (İTÜ)

ARALIK 2005

PREFACE

This Ph.D. education phase of my life has not been an easy one, being a full-time employee, and a mother. It is with deepest appreciation that I wish to acknowledge my advisor, Prof. Dr. Ahmet Hamdi KAYRAN, for his patient assistance and understanding manners over the course of this research. He was also my advisor during my M.Sc. thesis and he is the one who introduced me to the subject of two-dimensional signal processing and lattice modeling for which I am really grateful. I also wish to thank Prof. Dr. Lamine Mili whom we met during our scholarship in Virginia Tech-USA for his encouragement and his faith in me to complete this thesis.

I have a heart full of thanks for my husband and my best friend Faruk SARI, who most encouraged me and believed in me at times when I almost gave up. My daughter, Nur Irem, has been a great inspiration and motivation for me, always trying to tolerate the play times I had to steal from her. My brothers, Erkan YILDIZ and Ferhan YILDIZ, whom I am always proud of, were my never giving-up supporters.

I am grateful to my Mom, Nuran YILDIZ, for bringing me up to be who I am, supporting me and having an endless faith in me: nothing is enough to thank her.

And for the last, certainly not the least, I dedicate everything I have tried to make in this research to my Dad, Fehmi YILDIZ, whom we lost 12 years ago, his pain still being fresh in my heart.

DECEMBER 2005

Nursen YILDIZ SARI

P.S. We are now a family of four, welcoming our little son Asim Kerem.

CONTENTS

PREFACE.....	ii
CONTENTS	iii
LIST OF ABBREVIATIONS.....	vi
LIST OF TABLES	vii
LIST OF FIGURES	x
LIST OF SYMBOLS	xiv
SUMMARY.....	xvi
ÖZET	xxi
1. INTRODUCTION.....	1
2. TWO-DIMENSIONAL SYSTEM IDENTIFICATION: AN OVERVIEW OF THE PROPOSED STRUCTURE	36
3. HYBRID LATTICE STRUCTURE	42
3.1. Two-Dimensional Single Channel AR Lattice Structure	42
3.2. Two-Channel AR Lattice Structure for ARMA Modeling	44
3.3. Hybrid Lattice Structure Design Steps	46
3.3.1. Step 1.....	46
3.3.1.1. Step 1.1.....	47
3.3.1.2. Step 1.2.....	48
3.3.2. Step 2.....	49
3.4. Estimation of the Parameter b_0	49
4. A NEW METHOD TO CALCULATE THE 2-D ARMA (M, N) MODEL PARAMETERS.....	52
4.1. Calculation of the ARMA (M, N) Model Parameters: $M=N$	52
4.1.1. Prediction of the System Input Parameters	54
4.2. Calculation of the ARMA (M, N) Model Parameters: $M > N$	55
4.3. Calculation of the ARMA (M, N) Model Parameters: $M < N$	56
5. IMPLEMENTATION OF THE PROPOSED METHOD.....	60
5.1. Implementation of Step 1.....	60
5.2. Implementation of Step 1.1: $r = 1$	61
5.3. Implementation of Step 1.2: $r = 1$	66
5.4. Implementation of Step 1.1: $r = 2$	68

5.5.	Implementation of Step 1.2: $r=2$	69
5.6.	Implementation of Step 2: $r=3$	70
5.7.	The calculation of parameter b_0	71
6.	SIMULTANEOUS IDENTIFICATION OF IMAGE AND BLUR	
PARAMETERS.....		73
6.1.	Image and Blur Models.....	73
6.2.	Hybrid Lattice Design for Simultaneous Identification of Image and Blur Parameters	75
7.	PERFORMANCE EVALUATION OF THE PROPOSED METHOD ..	77
7.1.	The Least Squares Estimates	77
7.2.	The Itakura-Saito Distance Measure	79
7.3.	The L_1 , L_2 and L_∞ Norms.....	80
7.3.1.	L_1 -norm	81
7.3.2.	L_2 -norm	81
7.3.3.	L_∞ -norm.....	82
8.	EXPERIMENTAL RESULTS	83
8.1.	Computer Experiment 1	84
8.1.1.	Objectives of Computer Experiment 1.....	85
8.1.2.	Results of Computer Experiment 1	85
8.2.	Computer Experiment 2	90
8.2.1.	Objectives of Computer Experiment 2.....	91
8.2.2.	Results of Computer Experiment 2	91
8.3.	Computer Experiment 3	96
8.3.1.	Objectives of Computer Experiment 3.....	97
8.3.2.	Results of Computer Experiment 3	97
8.4.	Computer Experiment 4	105
8.4.1.	Objectives of Computer Experiment 4.....	105
8.4.2.	Results of Computer Experiment 4	106
8.5.	Computer Experiment 5	111
8.5.1.	Objectives of Computer Experiment 5.....	111
8.5.2.	Results of Computer Experiment 5	112
8.6.	Computer Experiment 6	117
8.6.1.	Objectives of Computer Experiment 6.....	117
8.6.2.	Results of Computer Experiment 6	118
8.7.	Computer Experiment 7	127
8.7.1.	Objectives of Computer Experiment 7.....	127
8.7.2.	Results of Computer Experiment 7	128
8.8.	Computer Experiment 8	133
8.8.1.	Objectives of Computer Experiment 8.....	134
8.8.2.	Results of Computer Experiment 8	134
8.9.	Computer Experiment 9	142
8.9.1.	Objectives of Computer Experiment 9.....	143
8.9.2.	Results of Computer Experiment 9	143
8.10.	Computer Experiment 10	150
8.10.1.	Objectives of Computer Experiment 10.....	150

8.10.2.	Results of Computer Experiment 10	151
8.11.	Computer Experiment 11	157
8.11.1.	Objectives of Computer Experiment 11	157
8.11.2.	Results of Computer Experiment 11	158
8.12.	Computer Experiment 12	164
8.12.1.	Objectives of Computer Experiment 12	164
8.12.2.	Results of Computer Experiment 12	164
8.13.	Computer Experiment 13	170
8.13.1.	Objectives of Computer Experiment 13	171
8.13.2.	Results of Computer Experiment 13	171
8.14.	Computer Experiment 14	176
8.14.1.	Objectives of Computer Experiment 14	177
8.14.2.	Results of Computer Experiment 14	177
8.15.	Computer Experiment 15	183
8.15.1.	Objectives of Computer Experiment 15	183
8.15.2.	Results of Computer Experiment 15	184
8.16.	Computer Experiment 16	190
8.16.1.	Objectives of Computer Experiment 16	191
8.16.2.	Results of Computer Experiment 16	191
8.17.	Computer Experiment 17	191
8.17.1.	Objectives of Computer Experiment 17	192
8.17.2.	Results of Computer Experiment 17	192
8.18.	Computer Experiment 18	193
8.18.1.	Objectives of Computer Experiment 18	194
8.18.2.	Results of Computer Experiment 18	194
9.	CONCLUDING REMARKS	195
10.	REFERENCES	197
APPENDIX-A: DERIVATION OF EQUATIONS (4.8a)- (4.8b)		217
APPENDIX B: DERIVATION OF EQUATIONS (4.12a)- (4.12b)		220
APPENDIX C: DERIVATION OF EQUATIONS (4.17a)- (4.17b)		223
CURRICULUM VITAE		227

LIST OF ABBREVIATIONS

1-D	: One-Dimensional
2-D	: Two-Dimensional
ACF	: AutoCorrelation Function
ADPCM	: Adaptive Differential Pulse Code Modulation
AIC	: Akaike Information Criterion
AR	: Autoregressive
ARMA	: Autoregressive Moving Average
ASHP	: Asymmetric Half Plane
CODEC	: Coder-Decoder
CRLB	: Cramer-Rao Lower Bounds
FIR	: Finite Impulse Response
FLRLS	: Fast Lattice Recursive Least Squares
FQP	: First Quarter Plane
FTF	: Fast Transversal Filter
GAL	: Gradient Adaptive Lattice
GLS	: Generalized Least Squares
HOS	: Higher Order Statistics
HMHV	: Harmonic Mean Horizontal Vertical
IFT	: Inverse Fourier Transform
IIR	: Infinite Impulse Response
ISAR	: Inverse Synthetic Aperture Radar
JPE	: Joint Process Estimation
MA	: Moving Average
ML	: Maximum Likelihood
PARCOR	: Partial Correlation
PORLA	: Pure Order Recursive Ladder Algorithm
PSF	: Point Spread Function
LMS	: Least Mean Squares
LS	: Least Squares
LSI	: Linear Shift Invariant
MMSE	: Minimum Mean Square Error
MYW	: Modified Yule-Walker
QP	: Quarter Plane
RLS	: Recursive Least Squares
SAR	: Synthetic Aperture Radar
SNR	: Signal to Noise Ratio
VLSI	: Very Large Scale Integration

LIST OF TABLES

	<u>Page Number</u>
Table 8.1	A summary of the Computer Experiments. 84
Table 8.2	Computer Experiment 1 - ARMA(3,3) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 86
Table 8.3	Computer Experiment 1 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 86
Table 8.4	Computer Experiment 2 - ARMA(3,3) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 92
Table 8.5	Computer Experiment 2 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 92
Table 8.6	Computer Experiment 3 - ARMA(3,3) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 98
Table 8.7	Computer Experiment 3 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 98
Table 8.8	Computer Experiment 3 – Hybrid Lattice parameter estimates of the input AR (3) process $x(k_1, k_2)$. 99
Table 8.9	Computer Experiment 3 - Performance criteria results in estimating the AR (3) parameters of $x(k_1, k_2)$. 99
Table 8.10	Computer Experiment 4 - ARMA(4,4) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 106
Table 8.11	Computer Experiment 4 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 107
Table 8.12	Computer Experiment 5 - ARMA(8,8) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 112
Table 8.13	Computer Experiment 5 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 113
Table 8.14	Computer Experiment 6- ARMA(8,8) parameter identification results. 119
Table 8.15	Computer Experiment 6 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. 120
Table 8.16	Computer Experiment 6 – Hybrid Lattice parameter estimates of the input AR (3) process 120
Table 8.17	Computer Experiment 6 - Performance criteria results in estimating the AR (3) parameters of $x(k_1, k_2)$. 120
Table 8.18	Computer Experiment 7 - ARMA(8,8) parameter identification results. 128

Table 8.19	Computer Experiment 7 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	129
Table 8.20	Computer Experiment 8 – ARMA(15,15) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the third row. Hybrid Lattice results with no b_0 correction are shown in the second row.	134
Table 8.21	Computer Experiment 8 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the third row. Hybrid Lattice results with no b_0 correction are shown in the second row.	137
Table 8.22	Computer Experiment 9 - ARMA(15,15) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	143
Table 8.23	Computer Experiment 9 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	145
Table 8.24	Computer Experiment 10 – ARMA(3,2) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the third row. Hybrid Lattice results with no b_0 correction are shown in the second row.	151
Table 8.25	Computer Experiment 10 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. Hybrid Lattice results with no b_0 correction are shown in the second row.	152
Table 8.26	Computer Experiment 11 - ARMA(15,8) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	158
Table 8.27	Computer Experiment 11 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	160
Table 8.28	Computer Experiment 12 - ARMA(15,8) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	165
Table 8.29	Computer Experiment 12 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	166
Table 8.30	Computer Experiment 13 – ARMA(2,3) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	172
Table 8.31	Computer Experiment 13 – Performance Criteria results of Hybrid Lattice algorithm.	172
Table 8.32	Computer Experiment 14 - ARMA(3,8) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	177
Table 8.33	Computer Experiment 14 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.	178
Table 8.34	Hybrid Lattice and LS identified versus original ARMA coefficients for Computer Experiment 15.	183
Table 8.35	Performance criteria results of Hybrid Lattice and LS algorithms for Computer Experiment 15.	185
Table 8.36	Original and identified image AR model coefficients: Simulated image, $v(m,n)$ neglected.	189
Table 8.37	Original and identified values of blur PSF: Simulated image, $v(m,n)$	190

	neglected.	
Table 8.38	Original and identified values of blur PSF: Cameraman image, $v(m,n)$ neglected.	191
Table 8.39	Original and identified image AR coefficients: Cameraman image, $v(m,n)$ neglected.	191
Table 8.40	Original and identified values of blur PSF: Cameraman image, $v(m,n)$ not neglected.	193
Table 8.41	Original and identified image AR coefficients: Cameraman image, $v(m,n)$ not neglected.	193

LIST OF FIGURES

	<u>Page Number</u>
Figure 2.1 : A sample ordering for (a) the system input $x(k_1, k_2)$ and (b) system output $y(k_1, k_2)$.	37
Figure 2.2 : Initial channel inputs of the proposed hybrid 2-D lattice model to identify the parameters of an ARMA (M, N) system for (a) $M \geq N$ and (b) $M < N$.	39
Figure 3.1 : Construction of the 2-D ARMA (M, N) hybrid lattice structure with respect to the running order index r and sample processing step p .	50
Figure 5.1 : Ordering arrangements for (a) $x(k_1, k_2)$ and (b) $y(k_1, k_2)$.	60
Figure 5.2 : The block interconnection for the hybrid lattice analysis model of an ARMA (3, 2) system.	72
Figure 6.1 : ASHP image model support S_a and noncausal PSF support S_b .	74
Figure 6.2 : Block diagram representation of the input-output model of the degraded images.	75
Figure 8.1 : FQP ordering schemes of the Computer Experiment 1 - ARMA(3,3) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	85
Figure 8.2 : Computer Experiment 1- Power Spectrums plotted using (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=1$ and the data field size is 10×10 .	88
Figure 8.3 : Computer Experiment 1- Contour plots of (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=1$ and the data field size is 10×10 .	90
Figure 8.4 : Computer Experiment 2- Power Spectrums plotted using (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=0.75$ and the data field size is 10×10 .	94
Figure 8.5 : Computer Experiment 2- Contour plots of (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=0.75$ and the data field size is 10×10 .	96
Figure 8.6 : Computer Experiment 3- Power Spectrums plotted using (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=-0.8$ and the data field size is 100×100 . The system input $x(k_1, k_2)$ is an AR process of order 3.	101
Figure 8.7 : Computer Experiment 3 - Contour plots of (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=-0.8$ and the data field size is 100×100 . The system input $x(k_1, k_2)$ is an AR process of order 3.	102
Figure 8.8 : Computer Experiment 3 -:(a) Original power spectrum of the input AR(3) data field $x(k_1, k_2)$ (b) Power spectrum of $x(k_1, k_2)$ plotted using parameters identified by the Hybrid Lattice algorithm (c) Original contour plot of $x(k_1, k_2)$ (d) Identified contour plot of $x(k_1, k_2)$.	104

Figure 8.9	: ASHP ordering schemes of the Computer Experiment 4 - ARMA(4,4) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	105
Figure 8.10	: Computer Experiment 4- Power Spectrums plotted using (a) Original ARMA(4,4) parameters (b) ARMA(4,4) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(4,4) parameters identified by the LS algorithm. The parameter $b_0=0.38$ and the data field size is 50×50 .	109
Figure 8.11	: Computer Experiment 4 - Contour plots of (a) Original ARMA(4,4) parameters (b) ARMA(4,4) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(4,4) parameters identified by the LS algorithm. The parameter $b_0=0.38$ and the data field size is 50×50 .	110
Figure 8.12	: FQP ordering schemes of the Computer Experiment 5 - ARMA(8,8) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	111
Figure 8.13	: Computer Experiment 5- Power Spectrums plotted using (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.1$ and the data field size is 100×100 .	115
Figure 8.14	: Computer Experiment 5 - Contour plots of (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.1$ and the data field size is 100×100 .	117
Figure 8.15	: Computer Experiment 6 – Power Spectrums plotted using (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.23$ and the data field size is 100×100 . The system input $x(k_1, k_2)$ is an AR process of order 8.	122
Figure 8.16	: Computer Experiment 6 – Contour plots of (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.23$ and the data field size is 100×100 . The system input $x(k_1, k_2)$ is an AR process of order 8.	124
Figure 8.17	: Computer Experiment 6 -(a) Original power spectrum of the input AR(8) data field $x(k_1, k_2)$ (b) Power spectrum of $x(k_1, k_2)$ plotted using parameters identified by the Hybrid Lattice algorithm (c) Original contour plot of $x(k_1, k_2)$ (d) Identified contour plot of $x(k_1, k_2)$.	126
Figure 8.18	: FQP ordering schemes of the Computer Experiment 7 - ARMA(8,8) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	126
Figure 8.19	: Computer Experiment 7 – Power Spectrums plotted using (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10×10 .	131
Figure 8.20	: Computer Experiment 7 – Contour plots of (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10×10 .	132
Figure 8.21	: FQP ordering schemes of the Computer Experiment 8 - ARMA(15,15) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	133
Figure 8.22	: Computer Experiment 8 – Power Spectrums plotted using (a) :	139

	Original ARMA(15,15) parameters (b) ARMA(15,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,15) parameters identified by the LS algorithm. The parameter $b_0=0.25$ and the data field size is 50x50.	
Figure 8.23	: Computer Experiment 8 – Contour plots of (a) Original ARMA(15,15) parameters (b) ARMA(15,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,15) parameters identified by the LS algorithm. The parameter $b_0=0.25$ and the data field size is 50x50.	140
Figure 8.24	: Computer Experiment 8 – (a) Power spectrum plotted using ARMA(15,15) parameters identified by the Hybrid Lattice algorithm with no b_0 correction (b) the related ARMA(15,15) contour plot.	141
Figure 8.25	: FQP ordering schemes of the Computer Experiment 9- ARMA(15,15) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	142
Figure 8.26	: Computer Experiment 9 – Power Spectrums plotted using (a) Original ARMA(15,15) parameters (b) ARMA(15,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,15) parameters identified by the LS algorithm. The parameter $b_0=0.25$ and the data field size is 50x50.	147
Figure 8.27	: Computer Experiment 9 – Contour plots of (a) Original ARMA(15,15) parameters (b) ARMA(15,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,15) parameters identified by the LS algorithm. The parameter $b_0=0.25$ and the data field size is 50x50.	149
Figure 8.28	: FQP ordering schemes of the Computer Experiment 10 - ARMA(3,2) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	150
Figure 8.29	: Computer Experiment 10 – Power Spectrums plotted using (a) Original ARMA(3,2) parameters (b) ARMA(3,2) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,2) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10x10.	154
Figure 8.30	: Computer Experiment 10 – Contour plots of (a) Original ARMA(3,2) parameters (b) ARMA(3,2) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,2) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10x10.	155
Figure 8.31	: Computer Experiment 10 – (a) Power spectrum plotted using ARMA(3,2) parameters identified by the Hybrid Lattice algorithm with no b_0 correction (b) the related ARMA(3,2) contour plot.	156
Figure 8.32	: FQP ordering schemes of the Computer Experiment 11 - ARMA(15,8) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	157
Figure 8.33	: Computer Experiment 11 – Power Spectrums plotted using (a) Original ARMA(15,8) parameters (b) ARMA(15,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,8) parameters identified by the LS algorithm. The parameter $b_0=1$ and the data field size is 10x10.	162
Figure 8.34	: Computer Experiment 11 – Contour plots of (a) Original ARMA(15,8) parameters (b) ARMA(15,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,8) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10x10.	163
Figure 8.35	: Computer Experiment 12 – Power Spectrums plotted using (a) Original ARMA(15,8) parameters (b) ARMA(15,8) parameters	168

	identified by the Hybrid Lattice algorithm and (c) ARMA(15,8) parameters identified by the LS algorithm. The parameter $b_0=0.3$ and the data field size is 100x100.	
Figure 8.36	: Computer Experiment 12 – Contour plots of (a) Original ARMA(15,8) parameters (b) ARMA(15,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,8) parameters identified by the LS algorithm. The parameter $b_0=0.3$ and the data field size is 100x100.	170
Figure 8.37	: FQP ordering schemes of the Computer Experiment 13 - ARMA(2,3) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	171
Figure 8.38	: Computer Experiment 13 – Power Spectrums plotted using (a) Original ARMA(2,3) parameters (b) ARMA(2,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(2,3) parameters identified by the LS algorithm. The parameter $b_0=0.75$ and the data field size is 35x35.	174
Figure 8.39	: Computer Experiment 13 – Contour Plots of (a) Original ARMA(2,3) parameters (b) ARMA(2,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(2,3) parameters identified by the LS algorithm. The parameter $b_0=0.75$ and the data field size is 35x35.	175
Figure 8.40	: FQP ordering schemes of the Computer Experiment 14 - ARMA(3,8) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	176
Figure 8.41	: Computer Experiment 14 – Power Spectrums plotted using (a) Original ARMA(3,8) parameters (b) ARMA(3,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,8) parameters identified by the LS algorithm. The parameter $b_0=0.8$ and the data field size is 50x50.	180
Figure 8.42	: Computer Experiment 14 – Contour Plots of (a) Original ARMA(3,8) parameters (b) ARMA(3,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,8) parameters identified by the LS algorithm. The parameter $b_0=0.8$ and the data field size is 50x50.	181
Figure 8.43	: FQP ordering schemes of the Computer Experiment 15 - ARMA(8,15) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.	182
Figure 8.44	: Computer Experiment 15 – Power Spectrums plotted using (a) Original ARMA(8,15) parameters (b) ARMA(8,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,15) parameters identified by the LS algorithm. The parameter $b_0=0.85$ and the data field size is 50x50.	186
Figure 8.45	: Computer Experiment 15 – Contour plots of (a) Original ARMA(8,15) parameters (b) ARMA(8,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,15) parameters identified by the LS algorithm. The parameter $b_0=0.85$ and the data field size is 50x50.	188
Figure 8.46	: Original and blurred simulated texture images. The effect of observation noise $v(m, n)$ has been neglected.	189
Figure 8.47	: Original and blurred cameraman images. The effect of observation noise $v(m, n)$ has been neglected.	190
Figure 8.48	: Original and blurred cameraman images. The effect of observation noise $v(m, n)$ has not been neglected.	192

LIST OF SYMBOLS

$\hat{\mathbf{a}}$	: Estimated AR parameter vector
$\hat{\mathbf{b}}$	: Estimated MA parameter vector
$\mathbf{a}_{u_0}^{(M)}$	: First channel M th order forward prediction error filter tap weight vector
$\mathbf{a}_{x_0}^{(M)}$	: Second channel M th order forward prediction error filter tap weight vector
$\mathbf{g}_{u_p}^{(M)}$	: First channel M th order backward prediction error filter tap weight vector
$\mathbf{g}_{x_p}^{(M)}$	: Second channel M th order backward prediction error filter tap weight vector
$\hat{b}_0$	: Estimated parameter b_0
$f_{u_{p-r}}^{(r)}(k_1, k_2)$	: r th lattice stage two-channel forward prediction error
$b_{u_p}^{(r)}(k_1, k_2)$	: r th lattice stage two-channel backward prediction error
$\mathbf{f}_{p-r}^{(r)}(k_1, k_2)$	: r th lattice stage two-channel forward prediction error vector
$\mathbf{b}_p^{(r)}(k_1, k_2)$	: r th lattice stage two-channel backward prediction error vector
$k_{b_p}^{(r)}$	: r th lattice stage backward reflection coefficient
$k_{f_{p-r}}^{(r)}$	: r th lattice stage forward reflection coefficient
$\mathbf{K}_{p-n}^{(r)}$	: r th lattice stage forward reflection coefficient matrix
$\bar{\mathbf{K}}_p^{(r)}$	: r th lattice stage backward reflection coefficient matrix
$\Delta_{f_{u_{p-r}} b_{u_p}}^{(r)}$	: r th lattice stage cross-correlation between forward and backward prediction errors
$\mathbf{?}_p^{(r)}$	: r th lattice stage cross-correlation matrix between forward and backward prediction errors
$P_{f_{u_{p-r}}}^{(r)}$	: r th lattice stage forward prediction error power
$P_{b_{u_p}}^{(r)}$	: r th lattice stage backward prediction error power
$\mathbf{P}_{p-r}^{(r)}$	: r th lattice stage forward prediction error power matrix
$\bar{\mathbf{P}}_p^{(r)}$	: r th lattice stage backward prediction error power matrix
M	: The order of the AR polynomial
N	: The order of the MA polynomial
S	: The smaller of the (M, N) pair
G	: The greater of the (M, N) pair
p	: Sample processing step

r	: Running order index
$s(m,n)$	: Original Image
$r(m,n)$	: Observed Image
$w(m,n)$	: Model Noise
$v(m,n)$	: Observation Noise
$P(w_m)$	: Given Model Spectrum
$\hat{P}(w_m)$	: Estimated Model Spectrum
E_{ISmin}	: Minimum Itakura-Saito Distance

TWO-DIMENSIONAL ARMA PARAMETER IDENTIFICATION USING TWO-CHANNEL AR LATTICE APPROACH

SUMMARY

The field of multi-dimensional digital signal processing has become increasingly important in recent years due to a number of trends in digital signal processing. Parametric representations of two-dimensional (2-D) random fields in the form of autoregressive (AR), moving average (MA) and autoregressive moving average (ARMA) models are useful in many applications such as image synthesis, classification, spectral estimation, radar imaging, etc.

There are a number of advantages and disadvantages related with AR and MA modelings. The major advantage of both models is that the solution for the model parameters involves only linear equations. In the MA models the solution is unbiased in the presence of additive noise on the system output as long as the noise and system input are uncorrelated. MA models are always stable since they are non-recursive. One of the most serious disadvantages of either AR or MA modeling is the fact that to adequately represent even simple linear systems, both methods may require a large number of parameters (a high order model). This problem arises since, from a transfer function standpoint, AR and MA models attempt to model the system using only poles or only zeros, in spite of the fact that physical systems may have both zeroes and poles. The ARMA (M , N) model is a generalization of the M th order AR and N th order MA models and accomplishes exactly modeling the unknown system with poles and zeroes, representing the system in rational transfer function form. Therefore this has motivated a considerable interest in the more general pole-zero (ARMA) model.

The fundamental problem of identifying a linear shift-invariant (LSI) system from measurements of its output response to known input excitation such as a white noise source is one that impacts on many important fields of interest. In the case of rational systems, the identification problem is to evaluate the degree (model order) of its numerator and denominator polynomials as well as their coefficients (system parameters). Because of its numerical robustness, linear identification under lattice form is of special interest. Two-dimensional orthogonal lattice filters are developed as a natural extension of the 1-D lattice parameter theory. But as there is no natural ordering of the data samples in the 2-D domain, lattice filter solution in 2-D is not unique. Many valid structures have been developed exhibiting different properties.

The primary concern of this research is the determination of discrete time models for 2-D LSI systems from sampled observations of the system input $x(k_1, k_2)$ and system output $y(k_1, k_2)$, using 2-D orthogonal lattice structures, assuming that the order of the ARMA(M , N) model is known. The ARMA model order is represented by the (M , N) pair, where M represents the order of the AR polynomial and N represents the order of the MA polynomial. The general approach underlying the model examined

here assumes that the system to be modeled is described by the following equation with input $x(k_1, k_2)$ and output $y(k_1, k_2)$, resulting with an ARMA(M, N) system.

$$y(k_1, k_2) = b_0 x(k_1, k_2) + \sum_{n=1}^N b_n x((k_1, k_2) - n) - \sum_{m=1}^M a_m y((k_1, k_2) - m) \quad (1)$$

Here the notation $y((k_1, k_2) - m)$ or $x((k_1, k_2) - n)$ denotes the m th or n th element behind $y(k_1, k_2)$ or $x(k_1, k_2)$ in the prediction mask.

In this thesis we present a “hybrid lattice” structure in order to identify the ARMA(M, N) system parameters, namely the a_n and b_n coefficients given in Equation (1), provided that $x(k_1, k_2)$ and $y(k_1, k_2)$ are given. The presented hybrid lattice structure is based on the two-channel AR lattice approach proposed by Kayran for equal AR and MA orders, briefly for $M=N$. The novelties brought about by this proposed structure can be listed as follows.

- We extend Kayran’s approach to the case where M and N can take arbitrary values different from each other. We accomplish this with the help of our proposed hybrid lattice structure where both 2-D two-channel AR and 2-D single-channel AR lattice stages are incorporated. We also propose a modification in terms of the channel inputs of the two-channel lattices. We drive the first channel input by a difference signal of $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$ instead of $y(k_1, k_2)$, which was formerly proposed. However, the use of such a difference signal brings with it the limitation that the data points of both the input and output signals in the prediction region should have the same ordering at least for the filter orders extending to the smaller of the (M, N) pair. The second channel input, which was formerly proposed as $x(k_1, k_2)$, is driven by a newly defined signal $t(k_1, k_2)$, which is related with the orders of the AR and MA polynomials. If the order of the AR polynomial is greater than the order of the MA polynomial ($M > N$), $t(k_1, k_2)$ is equal to $x(k_1, k_2)$, but if the order of the AR polynomial is smaller than the order of the MA polynomial ($M < N$), $t(k_1, k_2)$ is equal to $y(k_1, k_2)$.
- We propose modifications in the b_0 parameter estimates for the cases where AR order is greater or equal to MA order ($M \geq N$) and where AR order is smaller than the MA order ($M < N$) in accordance with our newly proposed channel inputs.
- We derive a new formulation for the ARMA(M, N) parameter estimates, namely $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ vectors, taking into account the estimated parameter $\hat{b}_0$ and the forward prediction error filters’ tap weights related with both channels.
- We emphasize the fact that AR modeled system input parameters and ARMA(M, N) modeled system output parameters can be identified simultaneously, provided that the system input is an AR model of order N , while the ARMA model system output order is (M, N) .

The proposed hybrid lattice structure can be considered as a two-stepped procedure. In order to explain this procedure, here we need to define S as the smaller, and G as the greater of the (M, N) pair.

The first step involves the “hybrid” part which is composed of two sub-steps. The first sub-step involves 2-D two-channel AR lattice structures, where the running order index r starts from 1 and continues up to S and the sample processing step p

starts from r and continues up to S . The compact form equations defining this first sub-step are given below.

$$\begin{bmatrix} \mathbf{f}_{p-r}^{(r)}(k_1, k_2) \\ \mathbf{b}_p^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{K}_{p-r}^{(r)T} \\ \bar{\mathbf{K}}_p^{(r)T} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{f}_{p-r}^{(r-1)}(k_1, k_2) \\ \mathbf{b}_p^{(r-1)}(k_1, k_2) \end{bmatrix} \quad r=1 \dots S, \quad p=r \dots S \quad (2)$$

Here, $\mathbf{f}_{p-r}^{(r)}(k_1, k_2)$ and $\mathbf{b}_p^{(r)}(k_1, k_2)$ denote the r th lattice stage two-channel forward and backward prediction error vectors and $\mathbf{K}_{p-r}^{(r)}$, $\bar{\mathbf{K}}_p^{(r)}$ denote the real valued 2x2 matrices composed of the r th lattice stage forward and backward reflection coefficients, respectively.

In the second sub-step, only single channel AR lattice structures are used. The running order index r starts from 1 and continues up to S and the sample processing step p starts from $(S+1)$ and continues up to $(S+r)$, provided that $(S+r) \leq G$. The compact form equations defining this second sub-step are given below.

$$\begin{bmatrix} f_{u_{p-r}}^{(r)}(k_1, k_2) \\ b_{u_p}^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_p}^{(r)} \\ k_{f_{p-r}}^{(r)} & 1 \end{bmatrix} \begin{bmatrix} f_{u_{p-r}}^{(r-1)}(k_1, k_2) \\ b_{u_p}^{(r-1)}(k_1, k_2) \end{bmatrix} \quad p=(S+1) \dots (S+r), (S+r) \leq G \quad (3)$$

The second step involves only 2-D single channel lattice stages. The running order index r starts from $(S+1)$ and continues up to G . The compact form equations defining the second step are given below.

$$\begin{bmatrix} f_{u_{p-r}}^{(r)}(k_1, k_2) \\ b_{u_p}^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_p}^{(r)} \\ k_{f_{p-r}}^{(r)} & 1 \end{bmatrix} \begin{bmatrix} f_{u_{p-r}}^{(r-1)}(k_1, k_2) \\ b_{u_p}^{(r-1)}(k_1, k_2) \end{bmatrix} \quad r=(S+1) \dots G, \quad p=r \dots G \quad (4)$$

It should be noted that in the special case when both AR and MA parts are of the same order, that is when $M=N$, the whole filter structure just reduces to two-channel lattice stages, hence there is no hybrid structure involved.

In the above two-stepped hybrid approach, the b_0 parameter is not readily available from the calculated lattice parameters, since all the lattice stages contain autoregressive recursions. We propose the following formulas to be used in the estimation of the parameter b_0 . We obtained these formulas by modifying Perry's formerly proposed method in accordance with our new channel inputs.

$$\hat{b}_0 = 1 + \frac{E[f_{u_0}^M(k_1, k_2) f_{x_0}^M(k_1, k_2)]}{E[f_{x_0}^M(k_1, k_2)^2]} \quad \text{for } M \geq N \quad (5)$$

$$\hat{b}_0 = \frac{E[(f_{y_0}^M(k_1, k_2)(f_{y_0}^M(k_1, k_2) - f_{u_0}^M(k_1, k_2)))]}{E[(f_{y_0}^M(k_1, k_2) - f_{u_0}^M(k_1, k_2))^2]} \quad \text{for } M < N \quad (6)$$

We also propose a new method to calculate the $[\hat{\mathbf{a}} \ \hat{\mathbf{b}}]^T$ vector of estimated ARMA(M, N) parameters, once we have obtained the lattice reflection coefficients, the parameter b_0 and the forward prediction error filter tap weights related with both channels. We derive new equations for the cases when $M \geq N$ and $M < N$. The proofs of these equations are given in Appendices. For the case $M=N$, we elaborate on the

situation where the system input $x(k_1, k_2)$ is not a white noise excitation, but an AR process of order N .

For the case when the AR and MA orders are equal ($M=N$), we derive the following equations for the ARMA(M, N) parameter estimates.

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \end{bmatrix} + (1-\hat{b}_0) \begin{bmatrix} \mathbf{a}_{x_0}^{(N)}(2) \\ \vdots \\ \mathbf{a}_{x_0}^{(N)}(M+1) \end{bmatrix} \text{ for } M=N \quad (7.a)$$

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(2M+2) \end{bmatrix} + (1-\hat{b}_0) \begin{bmatrix} \mathbf{a}_{x_0}^{(N)}(M+3) \\ \vdots \\ \mathbf{a}_{x_0}^{(N)}(2M+2) \end{bmatrix} \text{ for } M=N \quad (7.b)$$

Here $\mathbf{a}_{u_0}^{(M)}$ and $\mathbf{a}_{x_0}^{(N)}$ represent the M th order and N th order forward prediction error filter tap weights of the first and second channels, respectively.

For the case AR order is greater than the MA order ($M > N$), we derive the following matrix equations for the ARMA(M, N) parameter estimates.

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_N \\ \hat{a}_{N+1} \\ \vdots \\ \hat{a}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(2) + (1-\hat{b}_0)\mathbf{a}_{x_0}^{(N)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(N+1) + (1-\hat{b}_0)\mathbf{a}_{x_0}^{(N)}(N+1) \\ \mathbf{a}_{u_0}^{(M)}(N+2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \end{bmatrix} \text{ for } M > N \quad (8.a)$$

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(M+3) + (1-\hat{b}_0)\mathbf{a}_{x_0}^{(N)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+N+2) + (1-\hat{b}_0)\mathbf{a}_{x_0}^{(N)}(M+N+2) \end{bmatrix} \text{ for } M > N \quad (8.b)$$

For the case AR order is smaller than the MA order ($M < N$), we derive the following equations for the ARMA(M, N) parameter estimates.

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(N)}(2) + (1-\hat{b}_0)(\mathbf{a}_{y_0}^{(M)}(2) - \mathbf{a}_{u_0}^{(M)}(2)) \\ \vdots \\ \mathbf{a}_{u_0}^{(N)}(M+1) + (1-\hat{b}_0)(\mathbf{a}_{y_0}^{(M)}(M+1) - \mathbf{a}_{u_0}^{(M)}(M+1)) \end{bmatrix} \text{ for } M < N \quad (9.a)$$

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_M \\ \hat{b}_{M+1} \\ \vdots \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_{u_0}^{(N)}(M+3) + (1-\hat{b}_0)(\mathbf{a}_{u_0}^{(M)}(M+3) - \mathbf{a}_{y_0}^{(M)}(M+3)) \\ \vdots \\ -\mathbf{a}_{u_0}^{(N)}(2M+2) + (1-\hat{b}_0)(\mathbf{a}_{u_0}^{(M)}(2M+2) - \mathbf{a}_{y_0}^{(M)}(2M+2)) \\ -\mathbf{a}_{u_0}^{(N)}(2M+3) \\ \vdots \\ -\mathbf{a}_{u_0}^{(N)}(M+N+2) \end{bmatrix} \text{ for } M < N \quad (9.b)$$

Here $\mathbf{a}_{u_0}^M$ and $\mathbf{a}_{u_0}^N$ represent the M th order and N th order forward prediction error filter tap weights of the first channel respectively. The vector $\mathbf{a}_{y_0}^M$ denotes the N th order forward prediction error filter tap weights of the second channel.

In order to show the efficiency of the proposed method, we have carried out some computer simulations covering the prediction support regions in first quarter plane (FQP) and asymmetric half plane (ASHP) for $M = N$, $M > N$ and $M < N$. In order to make a comparison, we have plotted power spectrums and contours for each computer simulation, obtained using the original and identified parameters. We have used the Itakura-Saito distance measure, which indicates the similarity between the original and identified power spectrums; L_1 , L_2 and L_∞ vector norms as performance measures.

İKİ-KANALLI AR KAFES YAKLASIMI KULLANARAK İKİ-BOYUTLU ARMA PARAMETRE TANILAMA

ÖZET

Sayısal isaret islemedeki gelişmelere bağlı olarak, çok boyutlu sayısal isaret işleme son yıllarda oldukça önem kazanmıştır. İki-boyutlu rastgele alanların, özbağlanımlı (*autoregressive*-AR), kayan ortalamalı (*moving average*-MA) ve özbağlanımlı kayan ortalamalı (*autoregressive moving average*-ARMA) modeller şeklinde parametrik gösterimleri, imge sentezi, sınıflandırma, izgesel kestirim ve radar imgesi oluşturma gibi bir çok uygulama alanlarında yararlı olmaktadır.

AR ve MA modelleme ile ilgili olarak bir dizi üstünlükler ve sakıncalar bulunmaktadır. Her iki modelin baslıca üstünlüğü, model parametrelerinin çözümünün yalnızca doğrusal denklemleri içermesidir. MA modellerde, sistem çıkışında toplamsal gürültünün bulunduğu durumda, gürültü ve sistem girişi ilintisiz olduğu sürece, yansız çözüm elde edilir. MA modeller özyinelemeli olmadıklarından daima kararlıdır. AR ya da MA modellerin en ciddi kısıtlamalarından biri, basit doğrusal sistemleri uygun olarak modellemek için bile, her iki yöntemin de çok sayıda parametre (yüksek dereceli bir model) gerektirebileceği gerçektir. Aktarım işlevi açısından bakıldığında, fiziksel sistemlerde hem sıfır hem de kutup olabileceği halde, AR ve MA modeller sistemi yalnızca kutup veya yalnızca sıfır ile modellemeye çalıştığından böyle bir sorun ortaya çıkmaktadır. ARMA(M, N) modeli, M . dereceden AR ve N . dereceden MA modellerin genelleştirilmiş biçimidir ve bilinmeyen sistemi, aktarım işlevi şeklinde sıfır ve kutuplarla modelleme işlemini gerçekleştirmektedir. Bu nedenle, daha genel olan sıfır-kutup modeline (ARMA) ilgi artmıştır.

Öteleme ile değişmeyen doğrusal bir sistemi, beyaz gürültü kaynağı gibi bilinen bir giriş uyarımına karşı verdiği çıkış cevabından tanılama temel problemi, bir çok önemli ilgi alanına etkimektedir. Oransal ifade edilen sistemler söz konusu olduğunda, tanılama problemi, pay ve payda polinomlarının katsayılarını (sistem parametreleri) olduğu kadar derecesini (model derecesi) de belirleyebilmektir. Sayısal dayanıklılıkları nedeni ile, kafes yapıları altında doğrusal tanılamaya özel bir ilgi olmuştur. İki-boyutlu dikgen kafes süzgeçleri, bir-boyutlu kafes parametre kuramının doğal bir uzantısı olarak gerçekleştirilmiştir. Fakat iki-boyutta veri örnekleri doğal bir dizilime sahip olmadıklarından, iki-boyutlu kafes süzgeç çözümü tek değildir. Farklı özelliklere sahip bir çok geçerli yapı oluşturulmuştur.

Bu araştırmanın temel ilgi alanı, ARMA(M, N) model derecesinin bilindiği kabul edilerek, sistem girişi $x(k_1, k_2)$ ve sistem çıkışı $y(k_1, k_2)$ 'nin örneklenmiş gözlemlerinden, iki-boyutlu dikgen kafes yapıları kullanarak iki-boyutlu öteleme ile değişmeyen doğrusal sistemler için ayrık zamanlı modelleri belirlemektir. AR polinomun derecesi M ile, MA polinomun derecesi N ile gösterilmek üzere, ARMA

model derecesi (M,N) çifti ile temsil edilmektedir. Burada incelenen modelin altındaki genel yaklaşımlar, modellenen sistemin $x(k_1, k_2)$ girişli, $y(k_1, k_2)$ çıkışlı bir, ARMA(M, N) sistem oluşturacak şekilde aşağıdaki verilen bağıntı ile tanımlandığını kabul etmektedir.

$$y(k_1, k_2) = b_0 x(k_1, k_2) + \sum_{n=1}^N b_n x((k_1, k_2) - n) - \sum_{m=1}^M a_m y((k_1, k_2) - m) \quad (1)$$

Burada $y((k_1, k_2) - m)$ veya $x((k_1, k_2) - n)$ gösterimi, öngörü bölgesinde $y(k_1, k_2)$ veya $x(k_1, k_2)$ 'nin arkasında bulunan m . veya n . elemanı ifade etmektedir.

Bu tezde, $x(k_1, k_2)$ and $y(k_1, k_2)$ 'nin verildiği durumda, (1) bağıntısında gösterilen ARMA(M,N) sistem parametreleri a_n ve b_n katsayılarını tanılamak için, Kayran tarafından esit AR ve MA dereceli sistemler, kısaca $M=N$ durumu için önerilen iki-kanallı AR kafes yaklaşımını temel alan bir “karma kafes” yapısı sunmaktayız. Bu yapı ile getirilen yenilikler, aşağıdaki şekilde listelenebilir.

- Kayran tarafından kullanılan yaklaşımı M ve N 'in birbirinden farklı değerler alabileceği durum için genişletmekteyiz. Bunu, yeni teklif ettiğimiz, hem iki-boyutlu iki-kanallı AR ve iki-boyutlu tek-kanallı AR kafes katlarını içeren karma kafes yapısı yardımı ile gerçekleştirmekteyiz. Ayrıca, iki-kanallı kafeslerin kanal girişlerinde de bir değişiklik önermekteyiz. Birinci kanal girişini, önceden önerilen $y(k_1, k_2)$ yerine $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$ gibi bir fark isareti ile sürmekteyiz. Bununla beraber, böyle bir fark isaretinin kullanımı, hem giriş ve hem de çıkış isaretinin öngörü destek bölgesindeki veri noktalarının, en azından (M, N) çiftinin küçük olanına kadarki süzgeç derceleri için aynı dizilime sahip olması kısıtını getirmektedir. Önceden $x(k_1, k_2)$ olarak teklif edilen ikinci kanal girişi ise, AR ve MA derecelerine bağlı olan bir $t(k_1, k_2)$ isareti ile sürülmektedir. Eğer AR polinomunun derecesi MA polinomunun derecesinden büyükse ($M > N$), $t(k_1, k_2)$ isareti $x(k_1, k_2)$ 'ye esit olmaktadır, eğer AR polinomunun derecesi MA polinomunun derecesinden küçükse ($M < N$), $t(k_1, k_2)$ isareti $y(k_1, k_2)$ 'ye esit olmaktadır.
- Yeni teklif ettiğimiz kanal girişlerine uygun olarak, AR derecesinin MA derecesinden büyük veya esit ($M \geq N$) ve AR derecesinin MA derecesinden küçük ($M < N$) olduğu durumlar için, b_0 parametre kestirimlerinde değişiklikler önermekteyiz.
- ARMA(M,N) parametre kestirimleri, yani $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ vektörleri için, kestirilen parametre $\hat{b}_0$ ve her iki kanala ilişkin ileri yönde öngörü yanlışlığı süzgeçlerinin katsayı ağırlıklarını da göz önüne alan yeni bir formülasyon önermekteyiz.
- Sistem çıkışının ARMA model derecesi (M, N) iken, sistem girişinin N dereceli bir AR model olduğu durumda, AR modellenmiş sistem giriş parametrelerinin ve ARMA(M,N) modellenmiş sistem çıkış parametrelerinin aynı anda tanımlanabileceği gerçeğini vurgulamaktayız.

Önerilen karma kafes yapısı iki-basamaklı bir yordam olarak ele alınabilir. Bu yordamı açıklayabilmek için, burada S 'i (M, N) çiftinin küçük olanı, G 'yi ise büyük olanı olarak tanımlamamız gerekmektedir.

İlk basamak, iki alt-basamaktan oluşan “karma” olarak adlandırılan kısımdır. İlk alt basamak, iki-boyutlu iki-kanallı AR kafes yapılarını içermekte olup, değişken derece

indisi r , 1'den başlayıp S 'e kadar ve örnek isleme adımı p ise r 'den başlayıp S 'e kadar devam etmektedir. Bu alt basamağı tanımlayan denklemler aşağıda verilmektedir.

$$\begin{bmatrix} \mathbf{f}_{p-r}^{(r)}(k_1, k_2) \\ \mathbf{b}_p^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{K}_{p-r}^{(r)T} \\ \bar{\mathbf{K}}_p^{(r)T} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{f}_{p-r}^{(r-1)}(k_1, k_2) \\ \mathbf{b}_p^{(r-1)}(k_1, k_2) \end{bmatrix} \quad r=1 \dots S, \quad p=r \dots S \quad (2)$$

Burada $\mathbf{f}_{p-r}^{(r)}(k_1, k_2)$ ve $\mathbf{b}_p^{(r)}(k_1, k_2)$ r . kafes katinin iki-kanallı ileri ve geri yönde öngörü yanilgi vektörlerini ve $\mathbf{K}_{p-r}^{(r)}$ ile $\bar{\mathbf{K}}_p^{(r)}$ de, r . kafes katinin sirasi ile ileri ve geri yönde yansima katsayılarından olusan, 2x2 boyutlu gerçel degerli matrislerini göstermektedir.

İkinci alt basamakta yalnızca tek kanallı AR kafes yapıları kullanılmaktadır. Degisken derece indisi r , 1'den başlayıp S 'e kadar devam etmekte ve örnek isleme adımı p ise, $(S + r) \leq G$ olduğu sürece, $(S+1)$ 'den başlayıp $(S+r)$ 'ye kadar devam etmektedir. Bu alt basamağı tanımlayan denklemler aşağıda verilmektedir.

$$\begin{bmatrix} f_{u_{p-r}}^{(r)}(k_1, k_2) \\ b_{u_p}^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_p}^{(r)} \\ k_{f_{p-r}}^{(r)} & 1 \end{bmatrix} \begin{bmatrix} f_{u_{p-r}}^{(r-1)}(k_1, k_2) \\ b_{u_p}^{(r-1)}(k_1, k_2) \end{bmatrix} \quad p = (S+1) \dots (S+r), (S+r) \leq G \quad (3)$$

İkinci basamak, yukarıda olduğu gibi, yalnızca iki-boyutlu tek kanallı kafes katlarını içermektedir. Degisken derece indisi r , $(S+1)$ 'den başlayıp G 'ye kadar devam etmekte ve örnek isleme adımı p ise, r 'den başlayıp G 'ye kadar devam etmektedir. Bu alt basamağı tanımlayan denklemler aşağıda verilmektedir.

$$\begin{bmatrix} f_{u_{p-r}}^{(r)}(k_1, k_2) \\ b_{u_p}^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_p}^{(r)} \\ k_{f_{p-r}}^{(r)} & 1 \end{bmatrix} \begin{bmatrix} f_{u_{p-r}}^{(r-1)}(k_1, k_2) \\ b_{u_p}^{(r-1)}(k_1, k_2) \end{bmatrix} \quad r = (S+1) \dots G, \quad p = r \dots G \quad (4)$$

AR ve MA kısımlarının derecelerinin aynı olduğu $M=N$ durumunda, bütün bu yapının sadece iki-kanallı kafes katlarına indirgendigine ve bu durumda bir karma yapının oluşmadigina dikkat edilmelidir.

Yukarıdaki iki-basamaklı karma yaklaşımda, b_0 parametresi hesaplanan kafes parametrelerinden hemen elde edilemez, çünkü bütün kafes katları özbağlanımlı özyinelemeler içermektedirler. Önceden önerilen yöntemde, bizim yeni önerdiğimiz kanal girişlerine uygun olarak degisiklik yaparak, aşağıdaki formüllerin b_0 parametresi kestiriminde kullanılmasını önermekteyiz.

$$\hat{b}_0 = 1 + \frac{E[f_{u_0}^M(k_1, k_2) f_{x_0}^M(k_1, k_2)]}{E[f_{x_0}^M(k_1, k_2)^2]} \quad M \geq N \quad (5)$$

$$\hat{b}_0 = \frac{E[(f_{y_0}^M(k_1, k_2)(f_{y_0}^M(k_1, k_2) - f_{u_0}^M(k_1, k_2)))]}{E[(f_{y_0}^M(k_1, k_2) - f_{u_0}^M(k_1, k_2))^2]} \quad M < N \quad (6)$$

Ayrıca kafes yansima katsayılarını, her iki kanala ait ileri yönde öngörü yanilgi süzgeçlerinin katsayı ağırlıklarını ve b_0 parametresini elde ettikten sonra, ARMA parametreleri vektörü $\begin{bmatrix} \hat{\mathbf{a}} & \hat{\mathbf{b}} \end{bmatrix}^T$ 'yi hesaplayan yeni bir yöntem önermekteyiz. $M \geq N$

and $M < N$ durumları için, ispatları eklerde verilen yeni formüller çıkarmaktayız. $M = N$ durumunda, sistem girişi $x(k_1, k_2)$ 'nin beyaz gürültü uyarımı olmadığı koşul üzerinde inceleme yapmaktayız.

AR derecesinin MA derecesine esit ($M=N$) olduğu durumda, ARMA(M,N) parametre kestirimi için aşağıdaki bağıntılar elde edilmistir.

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \end{bmatrix} + (1-\hat{b}_0) \begin{bmatrix} \mathbf{a}_{x_0}^{(N)}(2) \\ \vdots \\ \mathbf{a}_{x_0}^{(M)}(M+1) \end{bmatrix} \quad M=N \quad (7.a)$$

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(2M+2) \end{bmatrix} + (1-\hat{b}_0) \begin{bmatrix} \mathbf{a}_{x_0}^{(N)}(M+3) \\ \vdots \\ \mathbf{a}_{x_0}^{(M)}(2M+2) \end{bmatrix} \quad M=N \quad (7.b)$$

Burada $\mathbf{a}_{u_0}^M$ ve $\mathbf{a}_{x_0}^N$, sırasıyla birinci ve ikinci kanallara ilişkin M . ve N . derecelerden ileri yönde öngörü yanılgi süzgeci katsayı ağırlıklarını göstermektedirler.

AR derecesinin MA derecesinden büyük ($M > N$) olduğu durumda, ARMA(M,N) parametre kestirimi için aşağıdaki bağıntılar elde edilmistir.

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_N \\ \hat{a}_{N+1} \\ \vdots \\ \hat{a}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(2) + (1-\hat{b}_0)\mathbf{a}_{x_0}^{(N)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(N+1) + (1-\hat{b}_0)\mathbf{a}_{x_0}^{(N)}(N+1) \\ \mathbf{a}_{u_0}^{(M)}(N+2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \end{bmatrix} \quad M > N \quad (8.a)$$

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(M+3) + (1-\hat{b}_0)\mathbf{a}_{x_0}^{(N)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+N+2) + (1-\hat{b}_0)\mathbf{a}_{x_0}^{(M)}(M+N+2) \end{bmatrix} \quad M > N \quad (8.b)$$

AR derecesinin MA derecesinden küçük ($M < N$) olduğu durumda, ARMA(M,N) parametre kestirimi için aşağıdaki bağıntılar elde edilmistir.

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(N)}(2) + (1-\hat{b}_0)(\mathbf{a}_{y_0}^{(M)}(2) - \mathbf{a}_{u_0}^{(M)}(2)) \\ \vdots \\ \mathbf{a}_{u_0}^{(N)}(M+1) + (1-\hat{b}_0)(\mathbf{a}_{y_0}^{(M)}(M+1) - \mathbf{a}_{u_0}^{(M)}(M+1)) \end{bmatrix} \quad M < N \quad (9.a)$$

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_M \\ \hat{b}_{M+1} \\ \vdots \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_{u_0}^{(N)}(M+3) + (1-\hat{b}_0)(\mathbf{a}_{u_0}^{(M)}(M+3) - \mathbf{a}_{y_0}^{(M)}(M+3)) \\ \vdots \\ -\mathbf{a}_{u_0}^N(2M+2) + (1-\hat{b}_0)(\mathbf{a}_{u_0}^{(M)}(2M+2) - \mathbf{a}_{y_0}^{(M)}(2M+2)) \\ -\mathbf{a}_{u_0}^{(N)}(2M+3) \\ \vdots \\ -\mathbf{a}_{u_0}^{(N)}(M+N+2) \end{bmatrix} \quad M < N \quad (9.b)$$

Burada $\mathbf{a}_{u_0}^M$ ve $\mathbf{a}_{u_0}^N$, sirasiyla birinci kanala iliskin M . ve N . derecelerden ileri yönde öngörü yanilgi süzgeci katsayi agirliklarini göstermektedirler. $\mathbf{a}_{y_0}^N$ ise ikinci kanalın N .dereceden ileri yönde öngörü yanilgi süzgeci katsayi agirliklarini göstermektedir.

Önerdigimiz yöntemin dogrulugunu gösterebilmek için, hem birinci çeyrek düzlem hem de simetrik olmayan yari düzlem öngörü destek bölgelerinde, $M = N$, $M > N$ ve $M < N$ için bilgisayar benzetimleri sunmaktayiz. Bir karsilastirma yapabilmek amaci ile, her bir benzetim için tanilanmis ve özgün parametrelere iliskin güç izgelerinin ve çevritlerin çizimlerini verdik. Özgün ve kestirilen güç izgeleri arasindaki benzerligi gösteren Itakura-Saito uzaklik ölçütü ile L_1 , L_2 ve L_∞ vector normlarini da basari ölçütleri olarak kullanmis bulunmaktayiz.

1. INTRODUCTION

The fundamental problem of fitting a model to a given data at hand is one that impacts on many important fields of interest. The data is supposed to fit an autoregressive (AR) model with an order M , a moving average (MA) model with an order N or an autoregressive moving average (ARMA) model with an order (M, N) .

There are a number of advantages and disadvantages related with AR and MA modeling. The major advantage of both models is that the solution for the model parameters involves only linear equations. In the MA models the solution is unbiased in the presence of additive noise on the system output as long as the noise and system input are uncorrelated. Regarding the stability, MA models are always stable since the model is non-recursive. One of the most serious disadvantages of either AR or MA modeling is the fact that to adequately represent even simple linear systems, both methods may require a large number of parameters (a high order model). This problem arises since, from a transfer function standpoint, AR and MA models attempt to model the system using only poles or only zeros, in spite of the fact that physical systems may have both zeroes and poles. While modeling the effects of a zero with a number of poles and visa versa has been analytically justified [1], it takes far more sense (both from the viewpoint of model accuracy and efficient use of model parameters) to let the model represent the system as it really is with both zeroes and poles if this is at all possible. Furthermore, in some situations, particularly in the presence of sharp zeroes in the transfer function of the unknown system, the AR model appears to have poor performance.

The ARMA(M,N) model is a generalization of the AR(M) and MA(N) models and accomplishes exactly modeling the unknown system with poles and zeroes, representing the system in rational transfer function form. Therefore this has motivated a considerable interest in the more general pole-zero (ARMA) model.

One-dimensional (1-D) ARMA processes have found wide applications in various signal-processing applications, like modeling [1]-[23], parameter estimation [24]-

[45], system identification [46]-[67], spectral estimation [68]-[86], digital filter design [87]-[98], speech processing [99]-[103], etc.

Perry [1] investigated the problem of obtaining parametric models for linear and nonlinear systems based on observations of the input and output of the system. Using the equation error formulation, he developed lattice solution methods in batch processing and adaptive form for both single and multichannel ARMA(M, N) models. Based on this equation error formulation, Perry and Parker [2] presented transitional formulas to provide solutions to the ARMA(M, N) modeling problem when given solutions to either AR(M) or MA(N) models. These formulas were useful in estimating ARMA(M, N) models, and they also provided insight into how the more general ARMA(M, N) modeling problem was related to the special cases of AR(M) and MA(N) modeling.

Lee *et al.* [3] presented the extension of the all-pole (AR) exact least-squares ladder algorithms to the pole-zero (ARMA) case. The algorithms were based on a general set of recursion obtained by a geometric approach. The recursions obtained were square-root normalized and had much simpler structures than the unnormalized case.

Moses *et al.* [4] presented a two-part fast recursive algorithm for ARMA(M, N) modeling. The autoregressive coefficient estimates were based on the solution of a set of extended Yule-Walker equations. These estimates were under appropriate conditions asymptotically unbiased and consistent. A recursive lattice filter implementation of this coefficient estimator was derived which required $O(p)$ computations per time update. Moreover, an exponential forgetting factor can be implemented which facilitates the tracking of variations in the data. The numerator spectral coefficients are then estimated from a forward prediction error sequence generated by the AR lattice. The latter are updated recursively in $O(q)$ computations, which can also be implemented with an exponential forgetting factor. The complete recursive algorithm is fast in the sense that $O(p + q)$ computations are required for each update.

Modeling of nonstationary signals through time-dependent ARMA(M, N) models and lattices was presented by Grenier [5], [6]. He first presented three methods for the modeling of nonstationary stochastic processes [5]. In each of them, the time dependent parameters were assumed to evolve in a space spanned by a basis of known deterministic functions of time. In the first method, ARMA modeling

introduced a vector process, which was the projection of the nonstationary scalar process on the basis of functions. Levinson's algorithm [7] for this vector process permitted the identification of the time-dependent AR coefficients. The MA part of the model was approximated by a long AR model. In the second method, the model consisted of a lattice filter with two time-varying reflection coefficients per cell. For each cell, the minimization of the variance of the outputs led to a linear system where, again, the covariance matrix of a vector process appeared, namely, the one obtained after decomposition of the inputs in the basis. By requiring the two reflection coefficients to be identical, one obtained an extension of Burg's [8] method. In the third method, the signal was the superposition of a white noise and the zero-input response of a time-varying system. In this extension of Prony's method, the vector of unknown parameters was the eigenvector associated to the smallest eigenvalue of a matrix derived from the covariance matrix of the same vector process of the projections. Grenier's latter presentation for nonstationary ARMA models [6] was an estimator, which realized a good compromise between cost and quality by simultaneous estimation of the AR and MA parts of the model.

Karlsson [9]-[10] considered AR, MA and ARMA modeling of linear time-varying systems with lattice filters. In particular, he presented an ARMA (M, N) lattice filter structure, which was fully consistent with the geometric characteristics of the AR and MA lattice filter structures in that it evaluated all optimal ARMA (i, j) filters of order lower than (M, N) , and each such filter was realized in terms of a fully orthogonal set of realization vectors. He also developed fast RLS algorithms for the evaluation of the lattice filter coefficients. The numerical properties of his proposed algorithms were given in a latter publication [11].

Ortigueira's [12] three-step ARMA modeling algorithm was based on a fundamental relationship, which showed that the AR polynomial of an ARMA (M, N) model belonged to the linear space spanned by the forward and backward linear predictors. The three steps involved were listed as follows: (a) computation of the forward and backward linear prediction errors, (b) use of the fundamental relationship together with an identification algorithm to compute the three referred polynomial, one of them being the AR polynomial, (c) obtain the MA polynomial by performing a limiting operation.

Rayala and Reddy [13] presented an approach that utilized the eigenfunction decomposition of the data matrix and made an effective use of the null space characterization of the autocovariance matrix of the response data, for the determination of the ARMA(M, N) model parameters.

The wide applicability of the multivariate parametric modeling motivated works on the problem of modeling multichannel ARMA processes [14]-[17]. Swami *et al.* [14]-[15] addressed multichannel ARMA modeling by using the compact Kronecker product representation [18] for cumulants of vector processes and obtained expressions for the polyspectra of multivariate linear processes in terms of system transfer function. Chakraborty and Prasad [16]-[17] developed an algorithm for multichannel ARMA modeling by least squares circular lattice filtering. They mapped the multivariate ARMA process to an equivalent scalar, periodic ARMA process and formulated the problem as the order recursive computation of the orthogonal projection of the input vector on an appropriate data space.

Therrien and May [19] compared several ARMA modeling methods for their ability to model the time response of systems characterized by exponentially damped resonant frequencies. Results were discussed for known ARMA systems with added noise and samples of recorded short duration acoustic data.

Prasad [20]-[21] presented a set of papers proposing a recursive pseudo least squares algorithm for ARMA filtering and modeling. In the first paper, he developed a recursive algorithm for ARMA filtering, resulting in a lattice-like structure [20]. As the coefficients of this filter did not constitute an exact match to the ARMA model parameters, the second paper was devoted to the derivation of the ARMA parameters [21].

Min and Un [22] presented a pole-zero (ARMA) modeling of discrete time linear systems using an RLS fast transversal filter (FTF) algorithm. This ARMA FTF algorithm was derived using geometric projections and forward backward prediction of the input signal. It was stated that this algorithm required less computations than RLS lattice filters.

Therrien and Velasco [23] presented an iterative version of the Prony method, being exceptionally effective in finding ARMA models for acoustic data in the time

domain. The method was based on a quadratic type of gradient algorithm, where it was shown that the gradient and Hessian were easily computed from the data.

From the ARMA parameter estimation point of view, Walker [24] proposed a solution based on the sample autocorrelation function (ACF). Walker's method was asymptotically efficient but efficiency was obtained only in the limit as the number of sample ACF lags went to infinity.

Solo [25] derived a lattice algorithm for finite data instrumental variable recursions and indicated its use in both ARMA and ARX time series models. He presented two basic finite data algorithm. One was for the fast solution of the equations satisfied by the AR parameters in a block identifiable multivariate ARMA model. The other algorithm was for the instrumental variable estimation of parameters in a multivariate lagged regression model with colored disturbance noise.

Bruzzzone and Kaveh [26] considered information tradeoffs in using the sample ACF in ARMA parameter estimation. It was seen that reducing the ARMA process data to a given set of consecutive lags of the popular lagged-product ACF estimates prior to parameter estimation increased Cramer-Rao bounds on the generalized error covariance.

A spectral matching technique for ARMA parameter estimation was presented by Friedlander and Porat [27]. They also considered the asymptotic accuracy of Gaussian ARMA parameter estimation methods based on a fixed number of sample covariances [28]. They presented a general asymptotic expression for the error covariance of the ARMA parameter estimates; a lower bound for the error covariance of these estimates; a theorem stating that this lower bound is strictly greater than the Cramer-Rao bound and an explicit ARMA estimation technique that achieves this bound. They proved that this lower bound approached the Cramer-Rao bound as the number of sample covariances tended to infinity.

Li and Dickinson [29] proposed an efficient method to compute consistent estimates of the AR parameters of an ARMA model based on the iterated least-squares approach proposed by Tsay and Tiao [30]. The method was derived from the lattice filter structure, which generated an orthogonal basis for the Hilbert space of interest. It was shown that for the ARMA (M, N) model, one could get consistent estimates of the AR parameters by using a least-squares lattice estimation algorithm and solving a

system of N linear equations. Another study on the application of the lattice filter to robust estimation of AR and ARMA models, was also presented by Li and Dickinson [31]. This presentation, which was basically a $(M+N)$ -stage lattice whitening filter, was also based on their own former work [29] and iterated least-squares approach of Tsay and Tiao [30].

Moura and Ribeiro [32] proposed an algorithm for linear estimation of ARMA processes. Given a finite length sample drawn from an ARMA (M, N) model, the technique provided the estimated values of the orders M and N , as well as the AR and MA coefficients. They were obtained from the reflection coefficient sequence estimated directly from the data.

Stoica *et al.* [33] presented a high-order Yule-Walker (HOYW) method for estimation of the AR parameters of an ARMA model to estimate the frequencies of sinusoidal signals from noisy measurements. They illustrated the performance of the proposed method by some numerical examples.

Generalized H -Lattice filter structure was also used for the ARMA parameter estimation [34]-[35]. It was shown that AR lattice structure of Itakura and Saito [99] was a special case of a generalized class of lattice structures where the delay element was replaced by all-pass filter sections. Hence recursive filters could be implemented by replacing the unit delay element by an all-pole filter.

Kwan and Lui [36] proposed a Levinson-type algorithm for estimating the 1-D ARMA model from the observed inputs and outputs of an unknown system. The algorithm was capable of computing models of any AR-order and MA-order with arbitrary arrangement of order-update recursions. When the input was a white process, the algorithm yielded a lattice model, which was canonic with respect to the number of parameters. Numerical examples were included to verify the formulation.

Zerubia and Alengrin [37] proposed a method based on a two-step approach for estimation of ARMA (M, N) parameters. First the AR parameters were estimated using a transient Kalman gain [38], then the MA parameters were obtained by a fast filtering algorithm [39].

Although the single-channel ARMA estimation can be extended to the multi-channel case in a straightforward manner, the resulting algorithms, however, employ extensive matrix operations such as inversion and Choleskey factorization and thus

become computationally unattractive especially from real-time application point of view. This problem was considered by Chakraborty and Prasad [16], where the given multi-channel ARMA process was converted into an equivalent scalar, periodic ARMA process. The authors later presented a method [40] to compute the Cramer-Rao lower bounds (CRLB) for the parameter estimates of the scalar periodic ARMA equivalent of a multivariate ARMA model. The main operation involved periodic AR filtering of certain signals, constructed by delaying and decimating the input and output sequences.

Sabiti [41] presented a fast and iterative batch algorithm for ARMA parameter estimation. This algorithm, which was a modified version of the method suggested by Mayne and Firoozan [52], provided estimates with the same asymptotic distribution as the maximum likelihood estimates.

Mari, Stoica and McKelvey [42]-[43] presented a vector ARMA parameter estimation method for finite-dimensional multivariate linear stochastic systems, which was guaranteed to produce valid models approximating the true underlying system in a computational time of a polynomial order in system dimension.

The ARMA parameter estimation algorithm of Chon and Lu [44] was based on the group method of data handling introduced formerly for solving high-order regression polynomials. Various simulation examples were given to show the efficacy of the proposed method.

Lu *et al.* [45] later proposed an algorithm for linear and nonlinear ARMA parameter estimation using the concepts of affine geometry in which the salient feature of the algorithm was to remove the linearly dependent ARMA vectors from the pool of candidate ARMA vectors. This algorithm, referred to as optimal parameter search (OPS), was able to extract only the correct parameters despite over-determined incorrect model-order selection.

The textbook of Box and Jenkins [46] was probably one of the most important contributions to the problem of identifying the parameters of an ARMA model. It was Graupe *et.al* [48] who later proposed a procedure for the identification of ARMA parameters of time series. Identification was performed by first identifying a purely autoregressive signal model. The parameters and orders of the mixed

autoregressive moving average process were then given from the solution of simple algebraic equations involving the purely autoregressive model parameters.

Brotherton and Caines [49] introduced a method, called the Choleskey Least Squares (CLS), for the identification of scalar ARMA models. This method, which iteratively estimated the coefficients of the ARMA model, given the input-output data sequences, was a variant of the Generalized Least Squares (GLS) identification method [104]-[105]. [60]

Benveniste and Chaura [50] proposed a simple innovations approach for Levinson identification algorithms, using abstract Levinson orthogonalization formulas in Hilbert spaces. This approach was allowed to obtain connection between classical (recursive) identification procedure for AR (least squares procedure) and ARMA (extended least squares procedure) processes, and the corresponding Levinson adaptive identification algorithms for polynomial and rational filters synthesized in lattice and ladder form.

Parker and Perry [51] presented a nonlinear extension of the discrete linear ARMA model for the identification of nonlinear systems from measurements of input and output signals. It was shown that this model was applicable to a broad class of nonlinear systems, and that the model parameters could be determined using multi-channel lattice techniques.

Mayne and Firoozan [52] presented a method for linear identification of ARMA processes. The method consisted of three linear least-squares estimations. In the first an autoregressive model was fitted to the observation sequence, yielding an estimate of the values of the driving white noise sequence. Linear least squares was then used to fit an ARMA model to the observation and estimated white noise sequences. This model was used to filter the observation and estimated white noise sequences. Finally an ARMA model was fitted to the filtered sequences. It was shown that the resultant estimator is ' p -consistent' (the asymptotic bias tends to zero as the degree p of the autoregressive model tends to infinity) and was ' p -efficient' (the asymptotic efficiency approaches the theoretical maximum as p tends to infinity). It was shown by Hannan and Kavalieris [53] that if, at the first stage of Mayne and Firoozan's method, the order of the fitted autoregression was allowed to depend on the number of time points, in a reasonable manner, then it would still be true that the

final estimate of the parameter vector would converge, almost surely, to the true value.

Lim and Parker [54] presented methods for the identification of systems from data measurements using ARMA lattice models. They constructed the ARMA structure by the use of two-channel AR lattices and showed that one-half of the lattice parameters were zero when the system's excitation signal was white noise.

Alengrin *et al.* [55] proposed an identification scheme for ARMA processes in view of some results on the asymptotic behavior of the prediction error covariance for a state variable system. The coefficients of the d -step predictor determined asymptotically the system moments; these moments were also nonlinear functions of the coefficients of the successive 1-step predictors. First, the Burg's technique [8] was used to find the estimates of the coefficients of the successive 1-step predictors. Second, the moments were computed by substitution of the estimates provided by the Burg technique for the coefficients in the nonlinear functions relating the moments with the 1-step predictor coefficients. Finally the Hankel matrix of moment estimates was used to determine the coefficients of the characteristic polynomial of the state transition matrix.

Strobach's method for ARMA system identification [56] was an extension of his formerly proposed PORLA (Pure Order Recursive Ladder Algorithm) technique [106]. As the time recursion in this method was limited on the calculation of the input data covariance matrix, round-off errors could not propagate in time in higher stages of the pure order recursively constructed lattice form. Furthermore, the algorithm allowed the use of higher order recursive windows on the data (e.g., recursive Hanning), which significantly improved the tracking as well as the steady-state behavior.

Miyanaga *et al.* [57] proposed an adaptive method, which identified a modified 1-D ARMA lattice model. Their modified model was designed as a compact structure with a limited number of parameters. It consisted of time and order update recursions for model identification, an estimate of a self-tuning input and a stability check on an estimation model. In addition the method employed a weighting factor in order to adjust a window length to observed data.

An LD^2 -ARMA identifier, which solved the essentially nonlinear ARMA identification problem with a linear procedure, was presented by Ribeiro and Moura [58]. The algorithm combined an order selection scheme with a linear, dual, decoupled algorithm for the estimation of the AR and MA components.

A fast algorithm of Chandrasekhar type for ARMA model identification was published by Du *et al* [59]. By appropriately defining the extended state vectors and corresponding matrices, a state-space model was obtained from the ARMA representation so that the Kalman filter could be used as a parameter estimator. Because the resulting system was time invariant, it was possible to apply Chandrasekhar factorization techniques to calculate the filter gain and thus produce a fast algorithm.

Haseyama *et al.* [60] presented a method for model identification with frequency weighting using an adaptive ARMA lattice filter. This method, which made the accuracy of model identification in the focused frequency bands higher, was realized in two-steps. In the first step, an ARMA lattice filter with frequency weighting was used. In the second step, the ARMA parameters were computed from the coefficients of the ARMA lattice filter.

Liang and Wilkes's recursive identification method for ARMA model estimation [61] was based on the linear identification method of Mayne and Firoozan [52]. As the linear identification method required the fitting of a large-order AR model to the data, the length chosen for this large-order model played an important role in the behavior of the algorithm. Therefore an algorithm was also proposed for the appropriate selection of the order.

Chen *et al.* [62] considered ARMA robust system identification problem with noisy output data. The system output was corrupted by measurement noise, and the noise distribution was assumed to be unknown. A generalized l_p -norm iterative estimation algorithm ($1 < p < \infty$) proposed to approach the maximum likelihood estimate of system parameters according to the noise distribution.

Monin and Salut [63] derived an optimal linear L^2 -predictor of ARMA type in the lattice form of arbitrarily fixed dimensions for a process whose autocorrelation function is known. Their algorithm preserved exact optimality at each step, as opposed to asymptotic convergence of more usual algorithms, at the expense of

hereditary computation. Later they derived an identification solution of the ARMA-type optimal linear predictor as a time-varying lattice of arbitrarily fixed dimension for a process whose output signal only is known [64]. The projection technique they introduced led to a hereditary algorithm that was the adaptive extension to raw data of the authors' previous results on lattice realization from given autocorrelation functions.

Zhang and Asada [65] presented an algorithm for blind multi-channel identification of a group of infinite impulse response (IIR) systems, represented by ARMA models, and driven by white noise with zero mean. This approach identified the AR and MA parameters separately by solving two sets of linear equations using the least-squares method.

An on-line and robust identification scheme for time-varying ARMA processes was discussed by Efe *et al.* [66]. The approach discussed assumed solely that the orders of the numerator and denominator polynomials were known. The algorithm was demonstrated to be stable in the sense of Lyapunov, furthermore it was shown that the evolution in the parameter space took place in a finite volume.

Papakos and Fassois [67] introduced a comprehensive AR and a linear multistage ARMA framework for effective multi-channel structural identification under unobservable excitation. Aircraft skeleton structure identification was chosen as an application area for this framework.

ARMA spectrum estimation studies dates back to Kaveh [68] who introduced an ad hoc method for estimating the ARMA spectrum based on reasonably accurate estimates of the correlation function. Bruzzone and Kaveh [69] described a suboptimum scheme for ARMA spectral estimation. They presented a least-squares method and compared it to the method based on the modified Yule-Walker equations [68]. A modification of the latter method was also given which improved its behavior in estimating spectra with narrow peaks.

Kay's ARMA spectral estimator [70] involved a procedure for obtaining a nonnegative spectral estimate given an estimate of the autoregressive parameters.

Cadzow [71]-[72] proposed a method for generating an ARMA model spectral estimate of a wide-sense stationary time series from a set of finite observations. The method was based upon a set of error equations, which were dependent on the

ARMA model's parameters. Minimization of a quadratic functional of these error equations with respect to the ARMA model's parameters produced the desired spectral estimate. This ARMA spectral estimator has provided significantly better performance when compared to such standard procedures as the maximum entropy and Box-Jenkins methods [46]. The computational requirements of this method basically entailed the solving of a system of p linear equations in the autoregressive coefficients where p denoted the order of the ARMA model. Since an ARMA model would typically be of lower order than its autoregressive model counterpart for a specified fidelity of match, the proposed ARMA procedure was generally computationally more efficient than the maximum entropy method.

Friedlander [73] proposed a class of lattice prediction filters for high-resolution spectral estimation. The square-root normalized lattice recursions were used to estimate a set of reflection coefficients from the data. The lattice variables determined the coefficients of a least-squares predictor, from which the spectrum could be evaluated. The pre-windowed and sliding window (covariance) cases were considered for both AR and ARMA spectra. The behavior of the proposed spectral estimator was illustrated by simulation results. It was also shown by Friedlander [74] in a later publication that the modified Yule-Walker technique of ARMA spectral estimation was a special case of the instrumental variable method of system identification. A more detailed overview on ARMA spectral estimation techniques was presented by Friedlander and Porat [75]. The importance of using order overestimation, as well as of using an over-determined set of equations, was emphasized. The Akaike information criterion (AIC) was proposed for determining the equation order [107]. The role of the singular value decomposition (SVD) method for solving the modified Yule-Walker equations was also discussed.

Talkhan *et.al.* [76] proposed an ARMA model for power spectrum estimation of noisy random ergodic zero mean discrete time signals. In this model the residual power, which was not covered by the AR polynomial was represented by a limited order MA polynomial.

Ogino [77] developed a recursive algorithm for finding the parameters of Cadzow's high performance ARMA spectral estimation model [71]. The development of the algorithm was based on the shift invariant structure of the covariance matrix and orthogonal projection operators. The number of arithmetic operations at each arrival

of new data points was reduced to the order of p , with p being the order of denominator coefficients.

Ortigueira and Tribolet [78] described the double Levinson recursion for the determination of the AR coefficients of an ARMA model and ARMA orders. The algorithm was described in terms of an inner product formulation, using the theory of orthogonal polynomials. The double Levinson recursion formulation was associated with a double lattice structure in a generalized Burg method [8] fashion.

Lagunas-Hernández *et al.* [80] proposed a solution concentrating on obtaining maximum entropy ARMA models for spectral estimation, using cepstral constraints and correlation constraints simultaneously. The nonlinearity, which these constraints introduced was avoided by a simple linearization, that provided an estimator which was easily implemented.

Zhang and Takeda [81] presented an approach to time series analysis and ARMA spectral estimation from only the output data corrupted by noise. It was shown that the generalized modified Yule-Walker (MYW) equations held when the residual was some correlated noise. To solve such equations, a new version of the GLS method was proposed, yielding AR parameter estimates with higher accuracy. Furthermore, a simple procedure for improving MA parameter estimates was studied.

Ribeiro and Moura's dual algorithm for ARMA spectral estimation [82] substituted the autocorrelation estimation sequence by the sequence of estimated reflection coefficients provided by the Burg's technique [8]. Then it fitted to the process both a sequence of higher order linear predictors (e.g. Levinson algorithm) and a sequence of higher order linear innovations filters (e.g. by recursive inversion). Finally it obtained the MA coefficients from the linear relations satisfied by the corresponding coefficients of the successive higher order linear predictors, and likewise obtained the AR coefficients from the relations satisfied by the corresponding coefficients of the successive higher order innovation filters.

Mandal *et al.*'s work about reduced order ARMA spectral estimation of ocean waves [83] was based on the calculation of modal energies. Considering only the higher energy modes, the AR part of the reduced order ARMA model was obtained. The MA part was determined based on partial fraction and recursive methods.

Haseyama *et al.* [84] proposed a method for designing an adaptive four-line lattice filter, which could perform frequency weighting spectral estimation, which provided more accurate spectral estimation for some frequency bands than for others.

Kaderli and Ayhan [85] considered the spectral estimation problem of stationary ARMA processes and proposed a new method for the estimation of the MA part. A simple recursion relating the ARMA parameters and the cepstral coefficients of an ARMA process was derived and utilized for the estimation of MA parameters. The method required neither any initial estimates nor fitting of a large order AR model, both of which required further *a priori* knowledge of the signal and increased the computational complexity. It was in later publication that Kaderli and Ayhan [86] considered the spectral estimation problem of non-stationary processed and proposed a new method for the estimation of the time-varying spectral content of such signals. The proposed method can be viewed as an extension of the estimator proposed earlier by the authors to the time-varying case [85].

Scharf and Luby [87]-[88] presented procedures for the systematic design of ARMA digital filters, introducing three key ideas. The first was that one could begin the design of an ARMA digital filter by specifying the power-spectral density of a stochastic sequence and then posing a classical linear prediction problem for the stochastic sequence. The second was that, this prediction problem could be solved, by designing a high-order MA prediction filter. This prediction filter was related to an MA whitening filter, which might, in turn, be “inverted” to give an AR filter termed the inverse filter. The third key idea was that all-pole filter might be used to generate consistent unit-pulse and covariance sequences for use in the Mullis-Roberts algorithm [108].

In the Scharf and Luby’s design, the ripple of the filter in the realized response was maximum at the edges of the transition band. Even if this effect was intended to overcome by employing a smooth functional transition from passband to stopband, no satisfactory results were obtained as the MA whitening filter for the given spectrum tried to whiten a spectrum with zero amplitude in the transition band. This was unavoidable since an AR model approximated peaks in a spectrum better than valleys. In a method proposed by Yegnanarayana [89], this property of AR modeling was exploited to effectively control the ripple at the band edges. Spectral information around the transition band was transformed into peaks by splitting the given squared-

magnitude frequency response into two component spectra. This splitting was accomplished using a pole-zero decomposition technique, which in turn used the properties of the derivative of phase spectrum of minimum phase filters.

Lim and Parker [90] considered the synthesis of lattice parameter digital filters from a given z -transform transfer function, including several versions of AR, MA, and ARMA structures. The approach was based upon partial fraction expansions and led to numerical algorithms for parameter calculation, which were numerically robust in terms of zeros and poles close to the unit circle as well as self-correcting for finite precision coefficients. The synthesis of two families of ARMA lattice filters were discussed. One had its denominator synthesized in the lattice form. The numerator was then formed from a weighted sum of the signals at appropriate points of the lattice realizing the denominator. The second ARMA lattice filter was based upon the two-channel AR lattice model.

The synthesis of two families of ARMA lattice filters, injected numerator filters and the tapped numerator filters were presented by Lim [91]. For both families of filters, the denominator of the z -transform transfer function was synthesized using the single channel AR lattice. The transfer function's numerator of the injected numerator filter was realized by weighting and injecting the input signal into $N + 1$ appropriate points of the AR lattice where N is the order of the filter. For the tapped numerator filter, the transfer function's numerator was realized by a weighted sum of the signals tapped from $N + 1$ appropriate points of the AR lattice. The lattice structure introduced by Gray and Markel [109] was a special case of the tapped numerator design. The injected numerator filters were observed to possess certain overflow properties superior to the tapped numerator filters.

Miyanaga *et al.* [93] proposed an ARMA digital lattice filter based on a generalized form of the Mullis-Roberts criterion [108] defined for designing the filter from given stochastic data. The ARMA digital lattice filter was directly designed by an ARMA parameter estimation algorithm. The proposed algorithm consisted of two recursive formulas. One was an AR-type recursive formula, which estimated ARMA parameters as the AR order of an ARMA estimation model increased by one. The other was an MA-type recursive formula, which estimated ARMA parameters as the MA order of an ARMA estimation model increases by one. The particularity of this

filter was the design of an ARMA model with an arbitrary AR order and an arbitrary MA order.

Mennecke and Tummala [94] later introduced an adaptive algorithm based on the LMS criterion for the lattice structure proposed by Miyanaga *et al.* [93]. The adaptive algorithm proposed considerably simplified the original filter computations because of the symmetries identified in computing the instantaneous gradient elements.

Sellami and Newcomb [95] introduced a method to obtain degree-one or real degree-two transfer scattering matrices of two-port lossless lattice filters through the use of complex Richard's function extractions for the minimum degree cascade synthesis of real, stable, single-input, single-output ARMA (M , N) filters from the transfer function or the input reflection coefficient.

Haseyama and Kitajima [96] presented the matrix identities that were inherent in the solution of the normal equations for an ARMA lattice filter. In order to obtain the matrix identities, the four-line ARMA lattice filter shown in [84] was modified and hence the relationship between the recursive least squares (RLS) method and the ARMA lattice filter realization algorithm was made clear. Further, as an application of the matrix identities, a new method for the model identification with frequency weighting was presented.

Kwan [97] proposed a minimal normalized ARMA lattice filter structure for the realization of an arbitrary digital filter from a given transfer function or impulse response. This ARMA lattice filter required a minimal number of coefficients and delays; was orthogonal and modular; exhibited low round-off noise, absence of internal overflow; and with reflection coefficients bounded by unity for ease of stability check. Kwan's later publication on minimal ARMA digital lattice filters [98] focused on the realization of arbitrary order IIR digital filters, having the same filters properties as mentioned above. He gave lowpass, highpass, bandpass, and bandstop digital filter realization examples.

Friedlander and Maitra [100] presented a pole-zero recursive lattice form for speech analysis. The algorithm was based on the formerly developed square-root normalized lattice forms [110].

Miyanaga *et al.*[101] presented an adaptive speech analysis method using an ARMA model and the recognition system in which a robust ARMA speech analysis was implemented. The proposed system consisted of robust ARMA speech analysis and speech clustering/labeling method. In the speech analysis part, an adaptive ARMA lattice modeling was introduced. This modeling method calculated an ARMA parameter set at every observed speech sample. By using these ARMA parameters, the time-varying speech characteristics could be accurately estimated.

Kwan and Li [102] introduced an ARMA lattice model for speech modeling. The speech characteristics were modeled and expressed in the form of lattice reflection coefficients for classification. Self Organization Map (SOM) was used to build codebooks for classification and recognition of the lattice reflection coefficients. Experimental results were based on an isolated word speech database of 10 word/names indicated that the ARMA lattice model achieved superior recognition performance as compared to those of the conventional AR model. The ARMA lattice structure formerly developed by Kwan and Lui [36] was a building block for this work. By using this same lattice structure Kwan and Wang [103] presented the ARMA lattice model for speech synthesis. Simulation results indicated that a more natural speech can be synthesized by a proper choice of parameters. In general, the quality of the synthesized speech was a function of the ARMA lattice order and its frame size.

The maturity and completeness of the algorithms in one-dimension motivated the theoretical developments in two-dimensional signal processing. It appeared that the most straightforward approach was to extend the results of one-dimensional linear prediction theory to the two-dimensional case [111]-[113]. However, the major difference besides the dimensionality was that of causality. A large number of 1-D signal processing methods were based on the fact that the observed data was the output of a causal system. For 2-D data fields, the data coordinates are spatial and any causality associated with a data field is purely due to its ordering or acquisition technique. There are also problems related with the correlation matching property and stability in two-dimensions. Despite these difficulties, two-dimensional linear prediction models in AR parametric representations have been used for various applications such as system modeling [111]-[125], digital filter design [126]-[144], spectral estimation [145]-[156], image coding and compression [157]-[164], image

restoration [165]-[167], image enhancement [168]-[170], radar imaging [171]-[174] and object detection [175]-[176]. Lattice filters and hence reflection coefficient models have especially found numerous applications in the above mentioned areas as they were simple to implement, exhibited excellent numerical behavior and had a modular structure that made them attractive candidates for VLSI implementation.

A fundamental approach to modeling 2-D fields by the reflection coefficients was proposed by Marzetta [112]. He approached the spectral factorization problem by extending the results of one-dimensional (1-D) linear prediction theory to the two-dimensional (2-D) case. The recursive factors obtained were of infinite order in at least one of the two dimensions and the exact solution could only be obtained by solving an infinite set of linear equations. A 2-D analog of Levinson's algorithm was devised to obtain the theoretically guaranteed solution. More importantly, it was shown that a one-to-one relationship existed between the reflection coefficients obtained in the Levinson algorithm, the associated prediction error filters, and the covariances of the random field used in the algorithm. Also, it was proven that reflection coefficients given on a finite support yielded finite support causal factors. These facts were used to design an approximate spectral factorization algorithm where the reflection coefficients were sequentially chosen on a finite lattice to minimize a prediction error functional. Marzetta's algorithm was successfully applied to 2-D recursive filter design [111] and to linear predictive coding of images [157].

A relation between the multi-channel 1-D AR model and the single-channel 2-D models with quadrant support was proposed by Therrien [113]. He derived a procedure for solving 2-D normal equations by relating the 2-D linear prediction problem, applying the multi-channel Levinson recursion. The procedure permitted computation of parameters for all quarter-plane filters simultaneously.

The modeling of 2-D AR data fields with quarter-plane (QP) lattice parameters was first proposed by Parker and Kayran [114]. Starting with a given data field, four prediction error fields were generated. Their linear combination was used to define and calculate 2-D lattice parameter reflection factors for successive lattice parameter model stages. In addition to developing the basic theory, their presentation included the relationships between the quarter-plane 2-D transfer function and the lattice parameters, the lattice parameter synthesis model for generating a 2-D field from a

random field input, and lattice parameter stability conditions. This structure was later generalized by Kayran [121], who developed an orthogonal four-field 2-D QP lattice structure with a complete set of reflection coefficients by employing appropriately defined auxiliary prediction errors. After the first stage four auxiliary forward and four auxiliary backward prediction errors were generated in order to obtain a growing number of 2-D reflection coefficients at successive stages. The theory was proved by using a geometrical formulation based on the mathematical concepts of vector space, orthogonal projection, and subspace decomposition. It was shown that all four quarter-plane filters were orthogonal and thus optimum for all stages. In addition to developing the basic theory a set of orthogonal backward prediction error fields for successive lattice parameter model stages were presented. In contrast with other existing four-field configurations [114]-[115], [131], the proposed structure could generate a set of orthogonal backward prediction error fields.

Lev-Ari and Parker [115] contributed to the 2-D AR lattice filter modeling efforts as an extension of one dimensional lattice filter theory by developing several distinct methods of extension, including a transfer function approach and a stochastic approach. The resulting lattice configurations spanned a wide range of structural complexity, all exhibiting the cascade structure of the conventional 1-D lattice filter but differing widely in the structure of their elementary sections. It appeared that orthogonality of prediction errors, which is essential for robust numerical behavior, required high structural complexity. As a result, they described a configuration that offered a trade-off between perfect orthogonality and structural complexity.

Zhao and Yu [117] extended the theories of the 2-D quarter plane and asymmetric half-plane AR lattice modeling developed by Parker and Kayran [114], [146] to the general noncausal whole-plane case. Their proposed 2-D noncausal AR lattice modeling involved nine prediction error fields and as many as 72 lattice parameter coefficients at each stage to be able to implement as many 2-D AR transfer functions as possible in the lattice configuration.

Two-dimensional orthogonal lattice structures developed by Kayran [119] offered a complete solution for the Levinson-type algorithm to compute the prediction error filter coefficients using lattice parameters from the given 2-D augmented normal equations. The proposed theory could be used for the quarter-plane and asymmetric half-plane models. Depending on the indexing scheme in the prediction region, it

was shown that the final order backward prediction error might correspond to different quarter-plane models. In addition to developing the basic theory, the presentation included the several properties of lattice models. Conditions for lattice model stability and an efficient method for factoring 2-D correlation matrix were given. It was shown that the unended forward and backward prediction errors formed orthogonal bases. The applicability of these lattice structures to the short data records was shown by Yildiz (Sari) [118].

Based on his orthogonal 2-D lattice structure [119], Kayran [120] extended the theory of the 1-D Schur algorithm, which was an efficient alternative to Levinson's recursion, to the 2-D case using a new generator matrix. Lenk and Parker [116] had also extended the 1-D Schur algorithm to the 2-D case but they had used tensor concepts to derive 2-D lattice filters by considering orthogonalizing coordinate transformations. Another important contribution on 2-D Schur algorithm was made by Kayran *et al.* [123] based on Parker and Kayran's four-field lattice approach [114]. Starting with given 2-D autocorrelation samples, four quarter-plane gapped functions were generated. Their linear combination was used to satisfy gap conditions and calculate 2-D lattice parameter reflection factors for the first stage. In order to determine the growing number of reflection coefficients at successive stages, appropriately defined auxiliary gapped functions were introduced after the first order.

Nakachi *et al* [122] proposed a 2-D Levinson algorithm and a lattice filter for the general case of AR model with an asymmetric half-plane (ASHP) support by extending Lenk and Parkers's Levinson algorithm [116]. The resulting Levinson algorithm and corresponding lattice filter solved the 2-D normal equations recursively.

Marple [124] presented an extension of the 1-D lattice technique of linear prediction parameter estimation, first popularized by Burg [8], to the 2-D case. The resulting fast recursive 2-D algorithm is a significant computational simplification over and an estimation improvement on previous attempts to extend the 1-D Burg linear prediction algorithm to 2-D by exploiting some newly discovered matrix structures. The technique presented was useful for high-resolution 2-D spectral analysis applications and the creation of high-resolution spotlight-mode synthetic aperture radar (SAR) imagery

Kayran and Erer [125] presented a new optimum asymmetric half-plane autoregressive lattice parameter modeling of two-dimensional random fields. This structure introduced $4N$ points into the prediction support region when the order of the model increased from $(N-1)$ to N . Starting with a given data field, a set of four auxiliary prediction errors were generated in order to obtain the growing number of 2-D ASHP reflection coefficients at successive stages. The theory was applied to the high-resolution radar imaging problem and was also proven using the concepts of vector space, orthogonal projection, and subspace decomposition. It was shown that the proposed ASHP structure generated the orthogonal realization subspaces for different recurse directions. In addition to developing the basic theory, the presentation included a comparison between the proposed theory and other alternative structures, both in terms of conceptual background and complexity. While the recently developed reduced-complexity ASHP lattice modeling structure required $O(4N^3)$ lattice sections with N equal to the order of the error filter [122], the proposed configuration required only $O(2N^2)$ lattice sections.

Kwan and Lui [127] presented the implementation of a quarter-plane autoregressive 2-D filter using a newly developed 2-D Levinson algorithm. The resulting structure inherited most of the nicer characteristics of the well-established 1-D lattice filter such as high modularity, low coefficient sensitivities, low roundoff noise, and elimination of internal overflow.

Qunshan and Xiyu [128] proposed a 2-D lattice structure and explicit formulas for the design of 2-D digital fan filters. They used complex transformation to map 1-D low-pass digital filter to 2-D fan filter. The relations of the frequency characteristics between 1-D low-pass filter and 2-D fan filter were introduced. Taking these results at hand, they developed simple and efficient explicit formulas for the design of fan filters. For an arbitrarily given fan filter with its pass-band ripple, stop-band loss and transmission band-width, the filter coefficients could easily be obtained.

Kayran [129] presented a method based on 2-D asymmetric half-plane lattice parameters for the design of 2-D recursive digital filters. The design procedure calculated the lattice parameter factors from the prescribed frequency characteristics. The order of the design was controlled by an a priori error criterion, corresponding to the minimum mean squared error between two successive stages. The stability of the resulting filter was determined at each successive lattice stage from the computed 2-

D reflection factors. The capabilities of the proposed approach were demonstrated by the design of a fan filter and circularly symmetric lowpass filter.

In order to decrease the amount of information loss, an in turn to model the AR data field in the best possible way, Ertüzün *et al.* [130]-[131] developed a lattice structure by extending the lattice structure of Parker and Kayran [114] and called this new structure “Extended Lattice Filter of Perpendicular Form (ELPF)”. ELPF generated four prediction error fields (one forward and three backward prediction error fields) at the first stage. After the first stage, using one of the backward prediction error fields, two additional prediction error fields were generated. It was concluded that ELPF approximated the maximum entropy more closely compared to the structure of Parker and Kayran [114]. The drawback of the Ertüzün’s lattice structure was that it could only model first and second order AR fields exactly and lacked a sufficient number of parameters to model all AR fields of higher order. Although it was shown that the number of reflection coefficients could be increased at every stage to model all AR fields exactly, this introduced a considerable amount of arithmetical complexity. A further improved 2-D lattice filter structure was proposed by Ertüzün *et al.* [132] to overcome this drawback. This structure could model all AR fields of degree three or less exactly and maximizes the entropy contained in the backward prediction error fields.

One of the issues related with the 2-D lattice structure of Parker and Kayran [114] was the fact that stability was not guaranteed. Ertüzün *et al.* [133] presented three different lattice filters, which were structurally stable for all parameters. These three filter structures were obtained using the structural stability concept introduced by Lev-Ari and Parker [126] and applying this concept to modify the lattice filter of Parker and Kayran.

Mahmoud *et al.* [134] described the realization of 2-D FIR filters using lower upper triangular (LU) decomposition, where 1-D filters resulting from the decomposition, were realized by 1-D lattices. The obtained 2-D digital filter was highly modular in structure. The idea of matrix decomposition was used to provide increased parallelism and regularity. The LU decomposition of the 2-D digital filter transfer function had many considerable properties for VLSI implementation. The lattice realization was chosen because of its superiority in finite word length performance.

Kawamata *et al.* [135] proposed separable-denominator two-dimensional (SD- 2D) adaptive filters that were realized in lattice and direct forms. The stability test for these SD-2D adaptive filters could be simplified to that for one-dimensional (1-D) recursive filters. Improvements of SD-2D adaptive filters over a non-recursive 2-D adaptive filter were shown by the experiments of 2D system identifications. Moreover it was shown by the experiments of the noise reduction that the lattice-form SD-2D adaptive filter was superior to the 2-D counterpart.

Kuduvalli and Rangayyan [136] presented some simplifications to the method of computing 2-D linear prediction coefficients (LPC's) directly from the image data using an extension of the multichannel version of the Burg algorithm presented by Therrien and El-Shaer [149]. This simplification resulted from forcing the structure of the 2-D auto-correlation matrices on the multichannel version of the Burg algorithm. In addition to computing the 2-D LPC's, the method could be used for the computation of prediction errors directly from the data. The proposed algorithm was also used to compute the LPC's estimating the 2-D power spectrum for the example given by Therrien and El-Shaer [150].

Kayran and Eksioglu [137] introduced a simple and computationally efficient algorithm for 2-D Wiener filter using the orthogonal backward prediction error fields of the 2-D lattice configuration in [119]. Moreover, it was shown that the set of normalized versions of the transfer functions of the backward prediction error filters were the 2-D analogues of the single-variable Szegö polynomials.

Antoniou [138] presented a circuit realization for 2-D lattice discrete filters with the minimal number of delay elements. Based on this circuit realization, the corresponding state-space realization was derived. The dimension of the 2-D state-space vector was minimal and the corresponding transfer function was characterized by the all-pass property.

Schnauffer and Jenkins [139] developed a 2-D adaptive filter structure, based on an extension of the 1-D joint process estimator (JPE). This structure, called the 2-D JPE, used the 2-D lattice filter introduced by Parker and Kayran [114] followed by a generalized lower tap-weight arrangement. The proposed 2-D JPE was shown to have superior convergence rate performance over the 2-D direct form adaptive filter when the parameters of the 2-D lattice were fixed at their optimal values, given the autocorrelation of the input field. This implied that when the structure was made

fully adaptive, it could offer better convergence rate performance for almost any input field. One observed drawback of the fully adaptive 2-D JPE was that it exhibited a rather high noise floor due to the adaptation of lattice reflection parameters.

Nakachi *et al.* [140] proposed a 2-D adaptive joint-process lattice estimator, which could represent a wide class of 2-D FIR systems and possessed orthogonality in the backward prediction error fields. The algorithm resulted in superior convergence and tracking properties versus, the transversal filter and other adaptive 2-D methods. The effectiveness of the proposed model was evaluated for noise cancellation through computer simulation.

Yamashita *et al.* [141] constructed a 2-D adaptive joint-process IIR filter with the generalized lattice structure. This filter was an IIR lattice filter with restricted feedback structure, but the backward prediction error fields for every stage, like FIR lattice filter, were orthogonal which produced fast convergence. More importantly it could borrow both FIR and IIR features and simultaneously held the well-known merits of lattice structure. The validity of the proposed model was evaluated through computer simulation for image restoration problem. Noise reduction was achieved by using the proposed IIR filter with the generalized lattice structure for the problem of 2-D adaptive noise canceller.

Moro *et al.* [142] proposed a 2-D lattice filter with six reflection coefficients at each stage by modifying the three-parameter structure of Parker and Kayran [114] and designed an adaptive lattice parameter scheme based on this improved lattice filter. They also proposed a method to improve the convergence rate of lattice parameter estimation.

Liu *et al.* [143] proposed a 2-D fast lattice recursive least squares (FLRLS) algorithm. This algorithm could update the filter coefficients in growing-order form with a computational complexity $O((M+1)K_1)$, where $(M+1)$ was the number of channels and K_1 was the total number of data used in the 2-D filter. A reduction in the computational cost was obtained when compared with the standard RLS algorithm ($1.5 K_1^2$). One major advantage of the 2-D FLRLS algorithm was that the model order could be adjusted dynamically. One did not need to determine the order, a priori, but just start with the orthogonalization procedure and stop as soon as the desired accuracy was reached.

Liu and Najim [144] derived a two-dimensional fast lattice recursive least-squares algorithm. This algorithm updated the filter coefficients in growing-order form with a linear computational complexity. After appropriately defining the order of the 2-D data and exploiting the relation with 1-D multichannel, order recursion relations and shift invariance property were derived. The geometrical approaches of the vector space and the orthogonal projection then could be used for solving this 2-D prediction problem.

Cadzow and Ogino [145] developed procedures for generating 2-D quarter-plane causal AR and ARMA spectral estimation models. These procedures were found to provide super resolution capabilities when compared to other more classical methods, such as the Fourier transform. ARMA model's autoregressive coefficients were selected to minimize a weighted least squares criterion composed of error elements while the moving average coefficients were obtained using an alternative approach. However there were some difficulties associated with these procedures, including weighting coefficient selection and a heavy computational burden for estimating the AR parameters.

Kayran, Parker and Klich [146] presented an approach for estimating the spectrum of 2-D fields from their auto-correlation matrix, based upon the formerly developed AR 2-D lattice parameter model [114].

Ranganath and Jain [147] presented results concerning realization of causal, semicausal, and noncausal models by linear prediction principles. For each class of models they presented two types of algorithms. In the first algorithm, they started with the given power spectrum density function (SDF) to obtain infinite order (irrational) realizations. A realization procedure was incorporated into the algorithms to provide stable models. As in all prior methods, these algorithms required solution of an infinite number of equations. A Levinson-type algorithm presented for semicausal models turned to be an extension of a similar algorithm [111]-[112] for causal models. The results for noncausal models did not require spectral factorization. The second algorithm yielded approximate rational realizations of 2-D SDF's, which were obtained by solving a finite number of equations.

Kimura and Honoki [148] proposed a practical approach to the spectrum estimation of 2-D homogeneous random fields that fully utilized the high resolution performance of the ME estimate, on the one hand, and avoided the highly nonlinear

nature of the ME estimation for 2-D signals, on the other hand. This algorithm was of the hybrid nature in the sense that the "time domain" estimation and the "frequency domain" estimation are mixed together. It was also the hybrid of the parametric and the nonparametric estimations.

Kayran [152] reported a method to obtain the 2-D ML spectrum estimation using orthogonal 2-D AR lattice filters [119]. It was shown that the reciprocal of the 2-D ML estimate was equal to the average of the reciprocals of the AR spectra obtained from the unended orthogonal forward or backward prediction error filters in the 2D AR lattice structure.

Alata *et al.* [153] proposed a multi-channel approach called Harmonic Mean Horizontal Vertical (HMHV) in the framework of high resolution 2-D spectrum analysis. It was based on the 2-D fast recursive least squares (FRLS) algorithms and their use for the computation of causal 2-D AR parameters. This HMHV spectrum revealed better frequency estimation results than harmonic mean (HM) spectrum. First, at low order and low SNRs, the HMHV estimate presented lower variances of frequency estimation in the case of a sinusoid in white Gaussian noise. Secondly, it had the same estimation properties in all directions, and finally it generated less spurious peaks than other methods.

Rouquete *et al.* [154] dealt with frequency estimation in the 2-D case in the case of only few data points. They proposed a method to estimate the frequencies of a sum of exponentials. This method was based on an original set of 2-D linear prediction models with new regions of support derived from the standard quarter plane support region. These models defined various spectra, which were finally combined by computing their harmonic mean. This method benefited from the subspace decomposition of the covariance matrix to perform well. It was demonstrated that the new regions of support improve the spectrum geometry and the estimation accuracy compared to the classical QP support regions.

Jakobsson, Marple and Stoica [155]-[156] presented a computationally efficient algorithm for computing the 2-D Capon spectral estimator. The implementation was based on the fact that the 2-D data covariance matrix would have a Toeplitz–Block–Toeplitz structure, with the result that the inverse covariance matrix could be expressed in closed form by using a special case of the Gohberg–Heinig formula that was a function of strictly the forward 2-D prediction matrix polynomials.

Furthermore, they presented a novel 2-D lattice algorithm to estimate the needed (forward) prediction matrices and the (forward) prediction error covariance matrix. The presented algorithm was considered as an improvement over the previous attempts to extend the linear prediction lattice-based 1-D techniques to two dimensions [136],[149],[151],[171].

Efficient compression algorithms can be devised for digital imagery data via an extension of the linear predictive coding methods, which exploit data redundancy and correlation properties. Poehler and Choi [157] used a linear prediction based approach for image compression employing 2-D lattice filter prediction, adaptive quantization and entropy coding. System implementation showed proof of feasibility and favorable performance in comparison with alternative transform techniques using the standard Minimum Mean Square Error (MSE) fidelity criterion.

Maragos *et al.* [158] summarized a study on two-dimensional linear prediction of images and its application to adaptive predictive coding of monochrome images. The study was focused on three major areas: two-dimensional linear prediction of images and its performance, implementation of an adaptive predictor and adaptive quantizer for use in image coding, and linear prediction and adaptive predictive coding of density (logarithm of intensity) images. Among the issues investigated were: AR modeling of 2-D image sequences, estimation of the nonzero average bias of the image samples, stability of the inverse prediction error filter, and estimation of the parameters of a 2-D separable linear predictor. The implementation of the adaptive predictor was based on the results of linear predictive analysis. The adaptive quantization of the prediction error signal was done by using a flexible three-level quantizer for code words of fixed or variable length. The above ideas were further applied to density images for exploiting the multiplicative structure of images. The results of this research indicated that by using adaptive prediction and quantization, intensity and density coded images of high quality could be obtained at information rates as low as 0.7 bits/pixel.

El-Shaer and Therrien [160] described 2-D multi-channel linear prediction and its application to predictive coding of color images, introducing two algorithms initially. In the first algorithm, the whole frame of image was divided into sub-frames and the predictor coefficients were computed separately for each. In the second, predictor coefficients were obtained for the whole frame of the image. Both of these methods

had the advantage that the linear prediction coefficients matrices had to be computed in real time and transmitted to the receiver as side information. This significantly increased complexity of the coding system. As an alternative, using a fixed set of prediction matrices, i.e. one that did not depend on the specific image being coded, was considered.

Onuk and Ertüzün [161] applied four different 2-D lattice filters to image compression. These were: three parameter lattice filter [114], extended lattice filter of perpendicular form [130], extended lattice filter of diagonal form [131] and further improved lattice filter [132]. Different images were compressed with these filters and their performances were compared with each other. The results of the computer simulations showed that further improved lattice filter produced somewhat better results.

Kwan and Chan [162] introduced a non-causal predictive lattice model (NCPLM) for image compression based on the structures in [119],[127]. Owing to its full plane support, the NCPLM appeared a more effective model in reducing redundancy for improved image compression as compared to causal predictive models. To avoid instability and nonrealizable synthesis problems in association with a noncausal support, the binary pyramid decomposition was adopted. The applications of the NCPLM for implementing lossless and lossy image CODEC were presented. Simulation results indicated that the proposed NCPLM is effective as a predictive algorithm for image compression.

Ives [163] proposed a lossless SAR compression scheme that used multiple passes of an adaptive filter to decorrelate the data prior to entropy coding. As a variation of this two-stage lossless algorithm, he later reported the results of the near-lossless compression of SAR-imagery using a three-stage predictive compression algorithm [164]. This first stage of this latter algorithm consisted of multiple passes of a gradient adaptive lattice (GAL) filter functioning as a predictor.

Youlal *et al.* [166] examined the joint process lattice (TDJPL) and its implementations for image restoration applications. They first developed a two-dimensional adaptive lattice filter algorithm (TDAL) exploiting the 2-D lattice modeling technique of Parker and Kayran [114]. Convergence properties of the algorithm were covered for the 2-D adaptive lattice least mean squares (TDAL-LMS) case. The complexity of the normalized algorithm (TDAL-NLMS) was slightly more

than that of the TDAL-LMS, but was a faster converging algorithm. Implementations of the proposed TDJPL estimator as a 2-D adaptive lattice noise canceller (TDALNC) and as a 2-D adaptive lattice line enhancer (TDALLE) were also considered. Performance evaluation of both schemes was undertaken using artificially degraded image data at different signal-to-noise ratios (SNR's). The results showed that substantial noise reduction was achieved, and high improvement in the mean square error (MSE), even at very low input SNR, was ensured. The results obtained consistently demonstrated the efficacy of the proposed TDJPL implementations, and illustrated the success in its use for adaptive restoration of images.

Youlal *et al.*'s adaptive lattice algorithms [166] lacked the property of orthogonality so that the cascaded stages would not lead to an optimum filter. Kayran [167] proposed an improved 2-D adaptive joint process lattice estimator using orthogonal quarter-plane 2-D lattice filters with a complete set of reflection coefficients for AR modeling of 2-D random fields.

Meylani *et al.* [168] used the eight-parameter 2-D adaptive lattice filter to detect defects in textures corresponding to raw textile fabrics. The eight-parameter 2-D adaptive lattice filter was derived, by applying the adaptation algorithm developed by Moro *et al.* [142]. A histogram modification technique was also applied for preprocessing the gray level texture image. Moreover, with the proposed scheme, it was possible to detect defects using the defective image only.

Hendekli and Ertüzün [169] presented a two-dimensional delta domain lattice filter and applied this filter to the adaptive image enhancement problem. The presented filter combined the concepts of 2-D lattice filter structure and the delta domain (DD) operator. A new 2-D lattice filter structure which was a transformation of the approach given in [114] into the DD, was developed and was applied in the form of a 2-D joint process lattice estimator, which acted as a noise canceller, to several images degraded by additive noise. Motivated by the success of 2-D delta-domain lattice filters, the authors later extended this work to the multi-channel case [170]. By a good choice of the transformation matrix that related the q -domain and delta-domain filters, the channels could be handled independently, which made the algorithm simple and attractive; however it was capable of handling multichannel problems, even if the noise was correlated between channels, resulting in extremely

low SNR values. The proposed algorithm provided the ability of handling multi-channel image frames and colored images in the so-called delta-domain that was known to be always better than the traditional q -domain in handling ill-conditioned problems. On-line image frames carrying Gaussian and multiplicative noise simultaneously on different channels, as well as color images, were taken into consideration, and successful results were obtained in terms of noise removal.

Erer, Kartal and Kayran [173] proposed a 2-D data extrapolation method for high resolution radar imaging data beyond the measurement range, using autoregressive lattice modeling proposed by Kayran [119]. As this technique did not guarantee a stable prediction filter, one might have to modify the prediction parameters to ensure stability. These modified parameters could be used for the 2-D extrapolation of the 2-D Cartesian backscattered data. An inverse Fourier transform (IFT) was applied on the extended data to obtain images with improved resolution in both range and cross-range. Since the proposed method was based on the 2-D IFT of the extrapolated data, it did not require any further peak searching or amplitude estimation. The authors also presented an algorithm to obtain high-resolution inverse synthetic aperture radar (ISAR) images in the case of limited frequency band and small angular section [174]. The method was also based on the 2-D AR asymmetric half-plane modeling of the backscattered data using the complex form of the 2-D orthogonal lattice predictor described [119]. The scattering centre locations were determined by the zeros of the transfer function of the prediction error filter.

Ffrench *et al.* [175]-[176] focused on the development of an improved 2-D adaptive lattice algorithm (2-D AL) and its application to the removal of correlated clutter to enhance the detectability of small objects in images. The two improvements proposed were the increased flexibility in the calculation of the reflection coefficients and a 2-D method to update the correlations used in the 2-D AL algorithm. The 2-D AL algorithm was shown to predict correlated clutter in image data and the resulting filter was compared with an ideal Wiener-Hopf filter. The results of the clutter removal were compared to previously published ones for a 2-D least mean-square (LMS) algorithm. 2-D AL was better able to predict spatially varying clutter than the 2-D LMS algorithm, since it converged faster to new image properties.

Kashyap [177] introduced a class of finite-order 2-D ARMA model that can represent any process with rational spectral density. In this model the driving noise

was correlated and need not be Gaussian. Two approaches were considered to estimate the parameters of a model to fit a given image. The first method used only the empirical correlations and involved the solution of linear equations. The second method was the likelihood approach. Since the exact likelihood function was difficult to compute, they resorted to approximations suggested by the toroidal models.

Kwan and Lui [178] proposed a Levinson-type algorithm for estimating the 2-D ARMA model from the observed inputs and outputs of an unknown 2-D system. By applying the concept of dummy prediction errors, their algorithm could be extended for computing 2-D models of any AR-order and MA-order. They have also shown that the algorithm could be greatly simplified when the input was a white noise process.

Zhang and Cheng [179] presented a practical algorithm for estimating the power spectrum of a 2-D homogeneous random field, based on 2-D autoregressive moving average (ARMA) modeling. This algorithm was a two-step approach: first the AR parameters were estimated by solving a new version of the 2-D modified Yule-Walker (MYW) equation, for which some existing efficient algorithms were available; then the MA spectrum parameters were obtained via simple computations.

Mikhael and Yu [180] presented an algorithm for 2D ARMA and MA modeling in the Fourier transform domain, which could also be implemented in other domains. The algorithm derived was based on the equationary model (recursive-like) structure which could match, in the least square (LS) sense, the unknown system. Simulation examples of 2-D system modeling were given to demonstrate the performance of this algorithm.

Chang and Kanefsky [181] adapted a 2-D linear predictor, which had an ARMA representation as well as a bias term for adaptive differential pulse code modulation (ADPCM) encoding of non-negative images. The predictor coefficients were updated by using a 2-D recursive LMS algorithm. A constraint on the optimum values for the convergence factors and an updating algorithm based on the constraint was developed. The coefficient updating algorithm could be modified with a stability control factor. This realization was able to operate in real time and in spatial domain. A comparison of three different types of predictors was made for real images. ARMA predictors were stated to show improved performance relative to an AR algorithm.

Gosh and Mikhael [182] presented 2-D optimal algorithms for image compression using ARMA predictors. The 2-D algorithms employing optimal convergence factors were employed using an adaptive differential pulse code modulation (ADPCM) scheme. Perceptual improvements and corresponding increases in SNRs over existing schemes were seen when the algorithms using optimal convergence factors were used in conjunction with an ARMA predictor.

Mikhael and Yu [183] presented a linear algorithm based on the equation error model for 2-D LS approximation in the frequency domain. The approximation yielded a 2-D approximation function in the complex variables, or equivalently a 2-D ARMA process. The proposed 2-D LS frequency domain was able to efficiently represent 2-D signals or images. It was also capable of accurately modeling 2-D linear shift invariant (LSI) stable systems when the model had a sufficient order relative to the unknown and the identification noise was negligible. However this algorithm was developed for real-valued data and for the 2-D ARMA model with the same model order for each dimension. When the algorithm was applied in areas such as radar processing, these limitations were exposed. Therefore, Zhang *et al.* [184] generalized this algorithm to the complex-valued data case and obtained a more general ARMA model structure. They also introduced a simple model order selection method, which was especially suitable for this generalized 2-D frequency domain LS algorithm. The authors later discussed the application of their generalized algorithm to modeling the ground clutter in airborne surveillance phased array radar scenarios [185].

Nijim *et al.* [185] proposed an algorithm for the lossless compression of 2-D signals. This approach was based on modeling the original signal by a rational function, which consisted of poles and zeros, or equivalently an ARMA process. The equation-error structure, which approximated the signal by minimizing the error in the least square sense, was used to obtain the optimal coefficients of the transfer function. This technique was implemented in the frequency domain.

Kizilkaya and Kayran [187] addressed the parameter estimation problem of a stationary QP 2-D ARMA model driven by an unobservable input noise and proposed a new algorithm for the estimation of the MA part of the 2-D ARMA process. The proposed method was based on the recursion relating the 2-D ARMA model parameters and the cepstral coefficients of the 2-D ARMA process. The MA parameters were obtained by using the estimated cepstral coefficients and the

autoregressive (AR) parameters of the 2-D ARMA process in the developed recursive equation. In the proposed method, the cepstral coefficients were found by utilizing the periodogram of the ARMA process. Then, the AR parameters were estimated by solving the 2-D modified Yule-Walker (MYW) equation and some numerical examples were included to verify the formulation.

Eksioglu and Kayran [188] developed a new lattice-ladder structure for the realization of 2-D ARMA digital filters. This structure utilized the formerly proposed 2-D AR lattice model [119] as the backbone and juxtaposed a ladder section to this 2-D AR model to create the full ARMA structure, eliminating any redundancy from the lattice reflection coefficients. The authors also presented a recursive algorithm to calculate the lattice-ladder coefficients for any given 2-D ARMA transfer function.

It was Kayran's work on 2-D ARMA lattices [189] that constituted a base for this thesis. He proposed a new 2-D ARMA modeling technique using two-channel AR lattice approach based on his orthogonal 2-D lattice filter structures [119]. Such an approach was formerly used for 1-D lattice structures by Lim and Parker [90]-[91]. The main drawback of this two-channel 2-D lattice structure was that it was only able to model 2-D ARMA processes where AR order was equal to the MA order. Also a solution for the ARMA parameter calculation was not included.

This thesis addresses the problem of identifying the parameters of a linear shift-invariant (LSI) 2D ARMA (M, N) system, where M and N can take arbitrary values different from each other. We have proposed a novel hybrid lattice structure, which incorporates both two-channel and single channel lattice stages in an interleaved manner in order to solve this problem [190]-[191]. The two-channel lattice part is based on the formerly proposed 2D ARMA lattice modeling approach where only the case $M=N$ was covered [189]. The single channel part is composed of 2-D orthogonal lattice structures proposed for AR modeling purposes [119]. We have estimated the parameter b_0 , which is generally assumed to be "one", by modifying the formulations originally derived by Perry [1] for 1-D systems and later proposed by Kayran [119] to use for 2-D systems. We have obtained "primitives" for ARMA system parameters at the output of the 2-D hybrid lattice analysis model. By using these primitive parameters and the estimated b_0 values, we have derived new formulations to obtain the "real" ARMA system parameters and have given the proofs in Appendices. We

have also applied our algorithm to the image identification problem [192] where the simultaneous identification of image and blur parameters has been possible.

In Chapter 2, we talk about 2-D system identification with a focus on the overview of our proposed hybrid lattice structure. We make the problem statement defining the ARMA (M, N) model to be identified and explain the key point in our proposed solution, which is to drive the first channel with a difference signal of system output and system input. This kind of an embedding approach was applied in 1-D recursive covariance ladder algorithms for ARMA system identification [56]. However, in the 2-D case, the use of such a difference signal brings with it the limitation that the data points of both the input and output signals in the prediction region should have the same ordering at least for the filter orders extending to the smaller one of the (M, N) pair. The second channel input is driven by either system input or system output, depending on the relation between the (M, N) pair. If $M \geq N$, the second channel is driven by the system input, but if $M < N$, the second channel is driven by the system output. The novel approach we present here can be called as a “hybrid lattice structure” involving both two-channel and single channel lattice stages. In the proposed procedure AR and MA parameters of the ARMA models are obtained simultaneously.

In Chapter 3, we explain in detail the lattice filter design steps needed to construct an ARMA system where M and N can take arbitrary values different from each other. We also modify the formulation for the estimation of the parameter b_0 , based on the approach proposed by Perry [1]. As our channel inputs change with the (M, N) pair, the estimation of the parameter b_0 takes two different values for the case $M \geq N$ and $M < N$.

In Chapter 4, we give the relations of the lattice parameters to the 2-D ARMA model coefficients. We also propose a new formulation for computing the ARMA parameter estimates in terms of the forward prediction error filter tap-weights of both channels and the parameter b_0 . The proof of all the formulation is given in Appendices.

In Chapter 5, we give an example to identify the parameters of an ARMA (3,2) system in order to illustrate the effectiveness of the proposed method. We solve for

the system step by step explaining in detail the way to compute the reflection coefficients and ARMA parameters.

In Chapter 6, we concentrate on the simultaneous identification of image and blur parameters using our hybrid lattice structures.

In Chapter 7, we focus on the performance evaluation of the proposed method using L_1 , L_2 and L_∞ vector norms and Itakura-Saito distance measure, which indicates the similarity between the original and estimated power spectrums. We also talk about Least Square estimation, which we use as a benchmark to compare our hybrid lattice identification results.

In Chapter 8, we give experimental results that make a verification of our proposed structure. We give simulation results for different data field sizes of the ARMA systems where $M=N$, $M>N$ and $M<N$ and also for first quarter-plane (FQP) and asymmetric half-plane (ASHP) supports. The results obtained show that quite satisfactory results are obtained using our proposed novel structure. We characterize the parameter identification results with their mean values, which have been obtained by 100 independent Monte-Carlo runs for each of the simulations. We also give the power spectrums and contour plots of the identified systems and compare it with the original ones. We calculate the Itakura-Saito distance measure, L_1 , L_2 and L_∞ vector norms and LS estimates for all of the simulations and show that successful results have been obtained.

In Chapter 9, we elaborate on the concluding remarks and future work.

2. TWO-DIMENSIONAL SYSTEM IDENTIFICATION: AN OVERVIEW OF THE PROPOSED STRUCTURE

The aim of system identification is to fit the given data, usually supplied in the form of a time series, with models from within a given model class. One can divide the main challenges of system identification into three successively stronger questions, as follows: As more and more data is provided to the identification algorithm,

- 1) Does the estimation error between the outputs of the identified model and the actual time series approach the minimum possible estimation error achievable by any model within the given model class?
- 2) Does the identified model converge to the best possible model within the given model class?
- 3) Assuming that the data is generated by a ‘true’ model, the output of which is corrupted by measurement noise, does the identified model converge to the true model?

The fundamental problem of identifying an LSI system from measurements of its output response to known input excitation such as a white noise source, is one that impacts on many important fields of interest. In the case of rational systems, the identification problem is to evaluate the degree (model order) of its numerator and denominator polynomials as well as their coefficients (system parameters). Because of its numerical robustness, linear identification under lattice form is of special interest. But as there is no natural ordering of the data samples in the 2-D domain, lattice filter solution in 2-D is not unique. Many valid structures have been developed exhibiting different properties [114]-[125].

This thesis addresses the problem of identifying the parameters of a 2D LSI ARMA (M, N) system, where M represents the order of the AR polynomial and N represents the order of the MA polynomial. In the proposed identification algorithm, where M and N can take arbitrary values different from each other, the sampled observations of the system input $x(k_1, k_2)$ and $y(k_1, k_2)$ are used, assuming that AR and MA orders

of the model are known. The general approach underlying the model examined here assumes that the system to be modeled is described by the following equation with input $x(k_1, k_2)$ and output $y(k_1, k_2)$, resulting with an ARMA system of order (M, N) .

$$y(k_1, k_2) = b_0 x(k_1, k_2) + \sum_{n=1}^N b_n x((k_1, k_2) - n) - \sum_{m=1}^M a_m y((k_1, k_2) - m) \quad (2.1)$$

The notation $x(k_1, k_2) - m$ or $y(k_1, k_2) - n$ denotes the m th or n th element behind $x(k_1, k_2)$ or $y(k_1, k_2)$ in the prediction mask and can be defined as follows.

$$x((k_1, k_2) - n) \triangleq x(k_1 - n_1, k_2 - n_2) \quad (2.2a)$$

$$y((k_1, k_2) - m) \triangleq y(k_1 - m_1, k_2 - m_2) \quad (2.2b)$$

The sample orderings for the pairs (n_1, n_2) and (m_1, m_2) , representing the shifts for $x(k_1, k_2)$ and $y(k_1, k_2)$ are illustrated in Figure 2.1

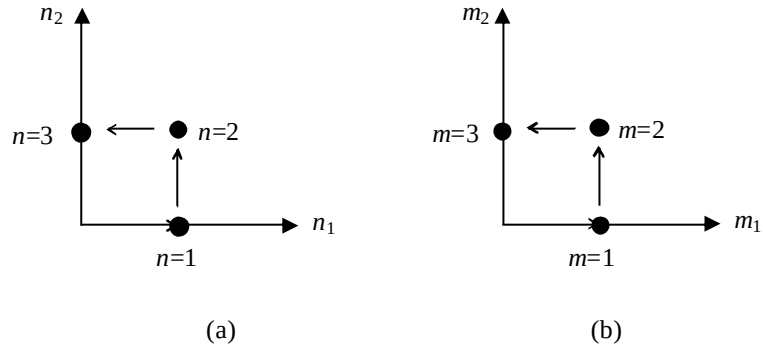


Figure 2-1: A sample ordering for (a) the system input $x(k_1, k_2)$ and (b) system output $y(k_1, k_2)$.

For an ordering of prediction support regions shown as in Figure 2.1 and using Equation (2.2a), we can elaborate on the given notation for $x(k_1, k_2)$ as follows.

$$n = 0 \Rightarrow n_1=0, n_2=0 \Rightarrow x((k_1, k_2) - 0) \triangleq x(k_1 - 0, k_2 - 0) = x(k_1, k_2)$$

$$n = 1 \Rightarrow n_1=1, n_2=0 \Rightarrow x((k_1, k_2) - 1) \triangleq x(k_1 - 1, k_2 - 0) = x(k_1 - 1, k_2)$$

$$n = 2 \Rightarrow n_1=1, n_2=1 \Rightarrow x((k_1, k_2) - 2) \triangleq x(k_1 - 1, k_2 - 1) = x(k_1 - 1, k_2 - 1)$$

$$n = 3 \Rightarrow n_1=0, n_2=1 \Rightarrow x((k_1, k_2) - 3) \triangleq x(k_1 - 0, k_2 - 1) = x(k_1, k_2 - 1)$$

In the same manner, using Figure 2.1 and Equation (2.2b), the given notation for $y(k_1, k_2)$ takes the following forms.

$$m = 0 \Rightarrow m_1=0, m_2=0 \Rightarrow y((k_1, k_2) - 0) \triangleq y(k_1 - 0, k_2 - 0) = y(k_1, k_2)$$

$$m = 1 \Rightarrow m_1=1, m_2=0 \Rightarrow y((k_1, k_2) - 1) \triangleq y(k_1 - 1, k_2 - 0) = y(k_1 - 1, k_2)$$

$$m = 2 \Rightarrow m_1=1, m_2=1 \Rightarrow y((k_1, k_2) - 2) \triangleq y(k_1 - 1, k_2 - 1) = y(k_1 - 1, k_2 - 1)$$

$$m = 3 \Rightarrow m_1=0, m_2=1 \Rightarrow y((k_1, k_2)-3) \triangleq y(k_1-0, k_2-1) = y(k_1, k_2-1)$$

In order to identify the ARMA(M, N) system parameters a_m , b_0 and b_n given in Equation (2.1), provided that $x(k_1, k_2)$ and $y(k_1, k_2)$ are given, we propose a “hybrid lattice” structure based on the two-channel AR lattice approach proposed by Kayran for equal AR and MA orders [189]. The novelties brought forward with this research can be listed as follows.

- We extend Kayran’s approach to the case where M and N can take arbitrary values different from each other. We accomplish this with the help of our proposed hybrid lattice structure where both 2-D two-channel AR and 2-D single-channel AR lattice stages are incorporated. We also propose a modification in terms of the channel inputs of the two-channel lattices. We drive the first channel input by a difference signal of $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$ instead of $y(k_1, k_2)$ which was formerly proposed. The point of using such a difference signal will be highlighted in the following paragraphs. The second channel input, which was formerly proposed as $x(k_1, k_2)$, is driven by $t(k_1, k_2)$, a signal which depends on the magnitude of orders of the AR and MA parts. If the order of the AR polynomial is greater than the order of the MA polynomial, $t(k_1, k_2)$ is equal to $x(k_1, k_2)$, but if the order of the MA polynomial is greater than the order of the AR polynomial, $t(k_1, k_2)$ is equal to $y(k_1, k_2)$.
- We propose modifications in the b_0 parameter estimates in accordance with our newly proposed channel inputs for $M \geq N$ and $M < N$.
- We derive a new formulation for the ARMA(M, N) parameter estimates, namely $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ vectors, taking into account the estimated parameter $\hat{b}_0$ and the forward prediction error filters’ tap weights related with both channels. We define $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ vectors as follows.

$$\hat{\mathbf{a}} = [\hat{a}_1 \quad \dots \quad \hat{a}_M] \quad (2.3a)$$

$$\hat{\mathbf{b}} = [\hat{b}_1 \quad \dots \quad \hat{b}_N] \quad (2.3b)$$

- We emphasize the fact that AR modeled system input parameters and ARMA(M, N) modeled system output parameters can be identified simultaneously, provided that the input is an AR model of order N , while the system output ARMA model order is (M, N) .

The initial channel inputs of our hybrid structure for identifying a 2-D ARMA(M,N) system of arbitrary AR and MA orders are illustrated in Figure 2.2.

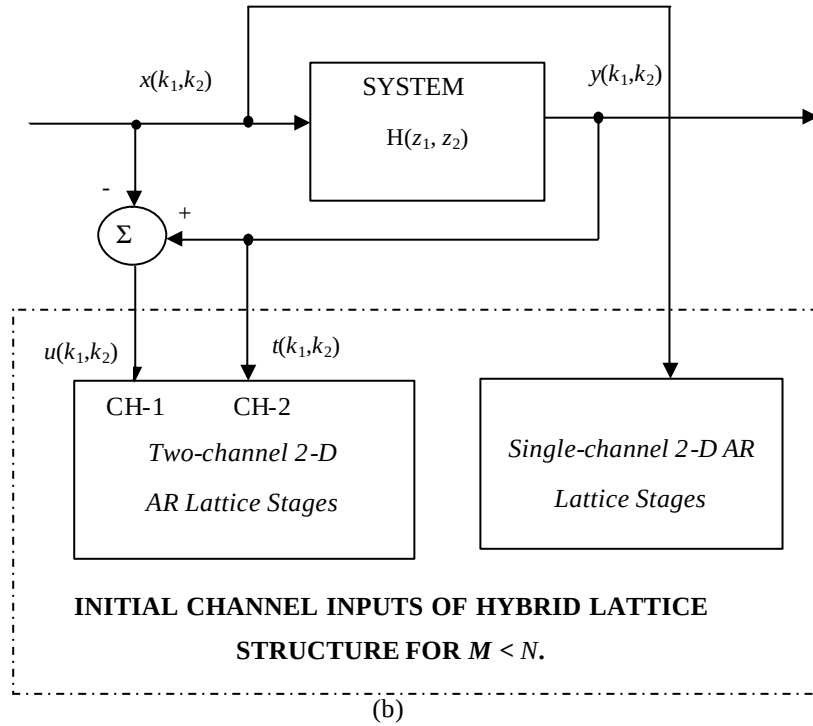
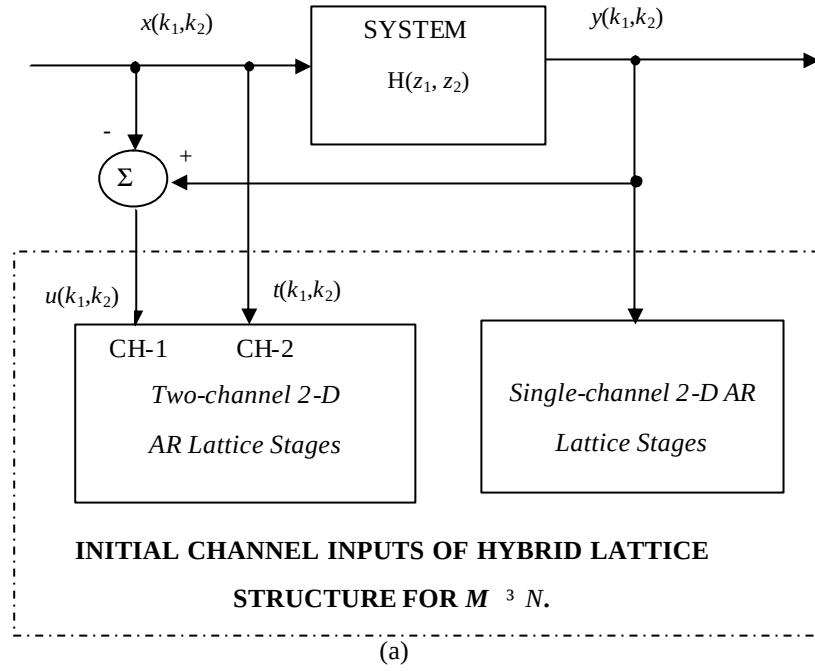


Figure 2-2: Initial channel inputs of the proposed hybrid 2-D lattice model to identify the parameters of an ARMA (M,N) system for (a) $M \geq N$ and (b) $M < N$.

As we have pointed out in the above paragraphs, in our procedure the first channel input of the first step is driven by a difference signal of $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$, instead of $y(k_1, k_2)$ only. This kind of an embedding approach was used by Strobach [56] in one-dimensional recursive covariance ladder algorithms for ARMA system identification. However, in the 2-D case, the use of such a difference signal brings with it the limitation that the data points of both the input and output signals in the prediction region should have the same ordering at least for the filter orders extending to S , which is the smaller of the (M, N) pair. By using such an embedding approach, we can make a more accurate analogy between the coefficients of the last stage forward prediction error filter and the ARMA(M, N) coefficients of the system to be identified. In order to prove this, we can proceed as follows.

The linear least-squares (LS) estimate of $y(k_1, k_2)$, denoted by $\hat{y}(k_1, k_2)$ is given by,

$$\hat{y}(k_1, k_2) = \sum_{n=0}^N \hat{b}_n x((k_1, k_2) - n) - \sum_{m=1}^M \hat{a}_m y((k_1, k_2) - m) \quad (2.4)$$

Now let us suppose we drive the first channel input with $y(k_1, k_2)$ and the second channel input with $x(k_1, k_2)$ and let us construct our two-channel 2-D AR equations accordingly. In this case the prediction error vector $\mathbf{f}_0^P(k_1, k_2)$ of any desired P th order takes the following form [119].

$$\mathbf{f}_0^P(k_1, k_2) = [y(k_1, k_2) - \hat{y}(k_1, k_2) \quad x(k_1, k_2) - \hat{x}(k_1, k_2)]^T \quad (2.5)$$

If we express this prediction error vector in terms of the ARMA(M, N) parameters we get the following.

$$\mathbf{f}_0^P(k_1, k_2) = \begin{bmatrix} y(k_1, k_2) - \sum_{n=1}^N \hat{b}_n x((k_1, k_2) - n) + \sum_{m=1}^M \hat{a}_m y((k_1, k_2) - m) \\ x(k_1, k_2) - \sum_{n=1}^N \hat{a}'_{x_n} x((k_1, k_2) - n) + \sum_{m=1}^M \hat{b}'_{x_m} y((k_1, k_2) - m) \end{bmatrix} \quad (2.6)$$

Here $\hat{a}'_{x_n}$ and $\hat{b}'_{x_m}$ are the model parameters of the system input $x(k_1, k_2)$. Now it can be said that the first row of the Equation (2.6), corresponds to the parameters of the 2-D ARMA system; but that can be misleading. If we compare the first row of Equation (2.6) with Equation (2.4), we see that $x(k_1, k_2)$ is not used to estimate $y(k_1, k_2)$. Therefore if we use a difference signal $u(k_1, k_2)$ defined above, we can overcome

this problem and improve the estimates. Hence, the forward prediction error vector can be rewritten as

$$\mathbf{f}_0^P(k_1, k_2) = \begin{bmatrix} y(k_1, k_2) - x(k_1, k_2) - \sum_{n=1}^N \hat{b}_n x((k_1, k_2) - n) + \sum_{m=1}^M \hat{a}_m y((k_1, k_2) - m) \\ x(k_1, k_2) - \sum_{n=1}^N \hat{a}'_{x_n} x((k_1, k_2) - n) + \sum_{m=1}^M \hat{b}'_{x_m} y((k_1, k_2) - m) \end{bmatrix} \quad (2.7)$$

As it can easily be seen from the first row of Equation (2.7), this time $x(k_1, k_2)$ is used to estimate $y(k_1, k_2)$ providing an exact correspondence to Equation (2.4). The second row of Equation (2.7) represents the estimation of $x(k_1, k_2)$, hence providing means to identify the system input and output parameters simultaneously.

3. HYBRID LATTICE STRUCTURE

The proposed hybrid lattice structure design has been explained briefly in Chapter 2. The 2-D AR lattice filters used in our procedure, both two-channel and single-channel, are based on the 2-D orthogonal lattice filters proposed by Kayran [119]. The very specific point in his design is that every point in the prediction mask is used in the prediction process; the number of coefficients is not fixed and the coefficients themselves are changed by the filter order update. It is actually this point-by-point processing that allows us to incorporate both two-channel and single channel lattice structures in a procedure that will result in a 2-channel 2-D ARMA(M,N) system identification framework.

3.1. Two-Dimensional Single Channel AR Lattice Structure

All 2-D lattice filter structures, whether they have quarter-plane support or half-plane support, consist of concatenated multi-input/multi-output stages defined in terms of reflection coefficients. The input and output fields are the forward and backward prediction error fields that are generated simultaneously. In order to calculate the lattice parameters at each stage, the prediction errors should be minimized in the mean-squared sense. The optimization can be performed in one of the fields (that is in the forward or in the backward prediction error fields) or in all the error fields simultaneously.

The original single-channel orthogonal 2-D lattice structure proposed by Kayran [119] for N th order AR modeling of random fields is given here for convenience.

$$\begin{bmatrix} f_{p-r}^{(r)}(k_1, k_2) \\ b_p^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_p}^{(r)} \\ k_{f_{p-n}}^{(r)} & 1 \end{bmatrix} \begin{bmatrix} f_{p-r}^{(r-1)}(k_1, k_2) \\ b_p^{(r-1)}(k_1, k_2) \end{bmatrix} \quad (3.1)$$

Here superscript r is the running order index, which starts from 1 and continues up to N , the final filter order. It is the current order of the prediction error filters, corresponding to the number of points used for the predictor. On the other hand, the

subscript p is the point-by-processing step mentioned above starting from r and continuing up to N . It denotes the number of the element to be predicted in the indexing scheme. $f_0^{(r)}(k_1, k_2)$ means the forward prediction error of $y((k_1, k_2)-0) = y(k_1, k_2)$ from previous r samples $y((k_1, k_2)-1), y((k_1, k_2)-2), \dots, y((k_1, k_2)-r)$. Similarly, $b_r^{(r)}(k_1, k_2)$ means the backward prediction of the sample $y((k_1, k_2)-r)$ from r samples prior to it, namely, $y((k_1, k_2)-r+1), \dots, y((k_1, k_2)-1), y(k_1, k_2)$.

$k_{b_p}^{(r)}$ and $k_{f_{p-r}}^{(r)}$ are the r th-order reflection coefficients of the forward and backward predictors and are shown to be calculated as follows [119]:

$$k_{b_p}^{(r)} = -\frac{\Delta_{f_{u_{p-r}} b_{u_p}}^{(r-1)}}{P_{b_{u_p}}^{(r-1)}} \quad (3.2a)$$

$$k_{f_{p-r}}^{(r)} = -\frac{\Delta_{f_{u_{p-r}} b_{u_p}}^{(r-1)}}{P_{f_{u_{p-r}}}^{(r-1)}} \quad (3.2b)$$

$P_{f_{u_{p-r}}}^{(r-1)}$ and $P_{b_{u_p}}^{(r-1)}$ are the forward and backward minimum mean-square errors, respectively are defined as below.

$$P_{f_{u_{p-r}}}^{(r-1)} \triangleq E \left[f_{u_{p-r}}^{(r-1)}(k_1, k_2) f_{u_{p-r}}^{(r-1)}(k_1, k_2) \right] \quad (3.3a)$$

$$P_{b_{u_p}}^{(r-1)} = E \left[b_{u_p}^{(r-1)}(k_1, k_2) b_{u_p}^{(r-1)}(k_1, k_2) \right] \quad (3.3b)$$

It can be shown that the scalars $\Delta_{f_0}^{(r-1)}$ and $\Delta_{b_r}^{(r-1)}$ used to obtain the reflection coefficients in (3.2) are equal to each other. On the other hand, it can be interpreted as a cross-correlation between the forward and backward prediction errors.

$$\Delta_{f_0 b_m}^{(r-1)} = \Delta_{f_0}^{(r-1)} = \Delta_{b_r}^{(r-1)} = E \left[f_0^{(r-1)}(k_1, k_2) b_r^{(r-1)}(k_1, k_2) \right] \quad (3.4)$$

Let the r -by-1 vectors $\mathbf{a}_0^{(r-1)}$ and $\mathbf{g}_r^{(r-1)}$ denote the tap weight vector of the corresponding forward and backward prediction error filters of order $(r-1)$, respectively. The tap weight vectors of 2-D forward and backward prediction error filters may be order updated as follows:

$$\mathbf{a}_0^{(r)} = \begin{bmatrix} \mathbf{a}_0^{(r-1)} \\ 0 \end{bmatrix} + k_{b_r}^{(r)} \begin{bmatrix} 0 \\ \mathbf{g}_r^{(r-1)} \end{bmatrix} \quad (3.5a)$$

and

$$\mathbf{g}_r^{(r)} = \begin{bmatrix} 0 \\ \mathbf{g}_r^{(r-1)} \end{bmatrix} + k_{f_0}^{(r)} \begin{bmatrix} \mathbf{a}_0^{(r-1)} \\ 0 \end{bmatrix} \quad (3.5b)$$

This method offers a complete solution for the Levinson-type algorithm to compute the prediction error filter coefficients using lattice parameter factors from the given 2-D augmented normal equation without requiring matrix inversion. Therefore, the proposed technique is exact 2-D equivalent of Burg's matrix formulation approach in the 1-D case. The superiority of this technique can be summarized as follows:

- 1) The determination of the stability of 2-D AR lattice model is made by using the 2-D lattice parameters computed at each stage, as in the 1-D case. Thus, complex stability testing procedures for large arrays are completely eliminated.
- 2) This procedure can be used for any shape and size of prediction support region covering quarter-plane and asymmetric half plane causal models.
- 3) The proposed 2-D lattice structure is amenable to systolic implementations. This is significant because the processing of 2-D data fields such as images in real time requires high data rates.
- 4) Since the unended forward and backward prediction errors form orthogonal bases, linear adaptive algorithms, such as least mean squares (LMS) and recursive least squares (RLS) can be applied to solve for 2-D system parameters.
- 5) The simplicity of the algorithm is less than all existing 2-D modeling techniques [111]-[112], [114]-[116], [130]-[132]. The only requirement is to select and ordering scheme with two types of shifts (vertical and horizontal) in the prediction support region. As a result of this, the first stages are 1-D lattice filters.

3.2. Two-Channel AR Lattice Structure for ARMA Modeling

Kayran proposed a 2-D ARMA(M,N) lattice filter modeling technique [189] based on the 2-D AR lattice structure explained in Chapter 3.1 using the two-channel AR approach for 1-D ARMA lattice synthesis proposed by Lim and Parker [54],[90]. He constructed the 2-D lattice analysis and synthesis models and adapted the estimator

of b_0 term, which was originally proposed by Perry [1], to his 2-D ARMA modeling technique.

Consider the transfer function of a 2D recursive system as a ratio of 2D polynomials in the following manner:

$$H(z_1, z_2) = \frac{b_0 + \sum_{(n_1, n_2) \in B} b_{n_1 n_2} z_1^{-n_1} z_2^{-n_2}}{1 + \sum_{(n_1, n_2) \in B} a_{n_1 n_2} z_1^{-n_1} z_2^{-n_2}} \quad (3.6)$$

where B is the region of support of the numerator and denominator polynomials composed of M data points and b_0 is the constant gain term for the model. The ordering of M data points in B can be made in various ways. If $y(k_1, k_2)$ and $x(k_1, k_2)$ are the output and input samples of the 2-D ARMA(M, N) system in Equation (3.6), atwo-channel linear forward prediction of estimating $y(k_1, k_2)$ and $x(k_1, k_2)$ from values of $y((k_1, k_2)-i)$ and $x((k_1, k_2)-i)$ for $1 \leq i \leq M$, and a two-channel linear backward prediction problem of estimating $y((k_1, k_2)-M)$ and $x((k_1, k_2)-M)$ using values of $y((k_1, k_2)-j)$ and $x((k_1, k_2)-j)$ for $M-1 > j \geq 0$ are formulated. The r th-stage prediction errors are related to the $(r-1)$ th stage prediction errors for $p=1, 2, \dots, M$ and $r=1, 2, \dots, p$ by the following relations.

$$\mathbf{f}_{p-r}^{(r)}(k_1, k_2) = \mathbf{f}_{p-r}^{(r-1)}(k_1, k_2) - \mathbf{K}_{p-r}^{(r)T} \mathbf{b}_p^{(r-1)}(k_1, k_2) \quad (3.7)$$

$$\mathbf{b}_p^{(r)}(k_1, k_2) = \mathbf{b}_p^{(r-1)}(k_1, k_2) - \bar{\mathbf{K}}_p^{(r)T} \mathbf{f}_{p-r}^{(r-1)}(k_1, k_2) \quad (3.8)$$

Here $\mathbf{f}_{p-r}^{(r)}(k_1, k_2)$ and $\mathbf{b}_p^{(r)}(k_1, k_2)$ denote the r th stage two-channel forward and backward prediction error vectors, respectively. The forward and backward prediction error vectors related with the zeroth stage are defined as follows.

$$\mathbf{f}_p^{(0)}(k_1, k_2) = \mathbf{b}_p^{(0)}(k_1, k_2) \triangleq \begin{bmatrix} y((k_1, k_2) - p) & x((k_1, k_2) - p) \end{bmatrix}^T \quad (3.9)$$

The analysis algorithm in Equations (3.7) and (3.8) starts from the zeroth order and continues up to the M th order as in the single channel case explained in the previous chapter. $\mathbf{K}_{p-r}^{(r)}$ and $\bar{\mathbf{K}}_p^{(r)}$ are the real-valued 2×2 matrices of the r th stage forward and backward reflection coefficients respectively and are defined as given in Equation (3.10). They are optimised to minimise the traces of the forward and backward prediction error covariances matrices.

$$\mathbf{K}_{p-r}^{(r)} \triangleq \begin{bmatrix} k_{p-r}^r(1,1) & k_{p-r}^r(1,2) \\ k_{p-r}^r(2,1) & k_{p-r}^r(2,2) \end{bmatrix} \quad (3.10a)$$

$$\bar{\mathbf{K}}_p^{(r)} \triangleq \begin{bmatrix} \bar{k}_p^r(1,1) & \bar{k}_p^r(1,2) \\ \bar{k}_p^r(2,1) & \bar{k}_p^r(2,2) \end{bmatrix} \quad (3.10b)$$

As it can be easily seen from Equation (3.6) that, in this proposed structure the numerator and denominator polynomials have the same region of support, B , meaning that only equal order AR and MA polynomials can be solved for. Therefore, in order to solve for arbitrary order AR and MA polynomials, we propose a novel hybrid structure in which both single channel 2-D lattice structures outlined in Chapter 3.2 and two-channel 2-D lattice structures explained briefly in this chapter are incorporated.

3.3. Hybrid Lattice Structure Design Steps

The proposed hybrid lattice structure incorporates the single channel and two-channel 2-D lattice structures explained in Chapter 3.1 and 3.2, respectively in an interleaved manner. In the whole lattice structure proposed, the running order index r , which determines the current filter order, starts from 1 and continues up to G , which is the greater of the (M, N) pair. We can split this whole structure into two steps as outlined in Chapter II namely, the first and the second steps. In the first step, the running order index r starts from 1 and continues up to S , which is the smaller of the (M, N) pair, while in the second step it starts from $(S + 1)$ and continues up to G . The parameter b_0 is not readily available from the calculated lattice parameters so it needs to be calculated separately after all the lattice stages have been realized.

It should be noted that in the special case when both AR and MA parts are of the same order, that is when $M = N$, the whole filter structure just reduces to two-channel lattice stages of the Step 1.1 with $r = 1 \dots M$ and $p = r \dots M$. Hence there is no hybrid structure involved and the second step is not applicable.

3.3.1. Step 1

The first step is composed of two sub-steps because of the point-by-point processing step p involved. For the whole first step design, the running order index r starts from

1 and continues up to S and the sample processing step p belonging to each running order index starts from r and continues up to G .

3.3.1.1. Step 1.1

The first sub-step is composed of 2-D two-channel AR lattice structures. In this sub-step the running order index r starts from 1 and continues up to S and the sample processing step p starts from r and continues up to S . The compact form equations defining this first sub-step are given below.

$$\begin{bmatrix} \mathbf{f}_{p-r}^{(r)}(k_1, k_2) \\ \mathbf{b}_p^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{K}_{p-r}^{(r)T} \\ \bar{\mathbf{K}}_p^{(r)T} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{f}_{p-r}^{(r-1)}(k_1, k_2) \\ \mathbf{b}_p^{(r-1)}(k_1, k_2) \end{bmatrix} \quad r = 1 \dots S, \quad p = r \dots S \quad (3.11)$$

Here, $\mathbf{f}_{p-r}^{(r)}(k_1, k_2)$ and $\mathbf{b}_p^{(r)}(k_1, k_2)$ denote the r th stage two-channel forward and backward prediction error vectors and $\mathbf{K}_{p-r}^{(r)}$, $\bar{\mathbf{K}}_p^{(r)}$ denote the real valued 2x2 matrices of the r th stage forward and backward reflection coefficients, respectively. The forward and backward prediction error vectors are defined as follows.

$$\mathbf{f}_{p-r}^{(r)}(k_1, k_2) \triangleq [f_{u_{p-r}}^{(r)}(k_1, k_2) \quad f_{t_{p-r}}^{(r)}(k_1, k_2)]^T \quad (3.12)$$

$$\mathbf{b}_p^{(r)}(k_1, k_2) \triangleq [b_{u_p}^{(r)}(k_1, k_2) \quad b_{t_p}^{(r)}(k_1, k_2)]^T \quad (3.13)$$

Here $f_{u_{p-r}}^{(r)}(k_1, k_2)$ and $f_{t_{p-r}}^{(r)}(k_1, k_2)$ are the forward prediction errors relating to the data fields $u(k_1, k_2)$ and $t(k_1, k_2)$. As defined in Chapter II, $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$ and $t(k_1, k_2)$ is $x(k_1, k_2)$ or $y(k_1, k_2)$ depending on the (M, N) pair. Hence, the following relation holds.

$$t(k_1, k_2) = \begin{cases} x(k_1, k_2) & M \geq N \\ y(k_1, k_2) & M < N \end{cases} \quad (3.14)$$

Initial order ($r = 0$) forward and backward prediction error vectors are defined as follows:

$$\mathbf{f}_p^{(0)}(k_1, k_2) = \mathbf{b}_p^{(0)}(k_1, k_2) \triangleq [u((k_1, k_2) - p) \quad t((k_1, k_2) - p)]^T \quad (3.15)$$

The equations defining the reflection coefficient matrices are given as follows. They are obtained by differentiating forward and backward error field vector equations in (3.8) and (3.9) with respect to $\mathbf{K}_{p-r}^{(r)}$ and $\bar{\mathbf{K}}_p^{(r)}$, respectively.

$$\mathbf{K}_{p-r}^{(r)} = -\bar{\mathbf{P}}_p^{(r-1)^{-1}} \mathbf{?}_p^{(r-1)^T} \quad (3.16a)$$

$$\bar{\mathbf{K}}_p^{(r)} = -\mathbf{P}_{p-r}^{(r-1)^{-1}} \mathbf{?}_p^{(r-1)} \quad (3.16b)$$

Here $\mathbf{P}_{p-r}^{(r-1)}$ and $\bar{\mathbf{P}}_p^{(r-1)}$ can be interpreted as 2x2 forward and backward prediction error power matrices, respectively and are defined as follows.

$$\mathbf{P}_{p-r}^{(r)} \triangleq E \left[\mathbf{f}_{p-r}^{(r)}(k_1, k_2) \mathbf{f}_{p-r}^{(r)T}(k_1, k_2) \right] \quad (3.17a)$$

$$\bar{\mathbf{P}}_p^{(r)} \triangleq E \left[\mathbf{b}_p^{(r)}(k_1, k_2) \mathbf{b}_p^{(r)T}(k_1, k_2) \right] \quad (3.17b)$$

$\mathbf{?}_p^{(r-1)}$ can be interpreted as the 2x2 cross-correlation matrix between forward and backward prediction error vectors and is defined as follows.

$$\mathbf{?}_p^{(r-1)} \triangleq E \left[\mathbf{f}_{p-r}^{(r-1)}(k_1, k_2) \mathbf{b}_p^{(r-1)T}(k_1, k_2) \right] \quad (3.18)$$

3.3.1.2. Step 1.2

The second sub-step is composed of single channel lattice structures. The running order index r starts from 1 and continues up to S and the sample processing step p starts from $(S + 1)$ and continues up to $(S + r)$ provided that $(S + r) \leq G$. Therefore the increment in p stops when $(S + r)$ is smaller or equal to G . The compact form equations defining this second sub-section are given below.

$$\begin{bmatrix} f_{u_{p-r}}^{(r)}(k_1, k_2) \\ b_{u_p}^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_p}^{(r)} \\ k_{f_{p-r}}^{(r)} & 1 \end{bmatrix} \begin{bmatrix} f_{u_{p-r}}^{(r-1)}(k_1, k_2) \\ b_{u_p}^{(r-1)}(k_1, k_2) \end{bmatrix} \quad p = (S + 1) \dots (S + r), (S + r) \leq G \quad (3.19)$$

Here in this step, it is important to note that $f_{u_{p-r}}^{(r)}(k_1, k_2)$ and $b_{u_p}^{(r)}(k_1, k_2)$ will not cover all the regions in the prediction support, as $u(k_1, k_2)$ is a difference signal of the system input and output. So it is important to form $f_{u_{p-r}}^{(r)}(k_1, k_2)$ and $b_{u_p}^{(r)}(k_1, k_2)$ values in the right way in the initial stage, that is for $r = 1$. As it can be seen from (3.19), $f_{u_{p-r}}^{(0)}(k_1, k_2)$ and $b_{u_p}^{(0)}(k_1, k_2)$ values are needed to compute $f_{u_{p-r}}^{(1)}(k_1, k_2)$ and $b_{u_p}^{(1)}(k_1, k_2)$, respectively.

As explained in the above paragraphs, $u(k_1, k_2)$ will be equal to the difference of $y(k_1, k_2)$ and $x(k_1, k_2)$ as long as the prediction supports of these signals are covered. For

the p values, which these supports are not covered, the related $y(k_1, k_2)$ or $x(k_1, k_2)$ values are taken to be zero. Therefore, initial order ($r = 0$) forward and backward prediction error vectors are defined as follows:

$$f_p^{(0)}(k_1, k_2) = b_p^{(0)}(k_1, k_2) \triangleq \begin{cases} y(k_1, k_2) & M \geq N \\ -x(k_1, k_2) & M < N \end{cases} \quad (3.20)$$

The reflection coefficients $k_{b_p}^{(r)}$ and $k_{f_{p-r}}^{(r)}$ in Equation (3.19) are the r th stage backward and forward reflection coefficients, respectively and are calculated as defined in Equation (3.2a) and (3.2b), respectively.

Here, like in the two-channel case $\Delta_{f_{u_{p-r}} b_{u_p}}^{(r-1)}$ can be interpreted as the cross-correlation between forward and backward prediction errors. $P_{b_{u_p}}^{(r-1)}$ and $P_{f_{u_{p-r}}}^{(r-1)}$ are the related backward and forward prediction error powers, respectively.

$$\Delta_{f_{u_{p-r}} b_{u_p}}^{(r-1)} \triangleq E[f_{u_{p-r}}^{(r-1)}(k_1, k_2) b_{u_p}^{(r-1)}(k_1, k_2)] \quad (3.18)$$

$$P_{b_{u_p}}^{(r-1)} \triangleq E[b_{u_p}^{(r-1)}(k_1, k_2) b_{u_p}^{(r-1)}(k_1, k_2)] \quad (3.19a)$$

$$P_{f_{u_{p-r}}}^{(r-1)} \triangleq E[f_{u_{p-r}}^{(r-1)}(k_1, k_2) f_{u_{p-r}}^{(r-1)}(k_1, k_2)] \quad (3.19b)$$

3.3.2. Step 2

The second step is composed of single channel lattice structures only. The running order index r starts from $(S+1)$ and continues up to G . The compact form equations defining this second step are given below.

$$\begin{bmatrix} f_{u_{p-r}}^{(r)}(k_1, k_2) \\ b_{u_p}^{(r)}(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_p}^{(r)} \\ k_{f_{p-r}}^{(r)} & 1 \end{bmatrix} \begin{bmatrix} f_{u_{p-r}}^{(r-1)}(k_1, k_2) \\ b_{u_p}^{(r-1)}(k_1, k_2) \end{bmatrix} \quad r = (S+1) \dots G, \quad p = r \dots G \quad (3.20)$$

3.4. Estimation of the Parameter b_0

In the above two-channel lattice approach, the b_0 parameter which is the coefficient of $x(k_1, k_2)$ in Equation (2.1), is not readily available from the calculated lattice parameters, since all the lattice stages contain autoregressive recursions. To overcome this difficulty, Kayran proposed the following method for the systems of

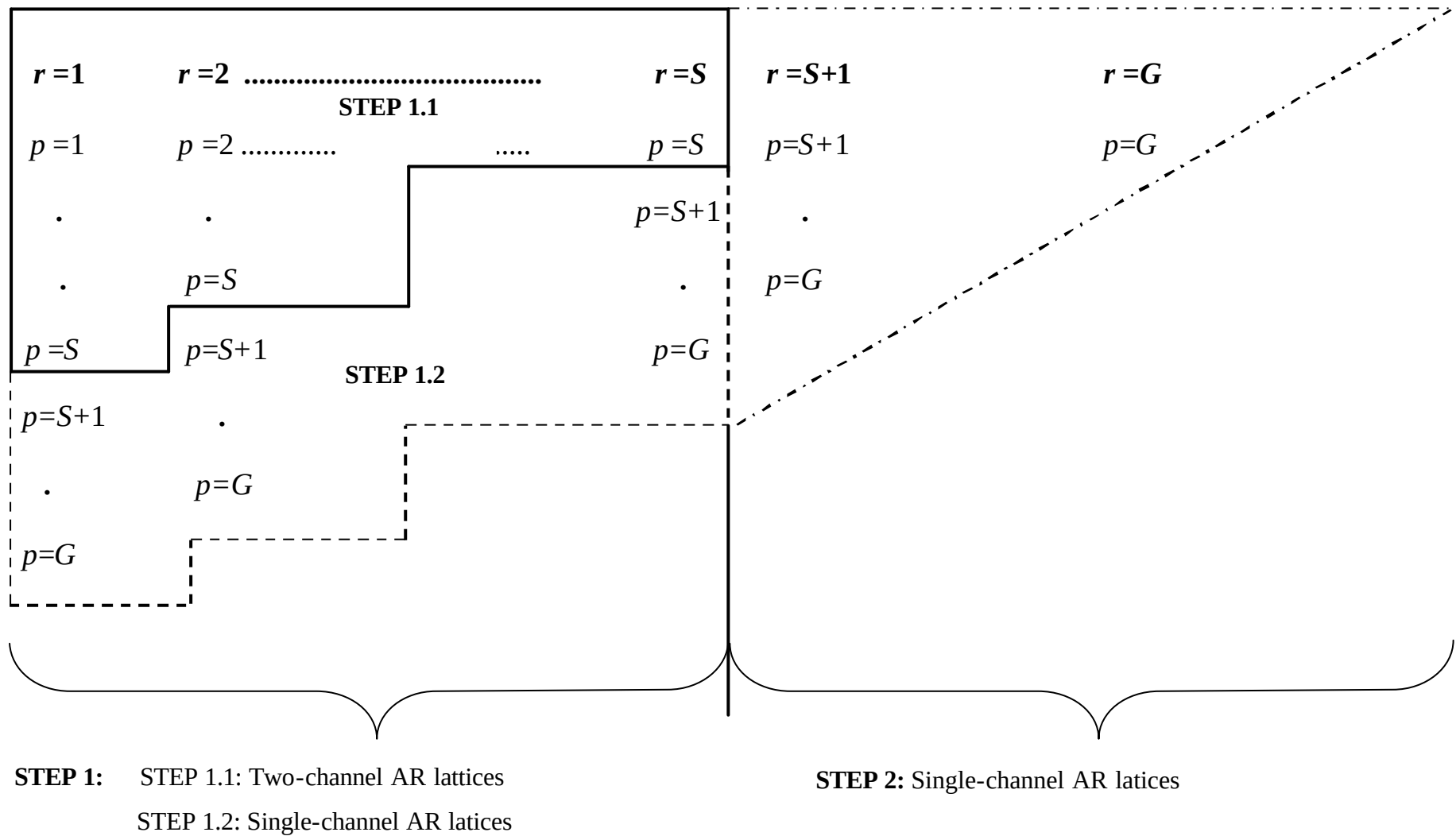


Figure 3-1: Construction of the 2-D ARMA (M, N) hybrid lattice structure with respect to the running order index r and sample processing step p .

equal AR and MA orders which was originally proposed by Perry [1] for 1-D systems.

$$\hat{b}_0 = \frac{E[f_{y_0}^M(k_1, k_2) f_{x_0}^M(k_1, k_2)]}{E[f_{x_0}^M(k_1, k_2)^2]} \quad (3.21)$$

As explained in the above paragraphs, we don't use the errors obtained from the data fields $x(k_1, k_2)$ and $y(k_1, k_2)$ directly. If we substitute, $f_{y_0}^M(k_1, k_2) = f_{u_0}^M(k_1, k_2) + f_{x_0}^M(k_1, k_2)$ in Equation (3.21), we obtain the following.

$$\hat{b}_0 = \frac{E[(f_{u_0}^M(k_1, k_2) + f_{x_0}^M(k_1, k_2))(f_{x_0}^M(k_1, k_2))]}{E[f_{x_0}^M(k_1, k_2)^2]} \quad (3.22)$$

$$\hat{b}_0 = 1 + \frac{E[f_{u_0}^M(k_1, k_2) f_{x_0}^M(k_1, k_2)]}{E[f_{x_0}^M(k_1, k_2)^2]} \text{ for } M \geq N \quad (3.23)$$

Therefore in order to estimate the parameter b_0 while solving for equal AR and MA orders or for AR order greater than MA order, the formula in Equation (3.23) should be used; $f_{u_0}^M(k_1, k_2)$ representing the forward prediction error related to the first channel and $f_{x_0}^M(k_1, k_2)$ representing the forward prediction error related to the second channel.

In order to solve for b_0 for the case when the AR order is smaller than the MA order, we substitute $f_{x_0}^M(k_1, k_2) = f_{y_0}^M(k_1, k_2) - f_{u_0}^M(k_1, k_2)$ in Equation (3.21) and we get the following.

$$\hat{b}_0 = \frac{E[(f_{y_0}^M(k_1, k_2)(f_{y_0}^M(k_1, k_2) - f_{u_0}^M(k_1, k_2)))]}{E[(f_{y_0}^M(k_1, k_2) - f_{u_0}^M(k_1, k_2))^2]} \text{ for } M < N \quad (3.24)$$

Here in equation (3.24), $f_{u_0}^M(k_1, k_2)$ represents the forward prediction error related to the first channel and $f_{y_0}^M(k_1, k_2)$ represents the forward prediction error related to the second channel.

4. A NEW METHOD TO CALCULATE THE 2-D ARMA (M, N) MODEL PARAMETERS

In this chapter, we propose a new method to calculate the ARMA (M,N) parameters once we have obtained the lattice reflection coefficients, the tap weights of the forward prediction error filters and the parameter b_0 which is the coefficient of $x(k_1, k_2)$ in the 2-D ARMA (M,N) model. We derive new equations for the cases when $M \geq N$ and $M < N$, the proofs of which are given in Appendices.

The ARMA (M,N) parameters we wish to calculate are defined by the following vectors, $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$, representing the AR and MA parameter estimates, respectively.

$$\hat{\mathbf{a}} = [\hat{a}_1 \quad \hat{a}_2 \quad \dots \quad \hat{a}_M]^T \quad (4.1a)$$

$$\hat{\mathbf{b}} = [\hat{b}_1 \quad \hat{b}_2 \quad \dots \quad \hat{b}_N]^T \quad (4.1b)$$

4.1. Calculation of the ARMA (M,N) Model Parameters: M=N

In order to calculate the forward and backward prediction error filter tap weights, we use the well-known two-channel Levinson-Durbin recursion given below.

$$\mathbf{a}_{p-r}^{(r)} = \begin{bmatrix} \mathbf{a}_{u_{p-r}}^{(r)} \\ \mathbf{a}_{x_{p-r}}^{(r)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{a}}_{u_{p-r}}^{(r)} \\ \hat{\mathbf{a}}_{x_{p-r}}^{(r)} \end{bmatrix} - \mathbf{K}_{p-r}^{(r)T} \begin{bmatrix} \hat{\mathbf{g}}_{u_p}^{(r)} \\ \hat{\mathbf{g}}_{x_p}^{(r)} \end{bmatrix} \quad (4.2a)$$

$$\mathbf{g}_p^{(r)} = \begin{bmatrix} \mathbf{g}_{u_p}^{(r)} \\ \mathbf{g}_{x_p}^{(r)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{g}}_{u_p}^{(r)} \\ \hat{\mathbf{g}}_{x_p}^{(r)} \end{bmatrix} - \bar{\mathbf{K}}_p^{(r)T} \begin{bmatrix} \hat{\mathbf{a}}_{u_{p-r}}^{(r)} \\ \hat{\mathbf{a}}_{x_{p-r}}^{(r)} \end{bmatrix} \quad (4.2b)$$

Here $\mathbf{a}_{p-r}^{(r)}$ and $\mathbf{g}_p^{(r)}$ denote the r th order tap weight coefficient vectors of the forward and backward prediction error filters related with the forward prediction of the $(p-r)$ th point and the backward prediction of the p th point, respectively.

Forward and backward prediction error filter tap weight coefficient vectors $\mathbf{a}_{p-r}^{(r)}$ and $\mathbf{g}_p^{(r)}$ are composed of forward and backward prediction error filter tap weights of the

first and second channels, which are driven by the signals $u(k_1, k_2)$ and $x(k_1, k_2)$, respectively.

In order to calculate the 2-D ARMA (M, N) parameters for the case when $M=N$, we propose that one should proceed as shown below.

- Set $r = 0$ and define the *zeroth* stage initial coefficient vectors as follows:

$$\mathbf{a}_{u_i}^{(0)} = \mathbf{g}_{u_i}^{(0)} = [1 \quad -1]^T \quad (4.3a)$$

$$\mathbf{a}_{x_i}^{(0)} = \mathbf{g}_{x_i}^{(0)} = [0 \quad 1]^T \quad (4.3b)$$

- Incrementing r by unity, form $\hat{\mathbf{a}}_{u_{p-r}}^{(r)}$, $\hat{\mathbf{a}}_{x_{p-r}}^{(r)}$ and $\hat{\mathbf{g}}_{u_p}^{(r)}$, $\hat{\mathbf{g}}_{x_p}^{(r)}$ from $\mathbf{a}_{u_{p-r}}^{(r-1)}$, $\mathbf{a}_{x_{p-r}}^{(r-1)}$ and $\mathbf{g}_{u_p}^{(r-1)}$, $\mathbf{g}_{x_p}^{(r-1)}$ using the augmented vectors for $r=1 \dots M$ and $p=r \dots M$.

$$\hat{\mathbf{a}}_{u_{p-r}}^{(r)} = \begin{bmatrix} \mathbf{a}_{u_{p-r}}^{(r-1)}(1) \\ \vdots \\ \mathbf{a}_{u_{p-r}}^{(r-1)}(r) \\ 0 \\ \mathbf{a}_{u_{p-r}}^{(r-1)}(r+1) \\ \vdots \\ \mathbf{a}_{u_{p-r}}^{(r-1)}(2r) \\ 0 \end{bmatrix} \quad \hat{\mathbf{a}}_{x_{p-r}}^{(r)} = \begin{bmatrix} \mathbf{a}_{x_{p-r}}^{(r-1)}(1) \\ \vdots \\ \mathbf{a}_{x_{p-r}}^{(r-1)}(r) \\ 0 \\ \mathbf{a}_{x_{p-r}}^{(r-1)}(r+1) \\ \vdots \\ \mathbf{a}_{x_{p-r}}^{(r-1)}(2r) \\ 0 \end{bmatrix} \quad (4.4a-b)$$

$$\hat{\mathbf{g}}_{u_p}^{(r)} = \begin{bmatrix} 0 \\ \mathbf{g}_{u_p}^{(r-1)}(1) \\ \vdots \\ \mathbf{g}_{u_p}^{(r-1)}(r) \\ 0 \\ \mathbf{g}_{u_p}^{(r-1)}(r+1) \\ \vdots \\ \mathbf{g}_{u_p}^{(r-1)}(2r) \end{bmatrix} \quad \hat{\mathbf{g}}_{x_p}^{(r)} = \begin{bmatrix} 0 \\ \mathbf{g}_{x_p}^{(r-1)}(1) \\ \vdots \\ \mathbf{g}_{x_p}^{(r-1)}(r) \\ 0 \\ \mathbf{g}_{x_p}^{(r-1)}(r+1) \\ \vdots \\ \mathbf{g}_{x_p}^{(r-1)}(2r) \end{bmatrix} \quad (4.5a-b)$$

- The tap-weight coefficient vectors of the 2-D forward and backward prediction error filters for $r=1 \dots M$ and $p=r \dots M$ may be order updated by using the two-channel Levinson-Durbin recursion given in Equation (4.2). For the last order, the following forward and backward tapweight coefficients are obtained.

$$\mathbf{a}_{u_0}^{(M)} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(1) = 1 \\ \mathbf{a}_{u_0}^{(M)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \\ \mathbf{a}_{u_0}^{(M)}(M+2) = -1 \\ \mathbf{a}_{u_0}^{(M)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+N+2) \end{bmatrix} \quad \mathbf{a}_{x_0}^{(N)} = \begin{bmatrix} \mathbf{a}_{x_0}^{(N)}(1) = 0 \\ \mathbf{a}_{x_0}^{(N)}(2) \\ \vdots \\ \mathbf{a}_{x_0}^{(N)}(M+1) \\ \mathbf{a}_{x_0}^{(N)}(M+2) = 1 \\ \mathbf{a}_{x_0}^{(N)}(M+3) \\ \vdots \\ \mathbf{a}_{x_0}^{(N)}(M+N+2) \end{bmatrix} \quad (4.6a-b)$$

$$\mathbf{g}_{u_M}^{(M)} = \begin{bmatrix} \mathbf{g}_{u_M}^{(M)}(1) \\ \vdots \\ \mathbf{g}_{u_M}^{(M)}(M) \\ \mathbf{g}_{u_M}^{(M)}(M+1) = 1 \\ \mathbf{g}_{u_M}^{(M)}(M+2) \\ \vdots \\ \mathbf{g}_{u_M}^{(M)}(M+N+1) \\ \mathbf{g}_{u_M}^{(M)}(M+N+2) = -1 \end{bmatrix} \quad \mathbf{g}_{x_N}^{(N)} = \begin{bmatrix} \mathbf{g}_{x_N}^{(N)}(1) \\ \vdots \\ \mathbf{g}_{x_N}^{(N)}(M) \\ \mathbf{g}_{x_N}^{(N)}(M+1) = 0 \\ \mathbf{g}_{x_N}^{(N)}(M+2) \\ \vdots \\ \mathbf{g}_{x_N}^{(N)}(M+N+1) \\ \mathbf{g}_{x_N}^{(N)}(M+N+2) = 1 \end{bmatrix} \quad (4.7a-b)$$

- ARMA (M,N) model coefficient vector $\begin{bmatrix} \hat{\mathbf{a}} & \hat{\mathbf{b}} \end{bmatrix}^T$, is calculated in terms of the final stage forward prediction error filter coefficients of the first channel $\mathbf{a}_{u_0}^M$ and the second channel $\mathbf{a}_{x_0}^N$, by using the following formulation, provided that the b_0 parameter has been estimated by using Equation (3.23).

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \end{bmatrix} + (1 - \hat{b}_0) \begin{bmatrix} \mathbf{a}_{x_0}^{(N)}(2) \\ \vdots \\ \mathbf{a}_{x_0}^{(N)}(M+1) \end{bmatrix} \quad \text{for } M=N \quad (4.8a)$$

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(2M+2) \end{bmatrix} + (1 - \hat{b}_0) \begin{bmatrix} \mathbf{a}_{x_0}^{(N)}(M+3) \\ \vdots \\ \mathbf{a}_{x_0}^{(N)}(2M+2) \end{bmatrix} \quad \text{for } M=N \quad (4.8b)$$

We give proof of Equation (4.8) in Appendix A.

4.1.1. Prediction of the System Input Parameters

If the system input $x(k_1, k_2)$ is different than a white noise process, for instance say an AR process, we can predict the AR model parameters of the input signal. This has

been shown in Chapter 2. The prediction gets quite accurate especially if $x(k_1, k_2)$ is an AR process of order N , which is the MA order of the ARMA (M, N) model.

4.2. Calculation of the ARMA (M, N) Model Parameters: $M > N$

In order to calculate the 2-D ARMA (M, N) parameters for the case when the AR order is greater than the MA order ($M > N$), we propose that one should proceed as shown below.

- Set $r = 0$ and define the initial stage forward and backward prediction error filter tap weight vectors as given in Equation (5.3).
- Incrementing r by unity, form $\hat{\mathbf{a}}_{u_{p-r}}^{(r)}$, $\hat{\mathbf{a}}_{x_{p-r}}^{(r)}$ and $\hat{\mathbf{g}}_{u_p}^{(r)}$, $\hat{\mathbf{g}}_{x_p}^{(r)}$ from $\mathbf{a}_{u_{p-r}}^{(r-1)}$, $\mathbf{a}_{x_{p-r}}^{(r-1)}$ and $\mathbf{g}_{u_p}^{(r-1)}$, $\mathbf{g}_{x_p}^{(r-1)}$ using the augmented vectors for $r = 1 \dots N$ and $p = r \dots (p + r) \leq N + 1$ as shown in Equations (4.4) and (4.5).
- The tap-weight coefficient vectors of the 2-D forward and backward prediction error filters for $r = 1 \dots N$ and $p = r \dots (p + r) \leq N + 1$ may be order updated by using the two-channel Levinson-Durbin recursion given in Equation (4.2).
- In order to order update the forward and backward prediction error filter coefficients for $r = (N+1) \dots M$ and $p = r \dots M$, we use the single channel Levinson-Durbin recursion given as follows:

$$\mathbf{a}_{u_{p-r}}^{(r)} = \hat{\mathbf{a}}_{u_{p-r}}^{(r)} - k_{b_p}^r \hat{\mathbf{g}}_{u_p}^{(r)} \quad (4.9a)$$

$$\mathbf{g}_{u_p}^{(r)} = \hat{\mathbf{g}}_{u_p}^{(r)} - k_{f_{p-r}}^r \hat{\mathbf{a}}_{u_{p-r}}^{(r)} \quad (4.9b)$$

- For the M th and N th orders, the following forward and backward tapweight coefficients are obtained.

$$\mathbf{a}_{u_0}^{(M)} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(1) = 1 \\ \mathbf{a}_{u_0}^{(M)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \\ \mathbf{a}_{u_0}^{(M)}(M+2) = -1 \\ \mathbf{a}_{u_0}^{(M)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+N+2) \end{bmatrix} \quad \mathbf{a}_{x_0}^{(N)} = \begin{bmatrix} \mathbf{a}_{x_0}^{(N)}(1) = 0 \\ \mathbf{a}_{x_0}^{(N)}(2) \\ \vdots \\ \mathbf{a}_{x_0}^{(N)}(N+1) \\ \mathbf{a}_{x_0}^{(N)}(N+2) = 1 \\ \mathbf{a}_{x_0}^{(N)}(N+3) \\ \vdots \\ \mathbf{a}_{x_0}^{(N)}(2N+2) \end{bmatrix} \quad (4.10a-b)$$

$$\mathbf{g}_{u_M}^{(M)} = \begin{bmatrix} \mathbf{g}_{u_M}^{(M)}(1) \\ \vdots \\ \mathbf{g}_{u_M}^{(M)}(M) \\ \mathbf{g}_{u_M}^{(M)}(M+1) = 1 \\ \mathbf{g}_{u_M}^{(M)}(M+2) \\ \vdots \\ \mathbf{g}_{u_M}^{(M)}(M+N+1) \\ \mathbf{g}_{u_M}^{(M)}(M+N+2) = -1 \end{bmatrix} \quad \mathbf{g}_{x_N}^{(N)} = \begin{bmatrix} \mathbf{g}_{x_N}^{(N)}(1) \\ \vdots \\ \mathbf{g}_{x_N}^{(N)}(N) \\ \mathbf{g}_{x_N}^{(N)}(N+1) = 0 \\ \mathbf{g}_{x_N}^{(N)}(N+2) \\ \vdots \\ \mathbf{g}_{x_N}^{(N)}(2N+1) \\ \mathbf{g}_{x_N}^{(N)}(2N+2) = 1 \end{bmatrix} \quad (4.11a-b)$$

- ARMA (M,N) model coefficient vector $[\hat{\mathbf{a}} \ \hat{\mathbf{b}}]^T$, is calculated in terms of $\mathbf{a}_{u_0}^{(M)}$ and $\mathbf{a}_{x_0}^{(N)}$ given in Equation (4.10) by using the following formulation, provided that the b_0 parameter has been estimated by using Equation (3.23).

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_N \\ \hat{a}_{N+1} \\ \vdots \\ \hat{a}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(2) + (1 - \hat{b}_0)\mathbf{a}_{x_0}^{(N)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(N+1) + (1 - \hat{b}_0)\mathbf{a}_{x_0}^{(N)}(N+1) \\ \mathbf{a}_{u_0}^{(M)}(N+2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \end{bmatrix} \text{ for } M > N \quad (4.12a)$$

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(M+3) + (1 - \hat{b}_0)\mathbf{a}_{x_0}^{(N)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+N+2) + (1 - \hat{b}_0)\mathbf{a}_{x_0}^{(N)}(M+N+2) \end{bmatrix} \text{ for } M > N \quad (4.12b)$$

We give proof of the Equation (4.12) in Appendix B.

4.3. Calculation of the ARMA (M,N) Model Parameters: $M < N$

In order to calculate the 2-D ARMA (M,N) parameters for the case when the AR order is smaller than the MA order ($M < N$), we propose that one should proceed as shown below.

- Set $r = 0$ and define the zeroth stage initial coefficient vectors as follows:

$$\mathbf{a}_{u_i}^{(0)} = \mathbf{g}_{u_i}^{(0)} = [1 \ -1]^T \quad (4.13a)$$

$$\mathbf{a}_{y_i}^{(0)} = \mathbf{g}_{y_i}^{(0)} = [1 \ 0]^T \quad (4.13b)$$

- Incrementing r by unity, form $\hat{\mathbf{a}}_{u_{p-r}}^{(r)}$, $\hat{\mathbf{a}}_{y_{p-r}}^{(r)}$ and $\hat{\mathbf{g}}_{u_p}^{(r)}$, $\hat{\mathbf{g}}_{y_p}^{(r)}$ from $\mathbf{a}_{u_{p-r}}^{(r-1)}$, $\mathbf{a}_{y_{p-r}}^{(r-1)}$ and $\mathbf{g}_{u_p}^{(r-1)}$, $\mathbf{g}_{y_p}^{(r-1)}$ using the augmented vectors for $r=1 \dots M$ and $p=r \dots (p+r) \leq M+1$.

$$\hat{\mathbf{a}}_{u_{p-r}}^{(r)} = \begin{bmatrix} \mathbf{a}_{u_{p-r}}^{(r-1)}(1) \\ \vdots \\ \mathbf{a}_{u_{p-r}}^{(r-1)}(r) \\ 0 \\ \mathbf{a}_{u_{p-r}}^{(r-1)}(r+1) \\ \vdots \\ \mathbf{a}_{u_{p-r}}^{(r-1)}(2r) \\ 0 \end{bmatrix} \quad \hat{\mathbf{a}}_{y_{p-r}}^{(r)} = \begin{bmatrix} \mathbf{a}_{y_{p-r}}^{(r-1)}(1) \\ \vdots \\ \mathbf{a}_{y_{p-r}}^{(r-1)}(r) \\ 0 \\ \mathbf{a}_{y_{p-r}}^{(r-1)}(r+1) \\ \vdots \\ \mathbf{a}_{y_{p-r}}^{(r-1)}(2r) \\ 0 \end{bmatrix} \quad (4.14a-b)$$

$$\hat{\mathbf{g}}_{u_p}^{(r)} = \begin{bmatrix} 0 \\ \mathbf{g}_{u_p}^{(r-1)}(1) \\ \vdots \\ \mathbf{g}_{u_p}^{(r-1)}(r) \\ 0 \\ \mathbf{g}_{u_p}^{(r-1)}(r+1) \\ \vdots \\ \mathbf{g}_{u_p}^{(r-1)}(2r) \end{bmatrix} \quad \hat{\mathbf{g}}_{y_p}^{(r)} = \begin{bmatrix} 0 \\ \mathbf{g}_{y_p}^{(r-1)}(1) \\ \vdots \\ \mathbf{g}_{y_p}^{(r-1)}(r) \\ 0 \\ \mathbf{g}_{y_p}^{(r-1)}(r+1) \\ \vdots \\ \mathbf{g}_{y_p}^{(r-1)}(2r) \end{bmatrix} \quad (4.15a-b)$$

- The tap-weight coefficient vectors of the 2-D forward and backward prediction error filters for $r=1 \dots M$ and $p=r \dots (p+r) \leq M+1$ may be order updated by using the two-channel Levinson-Durbin recursion given in Equation (4.2).
- To order update the forward and backward prediction error filter coefficients for $r=(M+1) \dots N$ and $p=r \dots N$, we use the single channel Levinson-Durbin recursion given in Equation (4.9).
- For the M th and N th orders, the following forward prediction error filter tapweight coefficient vectors are obtained.

$$\mathbf{a}_{u_0}^{(N)} = \begin{bmatrix} \mathbf{a}_{u_0}^{(N)}(1) = 1 \\ \mathbf{a}_{u_0}^{(N)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \\ \mathbf{a}_{u_0}^{(M)}(M+2) = -1 \\ \mathbf{a}_{u_0}^{(M)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+N+2) \end{bmatrix} \quad \mathbf{a}_{u_0}^{(M)} = \begin{bmatrix} \mathbf{a}_{u_0}^{(M)}(1) = 0 \\ \mathbf{a}_{u_0}^{(M)}(2) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(M+1) \\ \mathbf{a}_{u_0}^{(M)}(M+2) = 1 \\ \mathbf{a}_{u_0}^{(M)}(M+3) \\ \vdots \\ \mathbf{a}_{u_0}^{(M)}(2M+2) \end{bmatrix} \quad \mathbf{a}_{y_0}^{(M)} = \begin{bmatrix} \mathbf{a}_{y_0}^{(M)}(1) = 1 \\ \mathbf{a}_{y_0}^{(M)}(2) \\ \vdots \\ \mathbf{a}_{y_0}^{(M)}(M+1) \\ \mathbf{a}_{y_0}^{(M)}(M+2) = 0 \\ \mathbf{a}_{y_0}^{(M)}(M+3) \\ \vdots \\ \mathbf{a}_{y_0}^{(M)}(2M+2) \end{bmatrix} \quad (4.16a-b-c)$$

- ARMA (M,N) model coefficient vector $[\hat{\mathbf{a}} \ \hat{\mathbf{b}}]^T$, is calculated in terms of $\mathbf{a}_{u_0}^{(N)}$, $\mathbf{a}_{u_0}^{(M)}$ and $\mathbf{a}_{y_0}^{(M)}$ given in Equation (4.16), by using the following formulation, provided that the b_0 parameter has been estimated by using Equation (3.24).

$$\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{u_0}^{(N)}(2) + (1 - \hat{b}_0)(\mathbf{a}_{y_0}^{(M)}(2) - \mathbf{a}_{u_0}^{(M)}(2)) \\ \vdots \\ \mathbf{a}_{u_0}^{(N)}(M+1) + (1 - \hat{b}_0)(\mathbf{a}_{y_0}^{(M)}(M+1) - \mathbf{a}_{u_0}^{(M)}(M+1)) \end{bmatrix} \quad \text{for } M < N \quad (4.17a)$$

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_M \\ \hat{b}_{M+1} \\ \vdots \\ \hat{b}_N \end{bmatrix} = \begin{bmatrix} -\mathbf{a}_{u_0}^{(N)}(M+3) + (1 - \hat{b}_0)(\mathbf{a}_{u_0}^{(M)}(M+3) - \mathbf{a}_{y_0}^{(M)}(M+3)) \\ \vdots \\ -\mathbf{a}_{u_0}^{(N)}(2M+2) + (1 - \hat{b}_0)(\mathbf{a}_{u_0}^{(M)}(2M+2) - \mathbf{a}_{y_0}^{(M)}(2M+2)) \\ -\mathbf{a}_{u_0}^{(N)}(2M+3) \\ \vdots \\ -\mathbf{a}_{u_0}^{(N)}(M+N+2) \end{bmatrix} \quad \text{for } M < N \quad (4.17b)$$

We give proof of Equation (4.17) in Appendix C. The design steps in our proposed lattice structure are given in Algorithm 1. It should be noted that when $M=N$, there is no hybrid structure involved.

Step 1: If $M^1 N$, determine S and G .

$M > N \Rightarrow S=N$ and $G=M$.

$N > M \Rightarrow S=M$ and $G=N$.

Step 2: If $M^1 N$, determine the hybrid structure .

Two-channel lattice blocks: $r = 1, \dots, S$ and $p = r, \dots, S \quad \forall r$

Single channel lattice blocks: $r = 1, \dots, S$ and $p = S+1, \dots, G \quad \forall r$

$r = S+1, \dots, G$ and $p = r, \dots, G \quad \forall r$

Step 3: Initialization: $r = 0$

If $M^1 N$

For two-channel lattice blocks: $p = 1, \dots, S$ use Eq. (3.15)

For single-channel lattice blocks: $p = S+1, \dots, G$ use Eq. (3.20)

Else if $M=N$

For two-channel lattice blocks: $p = 1, \dots, M$ use Eq. (3.15)

Step 4: Start recursing on orders: $r = r + 1$. $p = r$.

If $M^1 N$

If $r > G$ go to Step 9.

Else if $M=N$

If $r > M$ go to Step 9.

Step 5: Calculate Reflection Coefficients

For two-channel lattice blocks, calculate $\mathbf{K}_{p-r}^{(r)}$ and $\bar{\mathbf{K}}_p^{(r)}$ using Equations (3.16a) and (3.16b), respectively.

For single channel lattice blocks, calculate $k_{f_{p-r}}^{(r)}$ and $k_{b_p}^{(r)}$ using Equations (3.2a) and (3.2b), respectively.

Step 6: Forward and Backward Prediction Errors

For two-channel lattice blocks, calculate $\mathbf{f}_{p-r}^{(r)}(k_1, k_2)$ and $\mathbf{b}_p^{(r)}(k_1, k_2)$ using Equations (3.7) and (3.8), respectively.

For single-channel lattice blocks, calculate $f_{u_{p-r}}^{(r)}(k_1, k_2)$ and $b_{u_p}^{(r)}(k_1, k_2)$ using Equation. (3.20).

Step 7: Calculate Tap-weight Coefficients

For two-channel lattice blocks, calculate $\mathbf{a}_{p-r}^{(r)}$ and $\mathbf{g}_p^{(r)}$ using Equations (4.2a) and (4.2b)

For single-channel lattice blocks, use Equations (4.9a) and (4.9b).

Step 8: Increment point-by-point processing step: $p = p + 1$.

If $p < G$

go to Step 5

else

go to Step 4.

Step 9: Calculate $\hat{b}_0$ using Equation (3.23) or (3.24).

Step 10: Obtain ARMA estimates using Equations (4.8), (4.12) or (4.17).

5. IMPLEMENTATION OF THE PROPOSED METHOD

In order to illustrate the functionality of the proposed 2-D two-channel hybrid lattice structure, we give here an example where AR order is greater than the MA order ($M=3$ and $N=2$). The problem we wish to solve is to simultaneously identify the AR and MA parameters of the unknown 2-D ARMA (3, 2) system, provided that the data fields $x(k_1, k_2)$ and $y(k_1, k_2)$ corresponding the system input and system output respectively, are given. The data field size for both the system input and system output is represented by $L_1 \times L_2$. The ordering of the data points in the prediction region is shown in Figure 5.1.

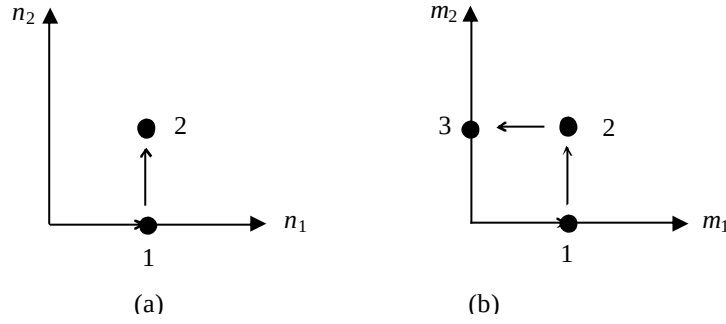


Figure 5-1: Ordering arrangements for (a) $x(k_1, k_2)$ and (b) $y(k_1, k_2)$.

We should design the lattice blocks in a structure where $M > N$, as illustrated in Figure 2.2(b). In terms of the design parameters defined in Chapter 3.2, the following relations hold: $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$, $t(k_1, k_2) = x(k_1, k_2)$, $S=2$ and $G=3$.

5.1. Implementation of Step 1

As explained in Chapter III, the first step is composed of two sub-steps because of the point-by-point processing step p involved. For the whole first step design, the running order index r starts from 1 and continues up to $S=2$ and the sample processing step p belonging to each running order index starts from r and continues up to $G=3$.

5.2. Implementation of Step 1.1: $r=1$

The first sub-step is composed of two-channel lattice structures. In this sub-step, the sample processing step p starts from r and continues up to $S=2$. Hence there are two sample processing steps involved, $p=1$ and $p=2$. The compact form equations defining this first sub-step are given below for these lattice stages.

$p=1$

$$\begin{bmatrix} f_{u_0}^1(k_1, k_2) \\ f_{x_0}^1(k_1, k_2) \end{bmatrix} = \begin{bmatrix} f_{u_0}^0(k_1, k_2) \\ f_{x_0}^0(k_1, k_2) \end{bmatrix} - \underbrace{\begin{bmatrix} k_0^1(1,1) & k_0^1(2,1) \\ k_0^1(1,2) & k_0^1(2,2) \end{bmatrix}}_{\mathbf{K}_0^1} \begin{bmatrix} b_{u_1}^0(k_1, k_2) \\ b_{x_1}^0(k_1, k_2) \end{bmatrix} \quad (5.1a)$$

$$\begin{bmatrix} b_{u_1}^1(k_1, k_2) \\ b_{x_1}^1(k_1, k_2) \end{bmatrix} = \begin{bmatrix} b_{u_1}^0(k_1, k_2) \\ b_{x_1}^0(k_1, k_2) \end{bmatrix} - \underbrace{\begin{bmatrix} \bar{k}_1^1(1,1) & \bar{k}_1^1(2,1) \\ \bar{k}_1^1(1,2) & \bar{k}_1^1(2,2) \end{bmatrix}}_{\bar{\mathbf{K}}_1^1} \begin{bmatrix} f_{u_0}^0(k_1, k_2) \\ f_{x_0}^0(k_1, k_2) \end{bmatrix} \quad (5.1b)$$

Now we need to calculate the forward and backward reflection coefficient matrices $\mathbf{K}_0^1$ and $\bar{\mathbf{K}}_1^1$, respectively using equations (3.11a) and (3.11b).

$$\mathbf{K}_0^1 = \bar{\mathbf{P}}_1^{0(-1)} \mathbf{?}_1^{0T} \quad (5.2a)$$

$$\bar{\mathbf{K}}_1^1 = \mathbf{P}_0^{0(-1)} \mathbf{?}_1^0 \quad (5.2b)$$

Forward and backward prediction error power matrices, $\mathbf{P}_0^0$, $\bar{\mathbf{P}}_1^0$ and $\mathbf{?}_1^0$ cross-correlation matrix are calculated using equations (3.12) and (3.13), respectively.

$$\bar{\mathbf{P}}_1^0 = E \begin{bmatrix} \mathbf{b}_1^0(k_1, k_2) & \mathbf{b}_1^{0T}(k_1, k_2) \end{bmatrix} = E \left[\begin{bmatrix} u(k_1-1, k_2) \\ x(k_1-1, k_2) \end{bmatrix} \begin{bmatrix} u(k_1-1, k_2) & x(k_1-1, k_2) \end{bmatrix} \right] \quad (5.3a)$$

$$\bar{\mathbf{P}}_1^0 = E \begin{bmatrix} u(k_1-1, k_2)u(k_1-1, k_2) & u(k_1-1, k_2)x(k_1-1, k_2) \\ x(k_1-1, k_2)u(k_1-1, k_2) & x(k_1-1, k_2)x(k_1-1, k_2) \end{bmatrix} \quad (5.3b)$$

The system input and system output fields, $x(k_1, k_2)$ and $y(k_1, k_2)$, respectively are of size $L_1 \times L_2$, as we have pointed out in the above paragraphs. Since we deal with the given data case, the expectation operations in Equation (5.3b) are replaced with arithmetic means. Hence we conclude with the following relation for $\bar{\mathbf{P}}_1^0$.

$$\bar{\mathbf{P}}_1^0 = \begin{bmatrix} \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1-1, k_2) u(k_1-1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1-1, k_2) x(k_1-1, k_2) \\ \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1-1, k_2) u(k_1-1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1-1, k_2) x(k_1-1, k_2) \end{bmatrix} \quad (5.4)$$

$$\mathbf{P}_0^0 = E \begin{bmatrix} \mathbf{f}_0^0(k_1, k_2) & \mathbf{f}_0^{0T}(k_1, k_2) \end{bmatrix} = E \begin{bmatrix} \begin{bmatrix} u(k_1, k_2) \\ x(k_1, k_2) \end{bmatrix} & \begin{bmatrix} u(k_1, k_2) & x(k_1, k_2) \end{bmatrix} \end{bmatrix} \quad (5.5a)$$

$$\mathbf{P}_0^0 = E \begin{bmatrix} u(k_1, k_2) u(k_1, k_2) & u(k_1, k_2) x(k_1, k_2) \\ x(k_1, k_2) u(k_1, k_2) & x(k_1, k_2) x(k_1, k_2) \end{bmatrix} \quad (5.5b)$$

For a data field size of $L_1 \times L_2$, the Equation (5.5b) takes the following form:

$$\mathbf{P}_0^0 = \begin{bmatrix} \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1, k_2) u(k_1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1, k_2) x(k_1, k_2) \\ \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1, k_2) u(k_1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1, k_2) x(k_1, k_2) \end{bmatrix} \quad (5.6)$$

$$\mathbf{P}_1^0 = E \begin{bmatrix} \mathbf{f}_0^0(k_1, k_2) & \mathbf{b}_1^{0T}(k_1, k_2) \end{bmatrix} = E \begin{bmatrix} \begin{bmatrix} u(k_1, k_2) \\ x(k_1, k_2) \end{bmatrix} & \begin{bmatrix} u(k_1-1, k_2) & x(k_1-1, k_2) \end{bmatrix} \end{bmatrix} \quad (5.7a)$$

$$\mathbf{P}_1^0 = E \begin{bmatrix} u(k_1, k_2) u(k_1-1, k_2) & u(k_1, k_2) x(k_1-1, k_2) \\ x(k_1, k_2) u(k_1-1, k_2) & x(k_1, k_2) x(k_1-1, k_2) \end{bmatrix} \quad (5.7b)$$

For a data size of $L_1 \times L_2$, the Equation (5.7b) takes the following form:

$$\mathbf{P}_1^0 = \begin{bmatrix} \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1, k_2) u(k_1-1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1, k_2) x(k_1-1, k_2) \\ \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1, k_2) u(k_1-1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1, k_2) x(k_1-1, k_2) \end{bmatrix} \quad (5.8)$$

Once the forward and backward reflection coefficient matrices have been calculated, we can write the compact form equations defining the forward and backward prediction error powers, as follows.

$$\begin{bmatrix} f_{u_0}^1(k_1, k_2) \\ f_{x_0}^1(k_1, k_2) \end{bmatrix} = \begin{bmatrix} u(k_1, k_2) \\ x(k_1, k_2) \end{bmatrix} - \begin{bmatrix} k_0^1(1,1) & k_0^1(2,1) \\ k_0^1(1,2) & k_0^1(2,2) \end{bmatrix} \begin{bmatrix} u(k_1-1, k_2) \\ x(k_1-1, k_2) \end{bmatrix} \quad (5.9a)$$

$$\begin{bmatrix} b_{u_1}^1(k_1, k_2) \\ b_{x_1}^1(k_1, k_2) \end{bmatrix} = \begin{bmatrix} u(k_1-1, k_2) \\ x(k_1-1, k_2) \end{bmatrix} - \begin{bmatrix} \bar{k}_1^1(1,1) & \bar{k}_1^1(2,1) \\ \bar{k}_1^1(1,2) & \bar{k}_1^1(2,2) \end{bmatrix} \begin{bmatrix} u(k_1, k_2) \\ x(k_1, k_2) \end{bmatrix} \quad (5.9b)$$

$$f_{u_0}^1(k_1, k_2) = u(k_1, k_2) - k_0^1(1,1)u(k_1-1, k_2) - k_0^1(2,1)x(k_1-1, k_2) \quad (5.10a)$$

$$f_{x_0}^1(k_1, k_2) = x(k_1, k_2) - k_0^1(1,2)u(k_1-1, k_2) - k_0^1(2,2)x(k_1-1, k_2) \quad (5.10b)$$

$$b_{u_1}^1(k_1, k_2) = u(k_1-1, k_2) - \bar{k}_1^1(1,1)u(k_1, k_2) - \bar{k}_1^1(2,1)x(k_1, k_2) \quad (5.10c)$$

$$b_{x_1}^1(k_1, k_2) = x(k_1-1, k_2) - \bar{k}_1^1(1,2)u(k_1, k_2) - \bar{k}_1^1(2,2)x(k_1, k_2) \quad (5.10d)$$

If we substitute $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$ and $u(k_1-1, k_2) = y(k_1-1, k_2) - x(k_1-1, k_2)$, we obtain the following equations.

$$f_{u_0}^1(k_1, k_2) = y(k_1, k_2) - k_0^1(1,1)y(k_1-1, k_2) - x(k_1, k_2) - [k_0^1(2,1) - k_0^1(1,1)]x(k_1-1, k_2) \quad (5.11a)$$

$$f_{x_0}^1(k_1, k_2) = -k_0^1(1,2)y(k_1-1, k_2) + x(k_1, k_2) - [k_0^1(2,2) - k_0^1(1,2)]x(k_1-1, k_2) \quad (5.11b)$$

$$b_{u_1}^1(k_1, k_2) = y(k_1-1, k_2) - \bar{k}_1^1(1,1)y(k_1, k_2) - x(k_1-1, k_2) - [\bar{k}_1^1(2,1) - \bar{k}_1^1(1,1)]x(k_1, k_2) \quad (5.11c)$$

$$b_{x_1}^1(k_1, k_2) = -\bar{k}_1^1(1,2)y(k_1, k_2) + x(k_1-1, k_2) - [\bar{k}_1^1(2,2) - \bar{k}_1^1(1,2)]x(k_1, k_2) \quad (5.11d)$$

To determine the tap-weight coefficients for the forward and backward prediction error filters, we use the equations (5.2a) and (5.2b).

$$\mathbf{a}_{u_0}^{(1)} = \begin{bmatrix} 1 \\ -k_0^1(1,1) \\ -1 \\ -[k_0^1(2,1) - k_0^1(1,1)] \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \begin{matrix} y(k_1, k_2) \\ y(k_1-1, k_2) \\ x(k_1, k_2) \\ x(k_1-1, k_2) \end{matrix} \begin{matrix} \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \end{matrix} \mathbf{a}_{x_0}^{(1)} = \begin{bmatrix} 0 \\ -k_0^1(1,2) \\ 1 \\ -[k_0^1(2,2) - k_0^1(1,2)] \end{bmatrix} \quad (5.12a-b)$$

$$\mathbf{g}_{u_1}^{(1)} = \begin{bmatrix} -\bar{k}_1^1(1,1) \\ 1 \\ -[\bar{k}_1^1(2,1) - \bar{k}_1^1(1,1)] \\ -1 \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \begin{matrix} y(k_1, k_2) \\ y(k_1-1, k_2) \\ x(k_1, k_2) \\ x(k_1-1, k_2) \end{matrix} \leftarrow \mathbf{g}_{x_1}^{(1)} = \begin{bmatrix} -\bar{k}_1^1(1,2) \\ 0 \\ -[\bar{k}_1^1(2,2) - \bar{k}_1^1(1,2)] \\ 1 \end{bmatrix} \quad (5.12c-d)$$

p=2

Here we are still in Step 1.1 with two-channel lattice structures. We conclude with the following equations going through the same path as above for $r=1$ and $p=1$.

$$\begin{bmatrix} f_{u_1}^1(k_1, k_2) \\ f_{x_1}^1(k_1, k_2) \end{bmatrix} = \begin{bmatrix} u(k_1-1, k_2) \\ x(k_1-1, k_2) \end{bmatrix} - \underbrace{\begin{bmatrix} k_1^1(1,1) & k_1^1(2,1) \\ k_1^1(1,2) & k_1^1(2,2) \end{bmatrix}}_{\mathbf{K}_1^1} \begin{bmatrix} u(k_1-1, k_2-1) \\ x(k_1-1, k_2-1) \end{bmatrix} \quad (5.13a)$$

$$\begin{bmatrix} b_{u_2}^1(k_1, k_2) \\ b_{x_2}^1(k_1, k_2) \end{bmatrix} = \begin{bmatrix} u(k_1-1, k_2-1) \\ x(k_1-1, k_2-1) \end{bmatrix} - \underbrace{\begin{bmatrix} \bar{k}_2^1(1,1) & \bar{k}_2^1(2,1) \\ \bar{k}_2^1(1,2) & \bar{k}_2^1(2,2) \end{bmatrix}}_{\bar{\mathbf{K}}_2^1} \begin{bmatrix} u(k_1-1, k_2) \\ x(k_1-1, k_2) \end{bmatrix} \quad (5.13b)$$

$$\mathbf{K}_1^1 = \bar{\mathbf{P}}_2^{0(-1)} \mathbf{?}_2^{0T} \quad (5.14a)$$

$$\mathbf{K}_1^1 = \bar{\mathbf{P}}_2^{0(-1)} \mathbf{?}_2^{0T} \quad (5.14b)$$

$$\bar{\mathbf{P}}_2^0 = E \left[\mathbf{b}_2^0(k_1, k_2) \mathbf{b}_2^{0T}(k_1, k_2) \right] = E \left[\begin{bmatrix} u(k_1-1, k_2-1) \\ x(k_1-1, k_2-1) \end{bmatrix} \begin{bmatrix} u(k_1-1, k_2-1) & x(k_1-1, k_2-1) \end{bmatrix} \right] \quad (5.15a)$$

$$\bar{\mathbf{P}}_2^0 = E \begin{bmatrix} u(k_1-1, k_2-1)u(k_1-1, k_2-1) & u(k_1-1, k_2-1)x(k_1-1, k_2-1) \\ x(k_1-1, k_2-1)u(k_1-1, k_2-1) & x(k_1-1, k_2-1)x(k_1-1, k_2-1) \end{bmatrix} \quad (5.15b)$$

For a data size of $L_1 \times L_2$, the Equation (5.15b) takes the following form:

$$\bar{\mathbf{P}}_2^0 = \begin{bmatrix} \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1-1, k_2-1)u(k_1-1, k_2-1) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1-1, k_2-1)x(k_1-1, k_2-1) \\ \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1-1, k_2-1)u(k_1-1, k_2-1) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1-1, k_2-1)x(k_1-1, k_2-1) \end{bmatrix} \quad (5.16)$$

$$\mathbf{P}_1^0 = E \left[\mathbf{f}_1^0(k_1, k_2) \mathbf{f}_1^{0T}(k_1, k_2) \right] = E \left[\begin{bmatrix} u(k_1-1, k_2) \\ x(k_1-1, k_2) \end{bmatrix} \begin{bmatrix} u(k_1-1, k_2) & x(k_1-1, k_2) \end{bmatrix} \right] \quad (5.17a)$$

$$\mathbf{P}_1^0 = \bar{\mathbf{P}}_1^0 \quad (5.17b)$$

$$\mathbf{P}_2^0 = E \left[\mathbf{f}_1^0(k_1, k_2) \mathbf{b}_2^{0T}(k_1, k_2) \right] = E \left[\begin{bmatrix} u(k_1-1, k_2) \\ x(k_1-1, k_2) \end{bmatrix} \begin{bmatrix} u(k_1-1, k_2-1) & x(k_1-1, k_2-1) \end{bmatrix} \right] \quad (5.18a)$$

$$\mathbf{P}_2^0 = E \begin{bmatrix} u(k_1-1, k_2)u(k_1-1, k_2-1) & u(k_1-1, k_2)x(k_1-1, k_2-1) \\ x(k_1-1, k_2)u(k_1-1, k_2-1) & x(k_1-1, k_2)x(k_1-1, k_2-1) \end{bmatrix} \quad (5.18b)$$

For a data field size of $L_1 \times L_2$, the Equation (5.18b) takes the following form:

$$\mathbf{P}_2^0 = \begin{bmatrix} \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1-1, k_2)u(k_1-1, k_2-1) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1-1, k_2)x(k_1-1, k_2-1) \\ \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1-1, k_2)u(k_1-1, k_2-1) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} x(k_1-1, k_2)x(k_1-1, k_2-1) \end{bmatrix} \quad (5.19)$$

$$f_{u_1}^1(k_1, k_2) = u(k_1-1, k_2) - k_1^1(1,1) \quad u(k_1-1, k_2-1) - k_1^1(2,1) \quad x(k_1-1, k_2-1) \quad (5.20a)$$

$$f_{x_1}^1(k_1, k_2) = x(k_1-1, k_2) - k_1^1(1,2) \quad u(k_1-1, k_2-1) - k_1^1(2,2) \quad x(k_1-1, k_2-1) \quad (5.20b)$$

$$b_{u_2}^1(k_1, k_2) = u(k_1-1, k_2-1) - \bar{k}_2^1(1,1) \quad u(k_1-1, k_2) - \bar{k}_2^1(2,1) \quad x(k_1-1, k_2) \quad (5.20c)$$

$$b_{x_2}^1(k_1, k_2) = x(k_1-1, k_2-1) - \bar{k}_2^1(1,2) \quad u(k_1-1, k_2) - \bar{k}_2^1(2,2) \quad x(k_1-1, k_2) \quad (5.20d)$$

If we substitute $u(k_1-1, k_2) = y(k_1-1, k_2) - x(k_1-1, k_2)$ and $u(k_1-1, k_2-1) = y(k_1-1, k_2-1) - x(k_1-1, k_2-1)$, we obtain the following equations.

$$f_{u_1}^1(k_1, k_2) = y(k_1-1, k_2) - k_1^1(1,1)y(k_1-1, k_2-1) - x(k_1-1, k_2) - [k_1^1(2,1) - k_1^1(1,1)]x(k_1-1, k_2-1) \quad (5.21a)$$

$$f_{x_1}^1(k_1, k_2) = -k_1^1(1,2) \quad y(k_1-1, k_2-1) + x(k_1-1, k_2) - [k_1^1(2,2) - k_1^1(1,2)] \quad x(k_1-1, k_2-1) \quad (5.21b)$$

$$b_{u_2}^1(k_1, k_2) = y(k_1-1, k_2-1) - \bar{k}_2^1(1,1)y(k_1-1, k_2) - x(k_1-1, k_2-1) - [\bar{k}_2^1(2,1) - \bar{k}_2^1(1,1)]x(k_1-1, k_2) \quad (5.21c)$$

$$b_{x_2}^1(k_1, k_2) = -\bar{k}_2^{-1}(1,2)y(k_1-1, k_2) + x(k_1-1, k_2-1) - [\bar{k}_2^{-1}(2,2) - \bar{k}_2^{-1}(1,2)]x(k_1-1, k_2) \quad (5.21d)$$

$$\mathbf{a}_{u_1}^{(1)} = \begin{bmatrix} 1 \\ -k_1^1(1,1) \\ -1 \\ -[k_1^1(2,1) - k_1^1(1,1)] \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \begin{matrix} y(k_1-1, k_2) \\ y(k_1-1, k_2-1) \\ x(k_1-1, k_2) \\ x(k_1-1, k_2-1) \end{matrix} \leftarrow \mathbf{a}_{x_1}^{(1)} = \begin{bmatrix} 0 \\ -k_1^1(1,2) \\ 1 \\ -[k_1^1(2,2) - k_1^1(1,2)] \end{bmatrix} \quad (5.22a-b)$$

$$\mathbf{g}_{u_2}^1 = \begin{bmatrix} -\bar{k}_2^{-1}(1,1) \\ 1 \\ -[\bar{k}_2^{-1}(2,1) - \bar{k}_2^{-1}(1,1)] \\ -1 \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \begin{matrix} y(k_1-1, k_2) \\ y(k_1-1, k_2-1) \\ x(k_1-1, k_2) \\ x(k_1-1, k_2-1) \end{matrix} \leftarrow \mathbf{g}_{x_2}^1 = \begin{bmatrix} -\bar{k}_2^{-1}(1,2) \\ 0 \\ -[\bar{k}_2^{-1}(2,2) - \bar{k}_2^{-1}(1,2)] \\ 1 \end{bmatrix} \quad (5.22c-d)$$

5.3. Implementation of Step 1.2: $r = 1$

This second sub-step is composed of single channel lattice structures as explained in Chapter III. The sample processing step p is $(S + 1) = 3$. Here we check the upper limit condition for p , which is $(S + r) \leq G$. As $(2+1) \leq 3$, this upper limit condition is readily satisfied. The compact form equations defining this second sub-step are given below.

$p=3$

$$\begin{bmatrix} f_{u_2}^1(k_1, k_2) \\ b_{u_3}^1(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_3}^1 \\ k_{f_2}^1 & 1 \end{bmatrix} \begin{bmatrix} f_{u_2}^0(k_1, k_2) \\ b_{u_3}^0(k_1, k_2) \end{bmatrix} \quad (5.23a)$$

$$\begin{bmatrix} f_{u_2}^1(k_1, k_2) \\ b_{u_3}^1(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_3}^1 \\ k_{f_2}^1 & 1 \end{bmatrix} \begin{bmatrix} u(k_1-1, k_2-1) \\ y(k_1, k_2-1) \end{bmatrix} \quad (5.23b)$$

It should be noted that, to write the prediction error fields here in this stage, in Equation (6.18b) we replace $f_{u_2}^0(k_1, k_2)$ with $u(k_1-1, k_2-1)$, since the prediction support allows to use such a difference signal.

$$f_{u_2}^1(k_1, k_2) = u(k_1-1, k_2-1) + k_{b_3}^1 y(k_1, k_2-1) \quad (5.24a)$$

$$b_{u_3}^1(k_1, k_2) = y(k_1, k_2-1) + k_{f_2}^1 u(k_1-1, k_2-1) \quad (5.24b)$$

$$k_{b_3}^1 = -\frac{\Delta_{f_{u_2} b_{u_3}}^0}{P_{b_{u_3}}^0} \quad (5.25a)$$

$$k_{f_2}^1 = -\frac{\Delta_{f_{u_2} b_{u_3}}^0}{P_{f_{u_2}}^0} \quad (5.25b)$$

$$\Delta_{f_{u_2} b_{u_3}}^0 = E[f_{u_2}^0(k_1, k_2) b_{u_3}^0(k_1, k_2)] = E[u(k_1 - 1, k_2 - 1) u(k_1, k_2 - 1)] \quad (5.26)$$

For a data size of $L_1 \times L_2$, the Equation (5.26) takes the following form:

$$\Delta_{f_{u_2} b_{u_3}}^0 = \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u(k_1 - 1, k_2 - 1) u(k_1, k_2 - 1) \quad (5.27)$$

$$P_{b_{u_3}}^0 = E[b_{u_3}^0(k_1, k_2) b_{u_3}^0(k_1, k_2)] = E[u^2(k_1, k_2 - 1)] \quad (5.28)$$

$$P_{b_{u_3}}^0 = \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u^2(k_1, k_2 - 1) \quad (5.29)$$

$$P_{f_{u_2}}^0 = E[f_{u_2}^0(k_1, k_2) f_{u_2}^0(k_1, k_2)] = E[u^2(k_1 - 1, k_2 - 1)] \quad (5.30)$$

$$P_{f_{u_2}}^0 = \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} u^2(k_1 - 1, k_2 - 1) \quad (5.31)$$

If we substitute $u(k_1 - 1, k_2 - 1) = y(k_1 - 1, k_2 - 1) - x(k_1 - 1, k_2 - 1)$, we obtain the following equations.

$$f_{u_2}^1(k_1, k_2) = y(k_1 - 1, k_2 - 1) - x(k_1 - 1, k_2 - 1) - k b_3^1 y(k_1, k_2 - 1) \quad (5.32a)$$

$$b_{u_3}^1(k_1, k_2) = y(k_1, k_2 - 1) - k f_2^1 y(k_1 - 1, k_2 - 1) - k f_2^1 x(k_1 - 1, k_2 - 1) \quad (5.32b)$$

$$\mathbf{a}_{u_2}^1 = \begin{bmatrix} 1 \\ -k b_3^1 \\ -1 \\ 0 \end{bmatrix} \rightarrow \begin{matrix} y(k_1 - 1, k_2 - 1) \\ y(k_1, k_2 - 1) \\ x(k_1 - 1, k_2 - 1) \\ x(k_1, k_2 - 1) \end{matrix} \leftarrow \mathbf{g}_{u_3}^1 = \begin{bmatrix} -k f_2^1 \\ 1 \\ k f_2^1 \\ 0 \end{bmatrix} \quad (5.33a-b)$$

5.4. Implementation of Step 1.1: $r=2$

$p=2$

$$\begin{bmatrix} f_{u_0}^2(k_1, k_2) \\ f_{x_0}^2(k_1, k_2) \end{bmatrix} = \begin{bmatrix} f_{u_0}^1(k_1, k_2) \\ f_{x_0}^1(k_1, k_2) \end{bmatrix} - \underbrace{\begin{bmatrix} k_0^2(1,1) & k_0^2(2,1) \\ k_0^2(1,2) & k_0^2(2,2) \end{bmatrix}}_{\mathbf{K}_0^2} \begin{bmatrix} b_{u_2}^1(k_1, k_2) \\ b_{x_2}^1(k_1, k_2) \end{bmatrix} \quad (5.34a)$$

$$\begin{bmatrix} b_{u_2}^2(k_1, k_2) \\ b_{x_2}^2(k_1, k_2) \end{bmatrix} = \begin{bmatrix} b_{u_2}^1(k_1, k_2) \\ b_{x_2}^1(k_1, k_2) \end{bmatrix} - \underbrace{\begin{bmatrix} \bar{k}_2^2(1,1) & \bar{k}_2^2(2,1) \\ \bar{k}_2^2(1,2) & \bar{k}_2^2(2,2) \end{bmatrix}}_{\bar{\mathbf{K}}_2^2} \begin{bmatrix} f_{u_0}^1(k_1, k_2) \\ f_{x_0}^1(k_1, k_2) \end{bmatrix} \quad (5.34b)$$

$$\mathbf{K}_0^2 = -\bar{\mathbf{P}}_2^{1(-1)} \mathbf{?}_2^{1T} \quad (5.35a)$$

$$\bar{\mathbf{K}}_2^1 = \mathbf{P}_0^{1(-1)} \mathbf{?}_2^1 \quad (5.35b)$$

$$\bar{\mathbf{P}}_2^1 = E \begin{bmatrix} \mathbf{b}_2^1(k_1, k_2) & \mathbf{b}_2^{1T}(k_1, k_2) \end{bmatrix} = E \left[\begin{bmatrix} b_{u_2}^1(k_1, k_2) \\ b_{x_2}^1(k_1, k_2) \end{bmatrix} \begin{bmatrix} b_{u_2}^1(k_1, k_2) & b_{x_2}^1(k_1, k_2) \end{bmatrix} \right] \quad (5.36)$$

For a data size of $L_1 \times L_2$, the Equation (5.36) takes the following form:

$$\bar{\mathbf{P}}_2^1 = \begin{bmatrix} \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} b_{u_2}^1(k_1, k_2) b_{u_2}^1(k_1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} b_{u_2}^1(k_1, k_2) b_{x_2}^1(k_1, k_2) \\ \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} b_{x_2}^1(k_1, k_2) b_{u_2}^1(k_1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} b_{x_2}^1(k_1, k_2) b_{x_2}^1(k_1, k_2) \end{bmatrix} \quad (5.37)$$

$$\mathbf{P}_0^1 = E \begin{bmatrix} \mathbf{f}_0^1(k_1, k_2) & \mathbf{f}_0^{1T}(k_1, k_2) \end{bmatrix} = E \left[\begin{bmatrix} f_{u_0}^1(k_1, k_2) \\ f_{x_0}^1(k_1, k_2) \end{bmatrix} \begin{bmatrix} f_{u_0}^1(k_1, k_2) & f_{x_0}^1(k_1, k_2) \end{bmatrix} \right] \quad (5.38)$$

For a data size of $L_1 \times L_2$, the Equation (5.38) takes the following form:

$$\mathbf{P}_0^1 = \begin{bmatrix} \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{u_0}^1(k_1, k_2) f_{u_0}^1(k_1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{u_0}^1(k_1, k_2) f_{x_0}^1(k_1, k_2) \\ \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{x_0}^1(k_1, k_2) f_{u_0}^1(k_1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{x_0}^1(k_1, k_2) f_{x_0}^1(k_1, k_2) \end{bmatrix} \quad (5.39)$$

$$\mathbf{?}_2^1 = E \begin{bmatrix} \mathbf{f}_0^1(k_1, k_2) & \mathbf{b}_2^{1T}(k_1, k_2) \end{bmatrix} = E \left[\begin{bmatrix} f_{u_0}^1(k_1, k_2) \\ f_{x_0}^1(k_1, k_2) \end{bmatrix} \begin{bmatrix} b_{u_2}^1(k_1, k_2) & b_{x_2}^1(k_1, k_2) \end{bmatrix} \right] \quad (5.40a)$$

$$\mathbf{?}_2^1 = E \begin{bmatrix} f_{u_0}^1(k_1, k_2) b_{u_2}^1(k_1, k_2) & f_{u_0}^1(k_1, k_2) b_{x_2}^1(k_1, k_2) \\ f_{x_0}^1(k_1, k_2) b_{u_2}^1(k_1, k_2) & f_{x_0}^1(k_1, k_2) b_{x_2}^1(k_1, k_2) \end{bmatrix} \quad (5.40b)$$

For a data size of $L_1 \times L_2$, the Equation (5.40b) takes the following form:

$$\mathbf{?}_2^1 = \begin{bmatrix} \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{u_0}^1(k_1, k_2) b_{u_2}^1(k_1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{u_0}^1(k_1, k_2) b_{x_2}^1(k_1, k_2) \\ \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{x_0}^1(k_1, k_2) b_{u_2}^1(k_1, k_2) & \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{x_0}^1(k_1, k_2) b_{x_2}^1(k_1, k_2) \end{bmatrix} \quad (5.41)$$

All the entries of matrix $\mathbf{?}_2^1$ are available from the step $r=1$; $p=1$ and $p=2$; therefore $\mathbf{?}_2^1$ can easily be computed.

5.5. Implementation of Step 1.2: $r=2$

$p=3$

$$\begin{bmatrix} f_{u_1}^2(k_1, k_2) \\ b_{u_3}^2(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_3}^2 \\ k_{f_1}^2 & 1 \end{bmatrix} \begin{bmatrix} f_{u_1}^1(k_1, k_2) \\ b_{u_3}^1(k_1, k_2) \end{bmatrix} \quad (5.42)$$

$$k_{b_3}^2 = -\frac{\Delta_{f_{u_1} b_{u_3}}^1}{P_{b_{u_3}}^1} \quad (5.43a)$$

$$k_{f_1}^2 = -\frac{\Delta_{f_{u_1} b_{u_3}}^1}{P_{f_{u_1}}^1} \quad (5.43b)$$

$$\Delta_{f_{u_1} b_{u_3}}^1 = E[f_{u_1}^1(k_1, k_2) b_{u_3}^1(k_1, k_2)] \quad (5.44)$$

For a data size of $L_1 \times L_2$, the Equation (5.44) takes the following form:

$$\Delta_{f_{u_1} b_{u_3}}^1 = \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{u_1}^1(k_1, k_2) b_{u_3}^1(k_1, k_2) \quad (5.45)$$

$$P_{b_{u_3}}^1 = E[b_{u_3}^1(k_1, k_2) b_{u_3}^1(k_1, k_2)] \quad (5.46)$$

$$P_{b_{u_3}}^1 = \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} b_{u_3}^1(k_1, k_2) b_{u_3}^1(k_1, k_2) \quad (5.47)$$

$$P_{f_{u_1}}^1 = E[f_{u_1}^1(k_1, k_2) f_{u_1}^1(k_1, k_2)] \quad (5.48)$$

$$P_{f_{u_1}}^1 = \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{u_1}^1(k_1, k_2) f_{u_1}^1(k_1, k_2) \quad (5.49)$$

5.6. Implementation of Step 2: $r=3$

As explained in the above paragraphs, the second step is composed of single channel lattice structures only. The running order index r starts from $(S+1)=3$ and continues up to $G=3$. The compact form equations defining this second step are given below.

$p=3$

$$\begin{bmatrix} f_{u_0}^3(k_1, k_2) \\ b_{u_3}^3(k_1, k_2) \end{bmatrix} = \begin{bmatrix} 1 & k_{b_3}^3 \\ k_{f_0}^3 & 1 \end{bmatrix} \begin{bmatrix} f_{u_0}^2(k_1, k_2) \\ b_{u_3}^2(k_1, k_2) \end{bmatrix} \quad (5.50)$$

$$k_{b_3}^3 = -\frac{\Delta_{f_{u_0} b_{u_3}}^2}{P_{b_{u_3}}^2} \quad (5.51a)$$

$$k_{f_0}^3 = -\frac{\Delta_{f_{u_0} b_{u_3}}^2}{P_{f_{u_0}}^2} \quad (5.51b)$$

$$\Delta_{f_{u_0} b_{u_3}}^2 = E[f_{u_0}^2(k_1, k_2) b_{u_3}^2(k_1, k_2)] \quad (5.52)$$

For a data size of $L_1 \times L_2$, the Equation (5.52) takes the following form:

$$\Delta_{f_{u_0} b_{u_3}}^2 = \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{u_0}^2(k_1, k_2) b_{u_3}^2(k_1, k_2) \quad (5.53)$$

$$P_{b_{u_3}}^2 = E[b_{u_3}^2(k_1, k_2) b_{u_3}^2(k_1, k_2)] \quad (5.53)$$

$$P_{b_{u_3}}^2 = \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} b_{u_3}^2(k_1, k_2) b_{u_3}^2(k_1, k_2) \quad (5.54)$$

$$P_{f_{u_0}}^2 = E[f_{u_0}^2(k_1, k_2) f_{u_0}^2(k_1, k_2)] \quad (5.55)$$

$$P_{f_{u_0}}^2 = \frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{u_0}^2(k_1, k_2) f_{u_0}^2(k_1, k_2) \quad (5.56)$$

Here, the solution for $M=3$ and $N=2$ has been completed. In Figure 5.3, we illustrate the inter-block connection for this specific example in the framework of our proposed hybrid lattice structure.

5.7. The calculation of parameter b_0

The parameter b_0 is calculated using the formulation given in Equation (3.23) by replacing the expectation operations with arithmetic mean. Hence the following relation is obtained.

$$\hat{b}_0 = 1 + \frac{\frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{u_0}^M(k_1, k_2) f_{x_0}^M(k_1, k_2)}{\frac{1}{L_1 L_2} \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} f_{x_0}^M(k_1, k_2)^2} \quad (5.57)$$

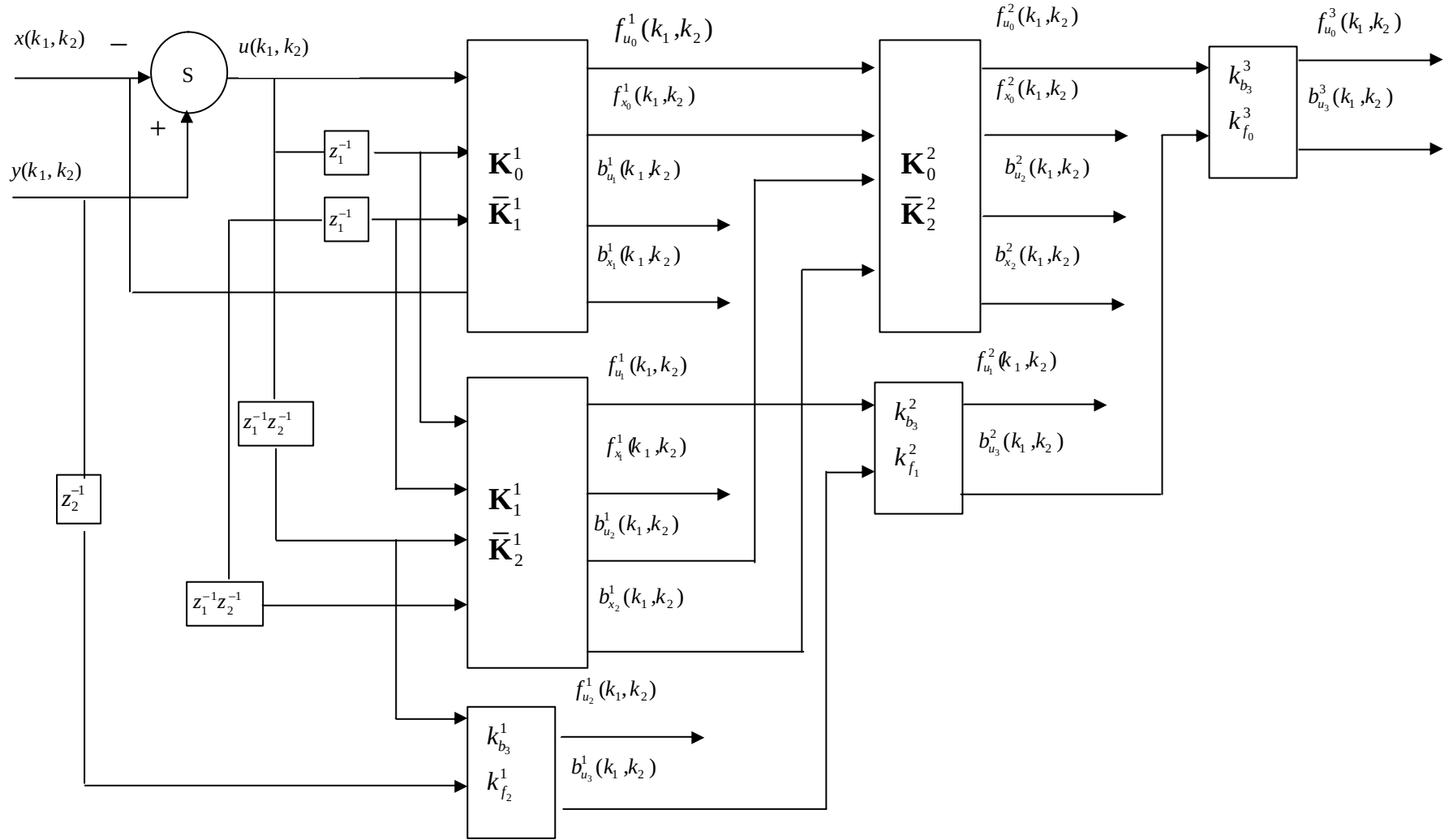


Figure 5-2: The block interconnection for the hybrid lattice analysis model of an ARMA (3, 2) system.

6. SIMULTANEOUS IDENTIFICATION OF IMAGE AND BLUR PARAMETERS

In many two-dimensional (2-D) signal processing applications, such as seismic signal processing or image processing, a zero-phase finite impulse response (FIR) linear system is used to model the distortion incurred in recording the signal. In most of these applications, it is not possible to have enough a priori information to determine the exact system impulse response. Therefore, the unknown system impulse response has to be identified from the degraded signal. One such application is the modeling of blurred images, where the blurred image is represented by the convolution of an ideal image with a symmetric, noncausal, finite-extent impulse response also known as the point spread function (PSF). Some examples of image blurs that can be modeled with a zero-phase PSF include linear motion blur and out-of-focus blur. Tekalp *et al.* [165] showed that the identification can be considered as a first step in image restoration, where one wishes to estimate the original image from the noisy and distorted observation. They also proved that noisy and blurred images can be represented by an ARMA(M,N) model where AR parameters represent the original image and MA parameters represent the blur.

This chapter is focused on the applicability of our proposed hybrid lattice structure to the simultaneous identification of the image and blur parameters by utilizing the concepts and derivations given in [165].

6.1. Image and Blur Models

Here we reproduce for convenience, the image and blur model equations and ARMA model derivations given in [165]. $N \times N$ discrete monochromatic images digitized with a conventional raster-type scan are considered. A pixel based random field model with asymmetric half-plane (ASHP) causality is considered for the image [197]. The degradation suffered during the image formation process can be modeled with a convolution summation having a noncausal support and additive white noise [198].

The noisy and blurred image pixel densities will thus be described by the following state-space model [199].

$$s(m, n) = \sum_{(k, l) \in S_a} a_{kl} s(m-k, n-l) + w(m, n) \quad (6.1)$$

$$r(m, n) = \sum_{(i, j) \in S_b} b_{ij} s(m-i, n-j) + v(m, n) \quad (6.2)$$

Here $s(m, n)$ is the undistorted image density at (m, n) , $r(m, n)$ is the observation degraded by space-invariant blur and additive noise, and $w(m, n)$ and $v(m, n)$ are independent, zero-mean, white Gaussian noise fields, with variances s_w^2 and s_v^2 , respectively.

The original image is described by Equation (6.1) with an (MXM) th-order AR model. The $\{a_{kl}\}$ are the model coefficients and S_a is the ASHP model support defined as:

$$S_a = \{(m-k, n-l) \mid (1 \leq k < M, 0 \leq l < M) \cup (-M \leq k \leq 0, 1 \leq l \leq M)\} \quad (6.3)$$

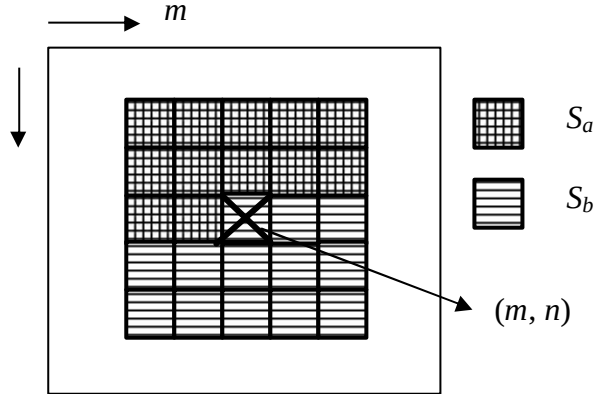


Figure 6.1: ASHP image model support S_a and noncausal PSF support S_b .

The blurring PSF is given by $\{b_{ij}\}$ and has noncausal support S_b . Both supports are shown in Figure 6.1. The PSF modeled in this manner is a FIR filter. The block diagram for this representation is given in Figure 6.2. In addition, the following equation should be satisfied in order to preserve the mean value of the blurred image.

$$\sum_{(i, j) \in R} b_{ij} = 1 \quad (6.4)$$

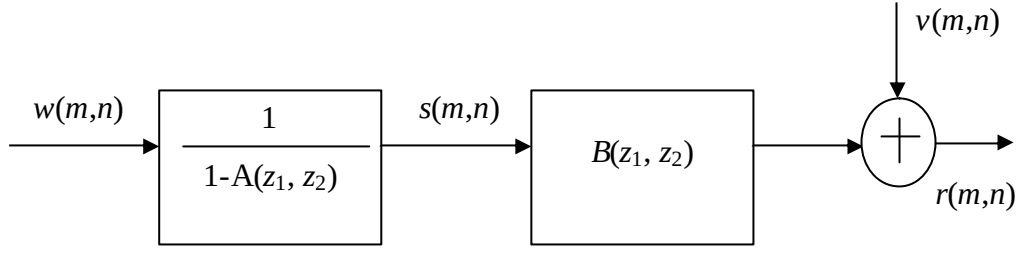


Figure 6. 2: Block diagram representation of the input-output model of the degraded images.

When the Equations (6.1) and (6.2) are combined, the summations are interchanged and the resulting equation is substituted in (6.2), the following input-output relation is obtained [165].

$$r(m, n) = \sum_{(k, l) \in S_a} a_{kl} r(m-k, n-l) + \sum_{(i, j) \in S_b} b_{ij} w(m-i, n-j) + v(m, n) - \sum_{(k, l) \in S_a} a_{kl} v(m-k, n-l) \quad (6.5)$$

If a new signal is defined as $x(m, n) = r(m, n) - v(m, n)$, the above equation takes the following form [165].

$$x(m, n) = \sum_{(k, l) \in S_a} a_{kl} x(m-k, n-l) + \sum_{(i, j) \in S_b} b_{ij} w(m-i, n-j) \quad (6.6)$$

It can easily be seen that Equation (6.6) represents an ARMA model with $x(m, n)$ as the system output and $w(m, n)$ as the system input. AR parameters represent the coefficients related with the original image and MA parameters represent the coefficients related with blur.

6.2. Hybrid Lattice Design for Simultaneous Identification of Image and Blur Parameters

As it can easily be deduced from Chapter 6.1 that, the problem of image and blur parameter identification turns out to be a problem of 2-D ARMA identification. Therefore, we propose to use our hybrid lattice algorithm for this 2-D identification problem [192]. In order to solve this problem, we assume that original image is defined in ASHP with four prediction points. The blur PSF is chosen to have a 3x3 noncausal ASHP support defined in all planes with four prediction points. Therefore we need to define a hybrid lattice structure where MA order is greater than the MA order.

As we have explained in Chapter 3, the first step in hybrid lattice design is the determination of the initial channel inputs, that is, $\mathbf{f}_0^{(0)}(k_1, k_2) = \mathbf{b}_0^{(0)}(k_1, k_2)$ vectors. When MA order is greater than the AR order, the following relation holds.

$$\mathbf{f}_0^{(0)}(k_1, k_2) = \mathbf{b}_0^{(0)}(k_1, k_2) = [x(m, n) - w(m, n) \quad w(m, n)]^T \quad (6.7)$$

The initial channel inputs of the single-channel lattices are defined as follows. $w(m, n)$ can be estimated by the use of a lattice whitening filter applied to the original image.

$$f_0^{(0)}(k_1, k_2) = b_0^{(0)}(k_1, k_2) = w(m, n) \quad (6.8)$$

7. PERFORMANCE EVALUATION OF THE PROPOSED METHOD

In order to evaluate the performance of the proposed method, we have used the Least Squares (LS) estimates as a benchmark for comparison. In terms of indicating the similarity between the original and identified power spectrums, we have utilized the discrete version of the Itakura-Saito measure. We have also calculated the distance between the original and identified ARMA parameter vectors by the use of L_1 , L_2 and L_∞ vector norms.

7.1. The Least Squares Estimates

The principle of least squares was introduced by the German mathematician Carl Friedrich Gauss, who used it to determine the orbit of the asteroid Ceres in 1821 by formulating the estimation problem as an optimization problem.

The design of optimum filters in the minimum mean square error (MMSE) sense, requires the a priori knowledge of second-order moments. However, such statistical information is simply not available in most practical applications, for which we can only obtain measurements of the input and the desired response signals. To avoid this problem, we can (1) estimate the required second-order moments from the available data, if possible, to obtain an estimate of the optimum MMSE filter, or (2) design an optimum filter by minimizing a criterion of performance that is a function of the available data.

The method, known as least-squares error (LSE) estimation, uses the minimization of the sum of the squares of the estimation error as the criterion of performance. The LSE method requires the measurement of both the input signal and the desired response signal.

For a 2-D ARMA system, the set of $K \times K$ linear equations, $K \times K$ determining the data field size, describing the LSE filter input/output relation can be written in matrix form as follows [200].

$$\begin{bmatrix} \hat{y}(1,1) \\ \hat{y}(1,2) \\ \vdots \\ \hat{y}(1,K) \\ \hat{y}(2,1) \\ \hat{y}(2,2) \\ \vdots \\ \hat{y}(2,K) \\ \vdots \\ \hat{y}(K,1) \\ \hat{y}(K,2) \\ \vdots \\ \hat{y}(K,K) \end{bmatrix} = \begin{bmatrix} y((1,1)-1) & y((1,1)-1) & \dots & y((1,1)-M) & x(1,1) & x((1,1)-1) & x((1,1)-2) & \dots & x((1,1)-N) \\ y((1,2)-1) & y((1,2)-1) & \dots & y((1,2)-M) & x(1,2) & x((1,2)-1) & x((1,2)-2) & \dots & x((1,2)-N) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y((1,K)-1) & y((1,K)-1) & \dots & y((1,K)-M) & x(1,K) & x((1,K)-1) & x((1,K)-2) & \dots & x((1,K)-N) \\ y((2,1)-1) & y((2,1)-2) & \dots & y((2,1)-M) & x(2,1) & x((2,1)-1) & x((2,1)-2) & \dots & x((2,1)-N) \\ y((2,2)-1) & y((2,2)-2) & \dots & y((2,2)-M) & x(2,2) & x((2,2)-1) & x((2,2)-2) & \dots & x((2,2)-N) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y((2,K)-1) & y((2,K)-2) & \dots & y((2,K)-M) & x(2,K) & x((2,K)-1) & x((2,K)-2) & \dots & x((2,K)-N) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y((K,1)-1) & y((K,1)-2) & \dots & y((K,1)-M) & x(K,1) & x((K,1)-1) & y((K,1)-2) & \dots & x((K,1)-N) \\ y((K,2)-1) & y((K,2)-2) & \dots & y((K,2)-M) & x(K,2) & x((K,2)-1) & y((K,2)-2) & \dots & x((K,2)-N) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y((K,K)-1) & y((K,K)-2) & \dots & y((K,K)-M) & x(K,K) & x((K,K)-1) & y((K,K)-2) & \dots & x((K,K)-N) \end{bmatrix} \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_N \\ b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \quad (7.1)$$

Equation (7.1) can be rewritten in compact matrix notation as

$$\hat{\mathbf{y}} = \mathbf{D}\mathbf{w} \quad (7.2)$$

The LSE error is the difference between the desired signal and the filter output defined as

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{D}\mathbf{w} \quad (7.3)$$

The energy of the error vector, that is the sum of the squared elements of the error vector, is given by the inner vector product as

$$\begin{aligned} \mathbf{e}^T \mathbf{e} &= (\mathbf{y} - \mathbf{D}\mathbf{w})^T (\mathbf{y} - \mathbf{D}\mathbf{w}) \\ &= \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{D}\mathbf{w} - \mathbf{w}^T \mathbf{D}^T \mathbf{y} + \mathbf{w}^T \mathbf{D}^T \mathbf{D} \mathbf{w} \end{aligned} \quad (7.4)$$

The gradient of the squared error function with respect to the coefficient vector $\mathbf{w}$ is obtained by differentiating Equation (7.4)

$$\frac{\partial \mathbf{e}^T \mathbf{e}}{\partial \mathbf{w}} = -2\mathbf{y}^T \mathbf{D} + 2\mathbf{w}^T \mathbf{D}^T \mathbf{D} \quad (7.5)$$

The LSE filter coefficients are obtained by setting the gradient of the squared error function of Equation (7.5) to zero, this yields

$$(\mathbf{D}^T \mathbf{D})\mathbf{w} = \mathbf{D}^T \mathbf{y} \quad (7.6)$$

$$\mathbf{w} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{y} \quad (7.7)$$

Note that the matrix $\mathbf{D}^T \mathbf{D}$ is a time-averaged estimate of the autocorrelation matrix of the filter input signal $\mathbf{R}_{yy}$, and the vector $\mathbf{D}^T \mathbf{y}$ is a time averaged estimate of $\mathbf{r}_{xy}$, the cross correlation vector of the input and the desired signals. Theoretically, the

Wiener filter is obtained from minimization of the squared error across the ensemble of different realizations of a process. For a correlation-ergodic process, as the data size $K \times K$ approaches infinity the LSE filter of Equation (7.8) approaches the Wiener filter solution.

$$\lim_{K \rightarrow \infty} [\mathbf{w} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{y}] = \mathbf{R}_{yy}^{-1} \mathbf{r}_{xy} \quad (7.8)$$

Since the LSE method described above require a block of $K \times K$ samples of the input and the desired signals, it is also referred to as the block least square error estimation method. The block estimation method is appropriate for processing of signals that can be considered as time-invariant over the duration of the block.

7.2. The Itakura-Saito Distance Measure

The Itakura-Saito (I-S) error measure was defined originally for continuous spectra [194]. However, it can be adapted to the discrete case as follows [195].

$$E_{IS} = \frac{1}{F_p} \sum_{m=1}^{F_p} \left[\frac{P(w_m)}{\hat{P}(w_m)} - \ln \frac{P(w_m)}{\hat{P}(w_m)} - 1 \right] \quad (7.9)$$

$P(w_m)$ is the given discrete spectrum defined at F_p frequency points, where $w_m \in \Omega$, and $\hat{P}(w_m)$ is the estimated model spectrum evaluated at the same frequencies. This error measure is always nonnegative and is equal to zero only when $\hat{P}(w_m) = P(w_m)$ for all $w_m \in \Omega$.

The continuous form of this error measure was originally presented as part of a maximum likelihood approach to linear prediction and was shown to produce the same result as linear prediction for continuous spectra. Later the discrete version shown in Equation (7.9) was derived by McAulay [196] for the maximum likelihood spectral modeling of periodic speech signals with Gaussian statistics.

It has been shown that minimizing the error in (7.9) is equivalent to maximizing the spectral flatness of the error spectrum $P(w_m)/\hat{P}(w_m)$, where the spectral flatness is defined as the geometric mean of the spectral samples divided by their arithmetic mean [195]. Therefore, at the minimum the following equation is satisfied.

$$\frac{1}{F_p} \sum_{m=1}^{F_p} \frac{P(w_m)}{\hat{P}(w_m)} = 1 \quad (7.10)$$

Consequently, the minimum of the Itakura-Saito error measure, E_{ISmin} is given in Equation (7.11).

$$E_{\text{ISmin}} = \frac{1}{F_p} \sum_{m=1}^{F_p} -\ln \frac{P(w_m)}{\hat{P}(w_m)} = \ln \frac{\left[\prod_{m=1}^{F_p} \hat{P}(w_m) \right]^{1/F_p}}{\left[\prod_{m=1}^{F_p} P(w_m) \right]^{1/F_p}} \quad (7.11)$$

It is concluded from (7.10) and (7.11) that, at the minimum, the energy in the residual spectrum $P(w_m)/\hat{P}(w_m)$ is automatically normalized to 1 and the minimum error is equal to the logarithm of the ratio of the geometric means of the model spectrum and the given spectrum.

7.3. The L_1 , L_2 and L_Ψ Norms

When dealing with vector spaces it is common to talk about the length and direction of the vector, and there is an intuitive geometric concept as to what the length and direction are. There are a variety of ways defining the length of a vector. The mathematical concept associated with the length of a vector is the **norm** of a vector $\mathbf{x}$, written $\|\mathbf{x}\|$. Although there is no unique definition, the following postulates must be satisfied.

1. $\|\mathbf{x}\| \geq 0$ for all $\mathbf{x}$, with equality if and only if $\mathbf{x} = \mathbf{0}$.
2. $\|a\mathbf{x}\| = |a|\|\mathbf{x}\|$ for any scalar a and for all $\mathbf{x}$.
3. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all $\mathbf{x}$ and $\mathbf{y}$.

We use vector norms in order to make a comparison between the original and identified ARMA parameter values. Therefore, we need to make two definitions here related with the original and identified ARMA (M, N) parameter vectors, $\mathbf{x}_o$ and $\mathbf{x}_i$, respectively.

$$\mathbf{x}_o = [\mathbf{a} \ \mathbf{b}] = [a_1 \ a_2 \ \dots \ a_M \ b_0 \ b_1 \ b_2 \ \dots \ b_N]^T \quad (7.12)$$

$$\mathbf{x}_i = [\hat{\mathbf{a}} \quad \hat{\mathbf{b}}] = [\hat{a}_1 \quad \hat{a}_2 \quad \dots \quad \hat{a}_M \quad \hat{b}_0 \quad \hat{b}_1 \quad \hat{b}_2 \quad \dots \quad \hat{b}_N]^T \quad (7.13)$$

7.3.1. L₁-norm

The L₁-norm $\|\mathbf{x}\|_1$ is a vector norm defined for a complex vector $\mathbf{x}=[x_1 \ x_2 \ \dots \ x_n]^T$ by

$$\|\mathbf{x}\|_1 = \sum_{r=1}^n |x_r| \quad (7.14)$$

We calculate the L₁-norm of our original and identified ARMA (M, N) parameter vectors, $\mathbf{x}_o$ and $\mathbf{x}_i$, as follows.

$$\|\mathbf{x}_o\|_1 = \sum_{r=1}^{M+N+1} |x_{o,r}| \quad (7.15a)$$

$$\|\mathbf{x}_i\|_1 = \sum_{r=1}^{M+N+1} |x_{i,r}| \quad (7.15b)$$

7.3.2. L₂-norm

The L₂-norm $\|\mathbf{x}\|_2$ is a vector norm defined for a complex vector $\mathbf{x}=[x_1 \ x_2 \ \dots \ x_n]^T$ by

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{k=1}^n |x_k|^2} \quad (7.16)$$

where $|x_k|$ on the right denotes the complex modulus. The L₂-norm is also known as the Euclidean norm. However, this terminology is not recommended since it may cause confusion with the Frobenius norm (a matrix norm) and is also sometimes called the Euclidean norm.

We calculate the L₂-norm of our original and identified ARMA (M, N) parameter vectors, $\mathbf{x}_o$ and $\mathbf{x}_i$, as follows.

$$\|\mathbf{x}_o\|_2 = \sqrt{\sum_{k=1}^{M+N+1} |x_{o,k}|^2} \quad (7.17a)$$

$$\|\mathbf{x}_i\|_2 = \sqrt{\sum_{k=1}^{M+N+1} |x_{i,k}|^2} \quad (7.17b)$$

7.3.3. L_∞ -norm

The L_∞ -norm $\|\mathbf{x}\|_\infty$ is a vector norm defined for a complex vector $\mathbf{x}=[x_1 \ x_2 \ \dots \ x_n]^T$ by

$$\|\mathbf{x}\|_\infty = \max_j |x_j| \quad (7.18)$$

We calculate the L_∞ -norm of our original and identified ARMA (M, N) parameter vectors, $\mathbf{x}_o$ and $\mathbf{x}_i$, as follows.

$$\|\mathbf{x}_o\|_\infty = \max_j |x_{o,j}| \quad (7.19a)$$

$$\|\mathbf{x}_i\|_\infty = \max_j |x_{i,j}| \quad (7.19b)$$

8. EXPERIMENTAL RESULTS

In order to make a verification of the proposed method, some simulations have been carried out to identify the ARMA (M,N) system parameters for the cases when the AR order is equal to the MA order ($M=N$), the AR order is greater than the MA order ($M>N$) and the AR order is smaller than the MA order ($M<N$). The prediction supports used in the simulations cover the FQP and ASHP regions and the orderings used have been illustrated in related figures. An estimate of each of the coefficients of the considered ARMA (M,N) processes has been characterized by the mean value. The acquired mean values correspond to the identified parameters of the considered 2-D ARMA (M,N) processes. These mean values have been acquired by running 100 independent Monte-Carlo simulations of the proposed algorithm. LS estimates have been calculated as a benchmark for comparison. The performance of the developed method has been evaluated in each of the computer experiments by the use of Itakura-Saito distance measure with 101 frequency points, and also L_1 , L_2 and L_∞ vector norms. In order to illustrate the performance by means of visual comparison, power spectrums and contours have been plotted using both the original and identified ARMA (M,N) parameter vectors.

The differences in computer experiments are listed in Table 8.1 in terms of ARMA(M, N) order, the parameter b_0 , system input, prediction region support and prediction region ordering schemes. The computer experiments numbered 1-15 are carried out using ARMA(M, N) difference equation formulation while the others are focused on image identification problem.

The ARMA parameters in all the computer experiments have been calculated based on the formulations given in Chapter 5.

Table 8.1: A summary of the Computer Experiments.

Computer Experiment No	ARMA (M,N) Order	b_0	System Input	Prediction Region Support	Prediction Region Ordering Scheme
1	(3,3)	1	White noise	FQP	Horizontal-Vertical
2	(3,3)	0.75	White noise	FQP	Horizontal-Vertical
3	(3,3)	-0.8	AR(3) process	FQP	Horizontal-Vertical
4	(4,4)	0.38	White noise	ASHP	Vertical-Horizontal
5	(8,8)	0.1	White noise	FQP	Horizontal-Vertical
6	(8,8)	0.23	AR(8) process	FQP	Horizontal-Vertical
7	(8,8)	0.22	White noise	FQP	Column-wise
8	(15,15)	0.25	White noise	FQP	Horizontal-Vertical
9	(15,15)	0.25	White noise	FQP	Column-wise
10	(3,2)	0.22	White noise	FQP	Horizontal-Vertical
11	(15,8)	1	White noise	FQP	Horizontal-Vertical
12	(15,8)	0.3	White noise	FQP	Horizontal-Vertical
13	(2,3)	0.75	White noise	FQP	Horizontal-Vertical
14	(3,8)	0.8	White noise	FQP	Horizontal-Vertical
15	(8,15)	0.85	White noise	FQP	Horizontal-Vertical
16	(4,8)	0.36	Simulated image $v(m,n)$ neglected.	ASHP	Vertical-Horizontal
17	(4,8)	0.36	Camerman image $v(m,n)$ neglected.	ASHP	Vertical-Horizontal
18	(4,8)	0.36	Camerman image $v(m,n)$ NOT neglected.	ASHP	Vertical-Horizontal

8.1. Computer Experiment 1

The unknown 2-D ARMA(3,3) system with FQP support has the following difference equation:

$$y(k_1, k_2) = x(k_1, k_2) + 0.15x(k_1 - 1, k_2) - 0.35x(k_1 - 1, k_2 - 1) + 0.1x(k_1, k_2 - 1) - 0.25y(k_1 - 1, k_2) + 0.25y(k_1 - 1, k_2 - 1) + 0.25y(k_1, k_2 - 1) \quad (8.1)$$

The original ARMA(3,3) parameters have been chosen so as to satisfy the stability conditions given in [119]. The system input $x(k_1, k_2)$ is white noise process with

variance s_x^2 and mean zero. The horizontal-vertical ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ are illustrated in Figure 8.1.

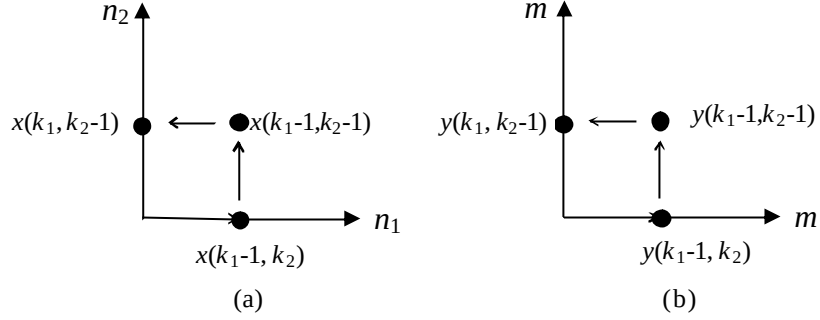


Figure 8-1: FQP ordering schemes of the Computer Experiment 1 - ARMA(3,3) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.1.1. Objectives of Computer Experiment 1

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm for the case when the AR order is equal to the MA order ($M=3, N=3$ and FQP support) and the parameter $b_0=1$.
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.1.2. Results of Computer Experiment 1

The results of the Computer Experiment 1 have been given in Table 8.2 and Table 8.3 for different data sizes. Table 8.2 lists the original ARMA(3,3) parameters versus the identified ARMA(3,3) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.3 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original

parameters versus the Lattice and LS identified parameters for a data field size of 10x10, have been given in Figure 8.2 and Figure 8.3, respectively.

Table 8.2: Computer Experiment 1 - ARMA(3,3) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

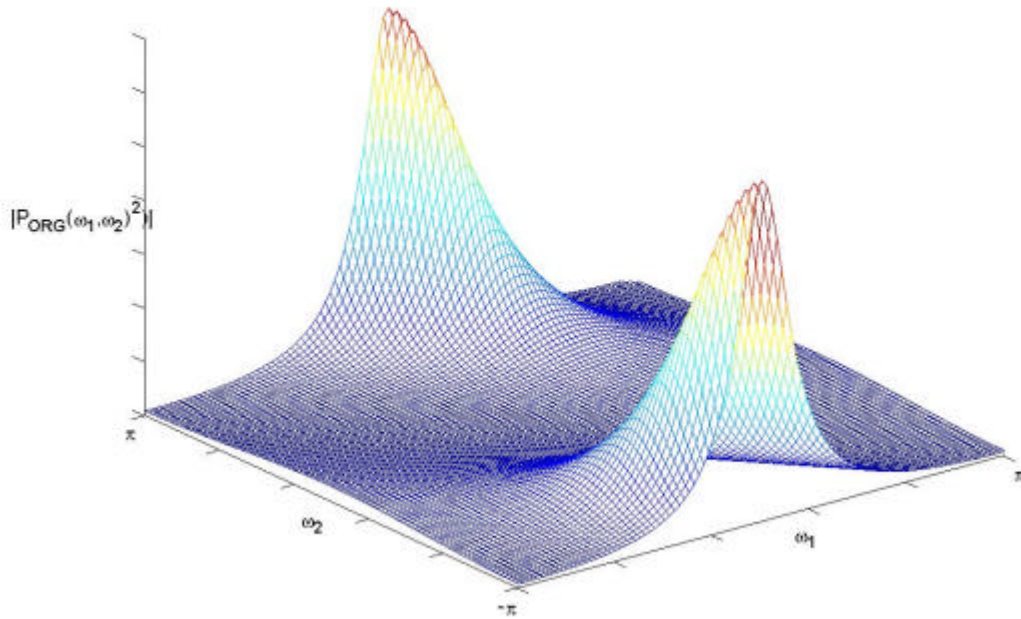
Original ARMA(3,3) Parameters		Identified ARMA Parameters (Hybrid Lattice and LS)		
		Data Field Size		
		10x10	35x35	100x100
AR Parameters	-0.25	-0.2517	-0.2473	-0.2492
		-0.2500	-0.2500	-0.2500
	0.25	0.2751	0.2587	0.2530
		0.2500	0.2500	0.2500
	0.25	0.2048	0.2375	0.2460
		0.2500	0.2500	0.2500
MA Parameters	0.15	0.1547	0.1527	0.1508
		0.1500	0.1500	0.1500
	-0.35	-0.3442	-0.3463	-0.3487
		-0.3500	-0.3500	-0.3500
	0.1	0.0706	0.0934	0.0973
		0.1000	0.1000	0.1000
b_0	1	0.9982	0.9991	0.9997
		1.0000	1.0000	1.0000

Table 8.3: Computer Experiment 1 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

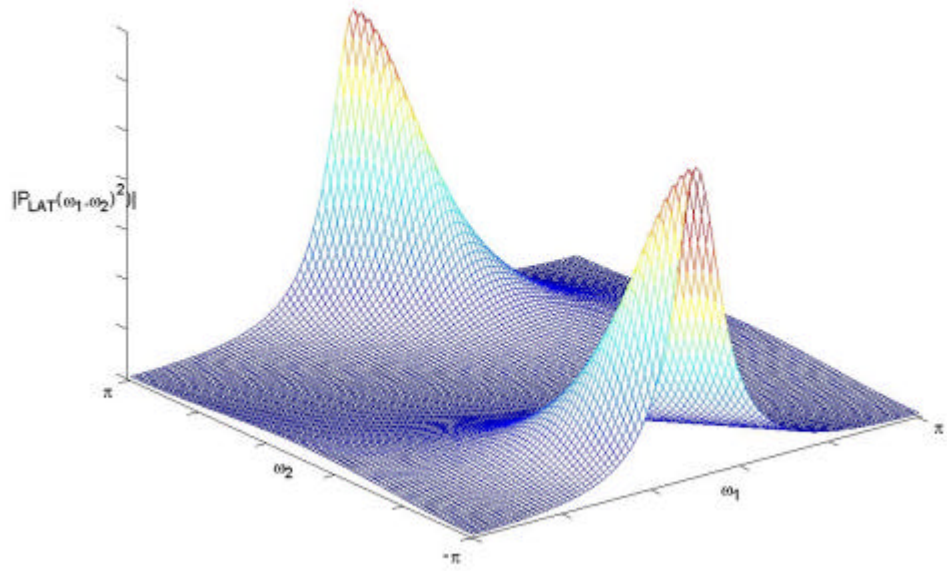
Performance Criteria (Hybrid Lattice and LS)	Data Field Size		
	10x10	35x35	100x100
L_1 -norm	0.1137	0.0378	0.0129
	0.0000	0.0000	0.0000
L_2 -norm	0.0600	0.0174	0.0059
	0.0000	0.0000	0.0000
L_∞ -norm	0.0452	0.0125	0.0040
	0.0000	0.0000	0.0000
Itakura-Saito Distance	0.001092	0.000091	0.000008
	0.000000	0.000000	0.000000

The obtained results yield the following conclusions for this experiment:

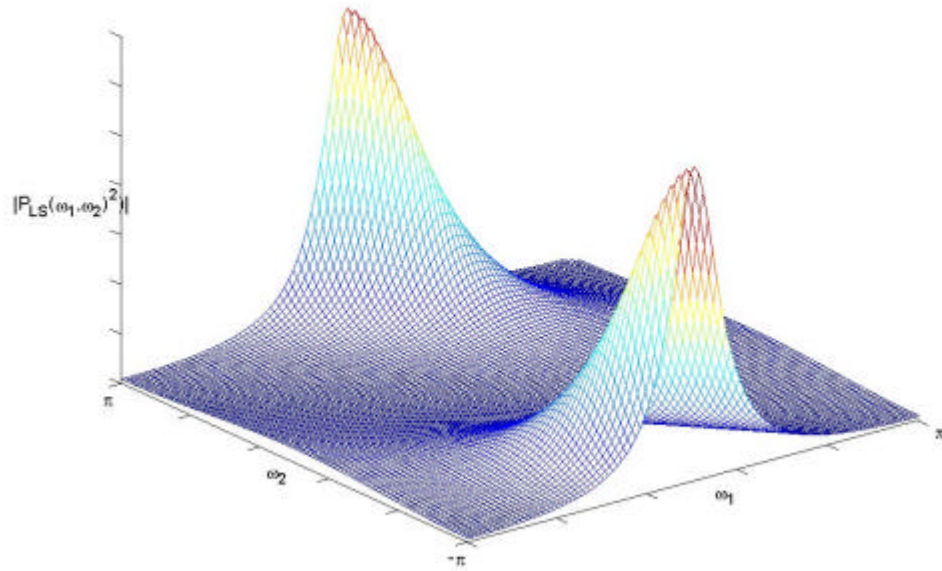
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results.
- The identification results obtained with the proposed 2-D hybrid lattice algorithm improve a great deal when the data size increases as it can easily be seen from Table 8.2 and Table 8.3.
- Even for a small data field size of 10x10, ARMA(3,3) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.
- By looking at the Itakura-Saito distances listed in Table 8.3 and the power spectrum and contour plots given in Figure 8.2 and Figure 8.3, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

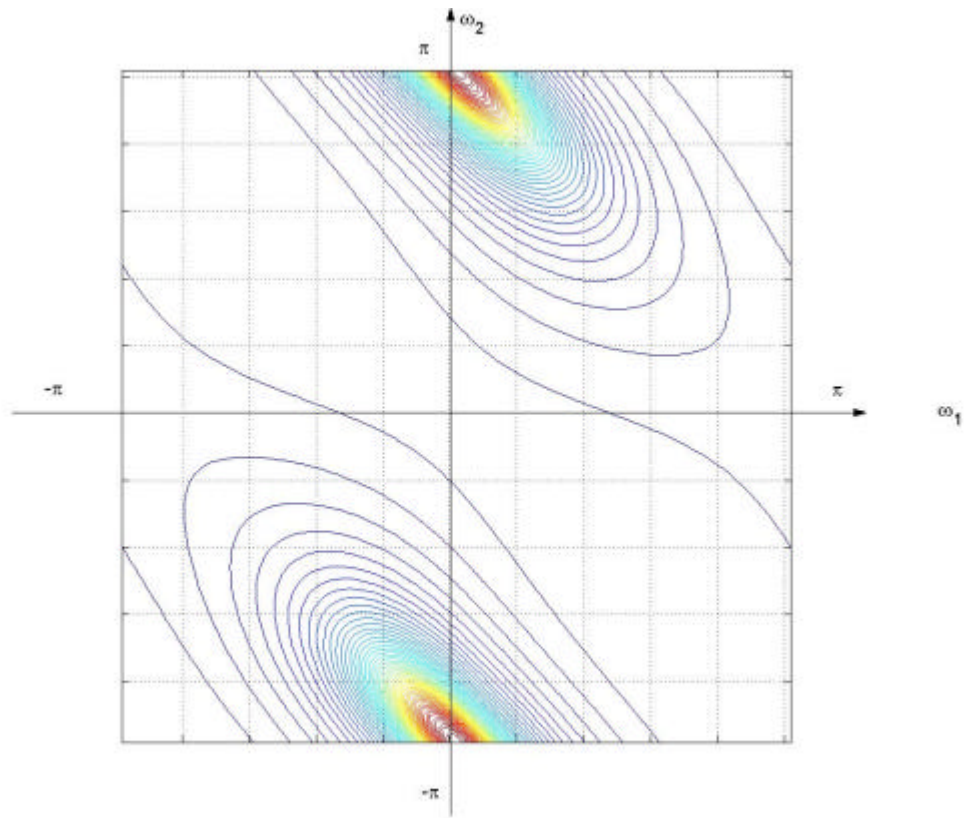


(b)

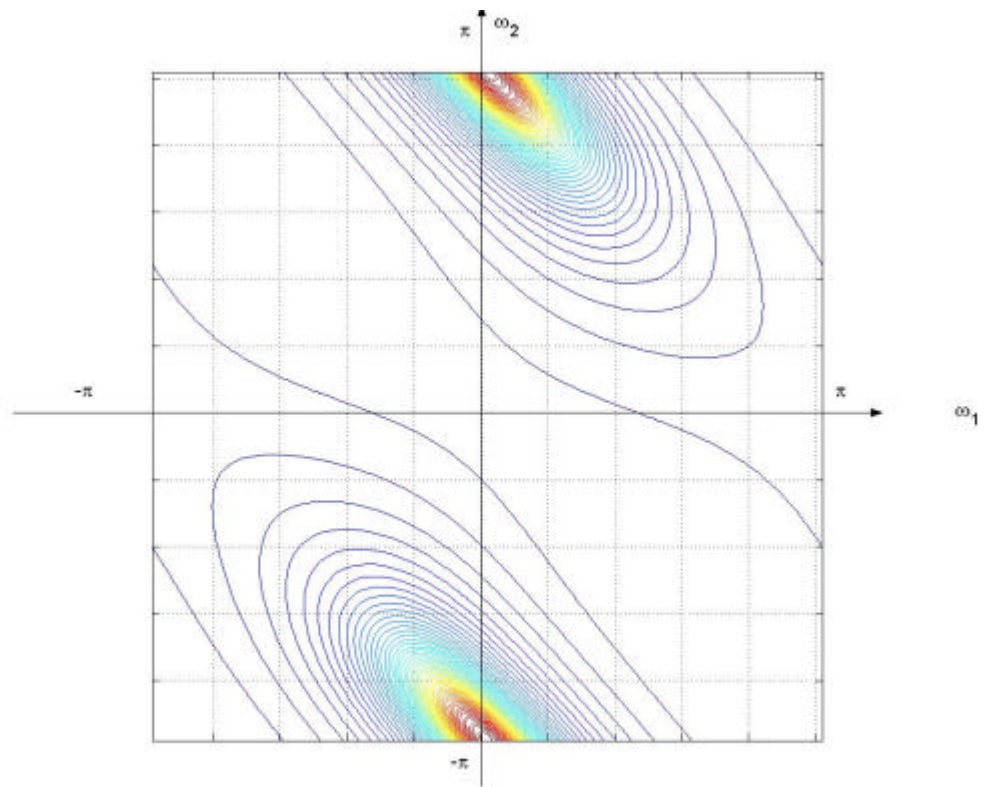


(c)

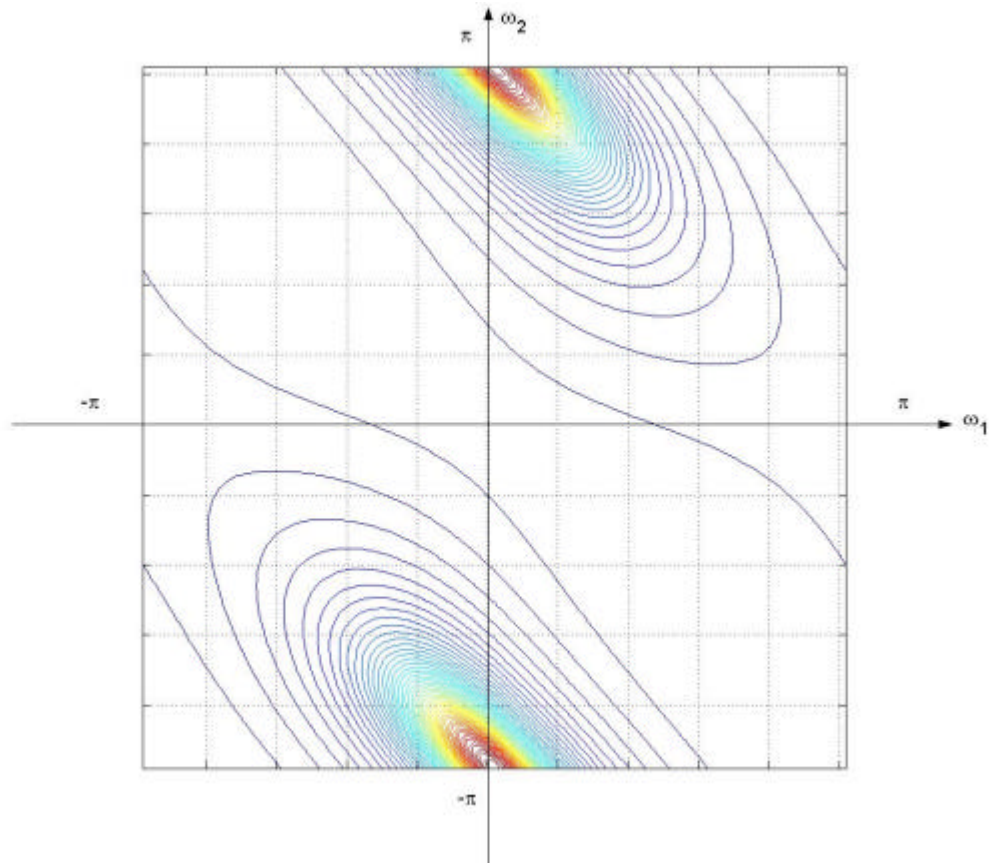
Figure 8-2: Computer Experiment 1- Power Spectrums plotted using (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=1$ and the data field size is 10×10 .



(a)



(b)



(c)

Figure 8-3: Computer Experiment 1- Contour plots of (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=1$ and the data field size is 10×10 .

8.2. Computer Experiment 2

The unknown 2-D ARMA(3,3) system with a FQP support has the following difference equation.

$$y(k_1, k_2) = 0.75x(k_1, k_2) + 0.7(k_1 - 1, k_2) + 0.3x(k_1 - 1, k_2 - 1) - 0.5x(k_1, k_2 - 1) + 0.4y(k_1 - 1, k_2) + 0.4y(k_1 - 1, k_2 - 1) + 0.2y(k_1, k_2 - 1) \quad (8.2)$$

The original ARMA(3,3) parameters have been chosen so as to satisfy the stability conditions given in [119]. The system input $x(k_1, k_2)$ is a white noise process driven by white-Gaussian noise with variance s_x^2 and mean zero. The horizontal-vertical ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ are the same as illustrated in Figure 8.1.

8.2.1. Objectives of Computer Experiment 2

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(3,3) parameter calculation formulas given in Equations (4.3) and (5.6), respectively, for the case which the AR order is equal to the MA order ($M=3$, $N=3$ and FQP support) and the parameter b_0 is different than 1 ($b_0=0.75$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.2.2. Results of Computer Experiment 2

The results of the Computer Experiment 2 have been given in Table 8.4 and Table 8.5 for different data sizes. Table 8.4 lists the original ARMA(3,3) parameters versus the identified ARMA(3,3) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.5 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the 2-D Hybrid Lattice and LS identified parameters for 10x10 data field size have been given in Figure 8.4 and Figure 8.5, respectively.

Table 8.4: Computer Experiment 2 - ARMA(3,3) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Original ARMA(3,3) Parameters		Identified ARMA Parameters (Hybrid Lattice and LS)		
		Data Field Size		
		10x10	35x35	100x100
AR Parameters	0.4	0.3967	0.3985	0.3996
		0.4000	0.4000	0.4000
	0.4	0.3908	0.3968	0.3990
		0.4000	0.4000	0.4000
	0.2	0.1723	0.1908	0.1970
		0.2000	0.2000	0.2000
MA Parameters	0.7	0.6973	0.6989	0.6997
		0.7000	0.7000	0.7000
	0.3	0.2846	0.2949	0.2983
		0.3000	0.3000	0.3000
	-0.5	-0.4656	-0.4896	-0.4960
		-0.5000	-0.5000	-0.5000
b_0	0.75	0.7506	0.7503	0.7500
		0.7500	0.7500	0.7500

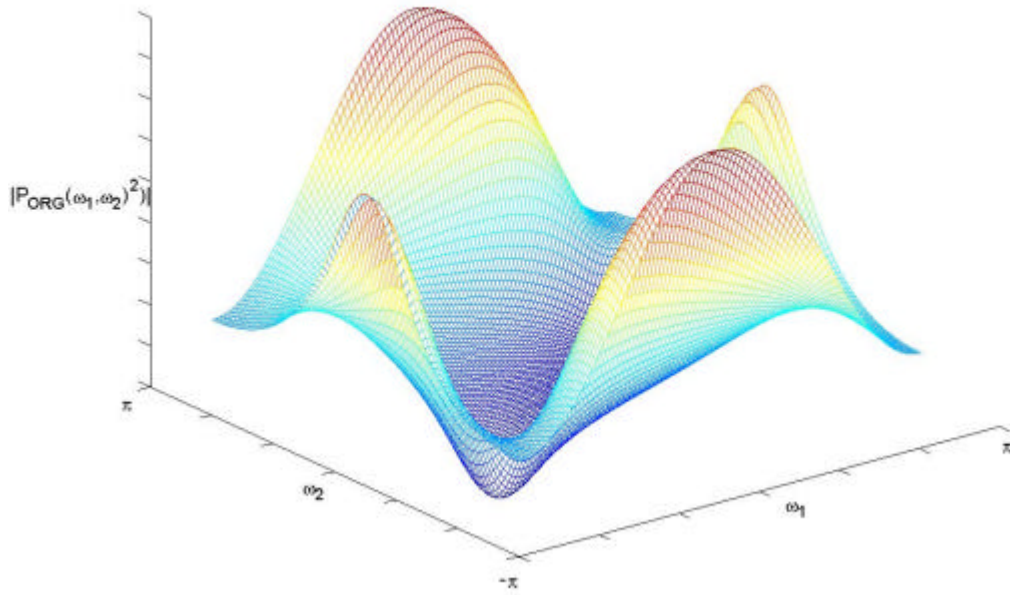
Table 8.5: Computer Experiment 2 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Performance Criteria (Hybrid Lattice and LS)	Data Field Size		
	10x10	35x35	100x100
L_1 -norm	0.0933	0.0308	0.0104
	0.0000	0.0000	0.0000
L_2 -norm	0.0479	0.0153	0.0054
	0.0000	0.0000	0.0000
L_∞ -norm	0.0344	0.0104	0.0040
	0.0000	0.0000	0.0000
Itakura-Saito Distance	0.009430	0.000693	0.000137
	0.000000	0.000000	0.000000

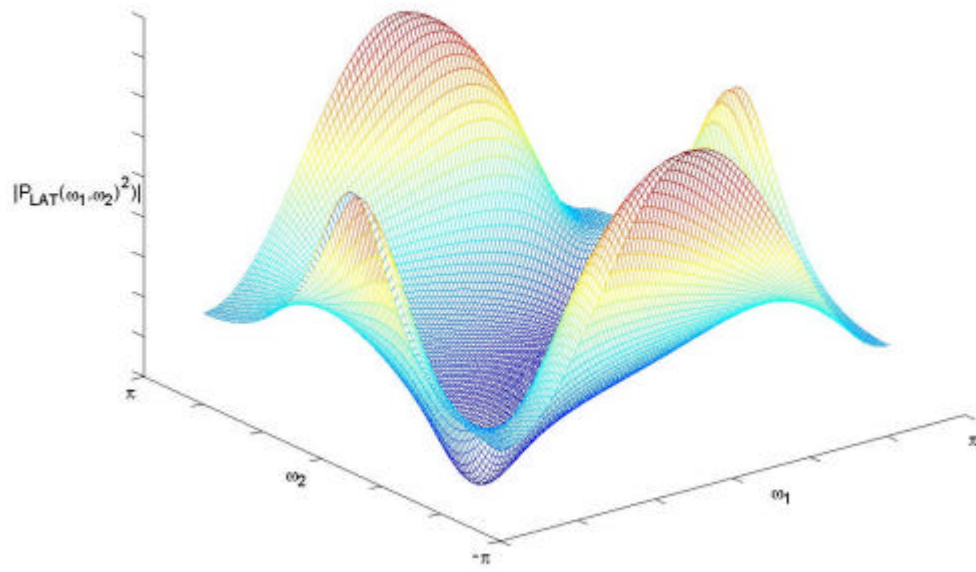
The obtained results yield the following conclusions for this experiment:

- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).

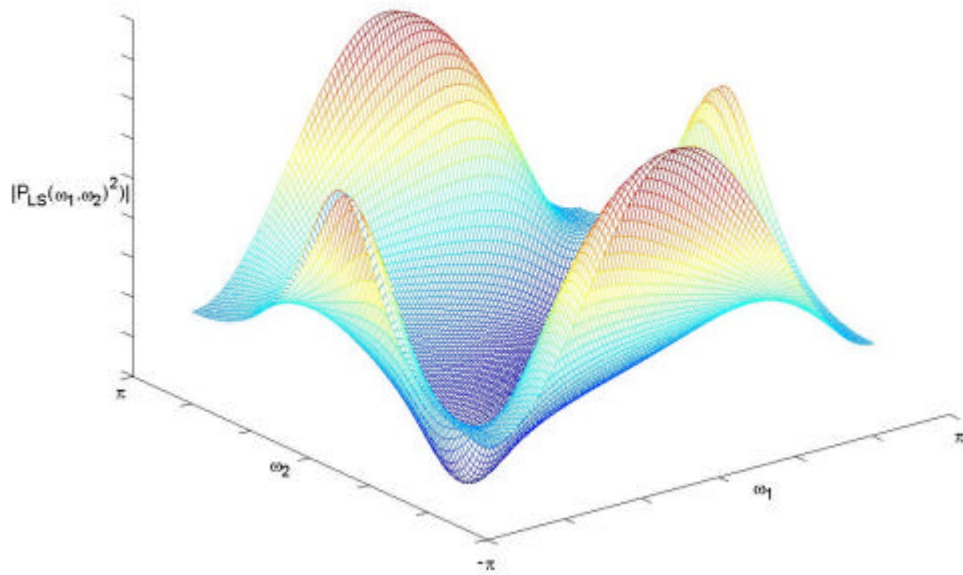
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm improve a great deal when the data size increases as it can easily be seen from Table 8.4 and Table 8.5.
- Even for a small data field size of 10x10, ARMA(3,3) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.
- By looking at the Itakura-Saito distances listed in Table 8.5 and the power spectrum and contour plots given in Figure 8.3, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

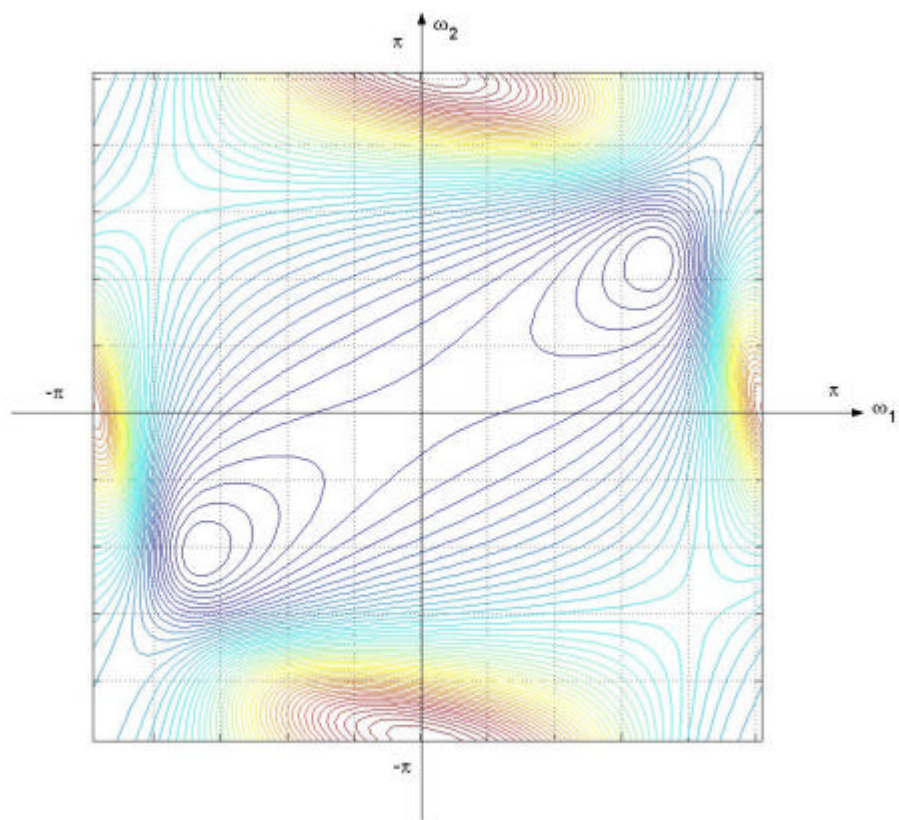


(b)

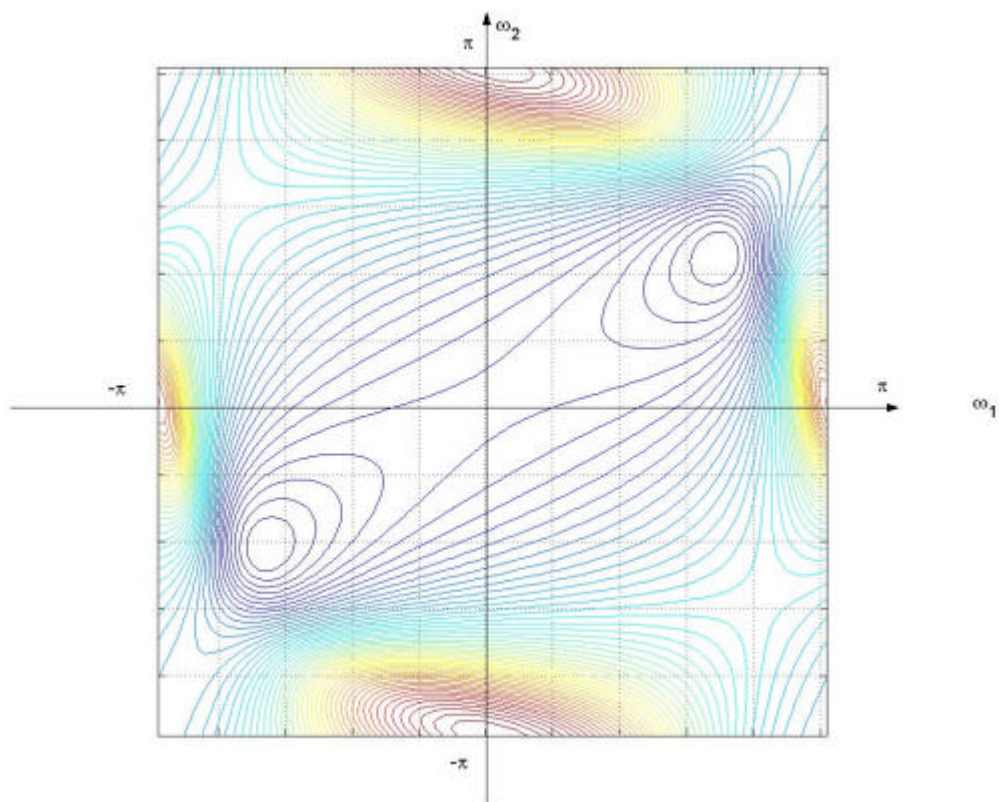


(c)

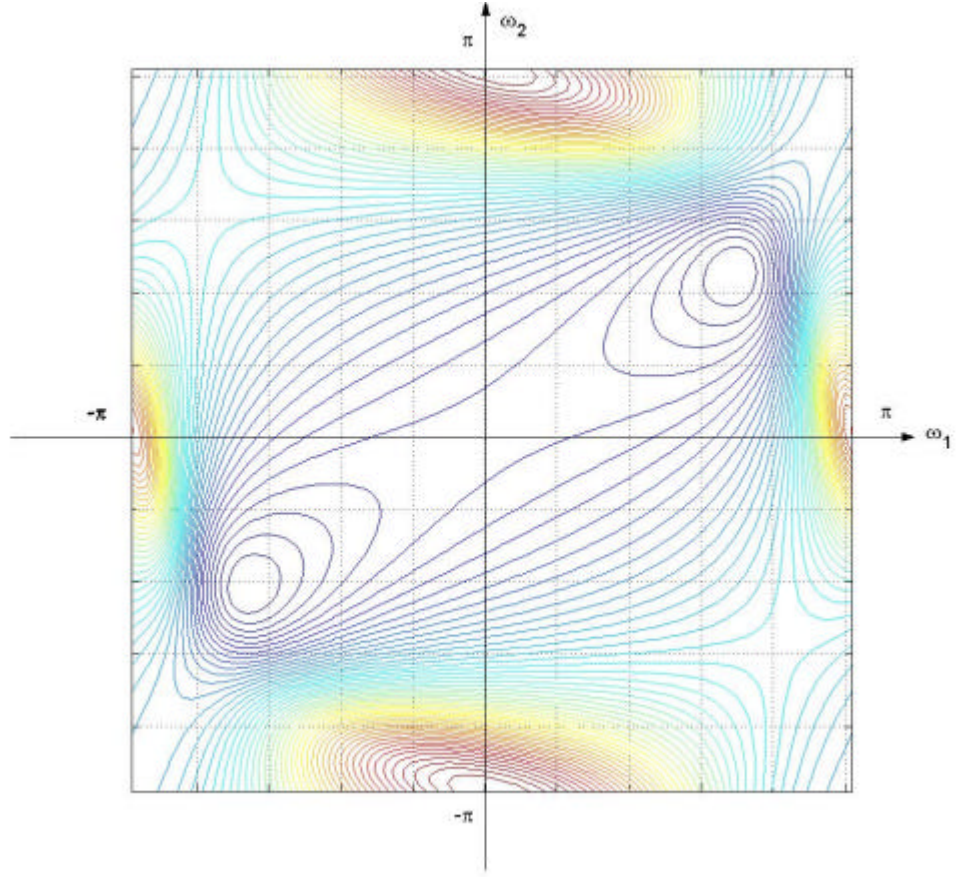
Figure 8-4: Computer Experiment 2- Power Spectrums plotted using (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=0.75$ and the data field size is 10×10 .



(a)



(b)



(c)

Figure 8-5: Computer Experiment 2 Contour plots of (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=0.75$ and the data field size is 10×10 .

8.3. Computer Experiment 3

The unknown 2-D ARMA(3,3) system with FQP support has the following difference equation.

$$y(k_1, k_2) = -0.8x(k_1, k_2) - 0.4x(k_1 - 1, k_2) + 0.03x(k_1 - 1, k_2 - 1) + 0.025x(k_1, k_2 - 1) - 0.4y(k_1 - 1, k_2) + 0.3y(k_1 - 1, k_2 - 1) + 0.25y(k_1, k_2 - 1) \quad (8.3)$$

The horizontal-vertical ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ are the same as illustrated in Figure 8.1. The ARMA(3,3) system input $x(k_1, k_2)$ is an AR process with $N=3$ satisfying the following difference equation.

$$x(k_1, k_2) = w(k_1, k_2) - 0.3x(k_1 - 1, k_2) + 0.3x(k_1 - 1, k_2 - 1) - 0.5x(k_1, k_2 - 1) \quad (8.4)$$

8.3.1. Objectives of Computer Experiment 3

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(3,3) parameter calculation formulas given in Equations (4.3) and (5.6), respectively, for the case which the AR order is equal to the MA order ($M=3$, $N=3$ and FQP support) and the parameter b_0 is different than 1 ($b_0=-0.8$).
- To verify the validity of the proposed 2-D hybrid lattice algorithm for the case when the input signal $x(k_1, k_2)$ is different than white noise, for this experiment a 2-D AR process, the difference equation of which is given in 8.4.
- To show that the AR parameters of the input signal $x(k_1, k_2)$ can be estimated simultaneously with the proposed 2-D hybrid lattice algorithm.
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.3.2. Results of Computer Experiment 3

The results of the Computer Experiment 3 have been given in Table 8.6-Table 8.9 for different data sizes. Table 8.6 lists the original ARMA(3,3) parameters versus the identified ARMA(3,3) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.7 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. Table 8.8 and Table 8.9 list the simultaneous AR parameter estimates of the input signal $x(k_1, k_2)$ and the related performance results, respectively. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for 100x100 data field size have been given in Figure 8.6 and Figure 8.7, respectively.

Table 8.6: Computer Experiment 3 - ARMA(3,3) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Original ARMA(3,3) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	35x35	100x100
AR Parameters	-0.4	-0.3929	-0.3962	-0.3985
		-0.4000	-0.4000	-0.4000
	0.3	0.3772	0.3360	0.3142
		0.3000	0.3000	0.3000
	0.25	0.1563	0.2101	0.2346
		0.2500	0.2500	0.2500
MA Parameters	-0.4	-0.4081	-0.4038	-0.4013
		-0.4000	-0.4000	-0.4000
	0.03	0.0286	0.0251	0.0276
		0.0300	0.0300	0.0300
	0.025	0.0629	0.0432	0.0324
		0.0250	0.0250	0.0250
b_0	-0.8	-0.7780	-0.7918	-0.7971
		-0.8000	-0.8000	-0.8000

Table 8.7: Computer Experiment 3 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Performance Criteria	Data Field Size		
	10x10	35x35	100x100
L_1 -norm	0.2474	0.1148	0.0451
	0.0000	0.0000	0.0000
L_2 -norm	0.1295	0.0578	0.0226
	0.0000	0.0000	0.0000
L_∞ -norm	0.0937	0.0399	0.0154
	0.0000	0.0000	0.0000
Itakura-Saito Distance	0.008272	0.001339	0.000189
	0.000000	0.000000	0.000000

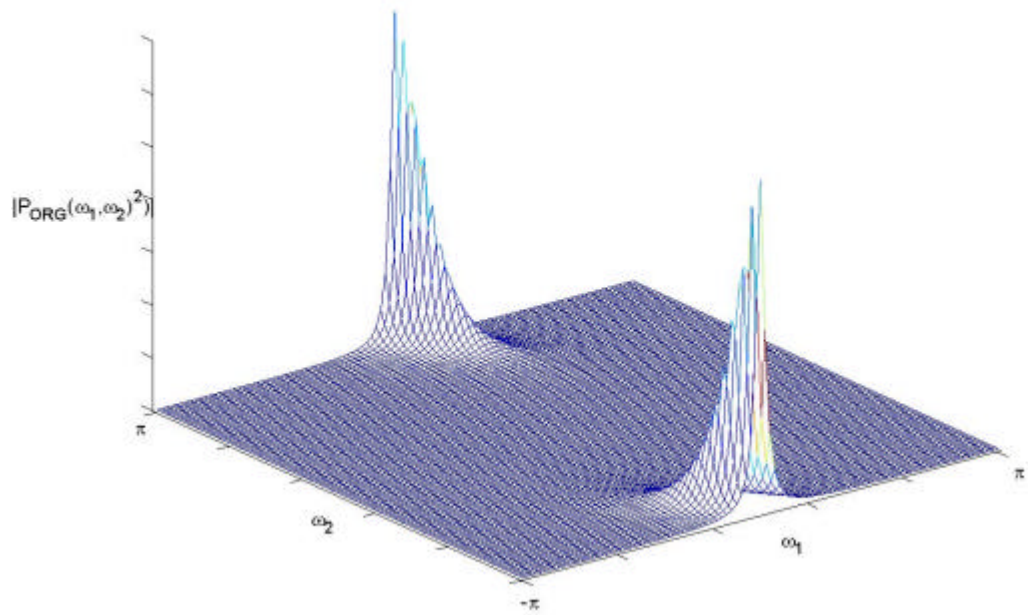
Table 8.8: Computer Experiment 3 – Hybrid Lattice parameter estimates of the input AR (3) process $x(k_1, k_2)$.

Original AR Parameters	Estimated AR Parameters (Mean Values)		
	Data Field Size		
	10x10	35x35	100x100
0.3	0.2978	0.3071	0.3005
-0.3	-0.2758	-0.2933	-0.3016
0.5	0.4315	0.4827	0.4945

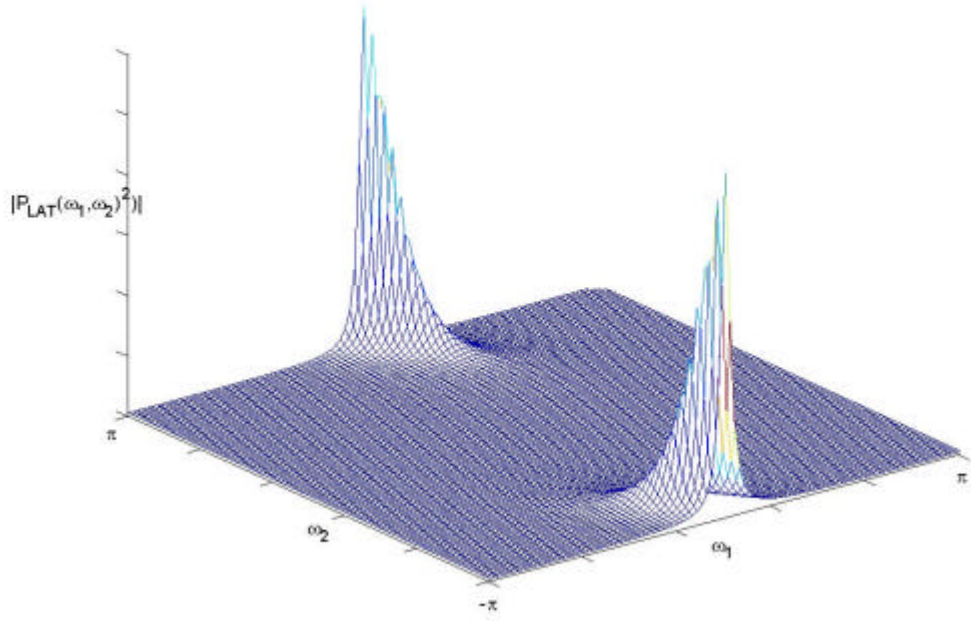
Table 8.9: Computer Experiment 3 - Performance criteria results in estimating the AR (3) parameters of $x(k_1, k_2)$.

Performance Criteria	Data Field Size		
	10x10	35x35	100x100
L₁-norm	0.0949	0.0311	0.0076
L₂-norm	0.0727	0.0199	0.0057
L_y-norm	0.0685	0.0173	0.0055
Itakura-Saito Distance	0.033288	0.001863	0.000118

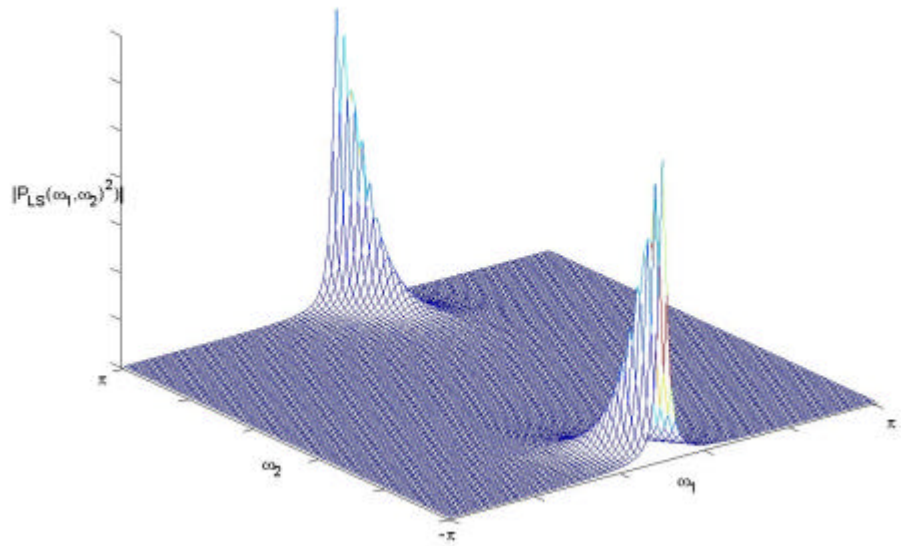
The identified versus original AR(3) power spectrums and contour plots of the AR(3) system input have been given in Figure 8.8.



(a)

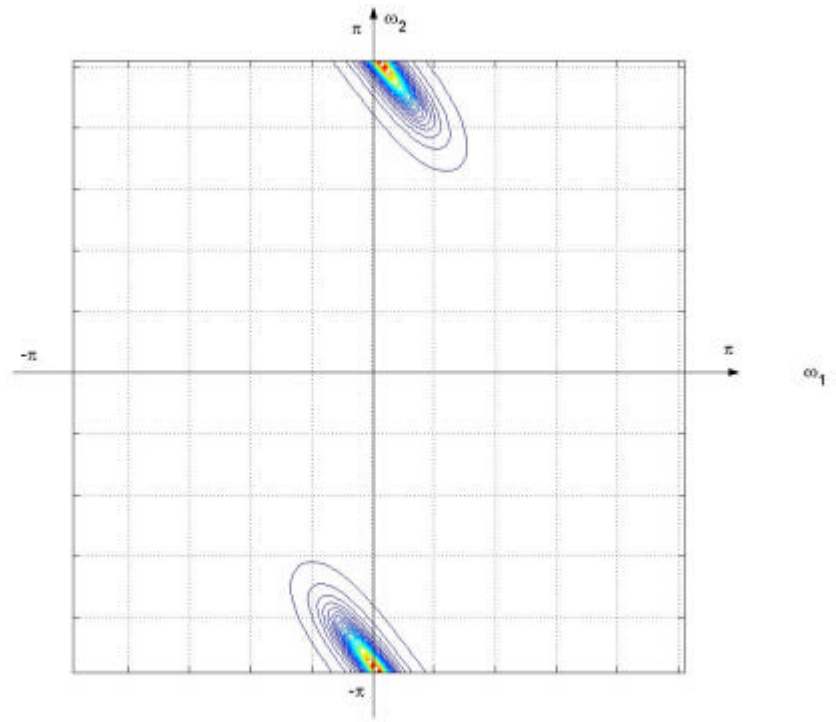


(b)

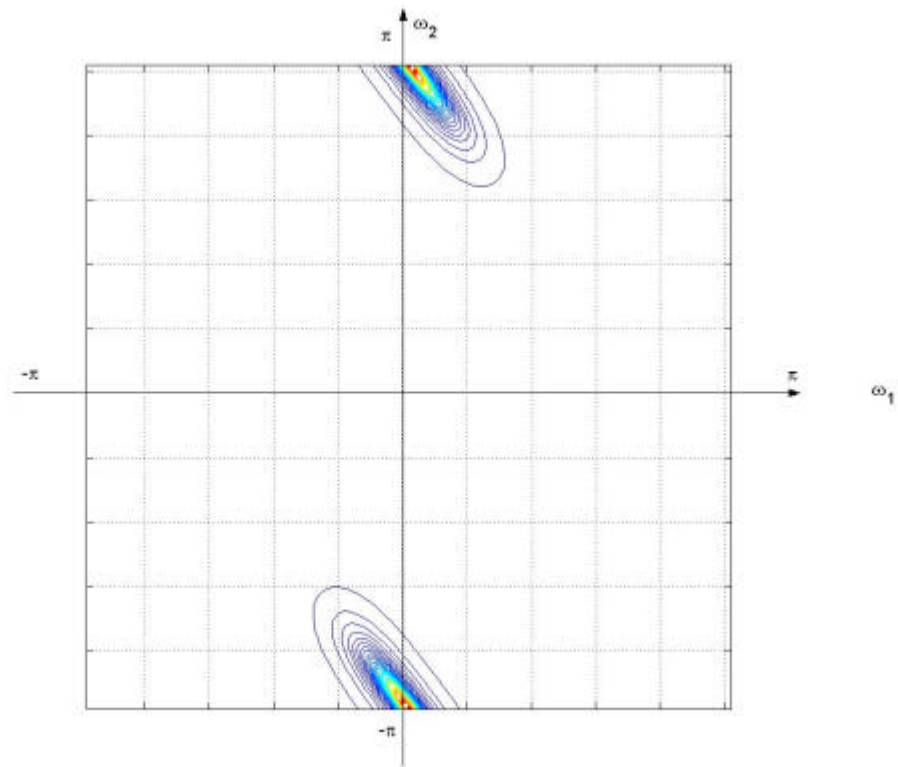


(c)

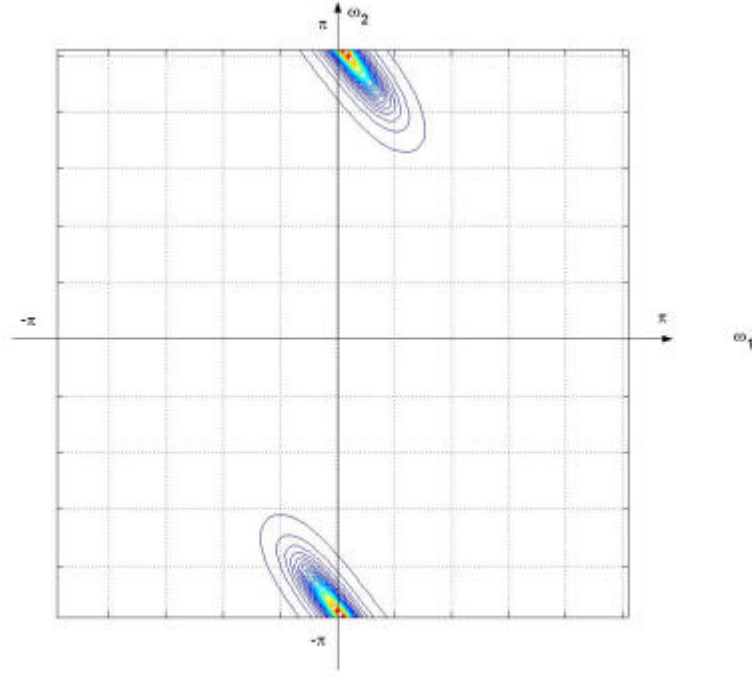
Figure 8-6: Computer Experiment 3- Power Spectrums plotted using (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0=-0.8$ and the data field size is 100×100 . The system input $x(k_1, k_2)$ is an AR process of order 3.



(a)



(b)

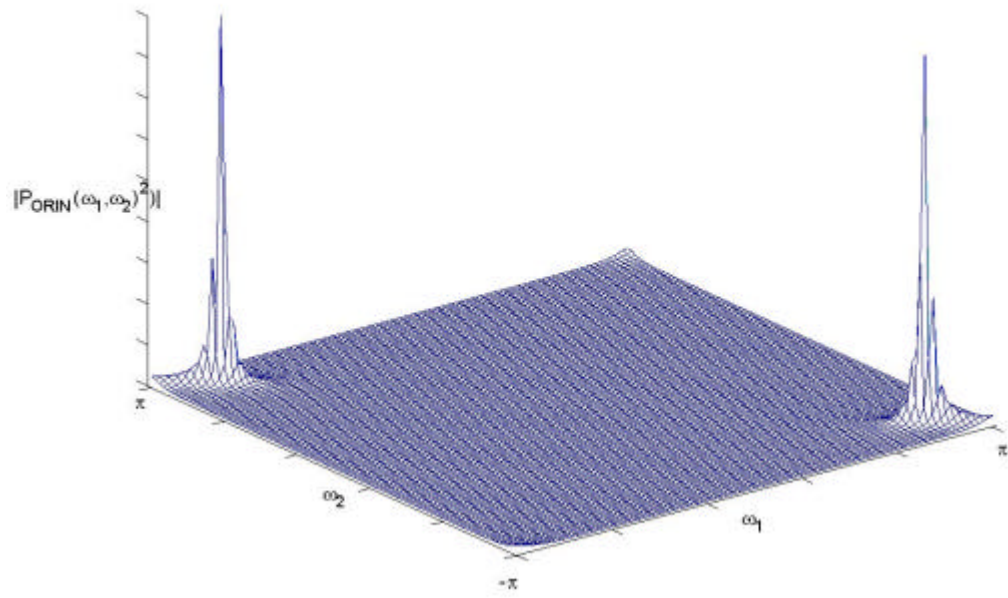


(c)

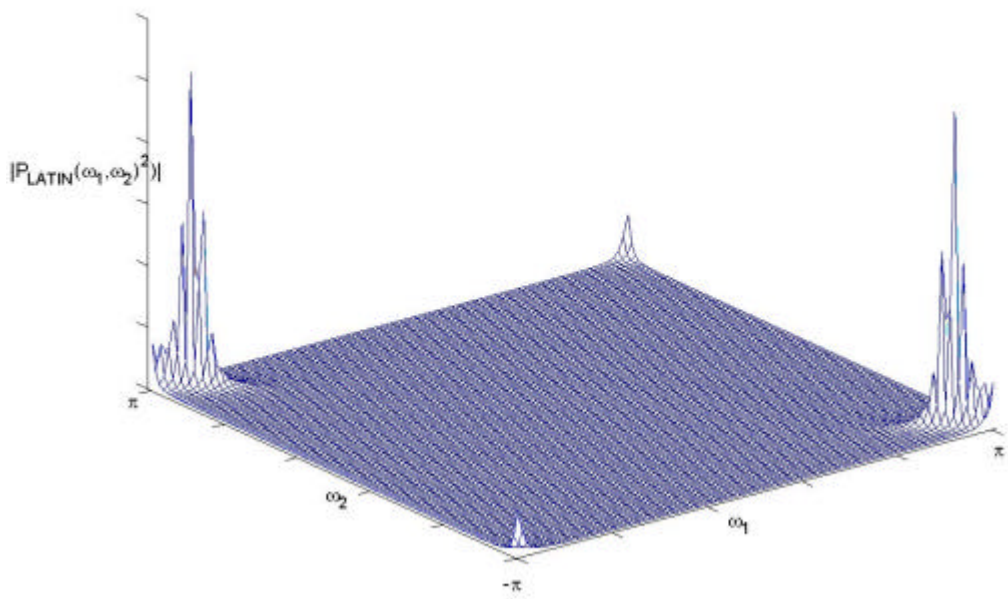
Figure 8-7: Computer Experiment 3 - Contour plots of (a) Original ARMA(3,3) parameters (b) ARMA(3,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,3) parameters identified by the LS algorithm. The parameter $b_0 = -0.8$ and the data field size is 100x100. The system input $x(k_1, k_2)$ is an AR process of order 3.

The obtained results yield the following conclusions for this experiment:

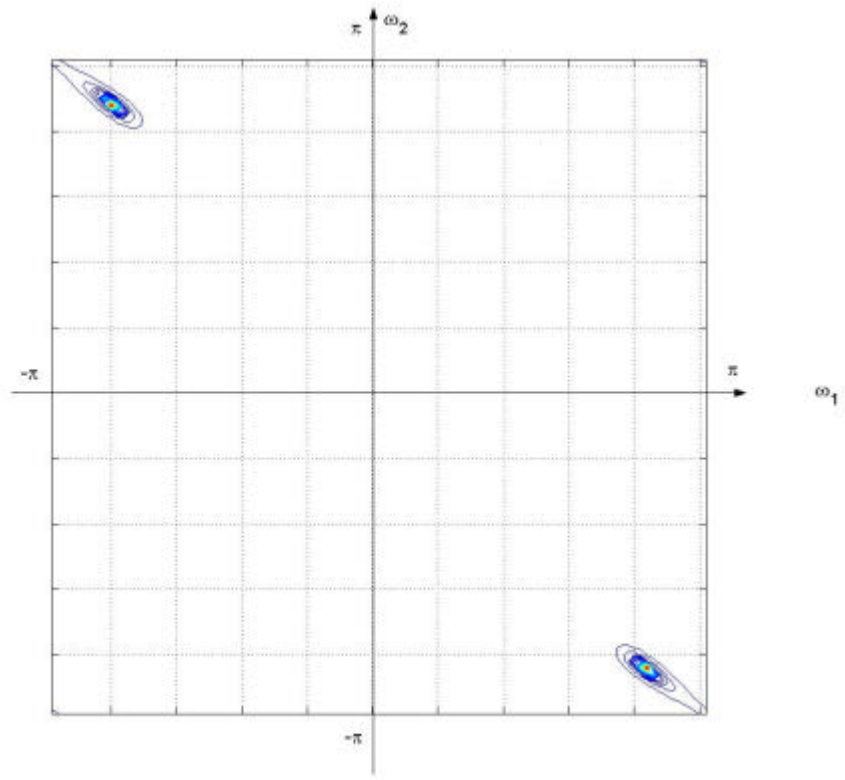
- The parameters of ARMA(3,3) system input $x(k_1, k_2)$ have been simultaneously identified with the proposed 2-D hybrid lattice algorithm provided that $x(k_1, k_2)$ is an AR process of order N . As it can easily be seen from Table 8.6 and Table 8.8, the identification results are quite accurate and improve with the increasing data size, thus verifying numerically the formulation derived in Equation (5.6).
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- Even for a small data field size of 10x10, ARMA(3,3) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.
- By looking at the Itakura-Saito distances listed in Table 8.7 and Table 8.9 and the power spectrum and contour plots given in Figure 8.6-Figure 8.8, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



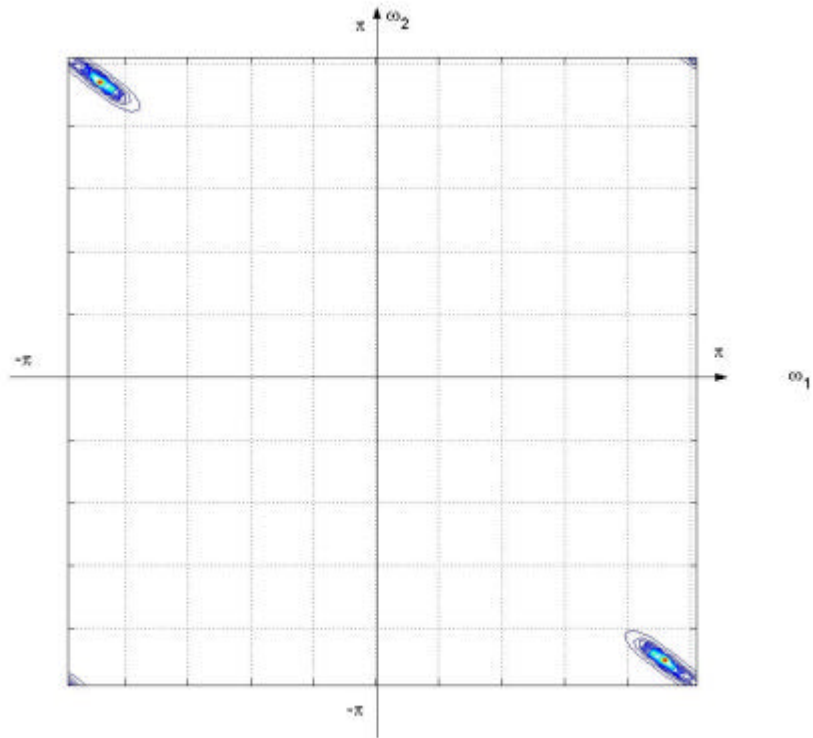
(a)



(b)



(c)



(d)

Figure 8-8 :Computer Experiment 3 -:(a) Original power spectrum of the input AR(3) data field $x(k_1, k_2)$ (b) Power spectrum of $x(k_1, k_2)$ plotted using parameters identified by the Hybrid Lattice algorithm (c) Original contour plot of $x(k_1, k_2)$ (d) Identified contour plot of $x(k_1, k_2)$.

8.4. Computer Experiment 4

The unknown 2-D ARMA(4,4) system with causal ASHP support has got the following difference equation:

$$\begin{aligned} y(k_1, k_2) = & 0.38x(k_1, k_2) - 0.3x(k_1 - 1, k_2) + 0.25x(k_1 - 1, k_2 - 1) - 0.2x(k_1, k_2 - 1) \\ & + 0.1x(k_1 + 1, k_2 - 1) + 0.25y(k_1 - 1, k_2) + 0.2y(k_1 - 1, k_2 - 1) + 0.3y(k_1, k_2 - 1) \\ & + 0.4y(k_1 + 1, k_2 - 1) \end{aligned} \quad (8.5)$$

The system input $x(k_1, k_2)$ is white-Gaussian noise with variance s_x^2 and mean zero. The vertical-horizontal ASHP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ are illustrated in Figure 8.9.

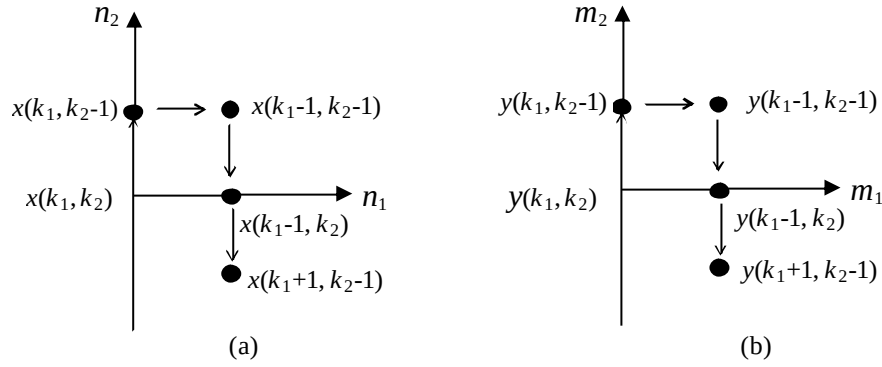


Figure 8-9: ASHP ordering schemes of the Computer Experiment 4 - ARMA(4,4) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.4.1. Objectives of Computer Experiment 4

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(4,4) parameter calculation formulas given in Equations (4.3) and (5.6), respectively, for the case which the AR order is equal to the MA order ($M=4$, $N=4$ and ASHP support) and the parameter b_0 is different than 1 ($b_0=0.38$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure and by use of L_1 , L_2 and L_∞ vector norms.

8.4.2. Results of Computer Experiment 4

The results of the Computer Experiment 4 have been given in Table 8.10 and Table 8.11 for different data sizes. Table 8.10 lists the original ARMA(4,4) parameters versus the identified ARMA(4,4) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.11 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for 50x50 data field size have been given in Figure 8.10 and Figure 8.11, respectively.

Table 8.10: Computer Experiment 4 - ARMA(4,4) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Original ARMA(4,4) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
AR Parameters	0.25	0.2029 0.2674	0.2372 0.2527	0.2556 0.2592
	0.2	0.1690 0.2011	0.1892 0.2016	0.2014 0.2048
	0.3	0.1839 0.3088	0.2786 0.3072	0.2960 0.3040
	0.4	0.3806 0.4260	0.3952 0.4000	0.4004 0.4011
MA Parameters	-0.3	-0.2934 -0.2930	-0.3016 -0.2984	-0.2975 -0.2966
	0.25	0.2840 0.2489	0.2546 0.2477	0.2498 0.2480
	-0.2	-0.2297 -0.1907	-0.2015 -0.1971	-0.1966 -0.1986
	0.1	0.1016 0.1079	0.0985 0.1004	0.1010 0.1013
b_0	0.38	0.3427 0.3299	0.3805 0.3799	0.3783 0.3791

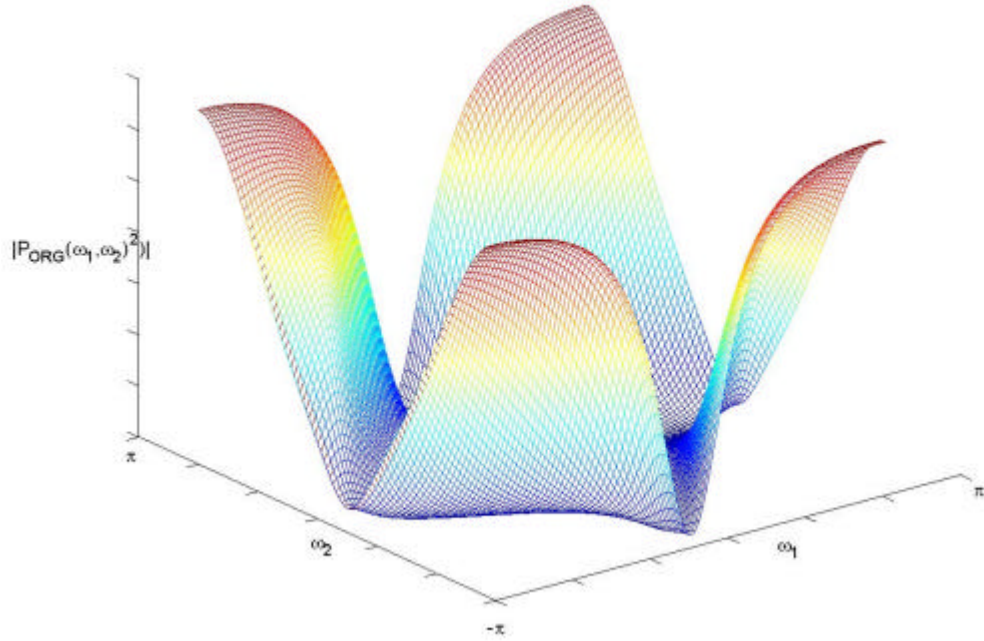
Table 8.11: Computer Experiment 4 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row..

Performance Criteria	Data Field Size		
	10x10	50x50	100x100
L₁-norm	0.3228	0.0597	0.0202
	0.1287	0.0186	0.0282
L₂-norm	0.1432	0.0281	0.0085
	0.0614	0.0088	0.0120
L_∞-norm	0.1161	0.0214	0.0056
	0.0501	0.0072	0.0092
Itakura-Saito Distance	0.183078	0.001002	0.000273
	0.094619	0.000081	0.000187

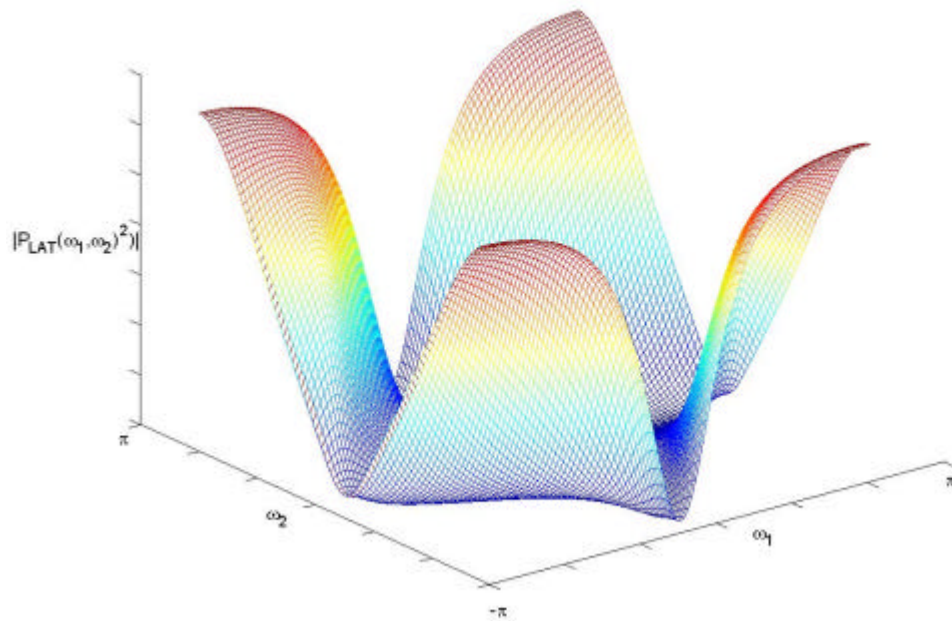
The obtained results yield the following conclusions for this experiment:

- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm for ASHP support compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm improve a great deal when the data size increases as it can easily be seen from Table 8.10 and Table 8.11.
- By looking at the Itakura-Saito distances listed in Table 8.11 and the power spectrum and contour plots given in Figure 8.10 and Figure 8.11, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.

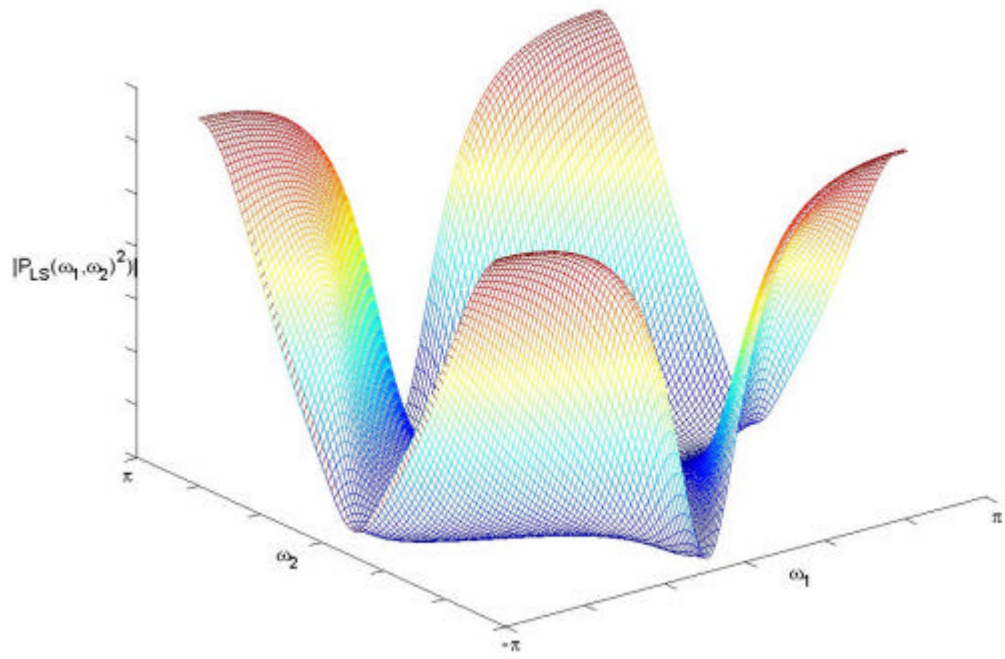
- Even for a small data field size of 10x10, ARMA(4,4) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.



(a)

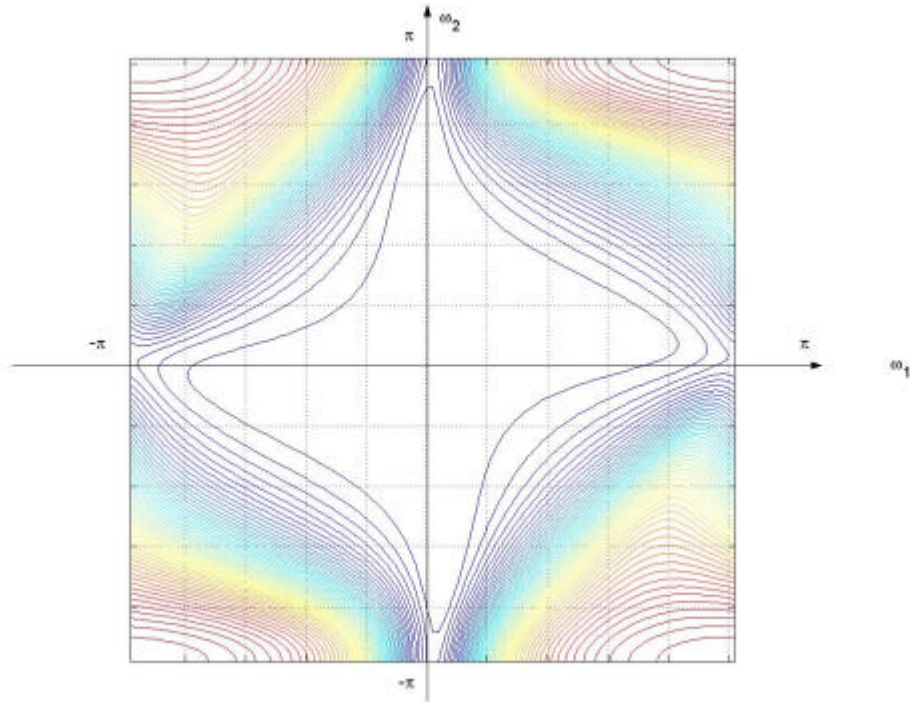


(b)

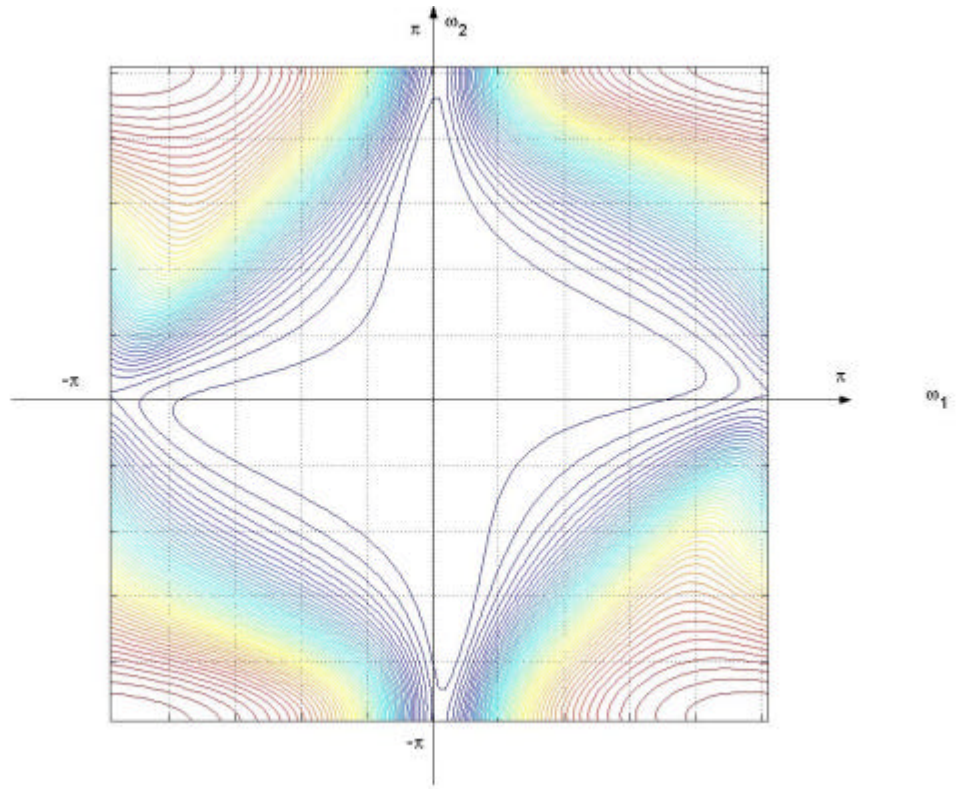


(c)

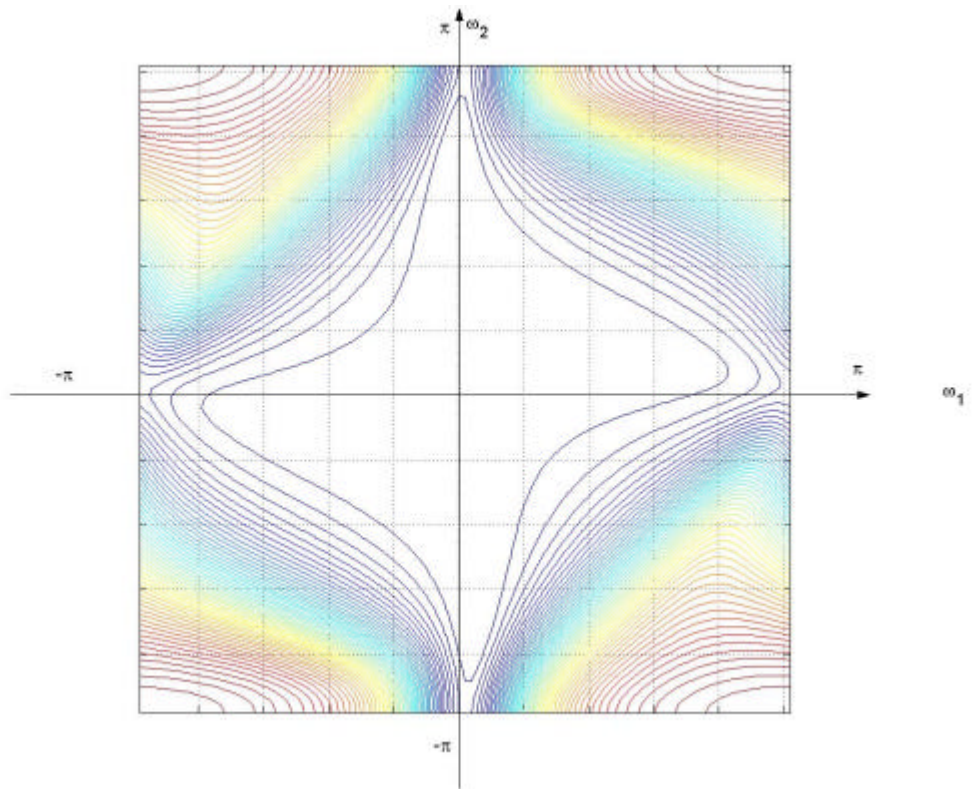
Figure 8-10: Computer Experiment 4- Power Spectrums plotted using (a) Original ARMA(4,4) parameters (b) ARMA(4,4) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(4,4) parameters identified by the LS algorithm. The parameter $b_0=0.38$ and the data field size is 50x50.



(a)



(b)



(c)

Figure 8-11: Computer Experiment 4 - Contour plots of (a) Original ARMA(4,4) parameters (b) ARMA(4,4) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(4,4) parameters identified by the LS algorithm. The parameter $b_0=0.38$ and the data field size is 50x50.

8.5. Computer Experiment 5

The unknown 2-D ARMA(8,8) system has a FQP support of the same order AR and MA polynomials ($M=8$ and $N=8$). The input $x(k_1, k_2)$ is a white noise process with mean zero and variances s_x^2 . The FQP horizontal-vertical ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=8$ and $N=8$ are illustrated in Figure 8.12.

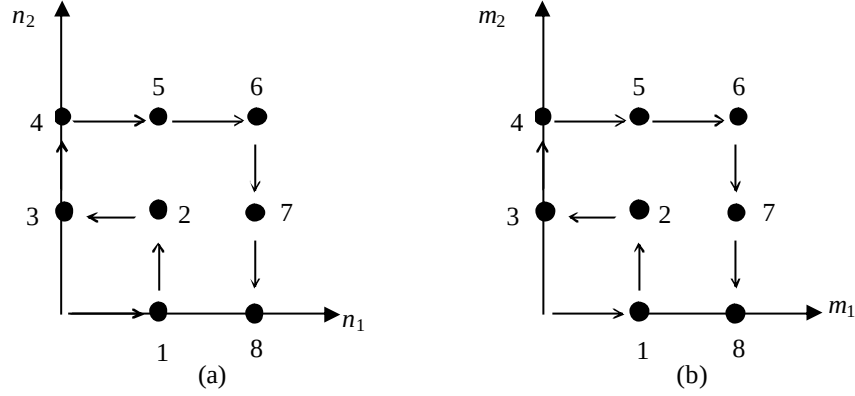


Figure 8-12: FQP ordering schemes of the Computer Experiment 5 - ARMA(8,8) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.5.1. Objectives of Computer Experiment 5

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(8,8) parameter calculation formulas given in Equations (4.3) and (5.6), respectively, for the case which the AR order is equal to the MA order ($M=8$, $N=8$ and FQP support) and the parameter b_0 is different than 1 ($b_0=0.1$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.5.2. Results of Computer Experiment 5

The results of the Computer Experiment 5 have been given in Table 8.12 and Table 8.13 for different data sizes. Table 8.12 lists the original ARMA(8,8) parameters versus the identified ARMA(8,8) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.13 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a 100x100 data field size have been given in Figure 8.13 and Figure 8.14, respectively.

Table 8.12: Computer Experiment 5 - ARMA(8,8) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Original ARMA(8,8) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
AR Parameters	0.1000	0.0565 0.0987	0.0849 0.0928	0.0948 0.0983
	0.1000	-0.1246 0.0538	0.0784 0.0971	0.0887 0.0977
	0.1000	0.2816 0.0217	0.1147 0.0869	0.1061 0.0940
	0.1000	0.0358 0.1304	0.0984 0.1046	0.0976 0.1023
	0.1000	0.2353 0.1170	0.1108 0.1001	0.1065 0.1003
	0.1000	0.0434 0.0701	0.0910 0.0957	0.0950 0.0979
	0.1000	0.1787 0.1414	0.1064 0.1030	0.1044 0.1019
	0.1000	0.1251 0.0626	0.1079 0.0981	0.1026 0.0985
	0.1000	0.0931 0.0989	0.0986 0.0993	0.0996 0.0999
	0.1000	0.0885 0.0881	0.0976 0.0979	0.0988 0.0990

Original ARMA(8,8) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
MA Parameters	0.1000	0.1161 0.0947	0.0993 0.0984	0.0999 0.0994
	0.1000	0.1035 0.0945	0.1005 0.0992	0.1001 0.0997
	0.1000	0.0858 0.0931	0.0979 0.0986	0.0990 0.0994
	0.1000	0.0938 0.0939	0.0989 0.0987	0.0994 0.0994
	0.1000	0.0940 0.0903	0.0987 0.0983	0.0995 0.0992
	0.1000	0.0966 0.0963	0.0993 0.0991	0.0996 0.0995
	0.1000	0.1013 0.1007	0.0992 0.09987	0.1002 0.1000
b_0	0.1000			

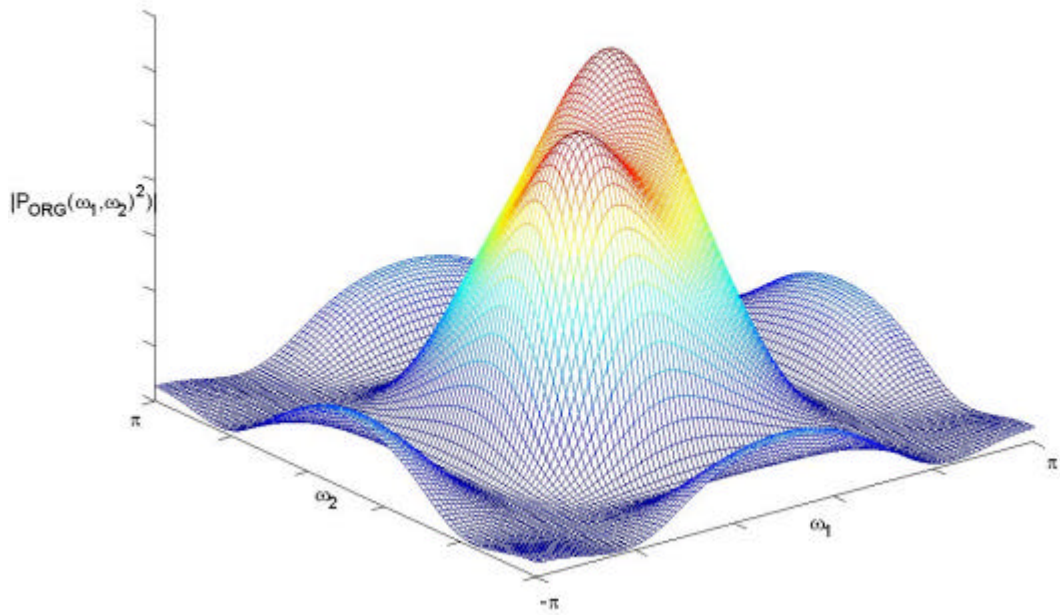
Table 8.13: Computer Experiment 5 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row..

Performance Criteria	Data Field Size		
	10x10	50x50	100x100
L_1 -norm	0.8787	0.0981	0.0480
	0.3328	0.0475	0.0225
L_2 -norm	0.3442	0.0350	0.0171
	0.1178	0.0173	0.0079
L_∞ -norm	0.2246	0.0216	0.0113
	0.0783	0.0131	0.0060
Itakura-Saito Distance	0.308972	0.05697	0.006214
	0.203778	0.01042	0.062238

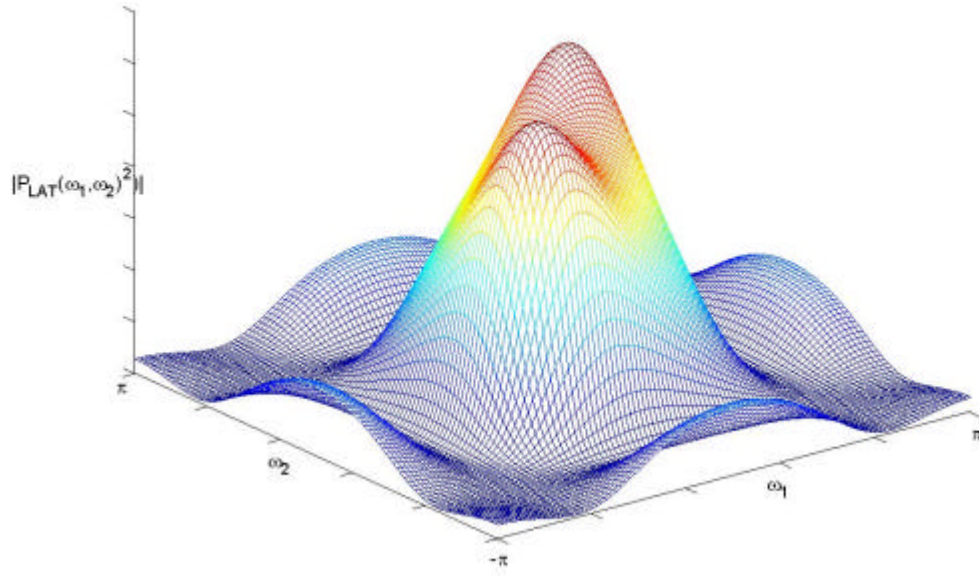
The obtained results yield the following conclusions for this experiment:

- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).

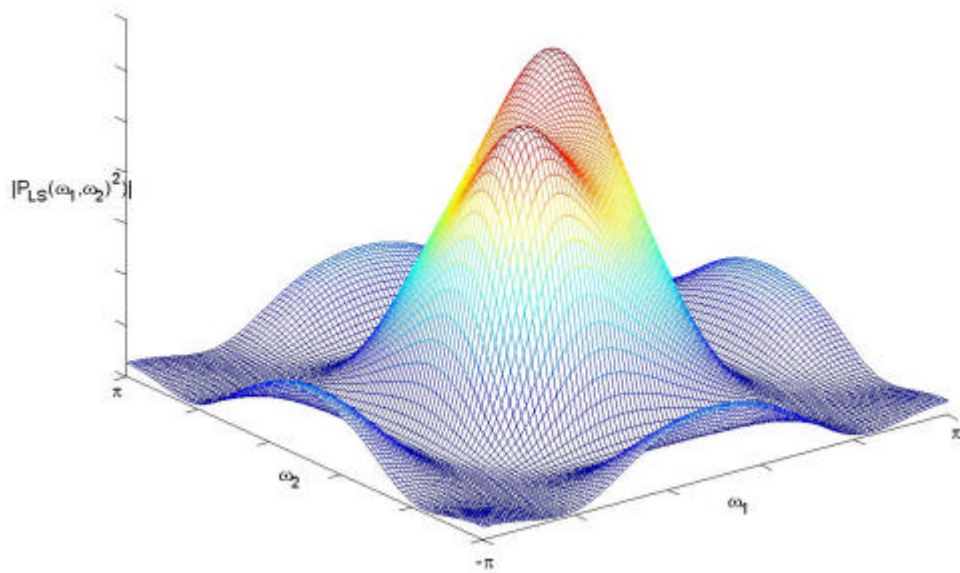
- The identification results obtained with the proposed 2-D hybrid lattice algorithm improve a great deal when the data size increases as it can easily be seen from Table 8.12 and Table 8.13.
- Even for a small data field size of 10x10, ARMA(8,8) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.
- By looking at the Itakura-Saito distances listed in Table 8.13 and the power spectrum and contour plots given in Figure 8.13 and Figure 8.14, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

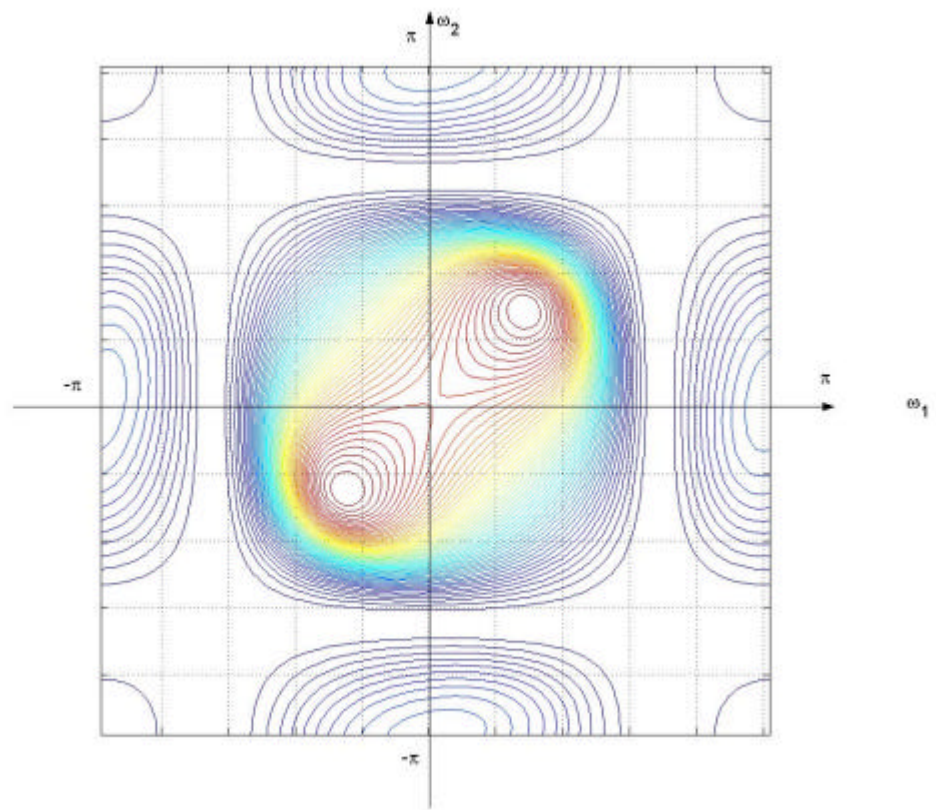


(b)

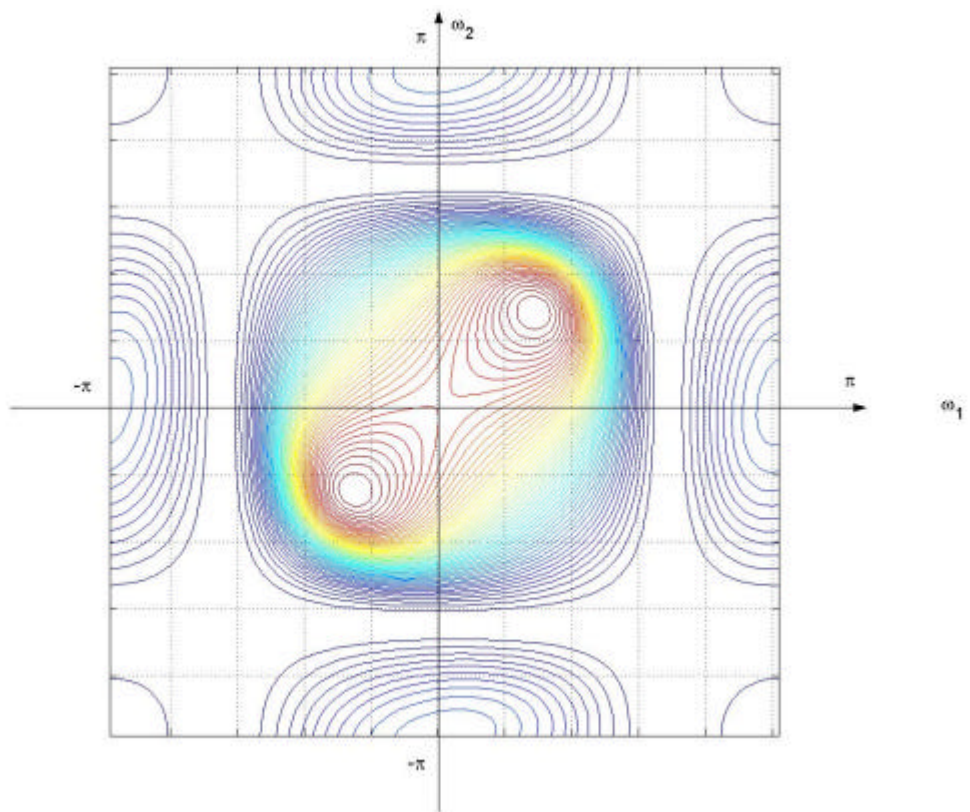


(c)

Figure 8-13: Computer Experiment 5 Power Spectrums plotted using (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.1$ and the data field size is 100x100.



(a)



(b)

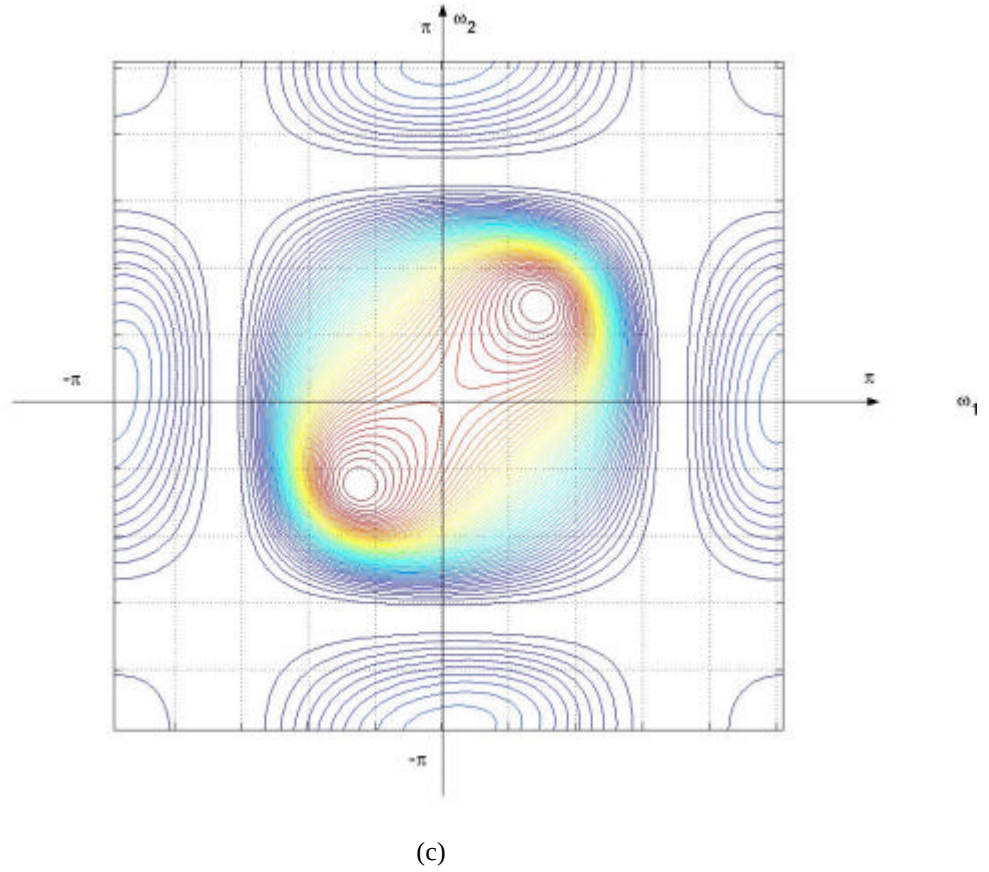


Figure 8-14: Computer Experiment 5 - Contour plots of (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.1$ and the data field size is 100x100.

8.6. Computer Experiment 6

The unknown 2-D ARMA(8,8) system has a FQP support of the same order AR and MA polynomials ($M=8$ and $N=8$). The input $x(k_1, k_2)$ is an AR process of order N . The horizontal-vertical FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=8$ and $N=8$ are illustrated in Figure 8.12.

8.6.1. Objectives of Computer Experiment 6

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(8,8) parameter calculation formulas given in Equations (4.3) and (5.6), respectively, for the case which the AR order is equal to the MA order ($M=8$, $N=8$ and QP support) and the parameter b_0 is different than 1 ($b_0=0.23$).

- To verify the validity of the proposed 2-D hybrid lattice algorithm for the case when the input signal $x(k_1, k_2)$ is different than white noise, for this experiment a 2-D AR process.
- To show that the AR parameters of the input signal $x(k_1, k_2)$ can be estimated simultaneously with the proposed 2-D hybrid lattice algorithm.
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.6.2. Results of Computer Experiment 6

The results of the Computer Experiment 6 have been given in Table 8.14- Table 8.17 for different data sizes. Table 8.14 lists the original ARMA(8,8) parameters versus the identified ARMA(8,8) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.15 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. Table 8.16 and Table 8.17 list the simultaneous AR parameter estimates of the input signal $x(k_1, k_2)$ and the related performance results, respectively. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a data field size 100x100 have been given in Figure 8.15 and Figure 8.16, respectively. The identified versus original AR power spectrums and contour plots of the AR system input have been given in Figure 8.17.

Table 8.14: Computer Experiment 6- ARMA(8,8) parameter identification results.

Original ARMA(8,8) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	35x35	100x100
AR Parameters	0.1000	0.2175 0.1701	0.1157 0.1201	0.1129 0.1063
	0.1100	0.1709 0.1292	0.1228 0.1209	0.1137 0.1150
	0.1230	0.1130 0.0686	0.0971 0.0821	0.1176 0.1141
	0.1370	0.1078 0.1290	0.1383 0.1308	0.1384 0.1352
	0.1450	0.1147 0.1033	0.1393 0.1243	0.1455 0.1395
	0.1600	0.1893 0.1810	0.1618 0.1667	0.1610 0.1633
	0.1200	0.1678 0.1516	0.1259 0.1300	0.1242 0.1236
	0.1000	0.1406 0.1468	0.1027 0.1213	0.1024 0.1053
MA Parameters	0.1400	0.1821 0.1584	0.1473 0.1491	0.1441 0.1427
	0.1500	0.1859 0.1574	0.1506 0.1492	0.1515 0.1508
	0.1600	0.1312 0.1501	0.1548 0.1546	0.1571 0.1587
	0.1700	0.1625 0.1661	0.1641 0.1641	0.1680 0.1685
	-0.1000	-0.0597 -0.1029	-0.0938 -0.1020	-0.0953 -0.1000
	-0.1100	-0.0974 -0.1031	-0.1069 -0.1049	-0.1093 -0.1081
	-0.1200	-0.0734 -0.0836	-0.1091 -0.1039	-0.1158 -0.1150
	-0.1300	-0.1086 -0.1105	-0.1273 -0.1224	-0.1275 -0.1273
b_0	0.2300	0.2087 0.2329	0.2214 0.2267	0.2258 0.2296

Table 8.15 Computer Experiment 6 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Performance Criteria	Data Field Size		
	10x10	35x35	100x100
L₁-norm	0.6219	0.1224	0.0583
	0.4011	0.1920	0.0559
L₂-norm	0.1816	0.0390	0.0182
	0.1263	0.0614	0.0166
L_∞-norm	0.1175	0.0259	0.0129
	0.0701	0.0409	0.0089
Itakura-Saito Distance	0.051115	0.005295	0.004884
	0.023006	0.009133	0.000470

Table 8.16: Computer Experiment 6 – Hybrid Lattice parameter estimates of the input AR (3) process $x(k_1, k_2)$.

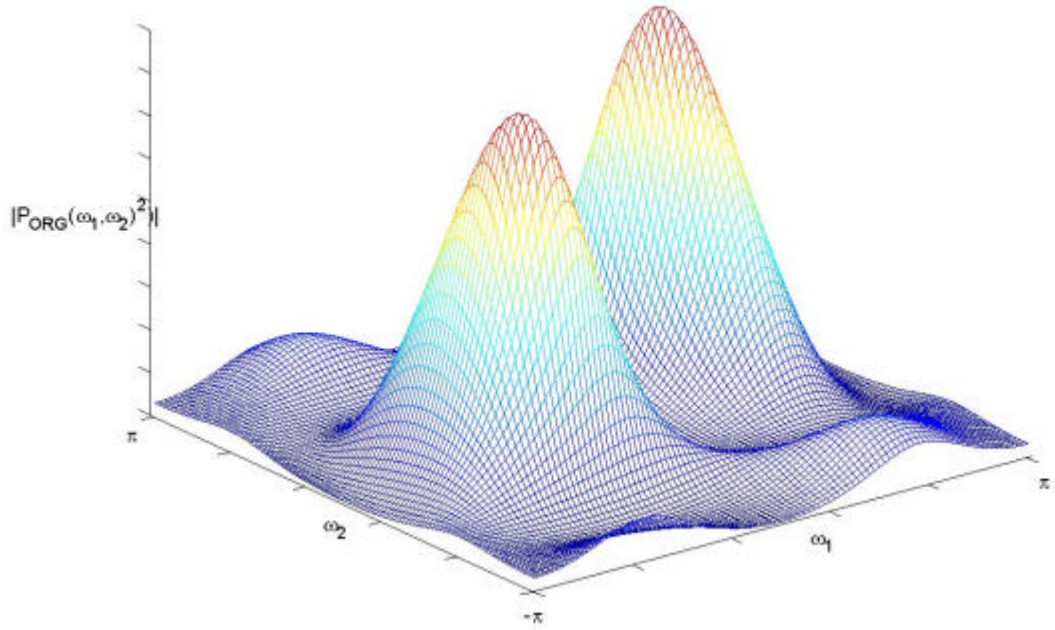
Original AR Parameters	Estimated AR Parameters (Mean Values)		
	Data Field Size		
	10x10	35x35	100x100
0.1000	0.1741	0.0643	0.1087
0.2000	0.7732	0.2165	0.2200
0.3000	0.4243	0.2940	0.3071
0.4000	0.3746	0.3563	0.4081
-0.4000	0.0810	-0.3828	-0.3822
-0.3000	-0.1171	-0.2287	-0.2805
-0.1000	-0.0362	-0.1238	-0.0855
-0.2000	0.0079	-0.2056	-0.1906

Table 8.17: Computer Experiment 6 - Performance criteria results in estimating the AR (3) parameters of $x(k_1, k_2)$.

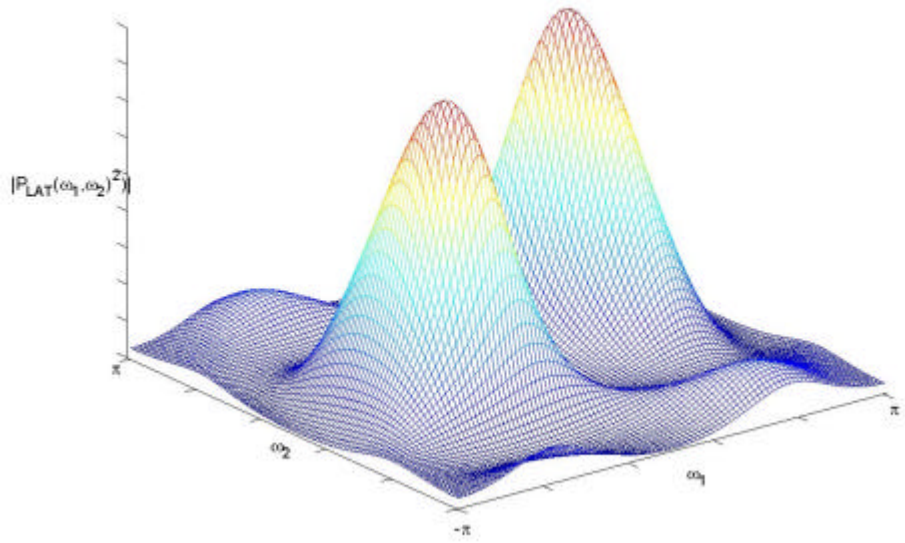
Performance Criteria	Data Field Size		
	10x10	35x35	100x100
L₁-norm	1.7327	0.2198	0.1051
L₂-norm	0.8138	0.0973	0.0399
L_∞-norm	0.5732	0.0713	0.0200
Itakura-Saito Distance	0.624233	0.020710	0.001886

The obtained results yield the following conclusions for this experiment:

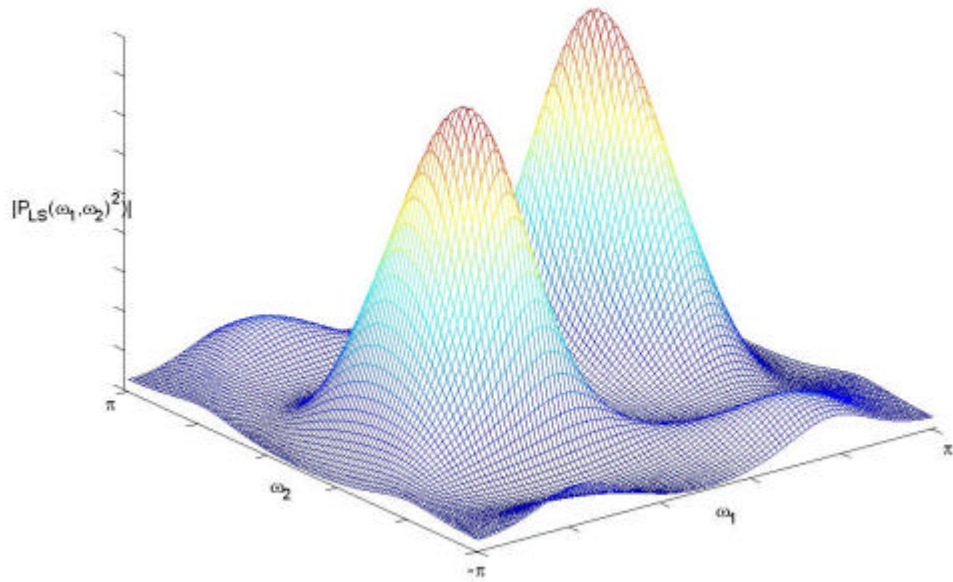
- The parameters of ARMA(8,8) system input $x(k_1, k_2)$ have been simultaneously identified with the proposed 2-D hybrid lattice algorithm provided that $x(k_1, k_2)$ is an AR process of order N .
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- By looking at the Itakura-Saito distances listed in Table 8.15 and Table 8.17 and the power spectrum and contour plots given in Figure 8.15-Figure 8.17, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

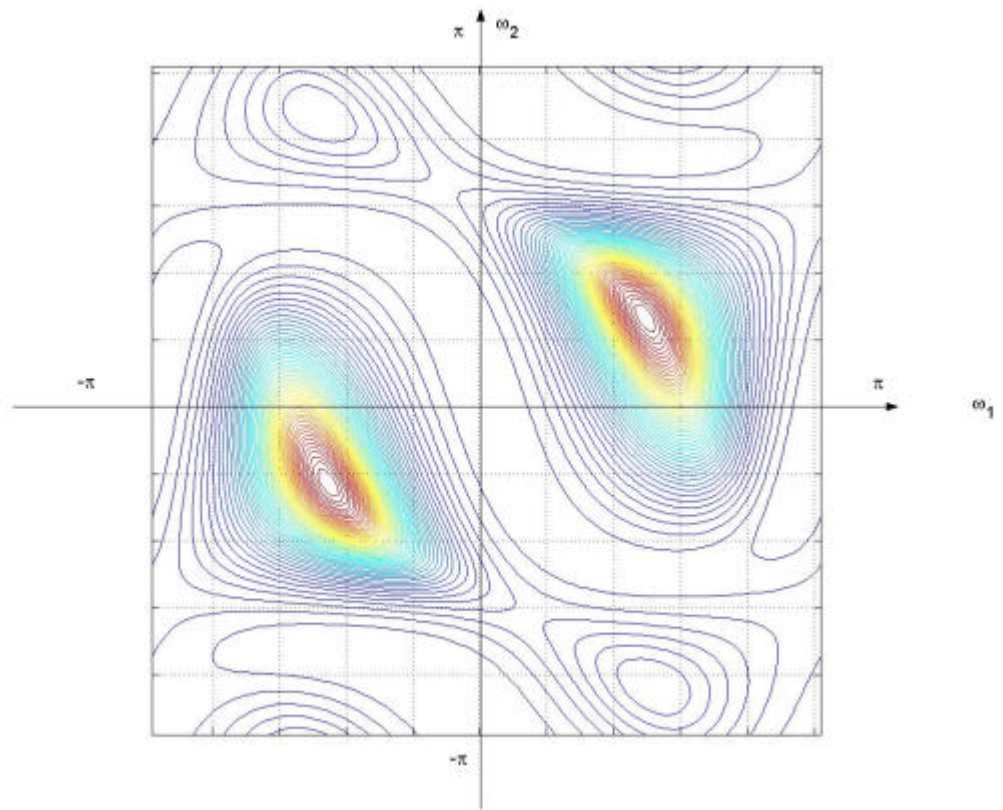


(b)

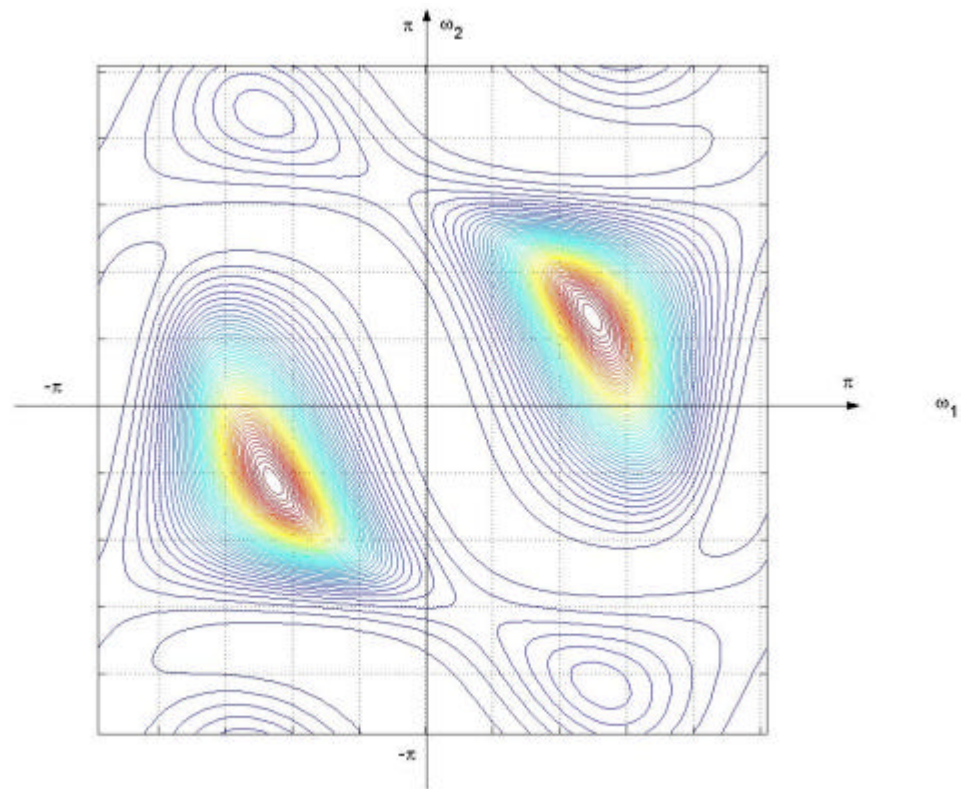


(c)

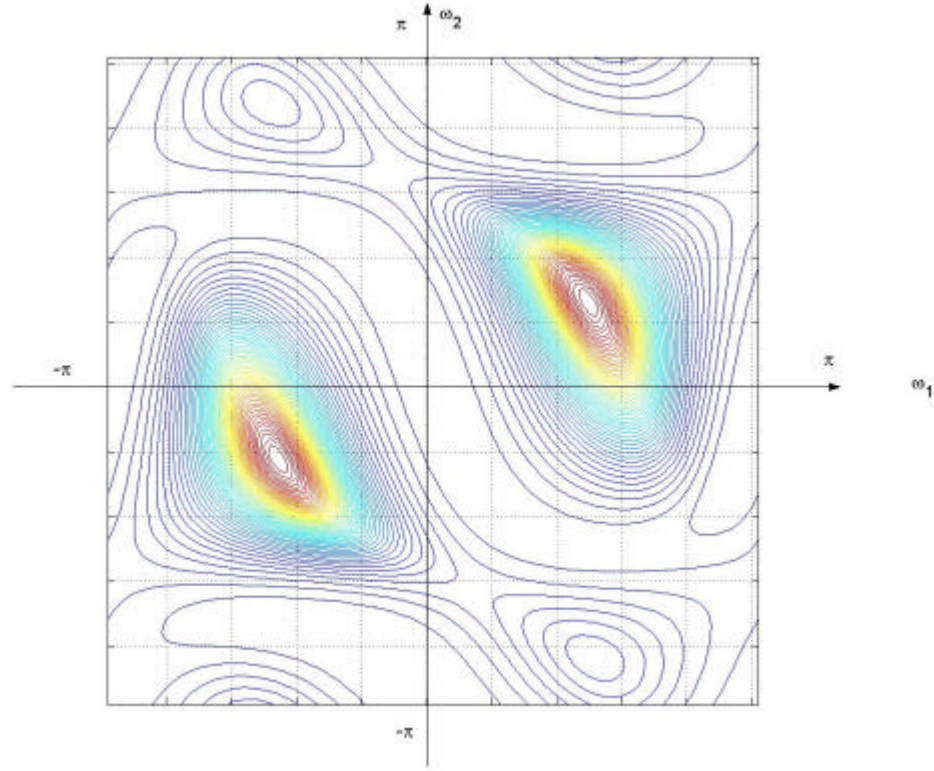
Figure 8-15: Computer Experiment 6 – Power Spectrums plotted using (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.23$ and the data field size is 100×100 . The system input $x(k_1, k_2)$ is an AR process of order 8.



(a)



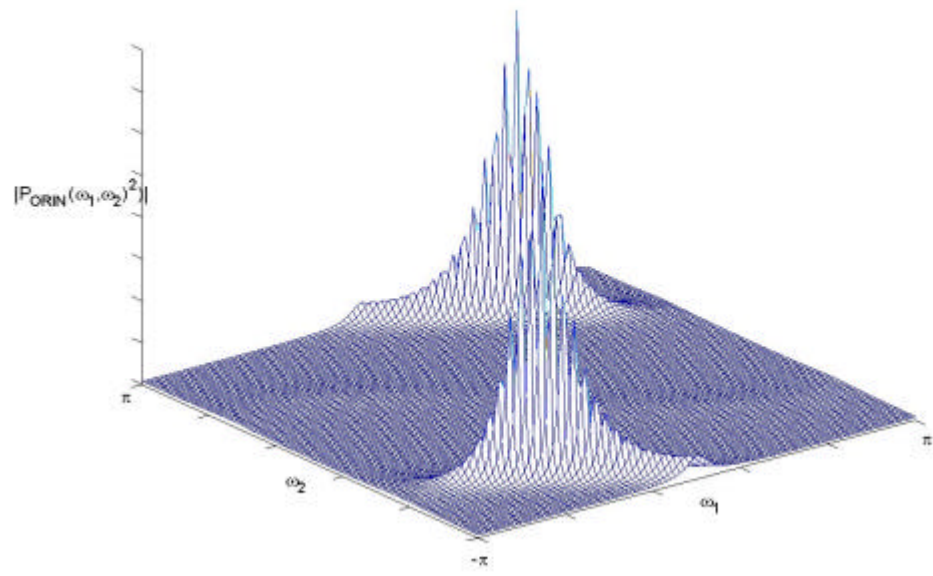
(b)



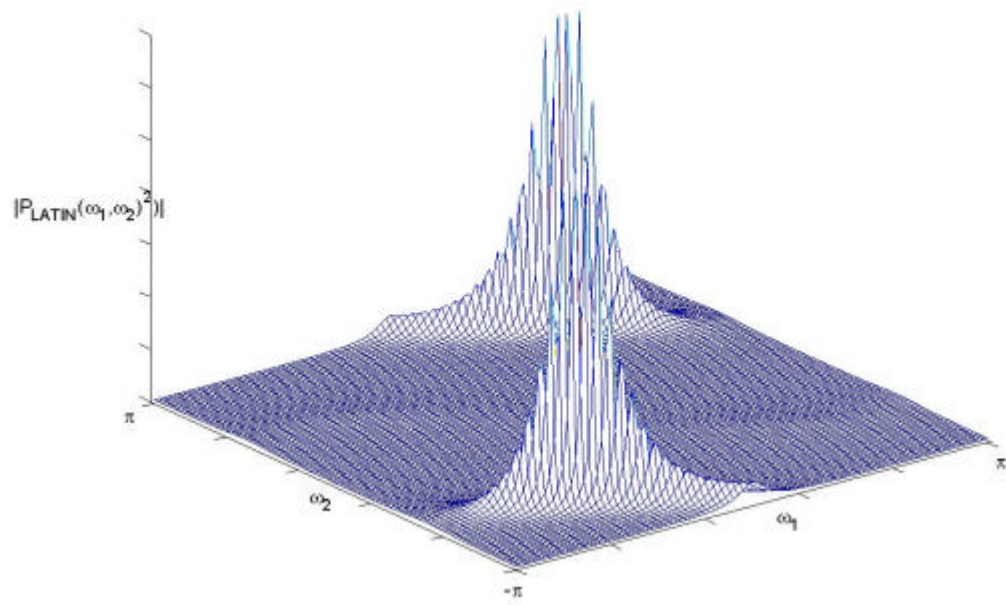
(c)

Figure 8-16: Computer Experiment 6 – Contour plots of (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.23$ and the data field size is 100x100. The system input $x(k_1, k_2)$ is an AR process of order 8.

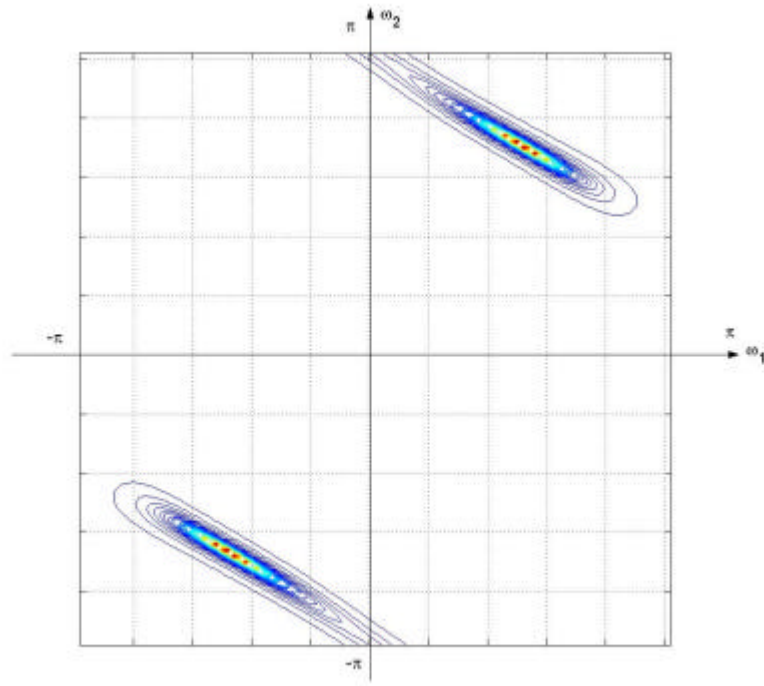
- As it can easily be seen from Table 8.14-Table 8.17, the identification results are quite accurate and improve with the increasing data size.
- Even for a small data field size of 10x10, ARMA(8,8) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.



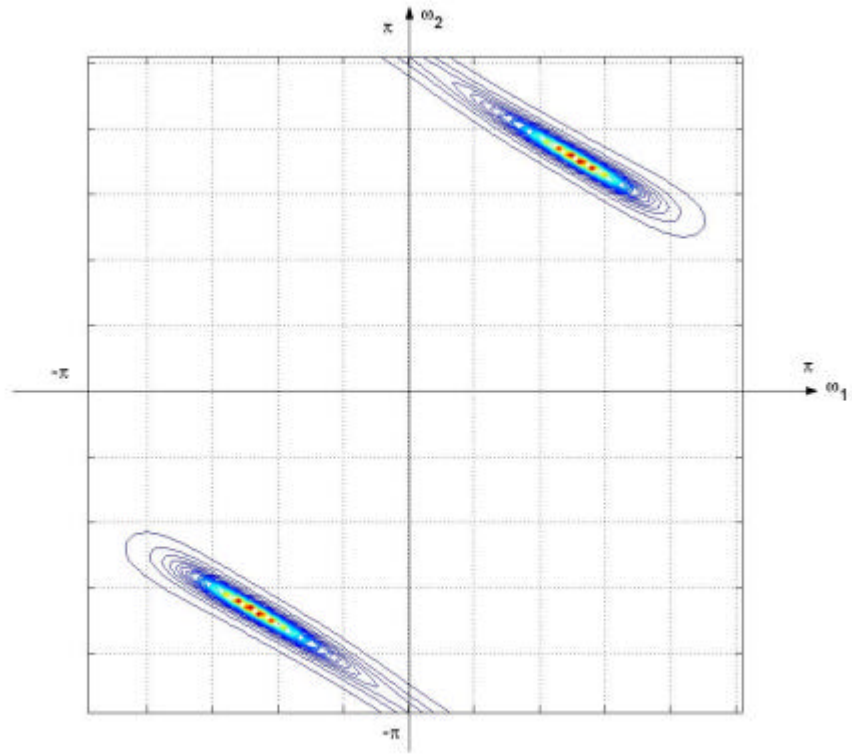
(a)



(b)



(c)



(d)

Figure 8-17: Computer Experiment 6 -(a) Original power spectrum of the input AR(8) data field $x(k_1, k_2)$ (b) Power spectrum of $x(k_1, k_2)$ plotted using parameters identified by the Hybrid Lattice algorithm (c) Original contour plot of $x(k_1, k_2)$ (d) Identified contour plot of $x(k_1, k_2)$.

8.7. Computer Experiment 7

The unknown 2-D ARMA(8,8) system has a FQP support of the same order AR and MA polynomials ($M=8$ and $N=8$). The input $x(k_1, k_2)$ is a white noise process with mean zero and variance s_x^2 . The column-wise FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=8$ and $N=8$ are illustrated in Figure 8.18.

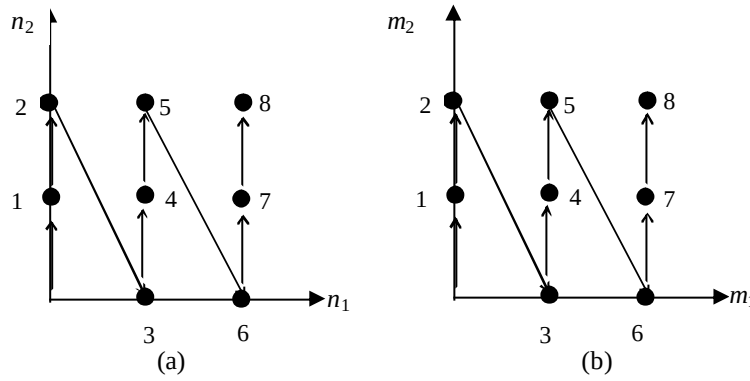


Figure 8-18: FQP ordering schemes of the Computer Experiment 7 - ARMA(8,8) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.7.1. Objectives of Computer Experiment 7

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(8,8) parameter calculation formulas given in Equations (4.3) and (5.6), respectively, for the case which the AR order is equal to the MA order ($M=8, N=8$, QP support and a different ordering scheme) and the parameter b_0 is different than 1 ($b_0=0.22$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.

- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.7.2. Results of Computer Experiment 7

The results of the Computer Experiment 7 have been given in Table 8.18 and Table 8.19 for different data sizes. Table 8.18 lists the original ARMA(8,8) parameters versus the identified ARMA(8,8) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.19 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a 10x10 data field size have been given in Figure 8.19 and Figure 8.20, respectively.

Table 8.18: Computer Experiment 7 - ARMA(8,8) parameter identification results.

Original ARMA(8,8) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	35x35	100x100
AR Parameters	-0.1250	-0.1250 -0.1075	-0.1250 -0.1206	-0.1250 -0.1233
	0.1970	0.1970 0.1920	0.1970 0.1970	0.1970 0.1970
	-0.1000	-0.1000 -0.1315	-0.1000 -0.1056	-0.1000 -0.1023
	-0.1100	-0.1100 -0.1112	-0.1100 -0.1109	-0.1100 -0.1103
	-0.1400	-0.1400 -0.1516	-0.1400 -0.1423	-0.1400 -0.1409
	-0.1950	-0.1950 -0.2111	-0.1950 -0.1964	-0.1950 -0.1954
	-0.1220	-0.1220 -0.1139	-0.1220 -0.1215	-0.1220 -0.1217
	-0.1030	-0.1030 -0.1112	-0.1030 -0.1036	-0.1030 -0.1031
MA Parameters	-0.1930	-0.1930 -0.1930	-0.1930 -0.1918	-0.1930 -0.1926

Original ARMA(8,8) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	35x35	100x100
	0.1460	0.1460 0.1413	0.1460 0.1452	0.1460 0.1457
	-0.1280	-0.1280 -0.1327	-0.1280 -0.1293	-0.1280 -0.1286
	-0.2300	-0.2300 -0.2280	-0.2300 -0.2298	-0.2300 -0.2298
	0.2120	0.2120 0.2019	0.2120 0.2100	0.2120 0.2113
	0.2090	0.2090 0.2104	0.2090 0.2107	0.2090 0.2094
	-0.1440	-0.1440 -0.1281	-0.1440 -0.1412	-0.1440 -0.1429
	-0.1870	-0.1870 -0.2030	-0.1870 -0.1887	-0.1870 -0.1877
b_0	0.2200	0.2200 0.2194	0.2200 0.2190	0.2200 0.2200

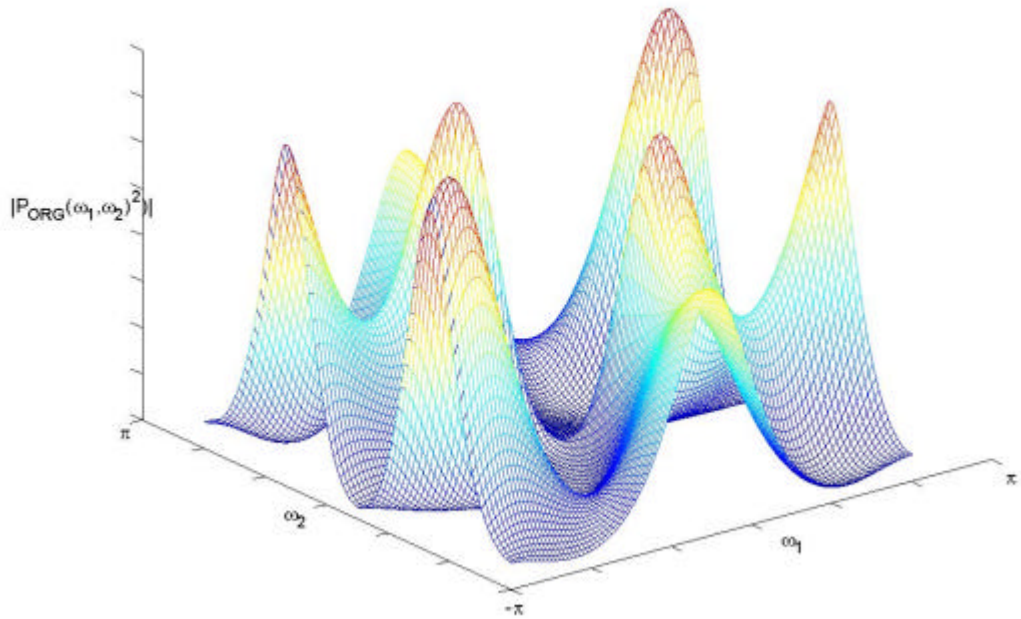
Table 8.19: Computer Experiment 7 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Performance Criteria	Data Field Size		
	10x10	35x35	100x100
L_1 -norm	0.0000	0.0000	0.0000
	0.1548	0.0283	0.0106
L_2 -norm	0.0000	0.0000	0.0000
	0.0502	0.0090	0.0035
L_∞ -norm	0.0000	0.0000	0.0000
	0.0315	0.0056	0.0023
Itakura-Saito Distance	0.000000	0.000000	0.000000
	0.004534	0.000239	0.000021

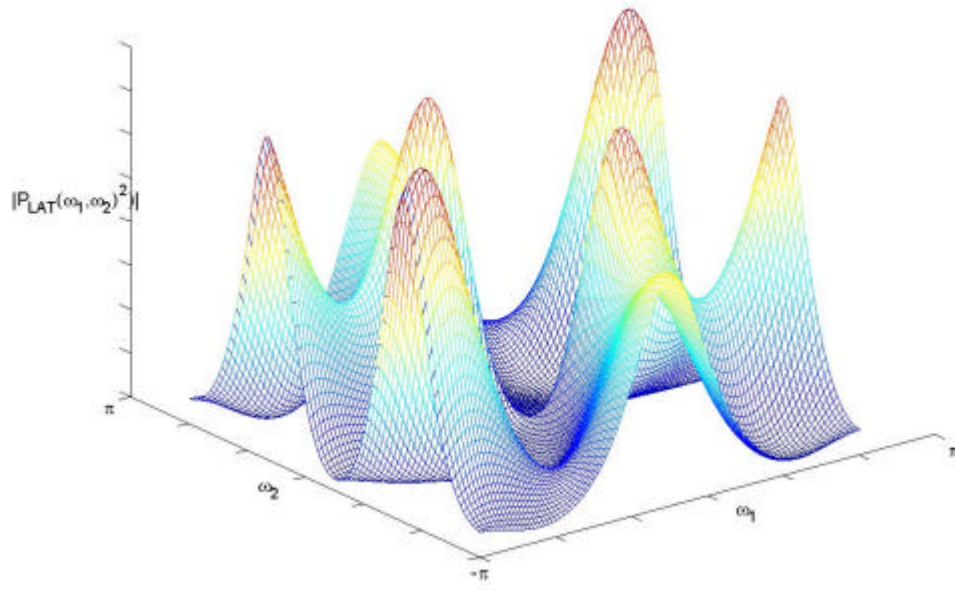
The obtained results yield the following conclusions for this experiment:

- The parameters of ARMA(8,8) system input $x(k_1, k_2)$ have been simultaneously identified with the proposed 2-D hybrid lattice algorithm provided that $x(k_1, k_2)$ is an AR process of order N .

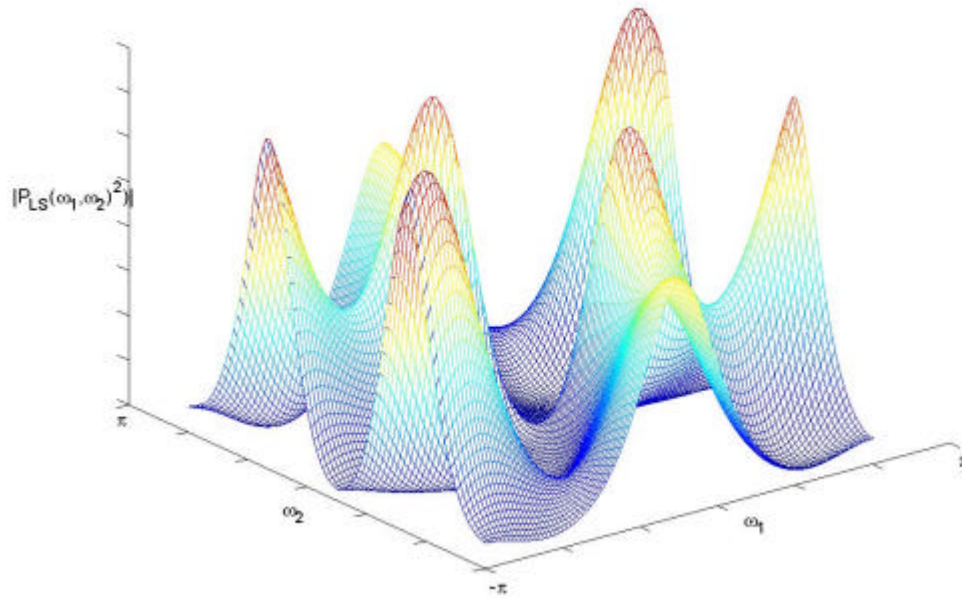
- As it can easily be seen from Table 8.18-Table 8.19, the identification results are quite accurate and improve with the increasing data size.
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- Even for a small data field size of 10x10, ARMA(8,8) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.
- By looking at the Itakura-Saito distances listed in Table 8.19 and the power spectrum and contour plots given in Figure 8.19 and Figure 8.20, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

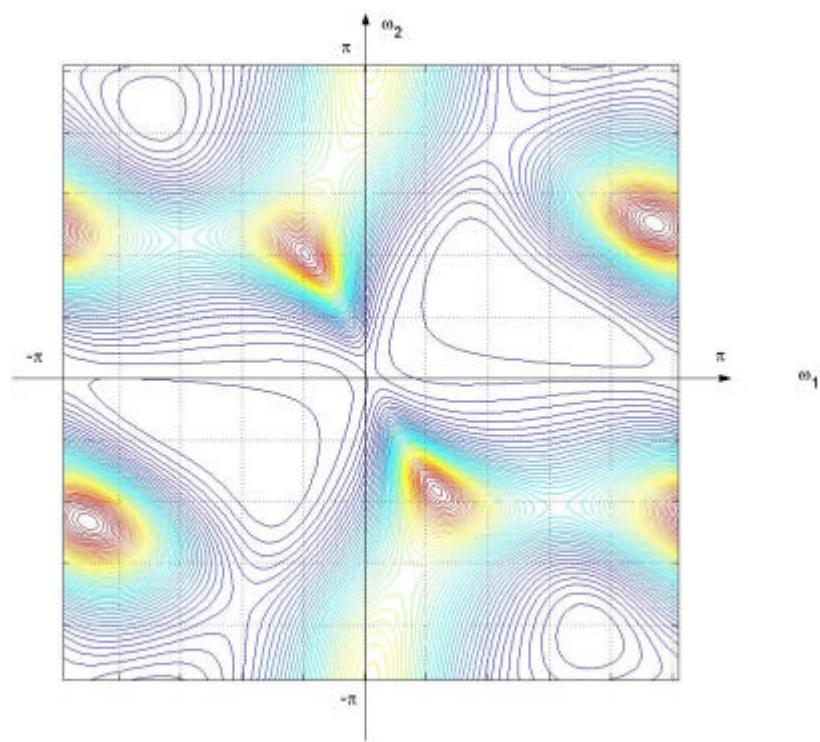


(b)

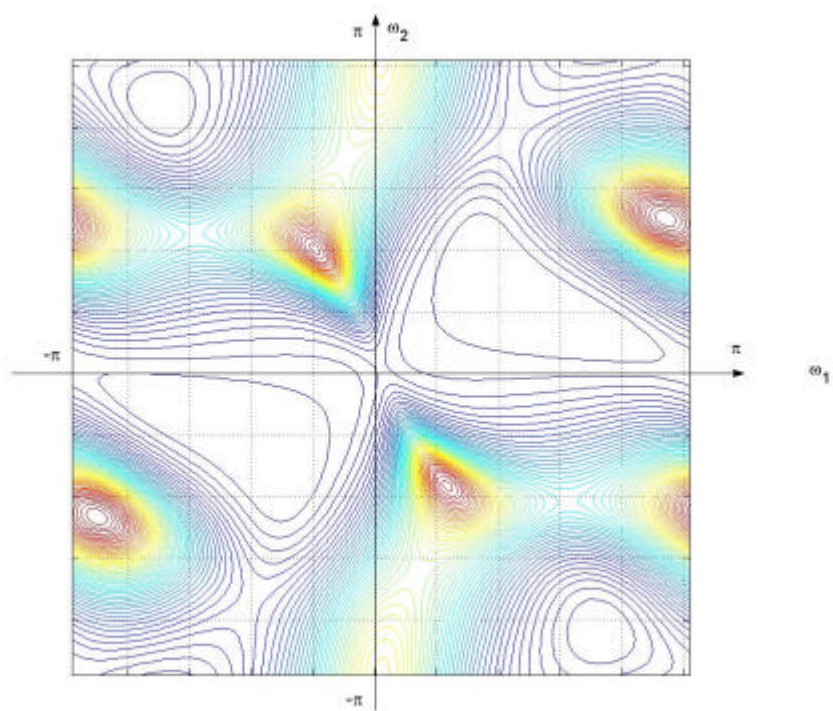


(c)

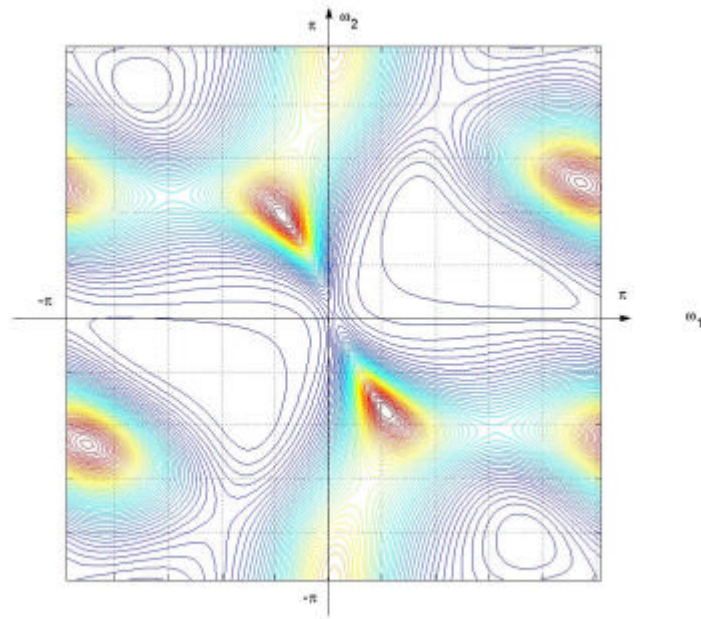
Figure 8-19: Computer Experiment 7 – Power Spectrums plotted using (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10×10 .



(a)



(b)



(c)

Figure 8-20: Computer Experiment 7 – Contour plots of (a) Original ARMA(8,8) parameters (b) ARMA(8,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,8) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10×10 .

8.8. Computer Experiment 8

The unknown 2-D ARMA(15,15) system has a FQP support of the same order AR and MA polynomials ($M=15$ and $N=15$). The input $x(k_1, k_2)$ is a white noise process with mean zero and variances σ_x^2 . The horizontal-vertical FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=15$ and $N=15$ are illustrated in Figure 8.21.

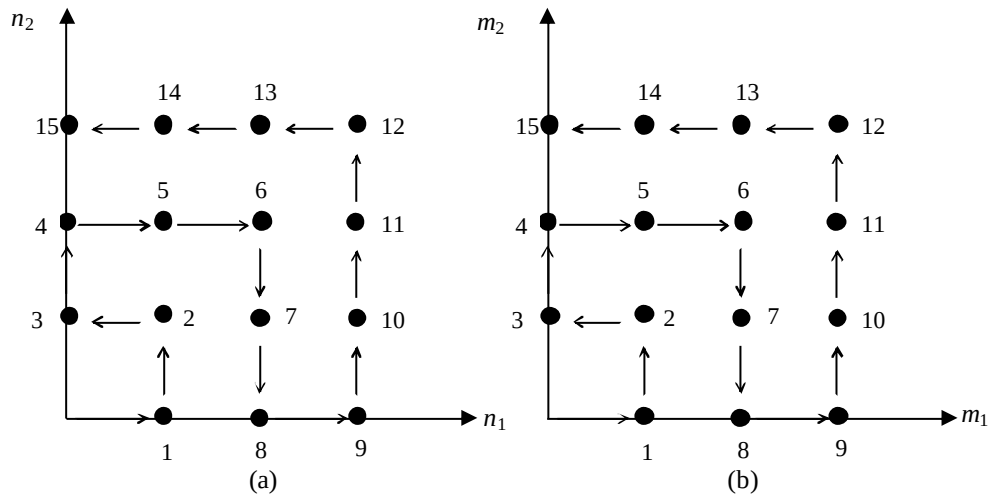


Figure 8-21: FQP ordering schemes of the Computer Experiment 8 - ARMA(15,15) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.8.1. Objectives of Computer Experiment 8

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(15,15) parameter calculation formulas given in Equations (4.3) and (5.6), respectively, for the case which the AR order is equal to the MA order ($M=8$, $N=8$ and QP support) and the parameter b_0 is different than 1 ($b_0=0.25$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.8.2. Results of Computer Experiment 8

Table 8.20: Computer Experiment 8 - ARMA(15,15) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the third row. Hybrid Lattice results with no b_0 correction are shown in the second row.

Original ARMA(15,15) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
AR Parameters	0.0715	0.0581	0.0658	0.0666
		0.4395	0.6603	-0.26159
		0.0452	0.0687	0.0700
	0.2770	0.2996	0.2673	0.2740
		0.7818	0.7784	0.28315
		0.2635	0.2705	0.2752
	-0.0275	-0.1619	-0.0549	-0.0384
		-2.9352	0.0085	0.0267
		-0.1905	-0.0533	-0.0374
	0.0018	-0.0126	-0.0001	0.0002
		0.3742	0.4121	-0.11346
		-0.0158	0.0001	0.0003

Original ARMA(15,15) Parameters	Identified ARMA Parameters (Mean Values)			
	Data Field Size			
	10x10	50x50	100x100	
-0.0946	-0.1540	-0.1053	-0.1014	
	-1.5009	-0.1501	-0.16342	
	-0.1950	-0.1052	-0.1008	
	0.1199	0.0915	0.0935	
	0.8647	0.7073	-0.028131	
	0.1030	0.0933	0.0940	
	-0.0300	-0.0359	-0.0335	
	-2.5342	-0.0434	-0.18892	
	-0.0183	-0.0333	-0.0309	
	0.0657	0.0505	0.0484	
	1.9323	0.2917	0.10452	
	0.0432	0.0476	0.0472	
	-0.0778	-0.0864	-0.0857	
	-1.8778	-0.1403	-0.23276	
	-0.0798	-0.0843	-0.0855	
0.0422	0.0407	0.0420	0.0431	
	1.1416	0.0275	0.073064	
	0.0592	0.0419	0.0424	
	-0.1082	-0.0824	-0.0821	
	-2.7745	-0.0809	-0.21455	
	-0.1065	-0.0813	-0.0808	
	-0.0336	-0.0528	-0.0517	
	0.5563	0.2980	-0.14925	
	-0.0500	-0.0510	-0.0506	
	0.0034	0.0792	0.0857	
	-0.4701	-0.0651	0.2279	
	-0.0347	0.0799	0.0852	
	-0.0712	-0.0947	-0.0982	
	0.1761	0.3395	-0.18327	
	-0.0797	-0.0943	-0.0978	
0.2276	0.2025	0.2290	0.2263	
	-1.6072	0.1821	0.31996	
	0.2145	0.2277	0.2260	
	-0.0437	-0.0410	-0.0411	
	-0.1074	-0.1270	0.1168	
	-0.0466	-0.0402	-0.0403	

Original ARMA(15,15) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
MA Parameters	0.0030	0.0162 -0.2832 0.0078	0.0025 -0.1411 0.0032	0.0029 0.0071 0.0032
	0.0025	-0.0334 0.7787 -0.0370	-0.0045 -0.0089 -0.0040	-0.0003 -0.0256 -0.0000
	0.0230	0.0204 -0.2101 0.0196	0.0228 -0.1339 0.0229	0.0227 0.0032 0.0227
	-0.0230	-0.0287 0.1854 -0.0373	-0.0246 0.0631 -0.0245	-0.0243 0.0635 -0.0241
	0.0110	0.0193 -0.3176 0.0186	0.0112 -0.1476 0.0114	0.0115 0.0151 0.0115
	-0.0110	-0.0149 0.6922 -0.0094	-0.0117 0.0810 -0.0114	-0.0115 0.0126 -0.0112
	0.0070	0.0146 -0.4882 0.0055	0.0084 -0.0391 0.0074	0.0076 -0.0415 0.0071
	-0.0070	-0.0029 0.4298 -0.0058	-0.0074 0.0155 -0.0066	-0.0071 0.0387 -0.0069
	0.0010	0.0044 -0.1620 0.0106	0.0097 -0.0045 0.0097	0.0100 -0.0176 0.0100
	-0.0230	-0.0314 0.5349 -0.0314	-0.0234 0.0255 -0.0232	-0.0234 0.0382 -0.0232
	0.0230	0.0266 -0.1113 0.0271	0.0231 -0.0733 0.0234	0.0229 0.0075 0.0232
	0.0400	0.0233 0.1329 0.0209	0.0380 0.0371 0.0381	0.0394 -0.0728 0.0393

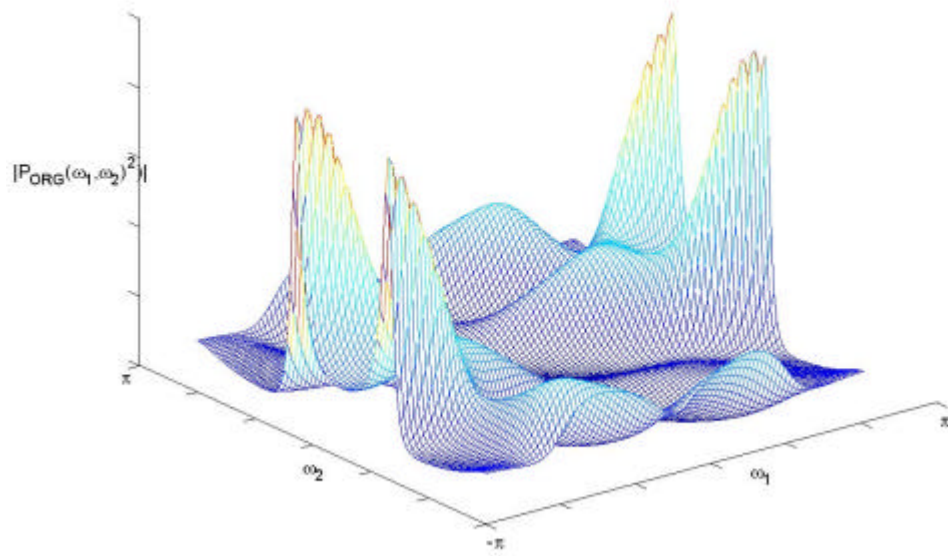
Original ARMA(15,15) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
	-0.0400	-0.0322	-0.0400	-0.0402
		-0.0805	-0.0481	0.0437
		-0.0372	-0.0399	-0.0402
	0.0210	0.0131	0.0211	0.0207
		0.6452	-0.0108	-0.0406
		0.0142	0.0209	0.0206
b_0	0.25	0.2489	0.2498	0.2500
		0.2492	0.2498	0.2499

Table 8.21: Computer Experiment 8 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the third row. Hybrid Lattice results with no b_0 correction are shown in the second row.

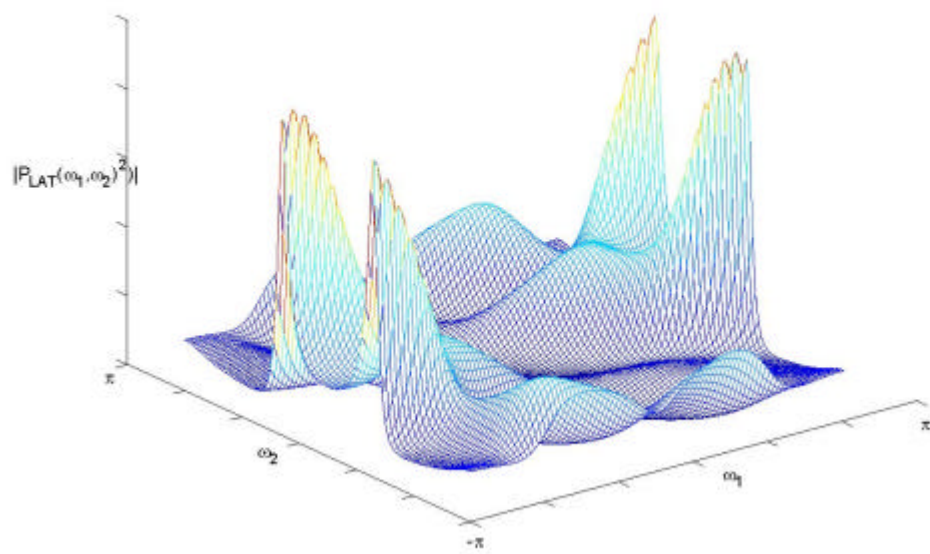
Performance Criteria	Data Field Size		
	10x10	50x50	100x100
L_1 -norm	0.6195	0.1039	0.0515
	24.7467	4.4657	2.0241
	0.6747	0.0820	0.0384
L_2 -norm	0.1889	0.0354	0.0161
	6.3019	1.2852	0.5248
	0.2398	0.0318	0.0134
L_∞ -norm	0.1344	0.0275	0.0109
	2.9077	0.6138	0.3331
	0.1631	0.0258	0.0099
Itakura-Saito Distance	0.003095	0.000100	0.000015
	0.464006	0.210280	0.045581
	0.003678	0.000086	0.000012

The results of the Computer Experiment 8 have been given in Table 8.20 and Table 8.21 for different data sizes. Table 8.20 lists the original ARMA(15,15) parameters versus the identified ARMA(15,15) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.21 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original

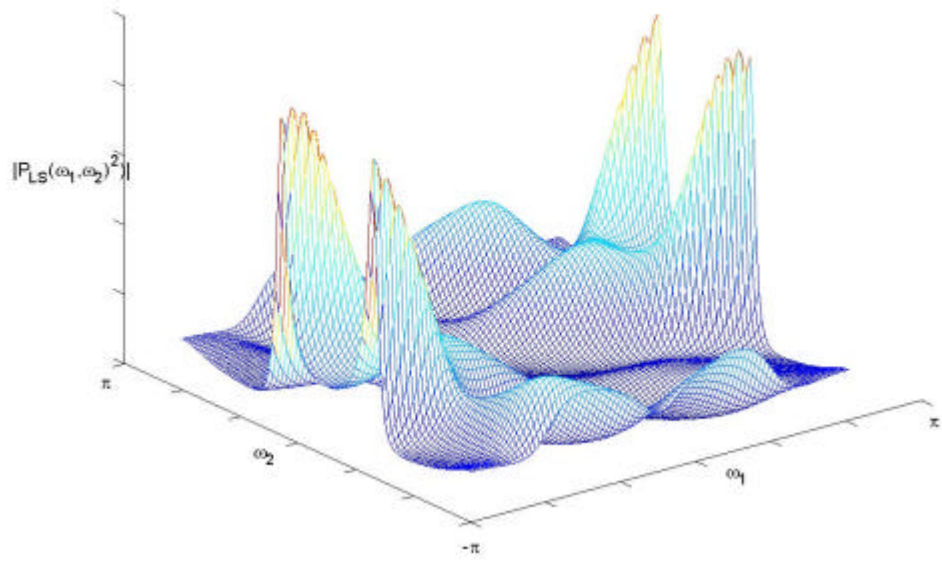
parameters versus the Lattice and LS identified parameters for a 50x50 data field size have been given in Figure 8.22-Figure 8.24.



(a)

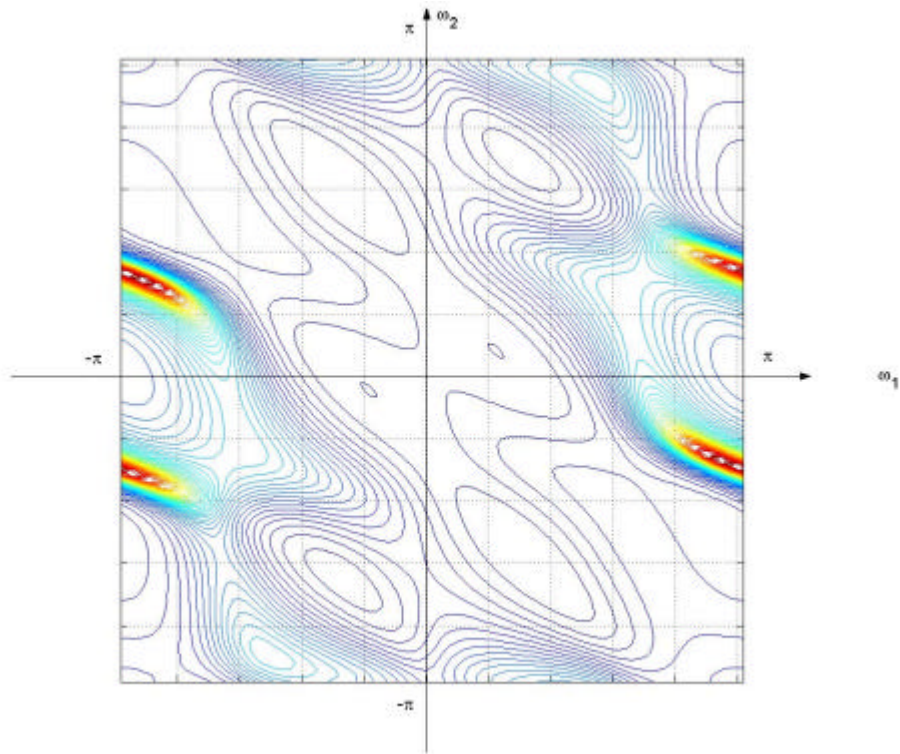


(b)

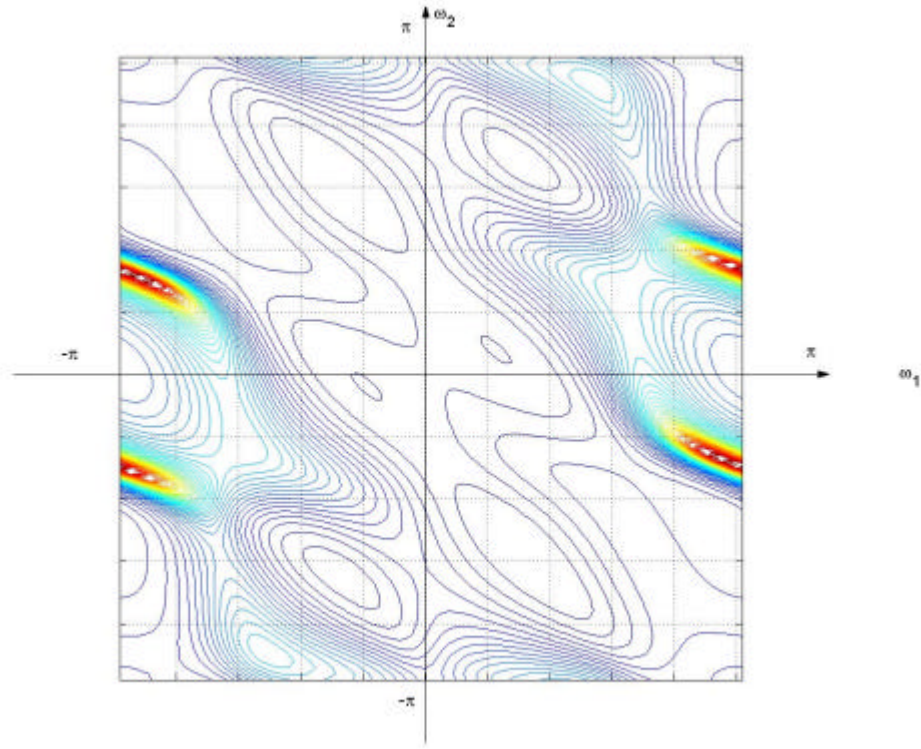


(c)

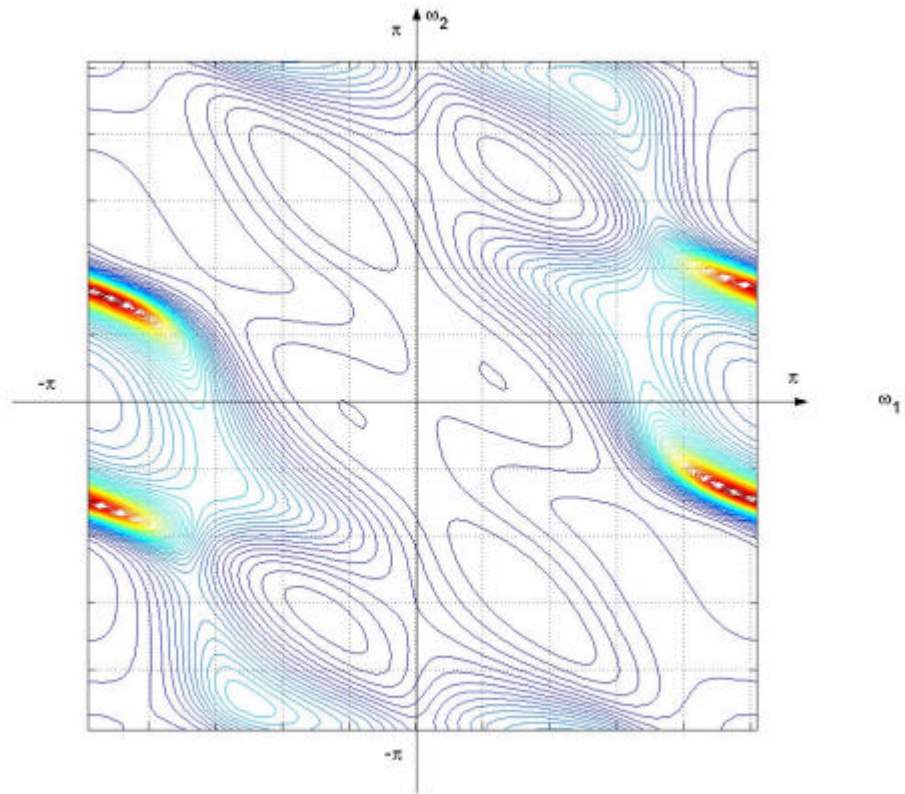
Figure 8-22: Computer Experiment 8 – Power Spectrums plotted using (a) Original ARMA(15,15) parameters (b) ARMA(15,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,15) parameters identified by the LS algorithm. The parameter $b_0=0.25$ and the data field size is 50x50.



(a)



(b)

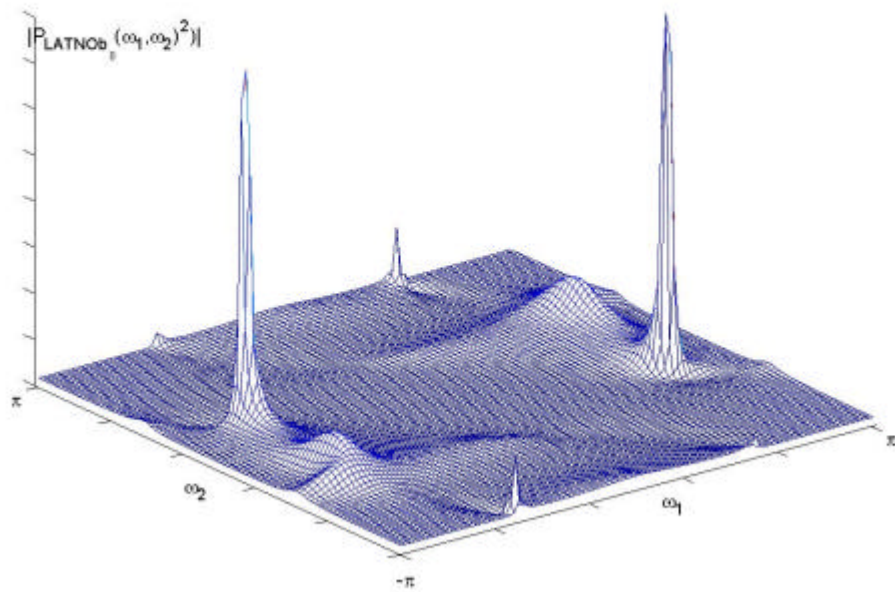


(c)

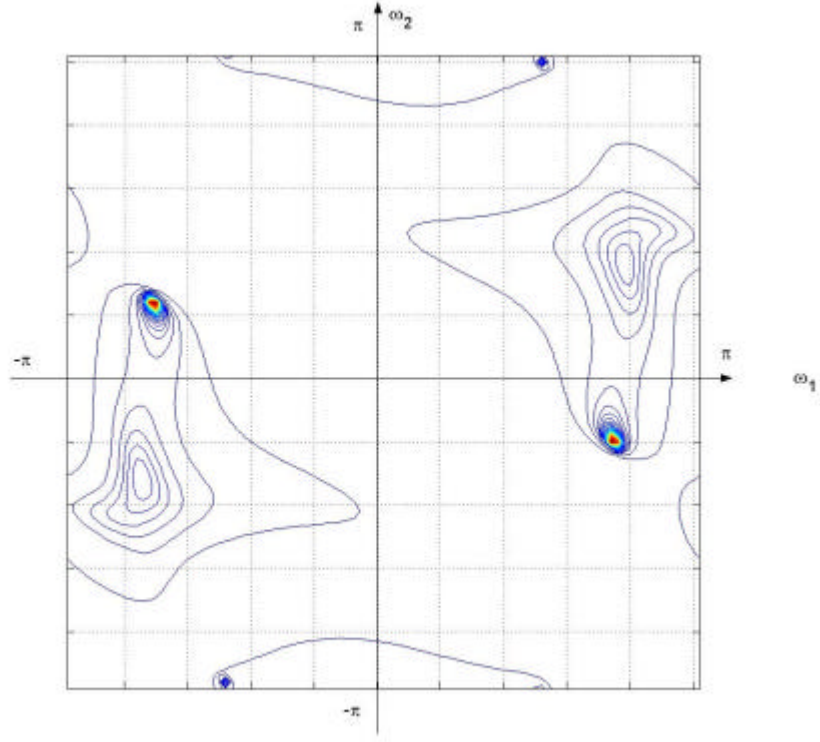
Figure 8-23: Computer Experiment 8 – Contour plots of (a) Original ARMA(15,15) parameters (b) ARMA(15,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,15) parameters identified by the LS algorithm. The parameter $b_0=0.25$ and the data field size is 50x50.

The obtained results yield the following conclusions for this experiment:

- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm improve a great deal when the data size increases as it can easily be seen from Table 8.20 and Table 8.21.
- Even for a small data field size of 10×10 , ARMA(15,15) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.
- By looking at the Itakura-Saito distances listed in Table 8.21 and the power spectrum and contour plots given in Figure 8.22 and Figure 8.23, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)



(b)

Figure 8-24: Computer Experiment 8 – (a) Power spectrum plotted using ARMA(15,15) parameters identified by the Hybrid Lattice algorithm with no b_0 correction (b) the related ARMA(15,15) contour plot.

8.9. Computer Experiment 9

The unknown 2-D ARMA(15,15) system has a FQP support of the same order AR and MA polynomials ($M=15$ and $N=15$). The input $x(k_1, k_2)$ is a white noise process with mean zero and variances σ_x^2 . The column-wise FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=15$ and $N=15$ are illustrated in Figure 8.25.

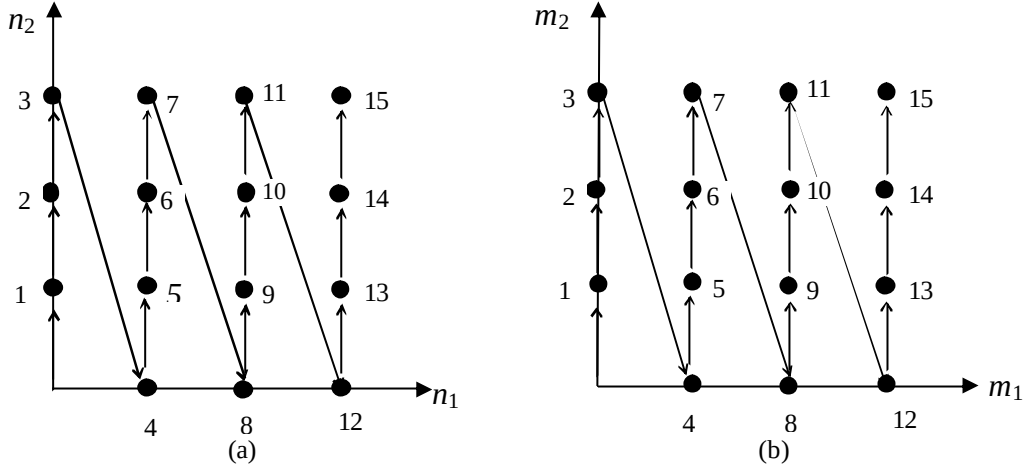


Figure 8-25: FQP ordering schemes of the Computer Experiment 9- ARMA(15,15) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.9.1. Objectives of Computer Experiment 9

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(15,15) parameter calculation formulas given in Equations (4.3) and (5.6), respectively, for the case which the AR order is equal to the MA order ($M=15$, $N=15$, QP support and a different ordering scheme) and the parameter b_0 is different than 1 ($b_0=0.25$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.9.2. Results of Computer Experiment 9

The results of the Computer Experiment 9 have been given in Table 8.22 and Table 8.23 for different data sizes. Table 8.22 lists the original ARMA(15,15) parameters versus the identified ARMA(15,15) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.23 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both

tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a 50x50 data field size have been given in Figure 8.26 and Figure 8.27, respectively.

Table 8.22: Computer Experiment 9 - ARMA(15,15) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Original ARMA(15,15) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
AR Parameters	0.0715	0.0511 0.0718	0.0702 0.0715	0.0704 0.0715
	0.2770	0.2727 0.2770	0.2767 0.2770	0.2768 0.2770
	-0.0275	-0.0327 -0.0273	-0.0279 -0.0275	-0.0278 -0.0275
	0.0018	-0.0081 0.0019	0.0016 0.0018	0.0016 0.0018
	-0.0946	-0.0975 -0.0944	-0.0949 -0.0946	-0.0949 -0.0946
	0.0935	0.0905 0.0935	0.0935 0.0935	0.0935 0.0935
	-0.0291	-0.0313 -0.0290	-0.0293 -0.0291	-0.0293 -0.0291
	0.0473	0.0509 0.0471	0.0477 0.0473	0.0476 0.0473
	-0.0857	-0.0901 -0.0855	-0.0859 -0.0857	-0.0859 -0.0857
	0.0422	0.0430 0.0421	0.0423 0.0422	0.0424 0.0422
	-0.0796	-0.0811 -0.0796	-0.0797 -0.0796	-0.0797 -0.0796
	-0.0510	-0.0460 -0.0507	-0.0510 -0.0510	-0.0510 -0.0510
	0.0896	0.0951 0.0895	0.0898 0.0896	0.0897 0.0896
	-0.0979	-0.0880 -0.0981	-0.0974 -0.0979	-0.0974 -0.0979

Original ARMA(15,15) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
	0.2276	0.2327	0.2278	0.2278
		0.2275	0.2276	0.2276
MA Parameters	-0.0400	-0.0450	-0.0403	-0.0403
		-0.0399	-0.0400	-0.0400
	0.0030	0.0030	0.0030	0.0030
		0.0030	0.0030	0.0030
	0.0025	0.0028	0.0025	0.0025
		0.0025	0.0025	0.0025
	0.0230	0.0204	0.0229	0.0230
		0.0229	0.0230	0.0230
	-0.0230	-0.0235	-0.0231	-0.0231
		-0.0229	-0.0230	-0.0230
	0.0110	0.0109	0.0110	0.0110
		0.0110	0.0110	0.0110
	-0.0110	-0.0111	-0.0110	-0.0110
		-0.0110	-0.0110	-0.0110
	0.0070	0.0079	0.0071	0.0071
		0.0069	0.0070	0.0070
	-0.0070	-0.0083	-0.0071	-0.0071
		-0.0069	-0.0070	-0.0070
	0.0010	0.0011	0.0010	0.0010
		0.0010	0.0010	0.0010
	-0.0230	-0.0227	-0.0230	-0.0230
		-0.0230	-0.0230	-0.0230
	0.0230	0.0246	0.0230	0.0230
		0.0230	0.0230	0.0230
	0.0400	0.0400	0.0400	0.0400
		0.0400	0.0400	0.0400
	-0.0400	-0.0388	-0.0399	-0.0399
		-0.0401	-0.0400	-0.0400
	0.0210	0.0221	0.0210	0.0210
		0.0210	0.0210	0.0210
b_0	0.25	0.2500	0.2499	0.2500
		0.2499	0.2499	0.2500

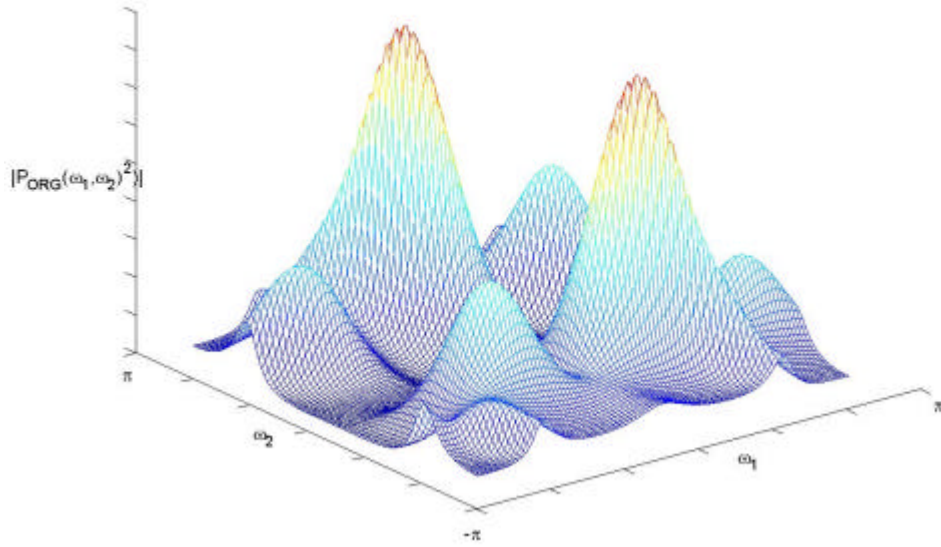
Table 8.23: Computer Experiment 9 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Performance Criteria	Data Field Size		
	10x10	50x50	100x100
L₁-norm	0.0990	0.00541	0.00463
	0.0029	0.00017	0.00007
L₂-norm	0.0289	0.00174	0.00152
	0.0007	0.00004	0.00002
L_∞-norm	0.0204	0.00132	0.00113
	0.0003	0.00002	0.00003
Itakura-Saito Distance	0.0000194	0.000000086	0.000000035
	0.0000003	0.000000013	0.000000012

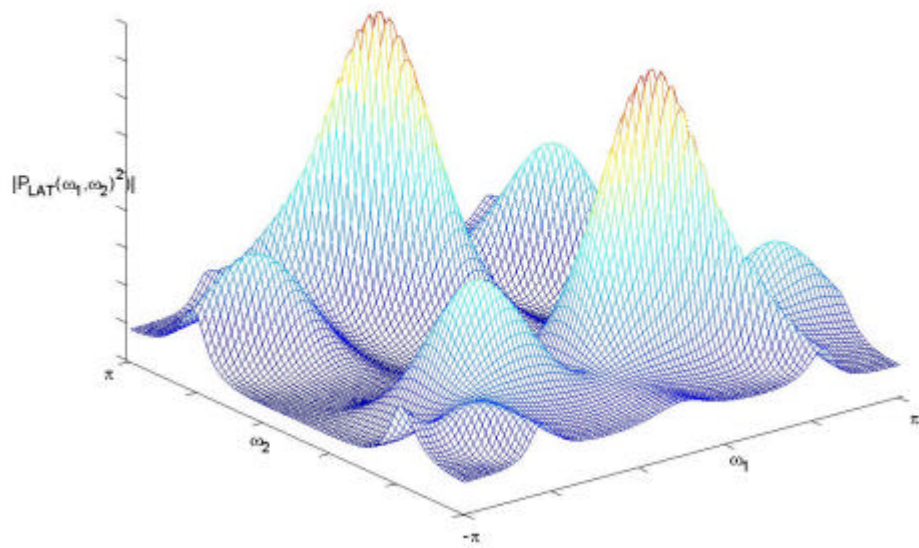
The obtained results yield the following conclusions for this experiment:

- The parameters of ARMA(8,8) system input $x(k_1, k_2)$ have been simultaneously identified with the proposed 2-D hybrid lattice algorithm provided that $x(k_1, k_2)$ is an AR process of order N .
- As it can easily be seen from Table 8.22 and Table 8.23, the identification results are quite accurate and improve with the increasing data size.
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- Even for a small data field size of 10x10, ARMA(8,8) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.

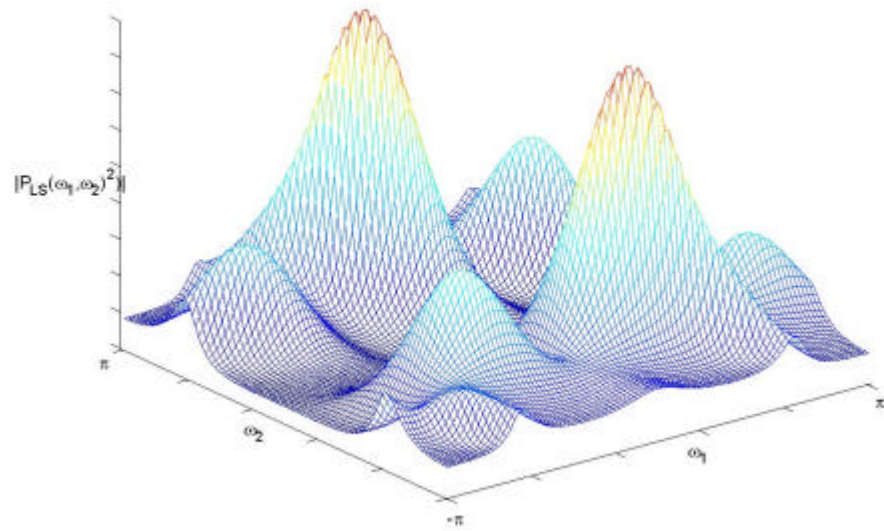
- By looking at the Itakura-Saito distances listed in Table 8.23 and the power spectrum and contour plots given in Figure 8.26 and Figure 8.27, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

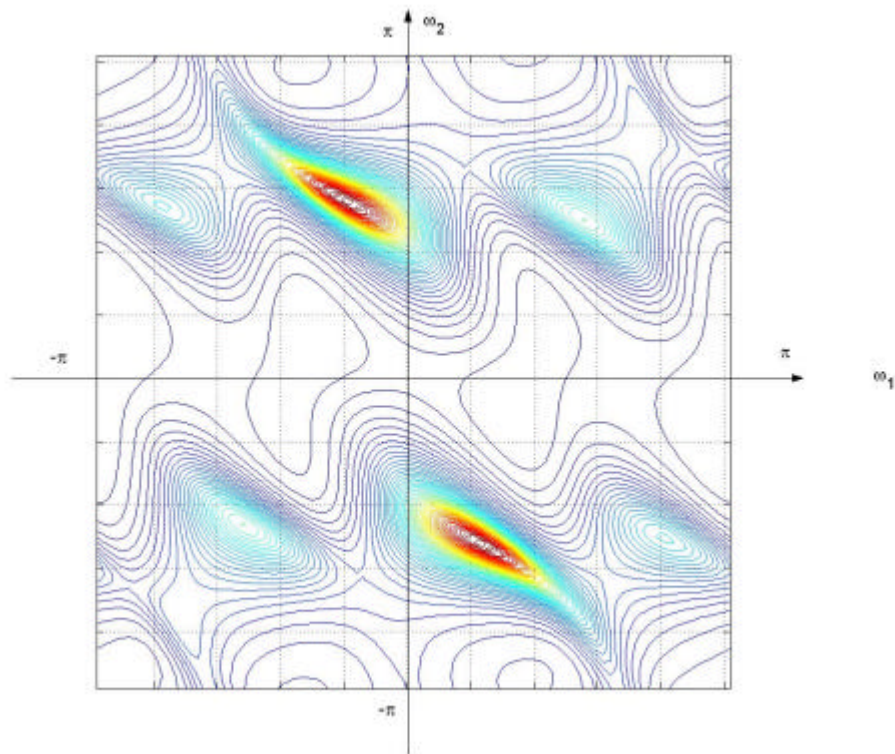


(b)

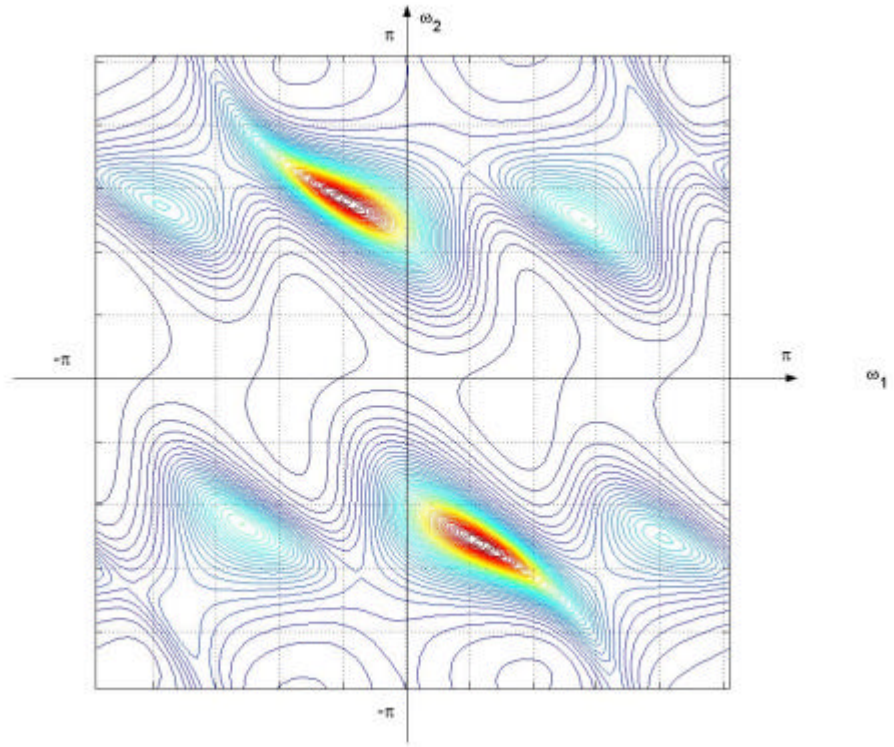


(c)

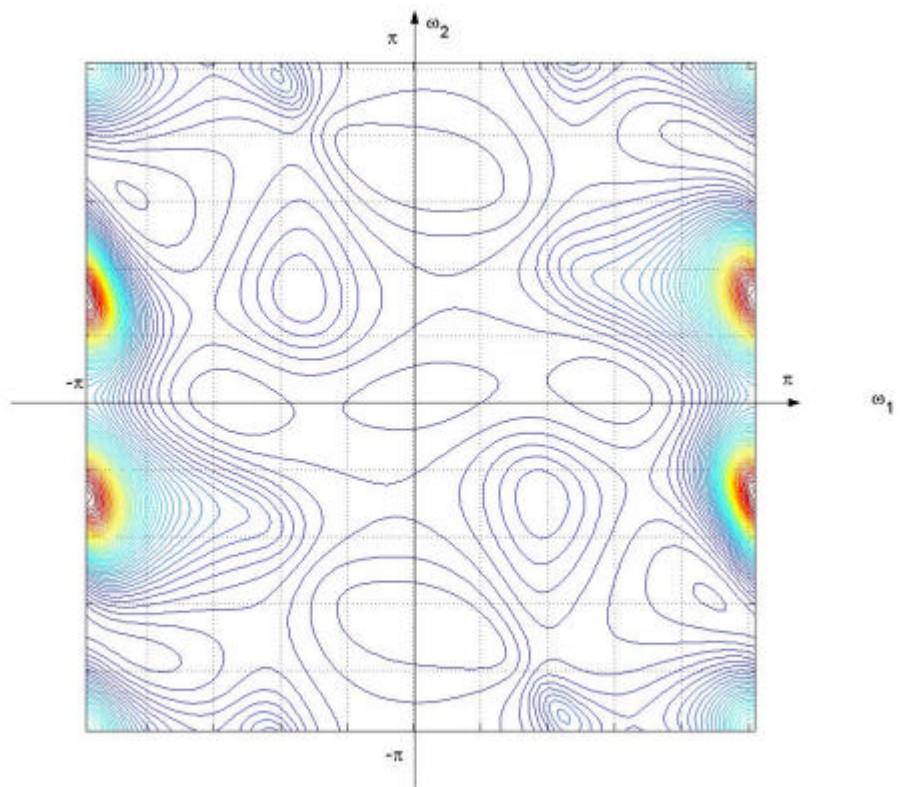
Figure 8-26: Computer Experiment 9 – Power Spectrums plotted using (a) Original ARMA(15,15) parameters (b) ARMA(15,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,15) parameters identified by the LS algorithm. The parameter $b_0=0.25$ and the data field size is 50x50.



(a)



(b)



(c)

Figure 8-27: Computer Experiment 9 – Contour plots of (a) Original ARMA(15,15) parameters (b) ARMA(15,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,15) parameters identified by the LS algorithm. The parameter $b_0=0.25$ and the data field size is 50x50.

8.10. Computer Experiment 10

The unknown 2-D ARMA(3,2) system with FQP support and of different order AR and MA polynomials ($M=3$ and $N=2$) has the following difference equation:

$$y(k_1, k_2) = 0.22x(k_1, k_2) - 0.1x(k_1 - 1, k_2) + 0.7x(k_1 - 1, k_2 - 1) - 0.2y(k_1 - 1, k_2) + 0.18y(k_1 - 1, k_2 - 1) + 0.4y(k_1, k_2 - 1) \quad (9.6)$$

The system input $x(k_1, k_2)$ is white-Gaussian noise with variance s_x^2 and mean zero. The horizontal-vertical FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=3$ and $N=2$ are illustrated in Figure 8.28.

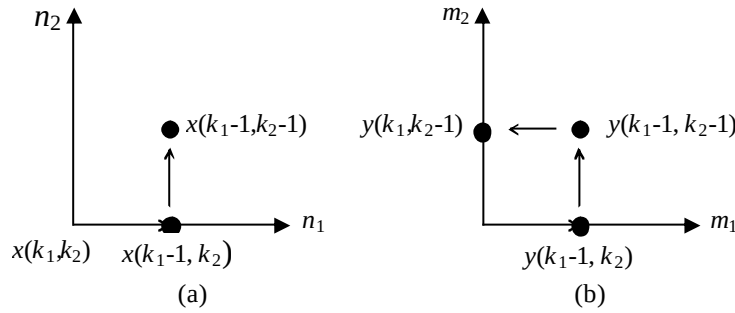


Figure 8-28: FQP ordering schemes of the Computer Experiment 10 - ARMA(3,2) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.10.1. Objectives of Computer Experiment 10

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(3,2) parameter calculation formulas given in Equations (4.3) and (5.13), respectively, for the case which the AR order is greater than the MA order ($M=3$, $N=2$ and QP support) and the parameter b_0 is different than 1 ($b_0=0.22$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.10.2. Results of Computer Experiment 10

The results of the Computer Experiment 10 have been given in Table 8.24 and Table 8.25 for different data sizes. Table 8.24 lists the original ARMA(3,2) parameters versus the identified ARMA(3,2) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.25 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a 50x50 data field size have been given in Figure 8.29-Figure 8.31, respectively.

Table 8.24: Computer Experiment 10 - ARMA(3,2) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the third row. Hybrid Lattice results with no b_0 correction are shown in the second row.

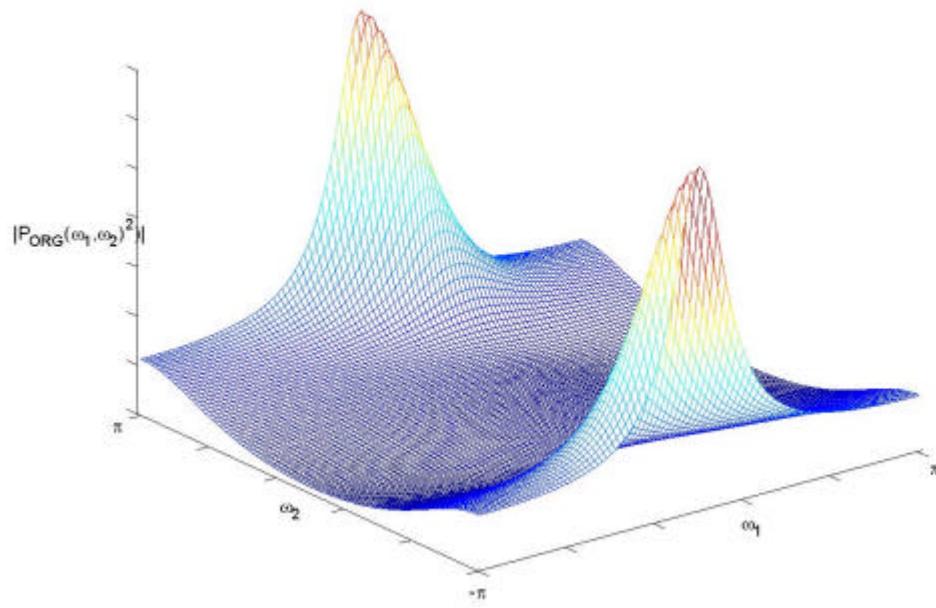
Original ARMA(3,2) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	35x35	100x100
AR Parameters	-0.2	-0.2020	-0.1948	-0.2013
		-0.0933	-0.2098	-0.2274
		-0.2000	-0.2000	-0.2000
	0.18	0.2170	0.1977	0.2172
		0.0973	0.1705	0.2154
		0.1800	0.1800	0.1800
	0.4	0.3893	0.3903	0.3897
		0.3893	0.3903	0.3897
		0.4000	0.4000	0.4000
MA Parameters	-0.1	-0.0993	-0.0997	-0.0995
		0.0209	-0.0652	-0.0940
		-0.1000	-0.1000	-0.1000
	0.7	0.6792	0.6780	0.6791
		-0.5809	0.6881	0.6770
		0.7000	0.7000	0.7000
b_0	0.22	0.2201	0.2239	0.2201
		0.2200	0.2200	0.2200

Table 8.25: Computer Experiment 10 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row. Hybrid Lattice results with no b_0 correction are shown in the second row.

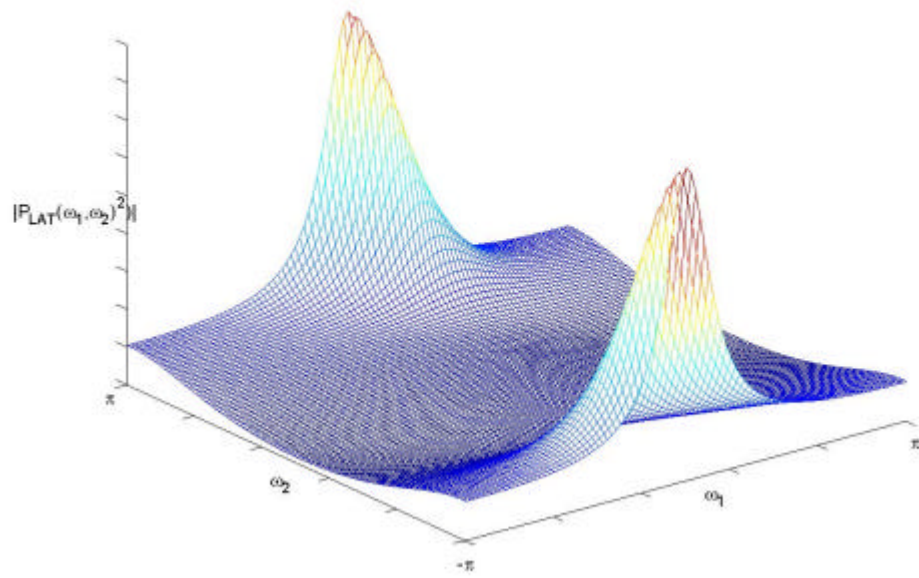
Performance Criteria	Data Field Size		
	10x10	35x35	100x100
L₁-norm	0.0713	0.0688	0.0703
	0.4169	0.1268	0.0969
	0.0000	0.0000	0.0000
L₂-norm	0.0438	0.0436	0.0439
	0.1983	0.0692	0.0504
	0.0000	0.0000	0.0000
L_∞-norm	0.0370	0.0371	0.0372
	0.1191	0.0568	0.0354
	0.0000	0.0000	0.0000
Itakura-Saito Distance	0.003287	0.003256	0.003286
	0.244200	0.002617	0.005969
	0.000000	0.000000	0.000000

The obtained results yield the following conclusions for this experiment:

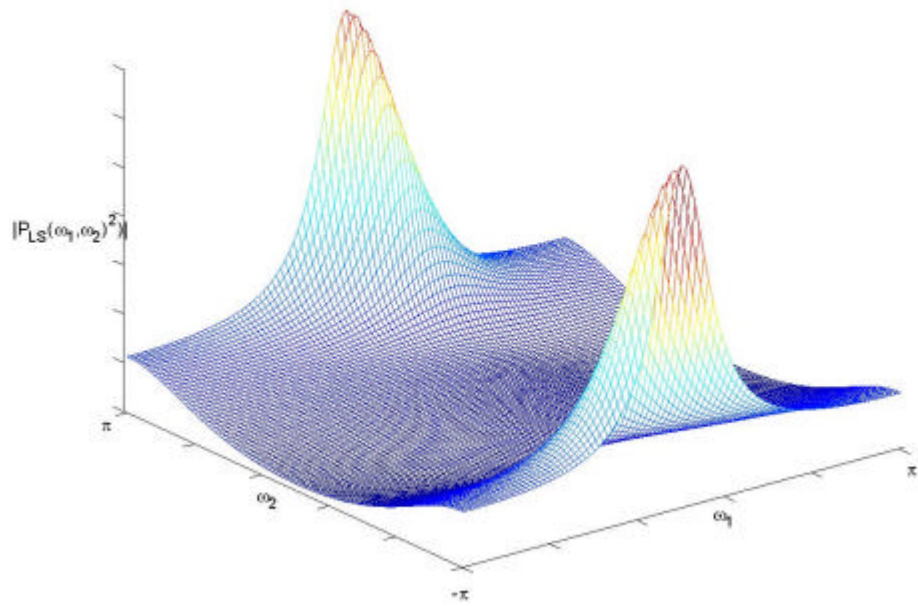
- As it can easily be seen from Table 8.24 and Table 8.25, the identification results are quite accurate and improve with the increasing data size.
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- By looking at the Itakura-Saito distances listed in Table 8.25 and the power spectrum and contour plots given in Figure 8.29 and Figure 8.30, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

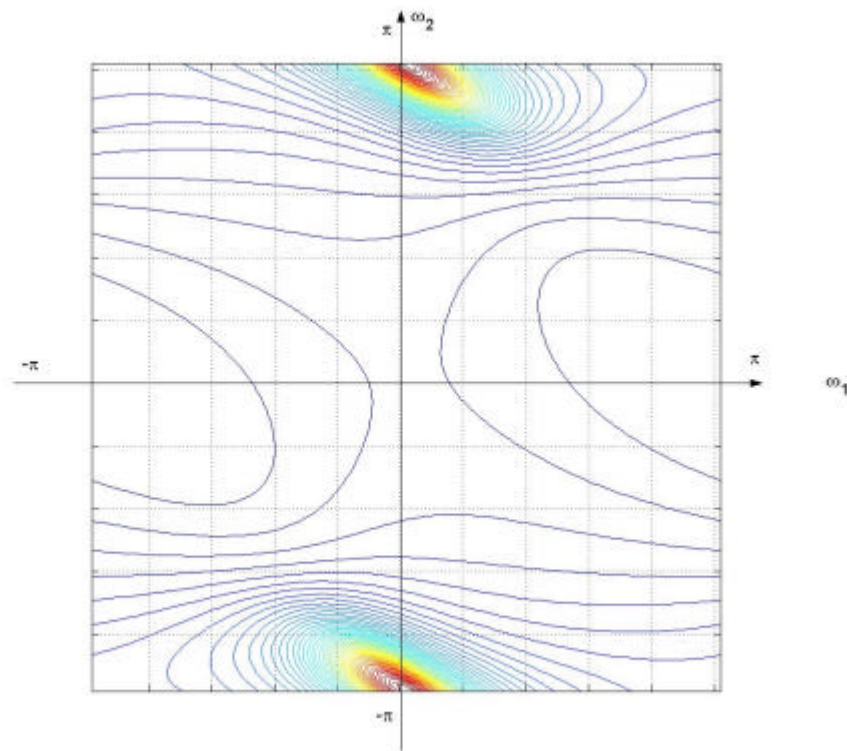


(b)

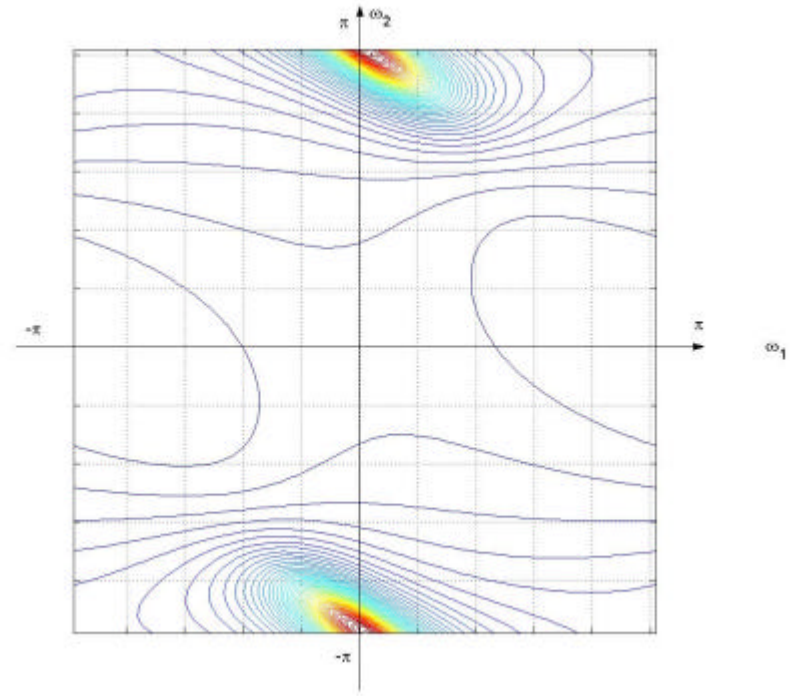


(c)

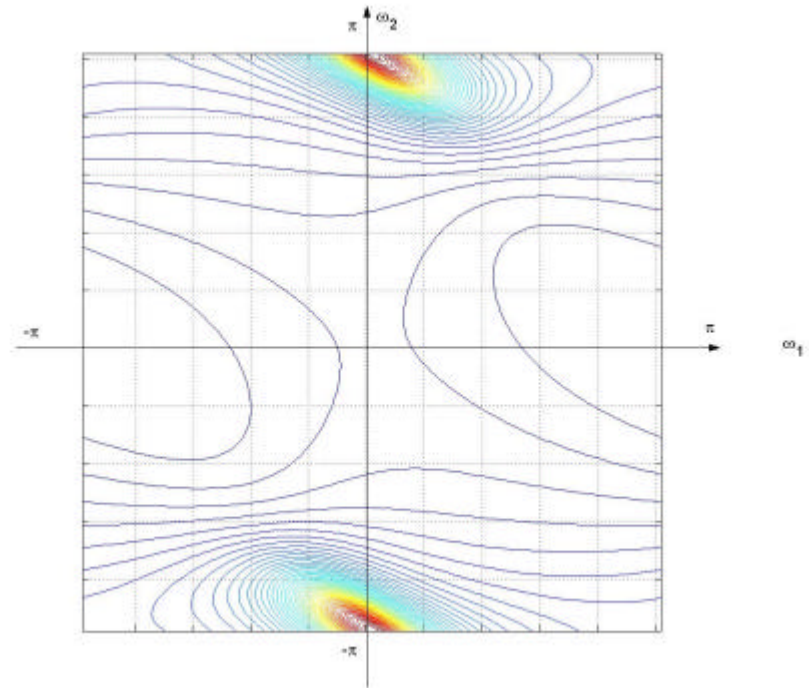
Figure 8-29: Computer Experiment 10 – Power Spectrums plotted using (a) Original ARMA(3,2) parameters (b) ARMA(3,2) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,2) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10×10 .



(a)



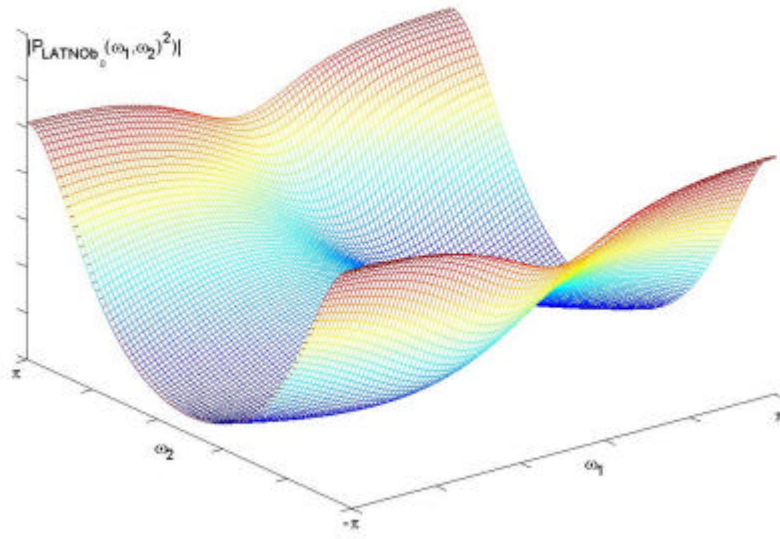
(b)



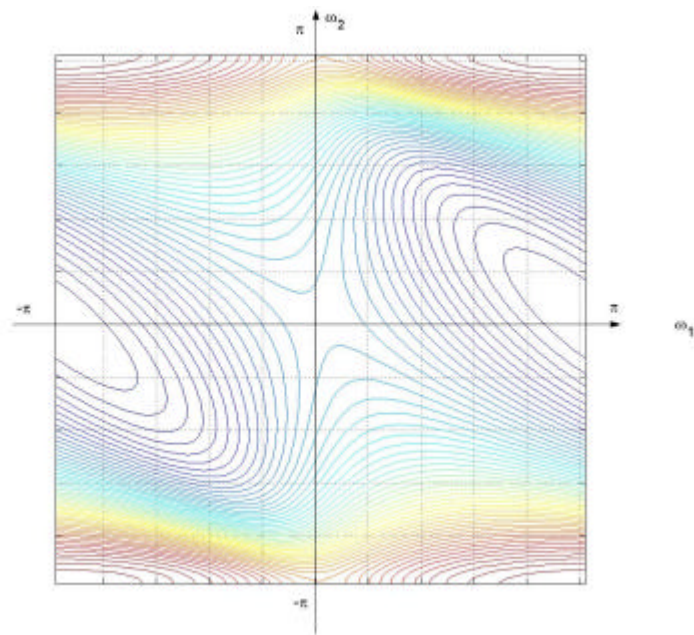
(c)

Figure 8-30: Computer Experiment 10 – Contour plots of (a) Original ARMA(3,2) parameters (b) ARMA(3,2) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,2) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10×10 .

- Even for a small data field size of 10×10 , ARMA(8,8) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.



(a)



(b)

Figure 8-31: Computer Experiment 10 – (a) Power spectrum plotted using ARMA(3,2) parameters identified by the Hybrid Lattice algorithm with no b_0 correction (b) the related ARMA(3,2) contour plot.

The unknown 2-D ARMA(15,8) system has a FQP support of the different order AR and MA polynomials, M being greater than N ($M=15$ and $N=8$). The input $x(k_1, k_2)$ is a white noise process with mean zero and variances σ_x^2 . The horizontal-vertical FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=15$ and $N=8$ are illustrated in Figure 8.32.

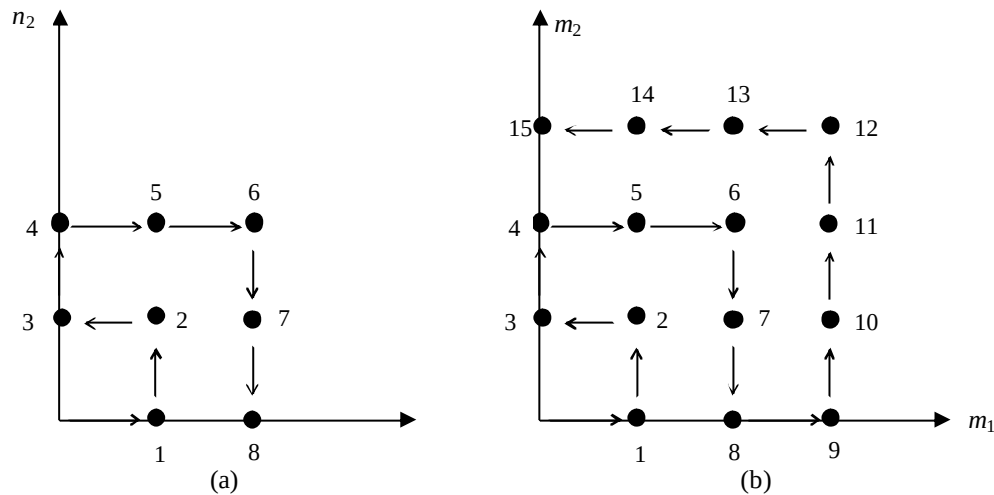


Figure 8-32: FQP ordering schemes of the Computer Experiment 11 - ARMA(15,8) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.11.1. Objectives of Computer Experiment 11

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(15,8) parameter calculation formulas given in Equations (4.3) and (5.13), respectively, for the case which the AR order is greater than the MA order ($M=15$, $N=8$ and QP support) and the parameter $b_0=1$.
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.

To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.11.2. Results of Computer Experiment 11

The results of the Computer Experiment 11 have been given in Table 8.26 and Table 8.27 for different data sizes. Table 8.26 lists the original ARMA(15,8) parameters versus the identified ARMA(15,8) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.27 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a 10x10 data field size have been given in Figure 8.33 and Figure 8.34, respectively.

Table 8.26: Computer Experiment 11 - ARMA(15,8) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Original ARMA(15,8) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
AR Parameters	0.2100	0.3575 0.2131	0.1750 0.2103	0.1772 0.2111
	0.2100	0.1908 0.2274	0.2028 0.2135	0.2020 0.2119
	0.1400	0.0849 0.1598	0.1422 0.1456	0.1395 0.1419
	0.1000	0.0683 0.0924	0.0973 0.1014	0.0992 0.1009
	0.1000	0.0736 0.0848	0.0906 0.0996	0.0927 0.0997
	0.1700	0.1791 0.1723	0.1607 0.1689	0.1632 0.1702
	0.1000	0.1354 0.1151	0.0898 0.1001	0.0916 0.1014
	0.1000	0.1083 0.1146	0.0756 0.1029	0.0747 0.1010
	0.1000	0.1193 0.1001	0.0942 0.1005	0.0923 0.1003
	-0.1300	-0.0951 -0.1220	-0.1388 -0.1294	-0.1396 -0.1293

Original ARMA(15,8) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
	-0.1100	-0.0806	-0.1257	-0.1246
		-0.1073	-0.1102	-0.1097
	-0.1200	-0.1760	-0.1180	-0.1176
		-0.1208	-0.1200	-0.1202
	-0.1300	-0.1684	-0.1259	-0.1270
		-0.1447	-0.1284	-0.1303
	-0.1000	-0.1155	-0.0963	-0.0971
		-0.1029	-0.0986	-0.0999
	-0.1000	-0.1172	-0.1063	-0.1013
		-0.1137	-0.1046	-0.1017
MA Parameters	0.1100	0.2781	0.0743	0.0715
		0.1184	0.1105	0.1110
	0.2000	0.0973	0.1992	0.2024
		0.2022	0.2022	0.2016
	-0.3100	-0.3769	-0.3066	-0.3054
		-0.2976	-0.3029	-0.3083
	0.7100	0.7075	0.7059	0.7044
		0.6965	0.7059	0.7092
	-0.2100	-0.1034	-0.2274	-0.2348
		-0.2146	-0.2089	-0.2096
	-0.1000	-0.1764	-0.1186	-0.1176
		-0.0981	-0.1000	-0.1001
	-0.1000	-0.0629	-0.1029	-0.0978
		-0.0992	-0.1026	-0.0995
	-0.1000	-0.0493	-0.1267	-0.1216
		-0.0753	-0.0970	-0.0989
b_0	1.0000	1.0117	0.9977	1.0001
		1.0003	0.9994	1.0000

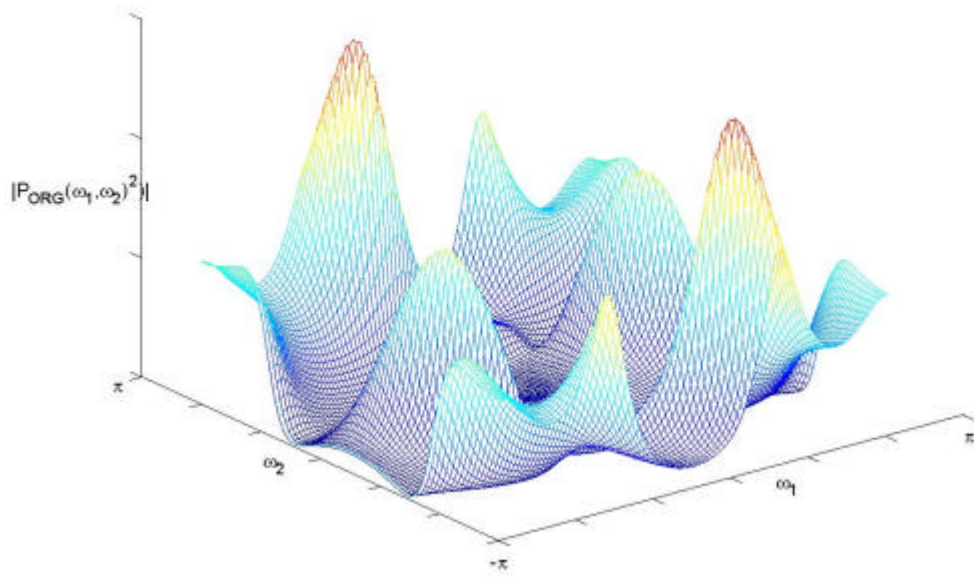
Table 8.27: Computer Experiment 11 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Performance Criteria	Data Field Size		
	10x10	50x50	100x100
L₁-norm	1.1661	0.2587	0.2487
	0.2068	0.0450	0.0198
L₂-norm	0.3169	0.0726	0.0726
	0.0545	0.0130	0.0050
L_∞-norm	0.1681	0.0357	0.0385
	0.0247	0.0071	0.0019
Itakura-Saito Distance	0.040194	0.001263	0.002100
	0.011703	0.000182	0.000013

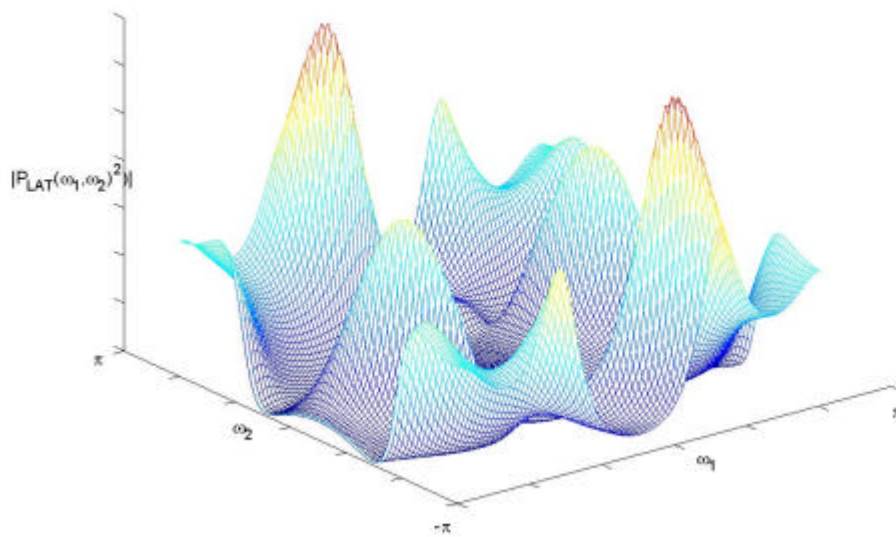
The obtained results yield the following conclusions for this experiment:

- As it can easily be seen from Table 8.26 and Table 8.27, the identification results are quite accurate and improve with the increasing data size.
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- Even for a small data field size of 10x10, ARMA(15,8) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.

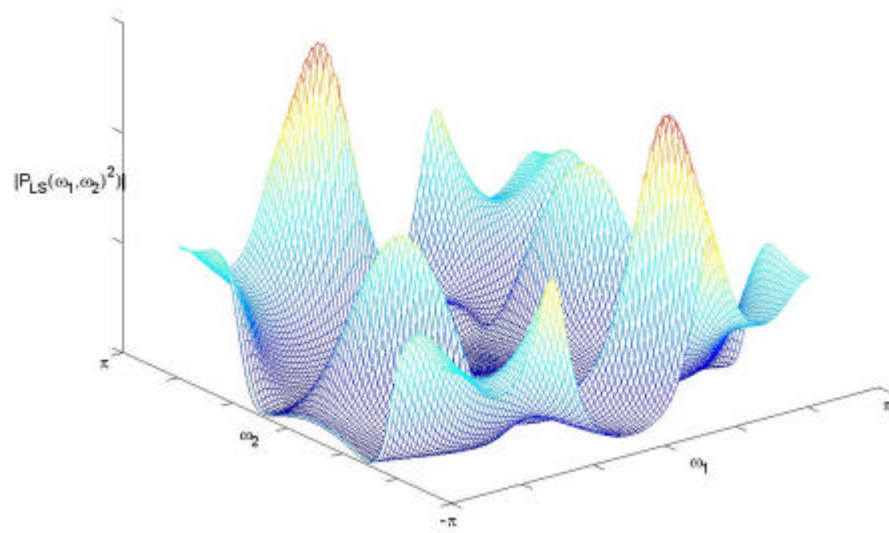
- By looking at the Itakura-Saito distances listed in Table 8.27 and the power spectrum and contour plots given in Figure 8.33 and Figure 8.34, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

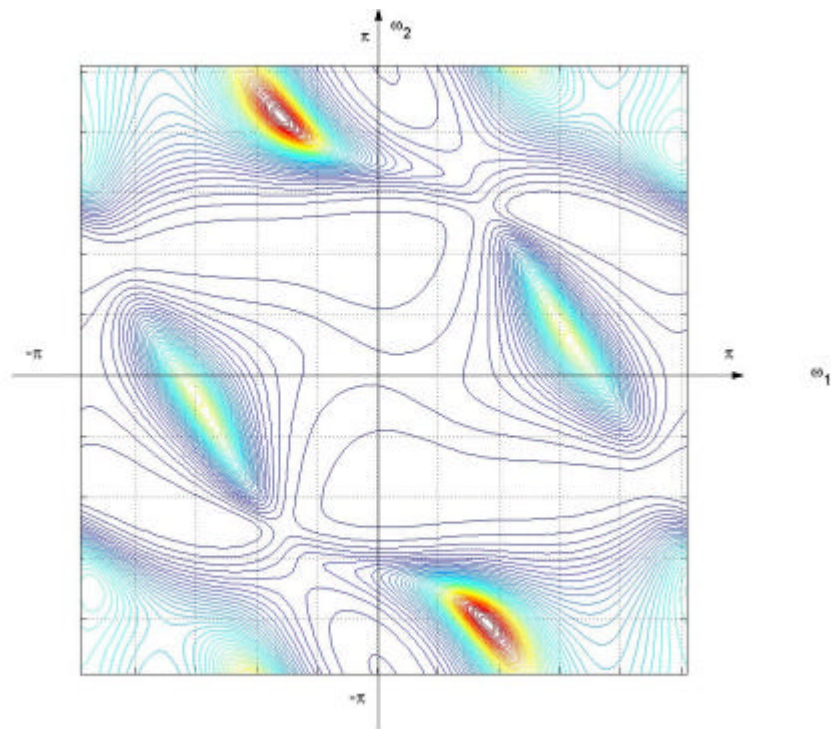


(b)

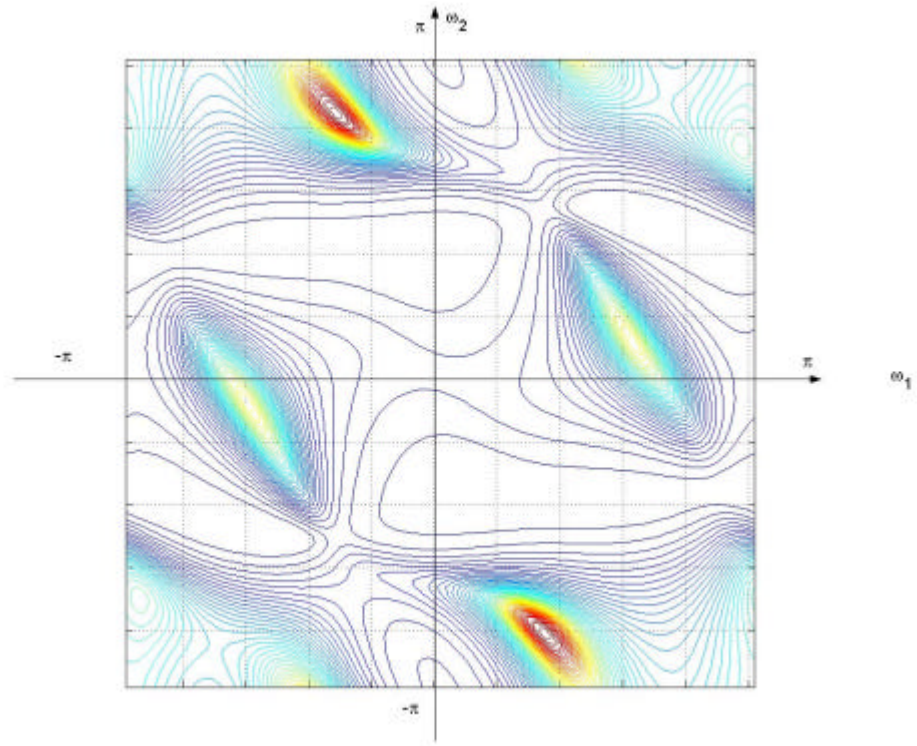


(c)

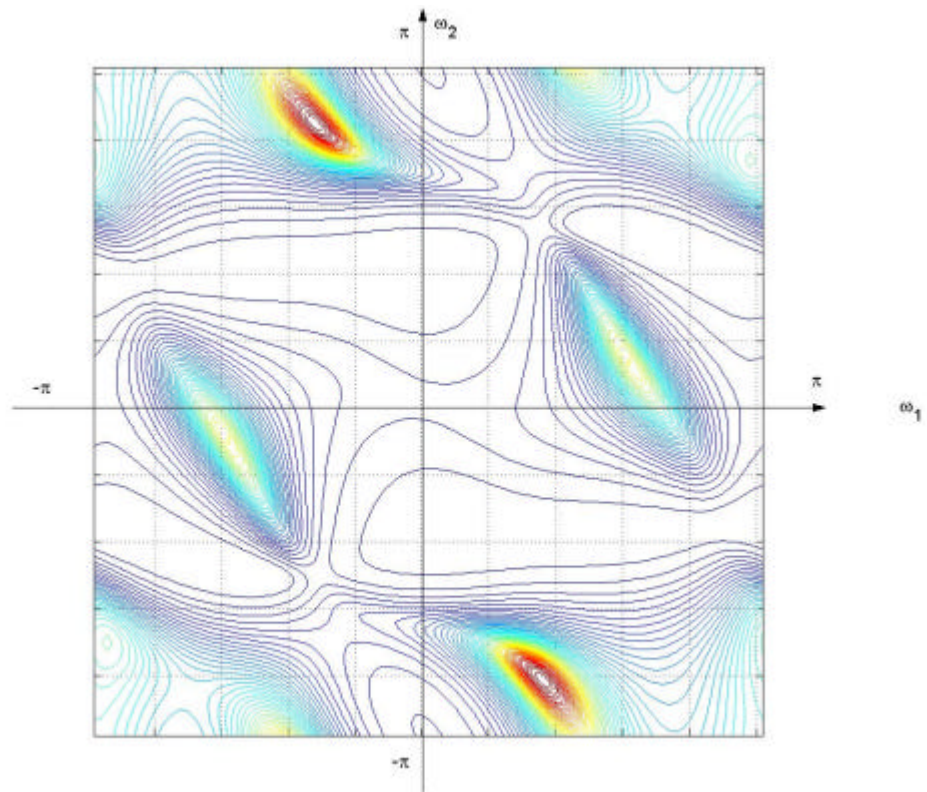
Figure 8-33: Computer Experiment 11 – Power Spectrums plotted using (a) Original ARMA(15,8) parameters (b) ARMA(15,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,8) parameters identified by the LS algorithm. The parameter $b_0=1$ and the data field size is 10×10 .



(a)



(b)



(c)

Figure 8-34: Computer Experiment 11 – Contour plots of (a) Original ARMA(15,8) parameters (b) ARMA(15,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,8) parameters identified by the LS algorithm. The parameter $b_0=0.22$ and the data field size is 10×10 .

8.12. Computer Experiment 12

The unknown 2-D ARMA(15,8) system has a FQP support of the different order AR and MA polynomials, M being greater than N ($M=15$ and $N=8$). The input $x(k_1, k_2)$ is a white noise process with mean zero and variance s_x^2 . The horizontal-vertical FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=15$ and $N=8$ are illustrated in Figure 8.32.

8.12.1. Objectives of Computer Experiment 12

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(15,8) parameter calculation formulas given in Equations (4.3) and (5.13), respectively, for the case which the AR order is greater than the MA order ($M=15$, $N=8$ and QP support) and the parameter b_0 is different than 1 ($b_0=0.3$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.

To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.12.2. Results of Computer Experiment 12

The results of the Computer Experiment 12 have been given in Table 8.28 and Table 8.29 for different data sizes. Table 8.28 lists the original ARMA(15,8) parameters versus the identified ARMA(15,8) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.29 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a 100x100 data field size have been given in Figure 8.35 and Figure 8.36, respectively.

Table 8.28: Computer Experiment 12 - ARMA(15,8) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Original ARMA(15,8) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
AR Parameters	0.3100	0.5633	0.4155	0.4073
		0.3345	0.3106	0.3107
	0.2100	0.2370	0.2804	0.2766
		0.2363	0.2119	0.2109
	0.1400	0.0864	0.1692	0.1665
		0.1593	0.1425	0.1415
	0.1400	0.1381	0.1692	0.1910
		0.1379	0.1420	0.1419
	0.1000	0.0556	0.1481	0.1740
		0.0782	0.0991	0.0998
	-0.1700	-0.1429	-0.1101	-0.0780
		-0.1546	-0.1701	-0.1700
	0.1000	0.2429	0.2039	0.2076
		0.1354	0.1002	0.1008
	0.1000	0.0048	0.0045	0.0310
		0.1014	0.1000	0.1001
	0.1000	-0.0519	0.0375	0.0550
		0.0944	0.1001	0.1000
	-0.1300	-0.0561	-0.0802	-0.0935
		-0.1078	-0.1296	-0.1293
	-0.1100	-0.1567	-0.1203	-0.1029
		-0.1034	-0.1104	-0.1102
	-0.1200	-0.0752	-0.1357	-0.1062
		-0.1466	-0.1214	-0.1212
	-0.1300	-0.1543	-0.1334	-0.0932
		-0.1638	-0.1300	-0.1310
	-0.1000	-0.2054	-0.1367	-0.1004
		-0.1205	-0.0999	-0.0997
	-0.1000	-0.1015	-0.1324	-0.0794
		-0.1232	-0.1034	-0.1012
	0.1100	0.1945	0.1381	0.1333
		0.1173	0.1101	0.1103

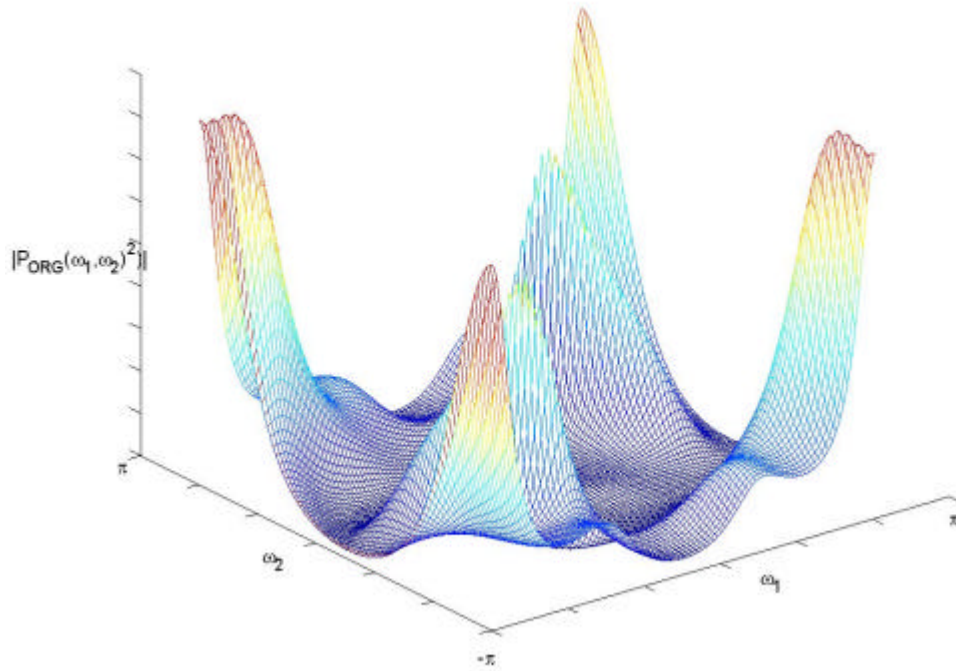
Original ARMA(15,8) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
MA Parameters	0.2000	0.0903 0.1863	0.1660 0.1996	0.1669 0.2003
	-0.3100	-0.3381 -0.3092	-0.2989 -0.3080	-0.3021 -0.3096
	0.7100	0.6998 0.7075	0.6715 0.7081	0.6897 0.7100
	-0.2100	-0.0477 -0.1884	-0.1564 -0.2092	-0.1532 -0.2095
	-0.1000	-0.2880 -0.1067	-0.2492 -0.1003	-0.2257 -0.1008
	-0.1000	0.0103 -0.0933	-0.0300 -0.1011	-0.0414 -0.0998
	-0.1000	-0.0995 -0.0883	-0.1204 -0.0996	-0.1076 -0.0996
b_0	0.3000	0.3016 0.2948	0.3039 0.3004	0.3004 0.2998

Table 8.29: Computer Experiment 12 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

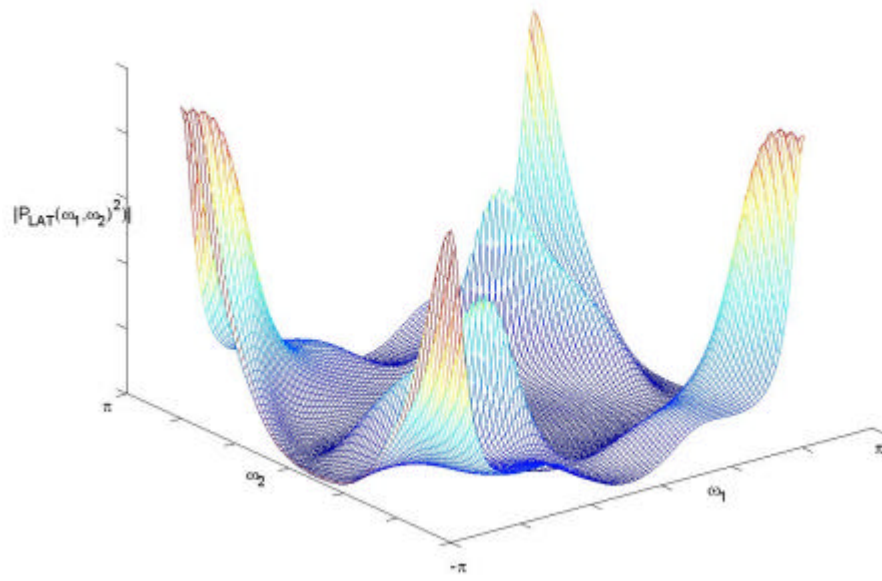
Performance Criteria	Data Field Size		
	10x10	50x50	100x100
L_1 -norm	1.7892	1.1610	1.0781
	0.3607	0.0215	0.0137
L_2 -norm	0.4882	0.2945	0.2784
	0.0893	0.0062	0.0037
L_∞ -norm	0.2533	0.1492	0.1257
	0.0354	0.0034	0.0019
Itakura-Saito Distance	0.231572	0.294239	0.279677
	0.005177	0.000074	0.000015

The obtained results yield the following conclusions for this experiment:

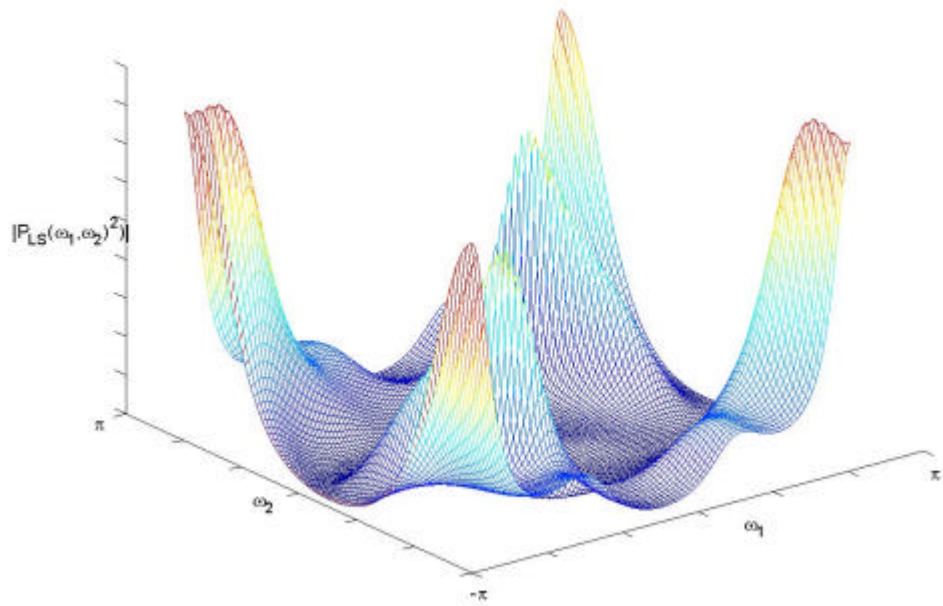
- As it can easily be seen from Table 8.28 and Table 8.29, the identification results are quite accurate and improve with the increasing data size.
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- Even for a small data field size of 10x10, ARMA(15,8) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.
- By looking at the Itakura-Saito distances listed in Table 8.29 and the power spectrum and contour plots given in Figure 8.35 and Figure 8.36, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

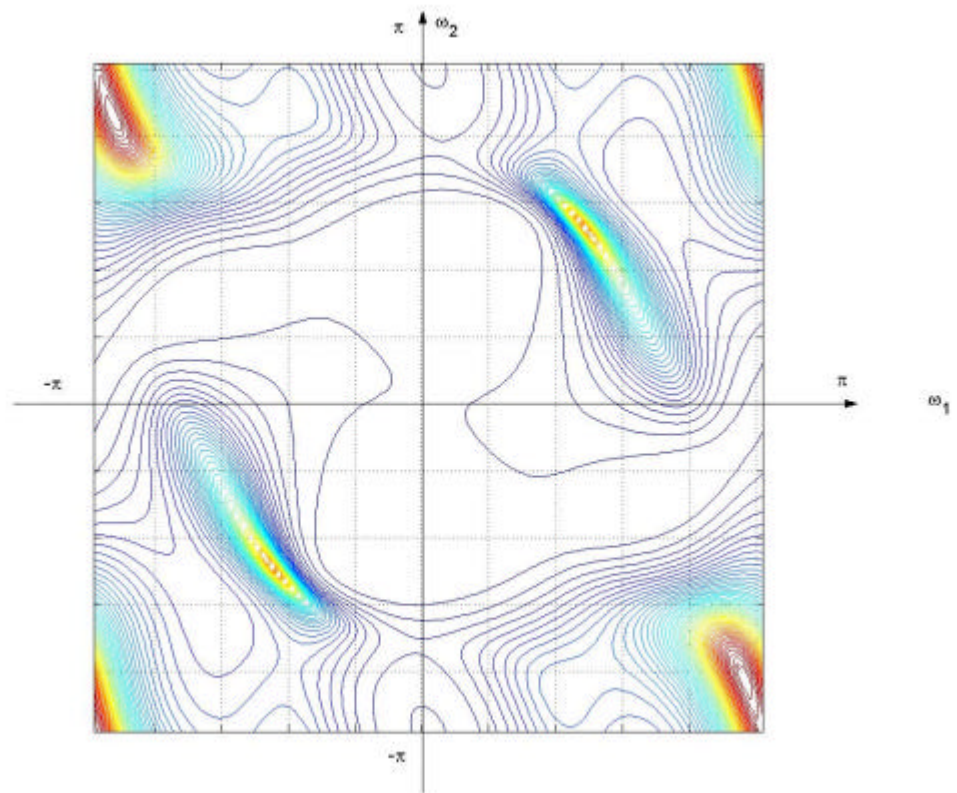


(b)

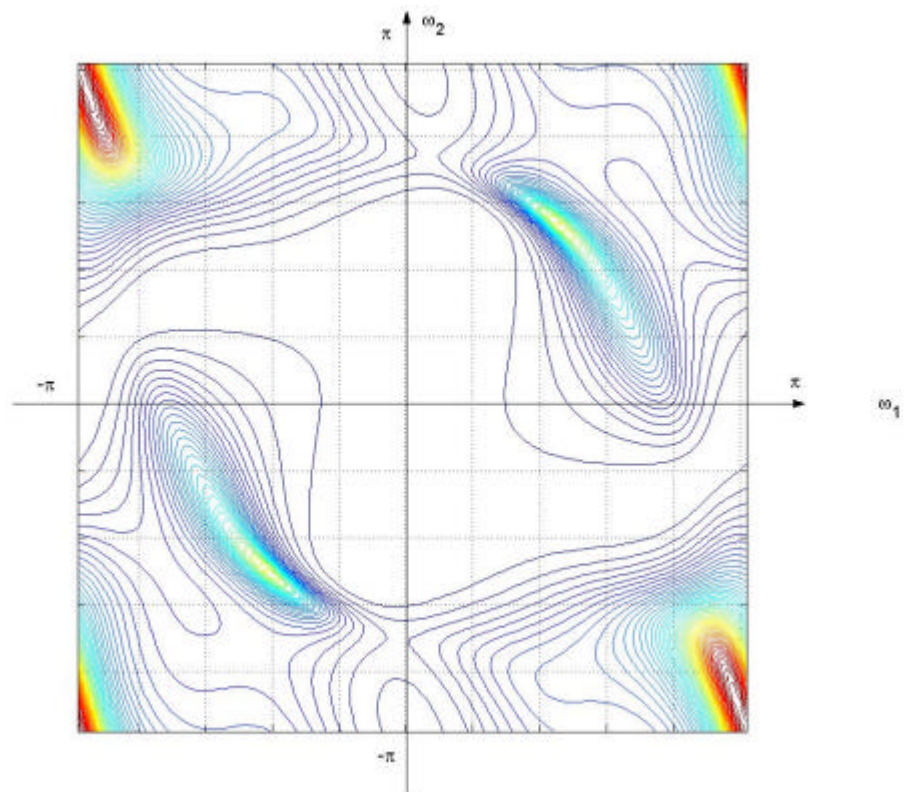


(c)

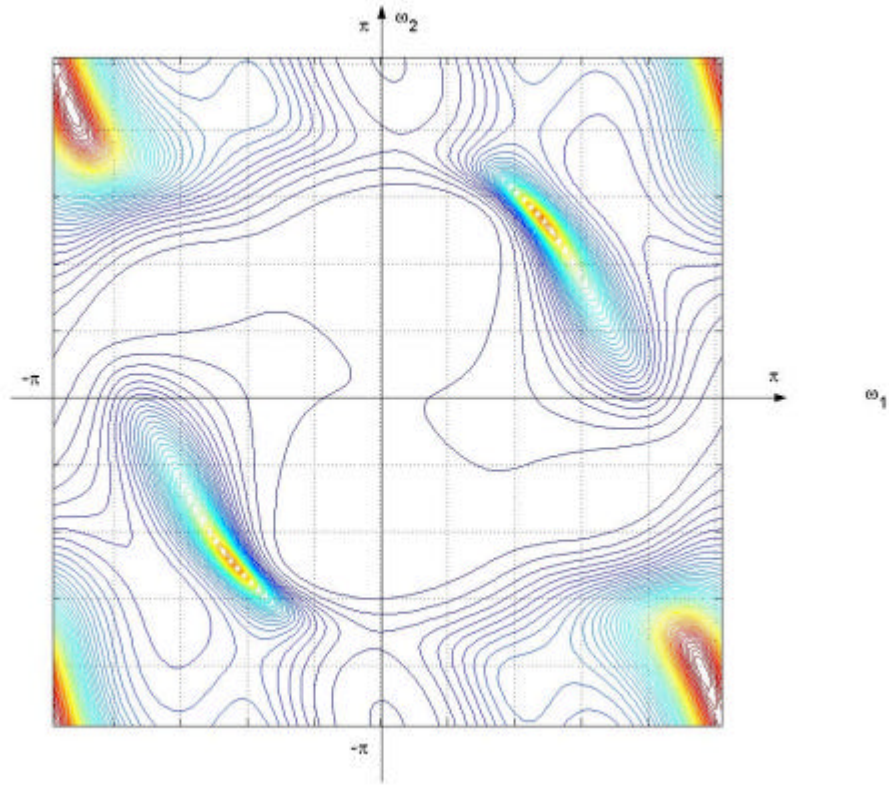
Figure 8-35: Computer Experiment 12 – Power Spectrums plotted using (a) Original ARMA(15,8) parameters (b) ARMA(15,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,8) parameters identified by the LS algorithm. The parameter $b_0=0.3$ and the data field size is 100x100.



(a)



(b)



(c)

Figure 8-36: Computer Experiment 12 – Contour plots of (a) Original ARMA(15,8) parameters (b) ARMA(15,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(15,8) parameters identified by the LS algorithm. The parameter $b_0=0.3$ and the data field size is 100x100.

8.13. Computer Experiment 13

The unknown 2-D ARMA(2,3) system with FQP support and of different order AR and MA polynomials ($M=2$ and $N=3$) has the following difference equation. The input $x(k_1, k_2)$ is a white noise process with mean zero and variance s_x^2 .

$$y(k_1, k_2) = 0.25x(k_1, k_2) - 0.1y(k_1 - 1, k_2) + 0.7y(k_1 - 1, k_2 - 1) - 0.2x(k_1 - 1, k_2) + 0.18x(k_1 - 1, k_2 - 1) + 0.4x(k_1, k_2 - 1) \quad (8.7)$$

The horizontal-vertical FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=2$ and $N=3$ are illustrated in Figure 8.37.

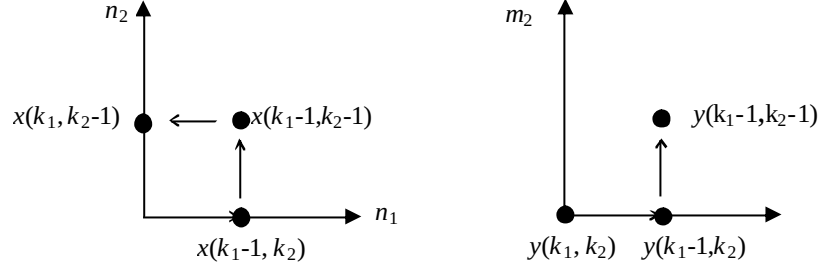


Figure 8-37: FQP ordering schemes of the Computer Experiment 13 - ARMA(2,3) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.13.1. Objectives of Computer Experiment 13

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(2,3) parameter calculation formulas given in Equations (4.3) and (5.13), respectively, for the case which the MA order is greater than the AR order ($M=2$, $N=3$ and QP support) and the parameter b_0 is different than 1 ($b_0=0.75$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.13.2. Results of Computer Experiment 13

The results of the Computer Experiment 13 have been given in Table 8.30 and Table 8.31 for different data sizes. Table 8.30 lists the original ARMA(2,3) parameters versus the identified ARMA(15,8) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.31 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a 10x10 data field size have been given in Figure 8.38 and Figure 8.39, respectively.

Table 8.30: Computer Experiment 13 - ARMA(2,3) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Original ARMA(2,3) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	35x35	100x100
AR Parameters	-0.1	-0.1015	-0.1017	-0.1001
		-0.1000	-0.1000	-0.1000
	0.7	0.6963	0.7033	0.7007
		0.7000	0.7000	0.7000
MA Parameters	-0.2	-0.1979	-0.1996	-0.2001
		-0.2000	-0.2000	-0.2000
	0.18	0.1801	0.1817	0.1803
		0.1800	0.1800	0.1800
	0.4	0.3459	0.3876	0.3958
		0.4000	0.4000	0.4000
b_0	0.75	0.7754	0.7820	0.7836
		0.7500	0.7500	0.7500

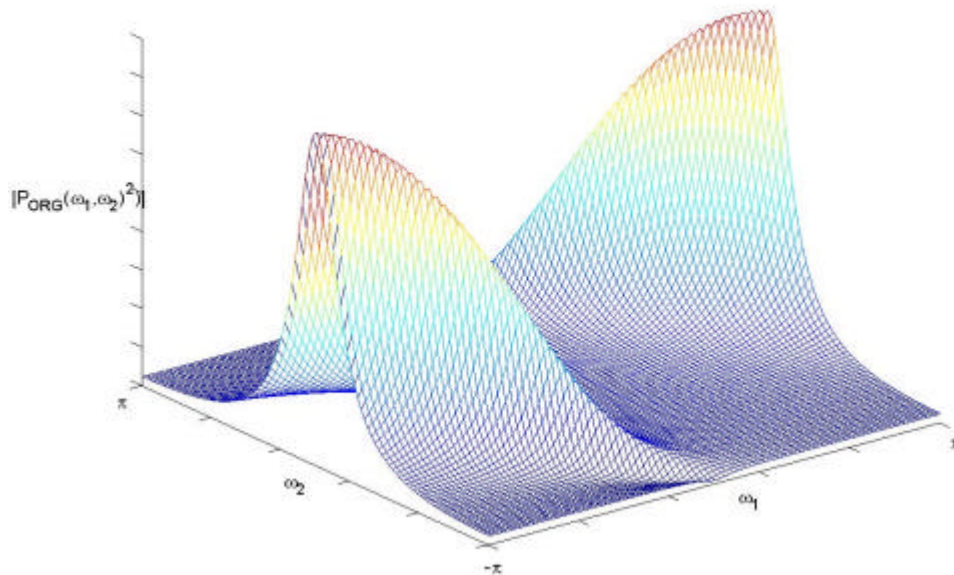
Table 8.31: Computer Experiment 13 – Performance Criteria results of Hybrid Lattice algorithm.

Performance Criteria	Data Field Size		
	10x10	35x35	100x100
L1-norm	0.0374	0.0515	0.0390
L₂-norm	0.0262	0.0346	0.0339
L_y-norm	0.0254	0.0320	0.0336
Itakura-Saito Distance	0.0505721	0.008144	0.010003

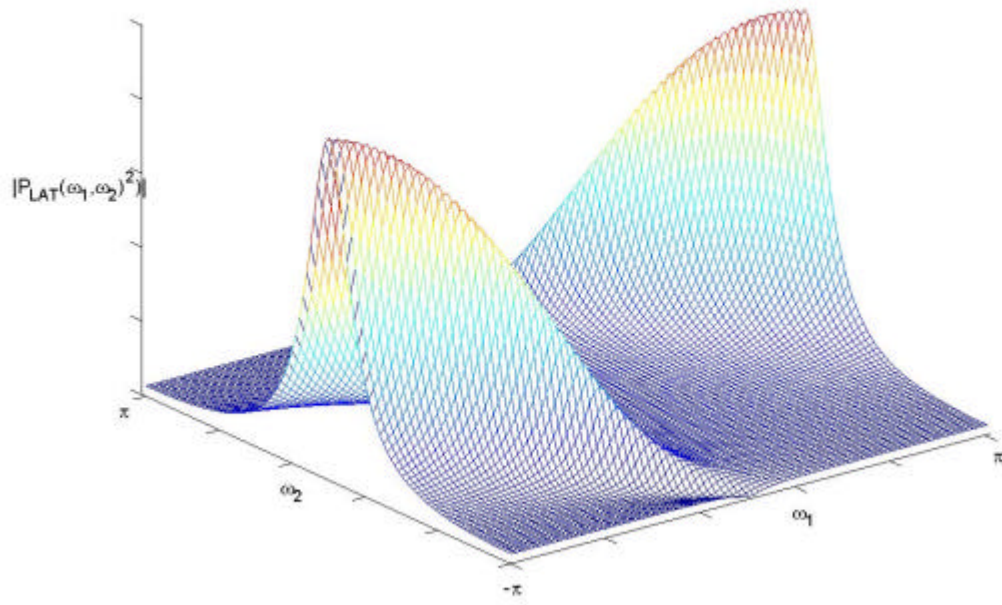
The original ARMA(2,3) parameters have been chosen so as to satisfy the stability conditions given in [119]. By looking at the Itakura-Saito distances, we can say that there is a close match between the power spectrums obtained using original and identified parameters.

The obtained results yield the following conclusions for this experiment:

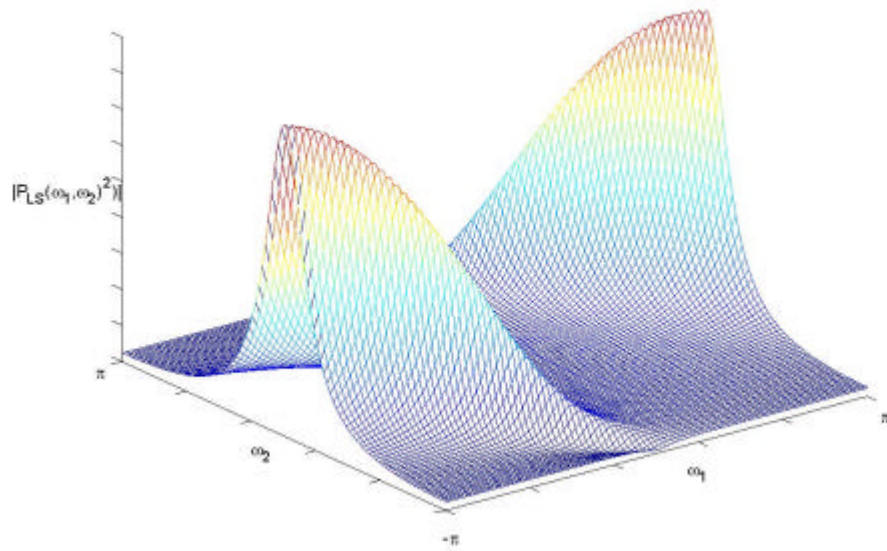
- As it can easily be seen from Table 8.30 and Table 8.31, the identification results are quite accurate and improve with the increasing data size.
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- Even for a small data field size of 10x10, ARMA(2,3) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.
- By looking at the Itakura-Saito distances listed in Table 8.29 and the power spectrum and contour plots given in Figure 8.35 and Figure 8.36, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

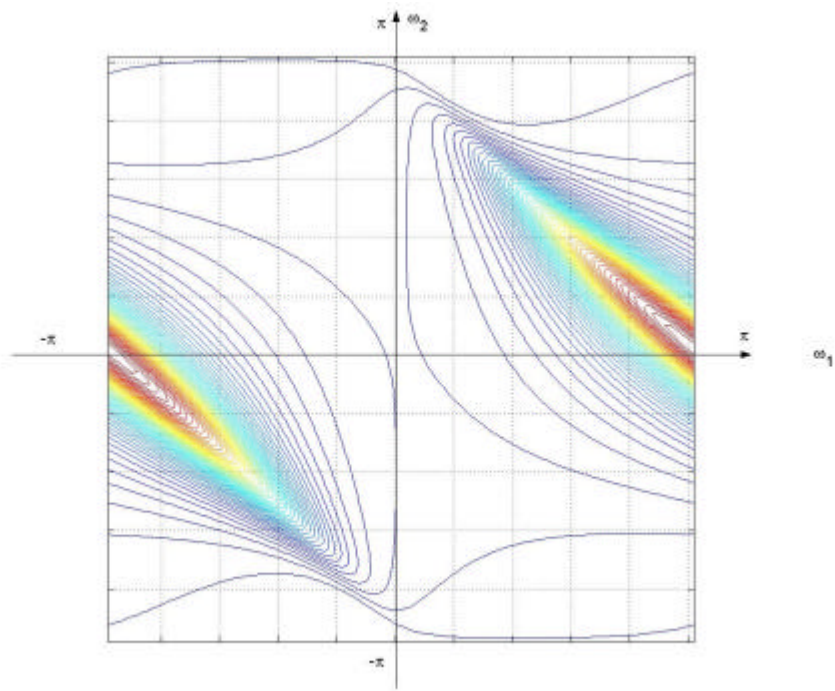


(b)

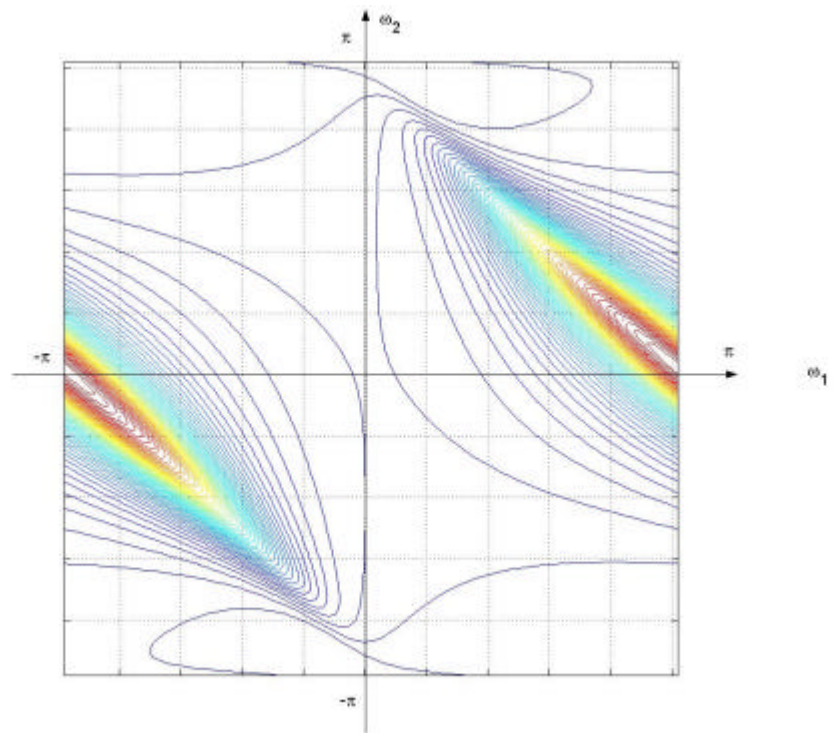


(c)

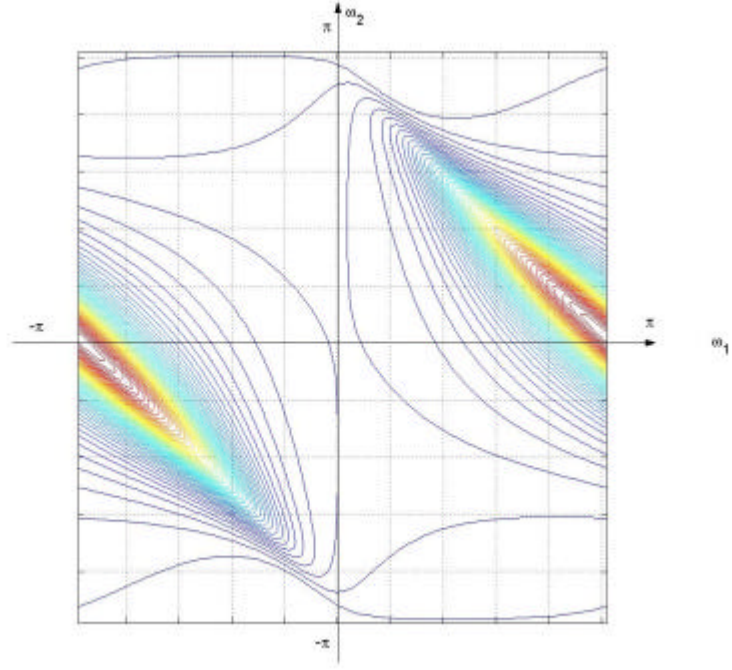
Figure 8-38: Computer Experiment 13 – Power Spectrums plotted using (a) Original ARMA(2,3) parameters (b) ARMA(2,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(2,3) parameters identified by the LS algorithm. The parameter $b_0=0.75$ and the data field size is 35×35 .



(a)



(b)



(c)

Figure 8-39: Computer Experiment 13 – Contour Plots of (a) Original ARMA(2,3) parameters (b) ARMA(2,3) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(2,3) parameters identified by the LS algorithm. The parameter $b_0=0.75$ and the data field size is 35×35 .

8.14. Computer Experiment 14

The unknown 2-D ARMA(3,8) system has a FQP support of the different order AR and MA polynomials, N being greater than M ($M=3$ and $N=8$). The input $x(k_1, k_2)$ is a white noise process with mean zero and variance s_x^2 . The horizontal-vertical FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=3$ and $N=8$ are illustrated in Figure 8.40.

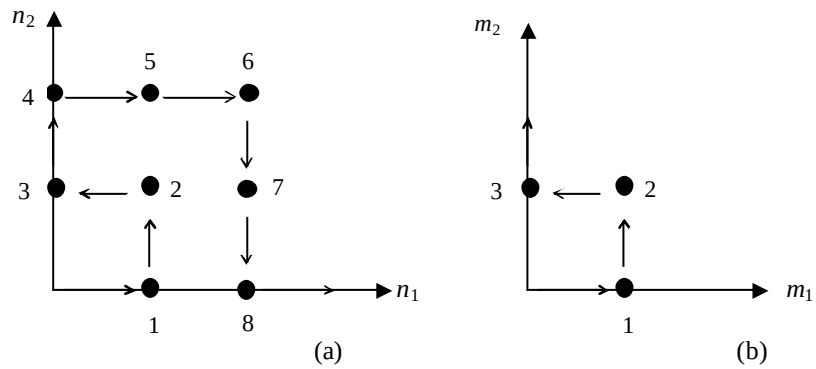


Figure 8-40: FQP ordering schemes of the Computer Experiment 14 - ARMA(3,8) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.14.1. Objectives of Computer Experiment 14

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(3,8) parameter calculation formulas given in Equations (4.3) and (5.13), respectively, for the case which the MA order is greater than the AR order ($M=3$, $N=8$ and QP support) and the parameter b_0 is different than 1 ($b_0=0.8$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.14.2. Results of Computer Experiment 14

The results of the Computer Experiment 14 have been given in Table 8.32 and Table 8.33 for different data sizes. Table 8.32 lists the original ARMA(3,8) parameters versus the identified ARMA(3,8) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.33 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a 10x10 data field size have been given in Figure 8.41 and Figure 8.42, respectively.

Table 8.32: Computer Experiment 14 - ARMA(3,8) parameter identification results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Original ARMA(3,8) Parameters		Identified ARMA Parameters		
		LAT and LS		
		Data Field Size		
		10x10	50x50	100x100
AR Parameters	0.3	0.3878	0.3063	0.3154
		0.3942	0.2986	0.3051
	-0.3	-0.0913	-0.2184	-0.2251
		-0.1303	-0.2431	-0.2751
	0.2	0.4097	0.2893	0.2830
		0.3274	0.2766	0.2297
MA Parameters	0.2	0.1929	0.2010	0.1994
		0.2355	0.2062	0.2058
	0.2	0.4131	0.2310	0.2258
		0.3985	0.2435	0.2195
	0.2	0.2981	0.4315	0.3745
		0.2996	0.2627	0.2233
	-0.2	-0.1681	-0.2038	-0.1932
		-0.2054	-0.1975	-0.1986
	0.2	0.3399	0.2430	0.2384
		0.2431	0.2352	0.2142
	-0.3	-0.1827	-0.2357	-0.2403
		-0.1874	-0.2556	-0.2808
	0.3	0.1952	0.2496	0.2592
		0.2655	0.2587	0.2868
-0.5	-0.4853	-0.4967	-0.5047	
	-0.5414	-0.5029	-0.5002	
b_0	0.8	0.8281	0.8585	0.8580
		0.7402	0.7949	0.7955

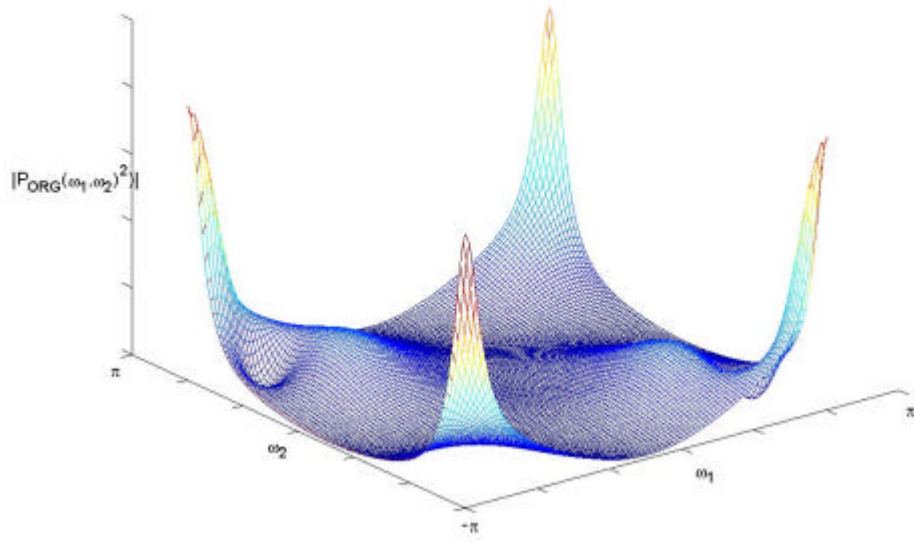
Table 8.33: Computer Experiment 14 – Performance Criteria results of Hybrid Lattice algorithm shown in the first row and LS algorithm shown in the second row.

Performance Criteria	Data Field Size		
	10x10	50x50	100x100
L₁-norm	1.2613	0.6639	0.5827
	1.0218	0.3787	0.1609
L₂-norm	0.4435	0.2849	0.2324
	0.3543	0.1412	0.0570
L_∞-norm	0.2131	0.2315	0.1745
	0.1985	0.0766	0.0297
Itakura-Saito Distance	0.364127	0.273299	0.271012
	0.447460	0.048981	0.004700

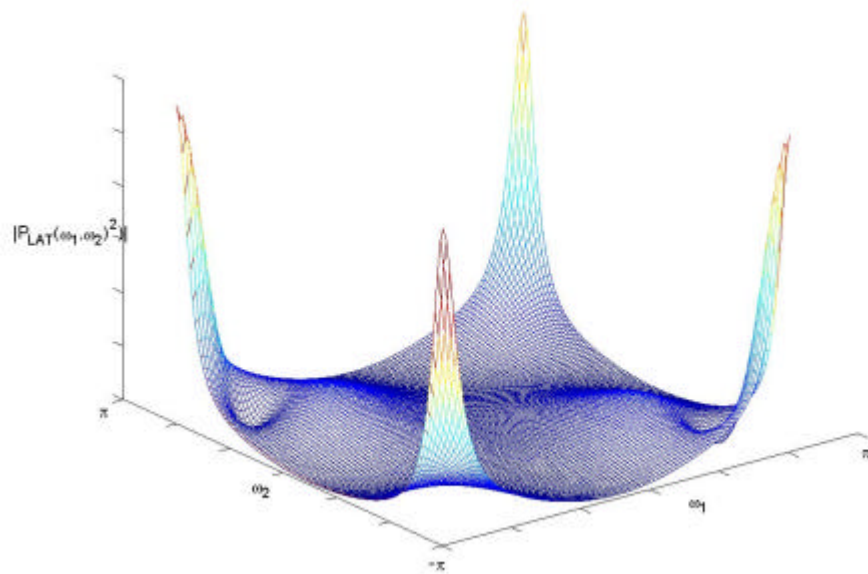
The obtained results yield the following conclusions for this experiment:

- As it can easily be seen from Table 8.28 and Table 8.29, the identification results are quite accurate and improve with the increasing data size.
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- Even for a small data field size of 10x10, ARMA(15,8) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.

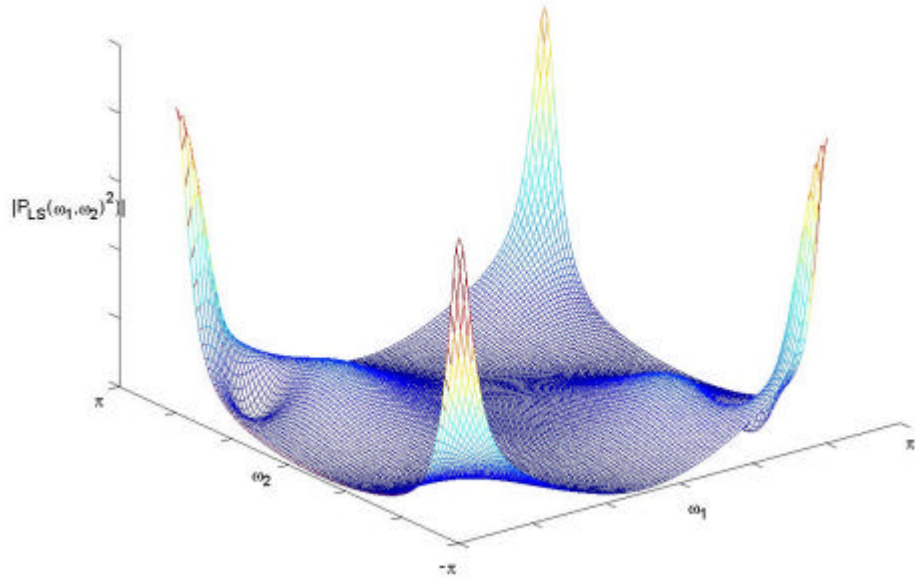
- By looking at the Itakura-Saito distances listed in Table 8.29 and the power spectrum and contour plots given in Figure 8.35 and Figure 8.36, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.



(a)

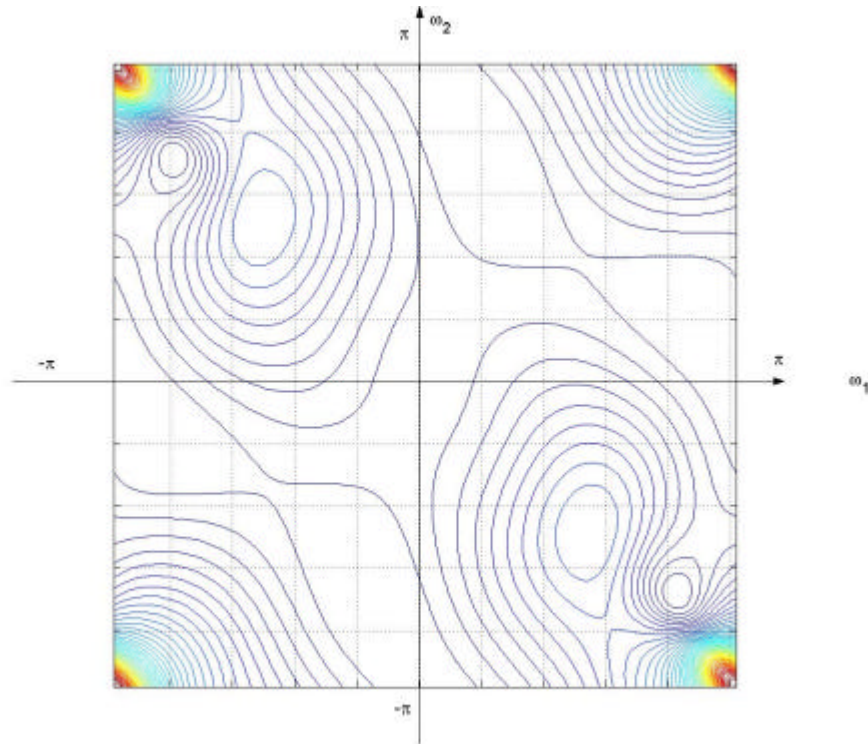


(b)

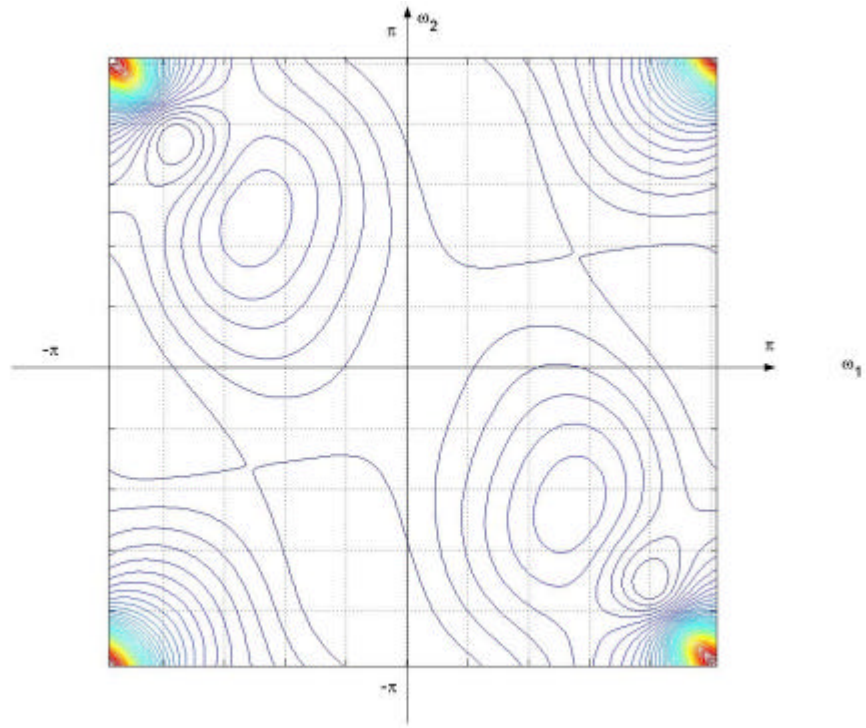


(c)

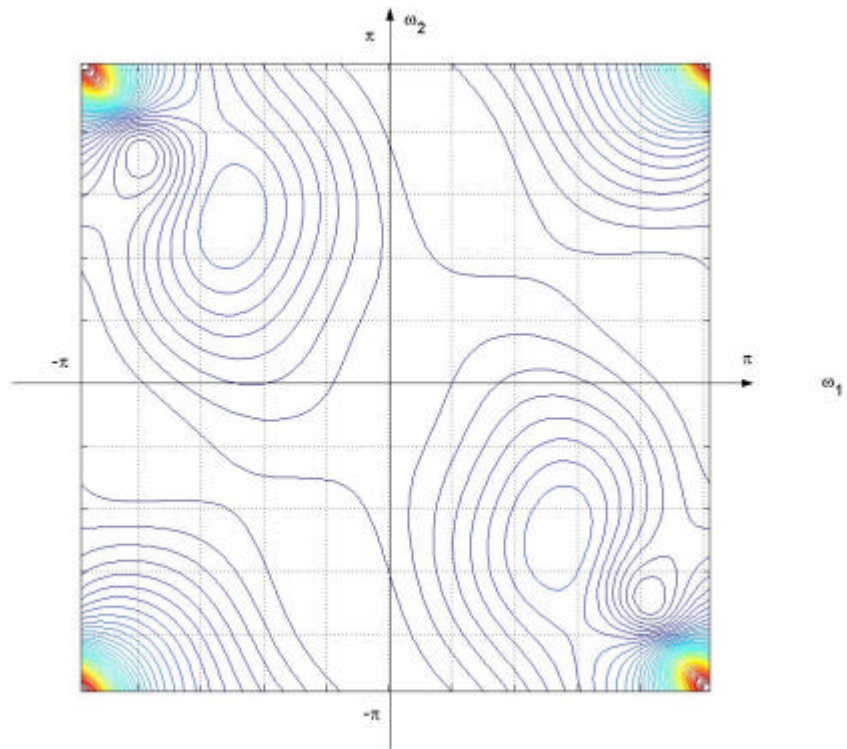
Figure 8-41: Computer Experiment 14 – Power Spectrums plotted using (a) Original ARMA(3,8) parameters (b) ARMA(3,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,8) parameters identified by the LS algorithm. The parameter $b_0=0.8$ and the data field size is 50x50.



(a)



(b)



(c)

Figure 8-42: Computer Experiment 14 – Contour Plots of (a) Original ARMA(3,8) parameters (b) ARMA(3,8) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(3,8) parameters identified by the LS algorithm. The parameter $b_0=0.8$ and the data field size is 50x50.

8.15. Computer Experiment 15

The unknown 2-D ARMA(8,15) system has a FQP support of the different order AR and MA polynomials, N being greater than M ($M=8$ and $N=15$). The input $x(k_1, k_2)$ is a white noise process with mean zero and variances s_x^2 . The horizontal-vertical FQP ordering schemes used for $x(k_1, k_2)$ and $y(k_1, k_2)$ for $M=8$ and $N=15$ are illustrated in Figure 8.43.

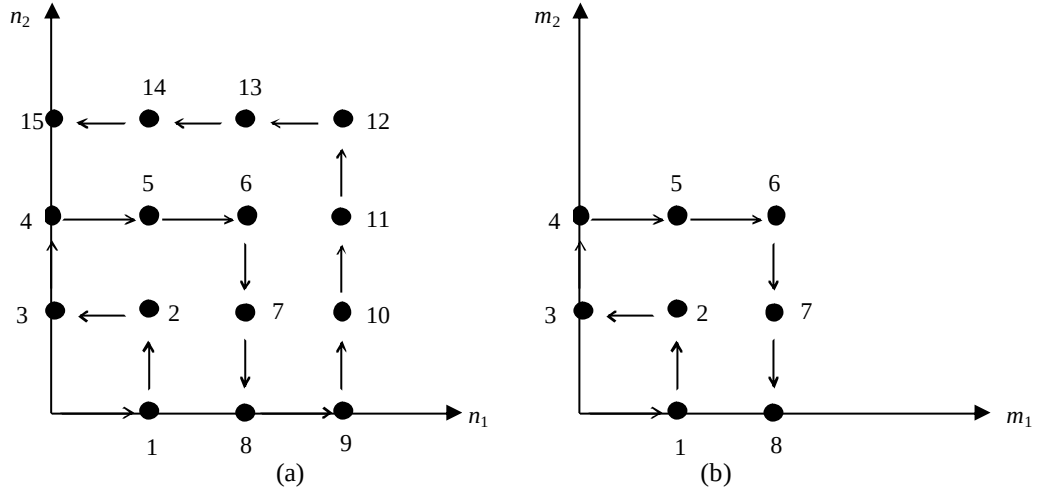


Figure 8-43: FQP ordering schemes of the Computer Experiment 15 - ARMA(8,15) system for (a) the system input $x(k_1, k_2)$ and (b) the system output $y(k_1, k_2)$.

8.15.1. Objectives of Computer Experiment 15

The objectives of this computer simulation experiment can be listed as follows:

- To verify the proposed 2-D hybrid lattice algorithm and to prove numerically the derived b_0 and ARMA(8,15) parameter calculation formulas given in Equations (4.3) and (5.13), respectively, for the case which the MA order is greater than the AR order ($M=8$, $N=15$ and QP support) and the parameter b_0 is different than 1 ($b_0=0.85$).
- To use the LS algorithm as a means for verification.
- To observe the proposed 2-D hybrid lattice performance with the increasing data size.
- To evaluate the proposed 2-D hybrid lattice performance in terms of power spectrum matching by use of Itakura-Saito distance measure.
- To evaluate the proposed 2-D hybrid lattice performance in terms of identified and original parameter vector matching by use of L_1 , L_2 and L_∞ vector norms.

8.15.2. Results of Computer Experiment 15

The results of the Computer Experiment 15 have been given in Table 8.34 and Table 8.35 for different data sizes. Table 8.34 lists the original ARMA(8,15) parameters versus the identified ARMA(8,15) parameters obtained by 2-D hybrid lattice and LS methods for different data sizes. Table 8.35 lists the performance evaluation results of 2-D hybrid lattice and LS methods versus the data field sizes. In both tables, the first row shows the Lattice identification results and the second row shows the LS identification results. The power spectrums and contours plots of the original parameters versus the Lattice and LS identified parameters for a 10x10 data field size have been given in Figure 8.44 and Figure 8.45, respectively.

Table 8.34: Hybrid Lattice and LS identified versus original ARMA coefficients for Computer Experiment 15.

Original ARMA(8,15) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
AR Parameters	0.1000	-0.0480 0.0991	0.0085 0.1001	0.0006 0.0999
	0.1500	0.0890 0.1397	0.1780 0.1496	0.1884 0.1495
	0.0500	0.0649 0.0347	0.0216 0.0491	0.0238 0.0487
	0.2000	0.3844 0.2354	0.2340 0.2015	0.2189 0.2020
	0.1000	0.0800 0.0720	0.0917 0.0976	0.1008 0.0990
	0.2500	0.2281 0.2612	0.2405 0.2510	0.2207 0.2496
	0.0500	-0.0863 0.0557	0.0156 0.0508	0.0231 0.0501
	0.1000	0.1525 0.1068	0.0947 0.1000	0.0697 0.1001
	0.1000	-0.0111 0.1019	0.0515 0.0997	0.0486 0.0998
	0.1200	0.0514 0.1123	0.1246 0.1196	0.1259 0.1196

Original ARMA(8,15) Parameters		Identified ARMA Parameters (Mean Values)		
		Data Field Size		
		10x10	50x50	100x100
MA Parameters	0.1300	0.1138 0.1176	0.1098 0.1291	0.1062 0.1290
	0.1400	0.2909 0.1631	0.1541 0.1393	0.1401 0.1407
	0.1500	0.1178 0.1283	0.1339 0.1500	0.1392 0.1500
	0.1600	0.1127 0.1628	0.1399 0.1614	0.1301 0.1598
	0.1700	0.0764 0.1776	0.1388 0.1709	0.1328 0.1699
	0.1800	0.2176 0.1882	0.1817 0.1795	0.1626 0.1800
	0.1900	0.2029 0.1903	0.1839 0.1899	0.1796 0.1901
	-0.1000	-0.1290 -0.1129	-0.1121 -0.1007	-0.1119 -0.1003
	-0.1000	-0.0503 -0.0916	-0.0761 -0.1000	-0.0817 -0.0994
	-0.1000	-0.1497 -0.1022	-0.0968 -0.0999	-0.0882 -0.1001
	-0.1000	-0.0734 -0.0980	-0.0840 -0.1005	-0.0827 -0.1000
	-0.1000	-0.0819 -0.0984	-0.0886 -0.1005	-0.0897 -0.0995
	-0.1000	-0.1146 -0.0787	-0.0986 -0.0980	-0.0982 -0.1001
b_0	0.8500	0.8415 0.8522	0.8585 0.8497	0.8597 0.8501

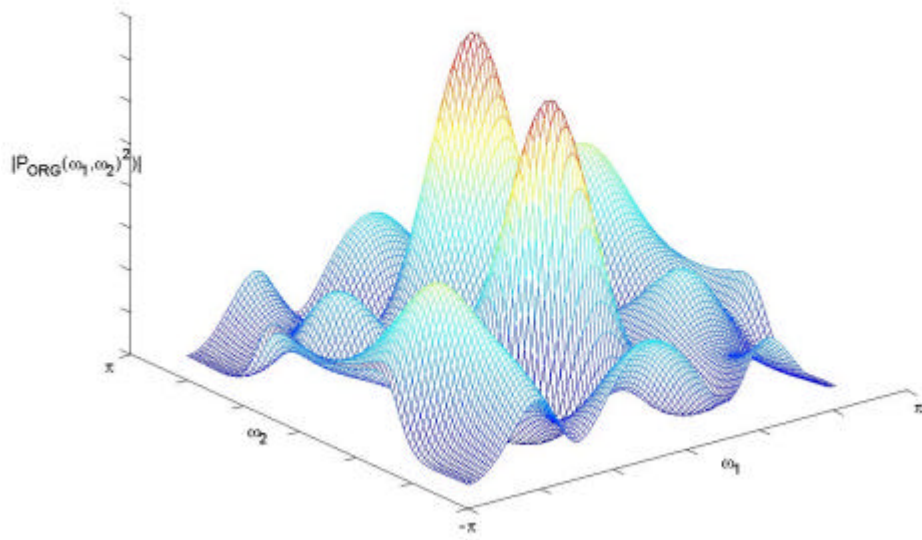
Table 8.35: Performance criteria results of Hybrid Lattice and LS algorithms for Computer Experiment 15.

Performance Criteria	Data Field Size		
	10x10	50x50	100x100
L₁-norm	4.3590	3.9624	3.9374
	0.2499	0.0163	0.0100
L₂-norm	1.0632	0.9997	0.9834
	0.0681	0.0045	0.0031
L_y-norm	0.4309	0.3739	0.3696
	0.0354	0.0024	0.0020
Itakura-Saito Distance	0.020283	0.005132	0.007182
	0.001917	0.000097	0.000007

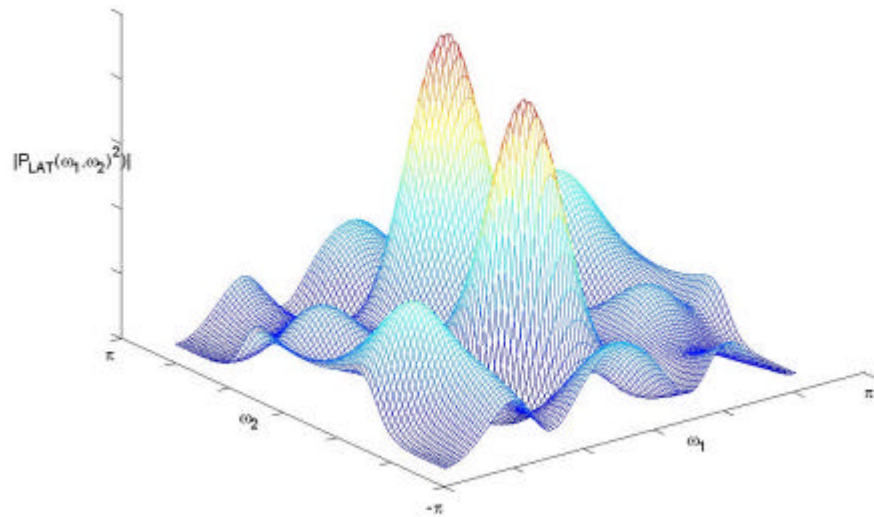
The obtained results yield the following conclusions for this experiment:

- As it can easily be seen from Table 8.34 and Table 8.35, the identification results are quite accurate and improve with the increasing data size.
- The b_0 parameter estimation results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original value, thus verifying numerically the formulation derived in Equation (4.3).
- The identification results obtained with the proposed 2-D hybrid lattice algorithm compares well with the original parameters and the LS identification results, thus verifying numerically the formulation derived in Equation (5.6).
- By looking at the Itakura-Saito distances listed in Table 8.29 and the power spectrum and contour plots given in Figure 8.35 and Figure 8.36, we can say that there is a close match between the power spectrums obtained using original and identified parameters obtained with the proposed 2-D hybrid lattice algorithm.

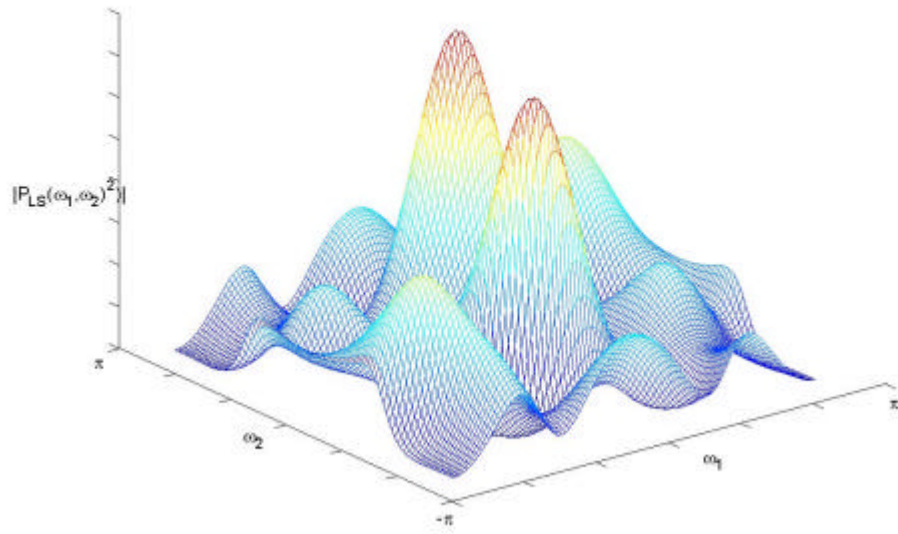
- Even for a small data field size of 10x10, ARMA(8,15) coefficients identified with the proposed 2-D hybrid lattice algorithm are quite acceptable.



(a)

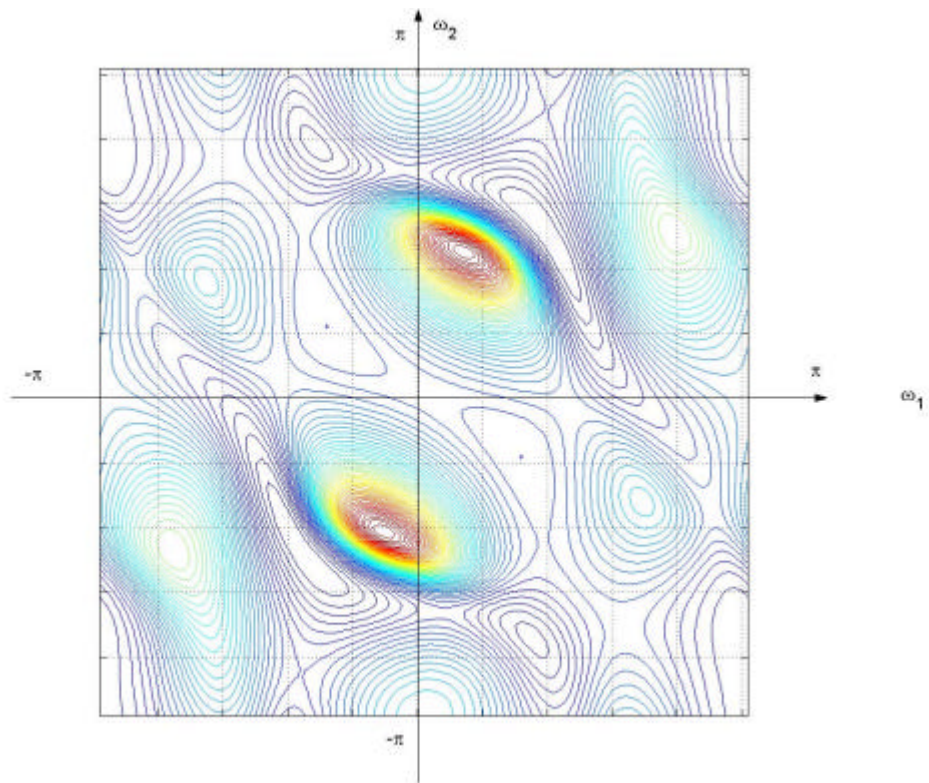


(b)

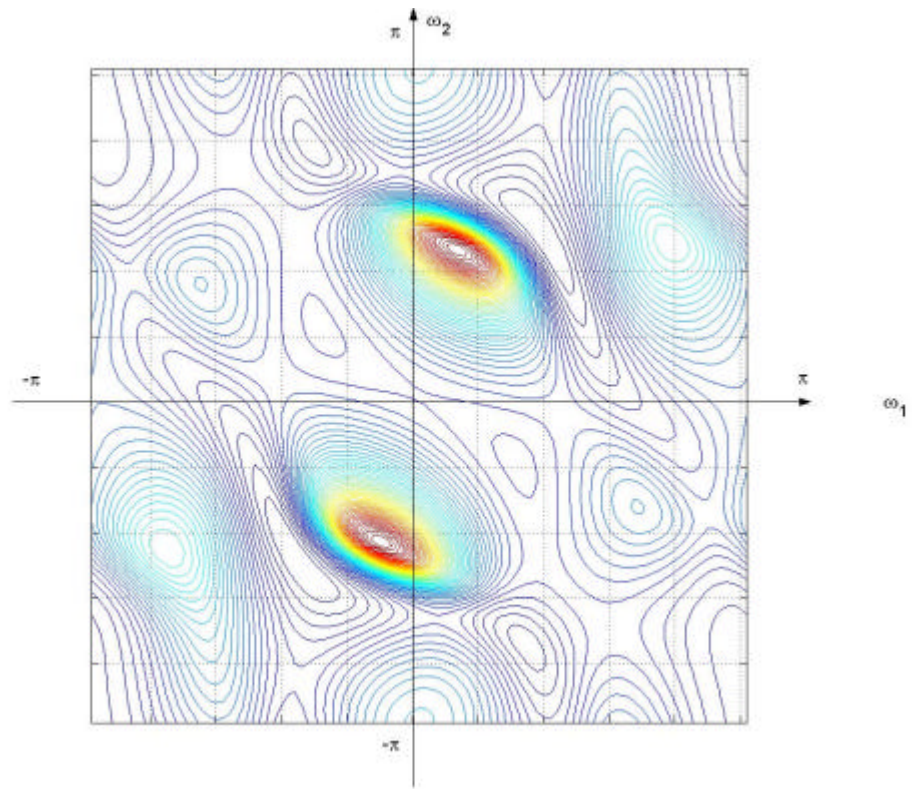


(c)

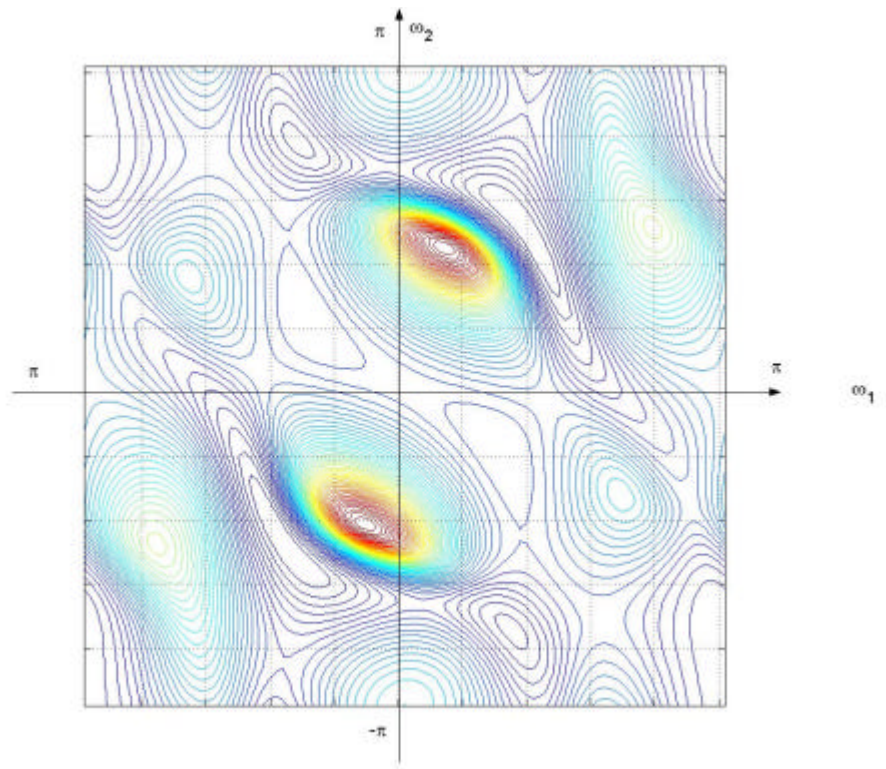
Figure 8-44: Computer Experiment 15 – Power Spectrums plotted using (a) Original ARMA(8,15) parameters (b) ARMA(8,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,15) parameters identified by the LS algorithm. The parameter $b_0=0.85$ and the data field size is 50x50.



(a)



(b)



(c)

Figure 8-45: Computer Experiment 15 – Contour plots of (a) Original ARMA(8,15) parameters (b) ARMA(8,15) parameters identified by the Hybrid Lattice algorithm and (c) ARMA(8,15) parameters identified by the LS algorithm. The parameter $b_0=0.85$ and the data field size is 50x50.

8.16. Computer Experiment 16

In this experiment a 100x100 simulated texture image has been generated using Equation (7.1). The AR coefficients and the ASHP ordering chosen are given in difference equation form in (8.8).

$$s(m, n) = 0.5s(m-1, n) + 0.3s(m-1, n-1) - 0.4s(m, n-1) - 0.4s(m-1, n+1) + w(m, n) \quad (8.8)$$

The original simulated image given in Equation (8.8) has been blurred using Equation (7.2), but the effect of observation noise $v(m, n)$ has been neglected. The 3x3 blur PSF coefficients (MA coefficients) are given in Equation (8.9) and they are chosen so as to satisfy Equation (7.4).

$$\begin{aligned} r(m, n) = & 0.36s(m, n) + 0.11s(m-1, n) + 0.05s(m-1, n-1) + 0.11s(m, n-1) \\ & + 0.05s(m+1, n-1) + 0.11s(m+1, n) + 0.05s(m+1, n+1) \\ & + 0.11s(m, n+1) + 0.05s(m-1, n+1) \end{aligned} \quad (8.9)$$

The original and blurred simulated texture images, which are $s(m, n)$ and $r(m, n)$ respectively, are given in Figure 8.46.

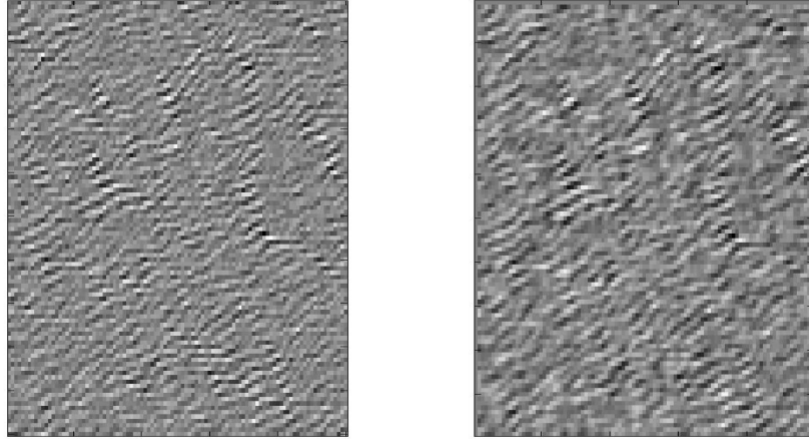


Figure 8-46: Original and blurred simulated texture images. The effect of observation noise $v(m, n)$ has been neglected.

The hybrid lattice structure designed for this computer experiment corresponds to the 2-D identification of an ARMA (4,8) model. The identification results are given in Table 8.36 and 8.37.

8.16.1. Objectives of Computer Experiment 16

The objectives of this computer simulation experiment can be listed as follows:

- To verify the applicability of the proposed 2-D hybrid lattice algorithm to the simultaneous image and blur parameter identification problem.
- To use a simulated image as a means for verification since the original image parameters are known.

8.16.2. Results of Computer Experiment 16

Table 8.36: Original and identified image AR model coefficients: Simulated image, $v(m,n)$ neglected.

$a_{k,l}$	Original Values	Identified Values
$a_{1,0}$	0.5	0.5923
$a_{1,1}$	0.3	0.2412
$a_{0,1}$	-0.4	-0.4271
$a_{1,-1}$	-0.4	-0.4672

Table 8.37: Original and identified values of blur PSF: Simulated image, $v(m,n)$ neglected.

b_{ij}	Original Values	Identified Values
$b_{-1,1}$	0.05	0.0936
$b_{0,1}$	0.11	0.1111
$b_{1,1}$	0.05	0.0405
$b_{-1,0}$	0.11	0.1163
$b_{0,0}$	0.36	0.3711
$b_{1,0}$	0.11	0.1189
$b_{-1,-1}$	0.05	0.0417
$b_{0,-1}$	0.11	0.1128
$b_{1,-1}$	0.05	0.0515

8.17. Computer Experiment 17

In this experiment the 128x128 “cameraman image” has been used. The image is assumed to have an ASHP support given in Equation (8.8). The original cameraman image has been blurred using Equation (8.9) and the effect of observation noise $v(m, n)$ has been neglected.

The hybrid lattice structure designed for this computer experiment corresponds to the 2-D identification of an ARMA (4,8) model. The identification results are given in Table 8.38 and Table 8.39.

The original and blurred “cameraman” images are given in Figure 8.47.



Figure 8-47: Original and blurred cameraman images. The effect of observation noise $v(m, n)$ has been neglected.

8.17.1. Objectives of Computer Experiment 17

The objectives of this computer simulation experiment can be listed as follows:

- To verify the applicability of the proposed 2-D hybrid lattice algorithm to the simultaneous image and blur parameter identification problem.
- To test the validity of the algorithm for real images.

8.17.2. Results of Computer Experiment 17

Table 8.38: Original and identified values of blur PSF: Cameraman image, $v(m,n)$ neglected.

b_{ij}	Original Values	Identified Values
$b_{-1,1}$	0.05	0.0697
$b_{0,1}$	0.11	0.0616
$b_{1,1}$	0.05	0.0124
$b_{-1,0}$	0.11	0.1402
$b_{0,0}$	0.36	0.3369
$b_{1,0}$	0.11	0.0755
$b_{-1,-1}$	0.05	0.0315
$b_{0,-1}$	0.11	0.1525
$b_{1,-1}$	0.05	0.0621

Table 8.39: Original and identified image AR coefficients:Ca meraman image, $v(m,n)$ neglected.

$a_{k,l}$	Original Values	Identified Values
$a_{1,0}$	0.6350	0.7076
$a_{1,1}$	-0.3462	-0.3138
$a_{0,1}$	0.5604	0.5493
$a_{1,-1}$	0.1373	0.0471

8.18. Computer Experiment 18

In this experiment the 128x128 “cameraman image” has been used. The image is assumed to have an ASHP support given in Equation (8.8). The original cameraman image has been blurred using Equation (8.9), but effect of observation noise $v(m, n)$ has not been neglected.

The hybrid lattice structure designed for this computer experiment corresponds to the 2-D identification of an ARMA (4,8) model. The identification results are given in Table 8.40 and Table 8.41.

The original and blurred “cameraman” images are given in Figure 8.48. The SNR in blurred image is 25 dB.

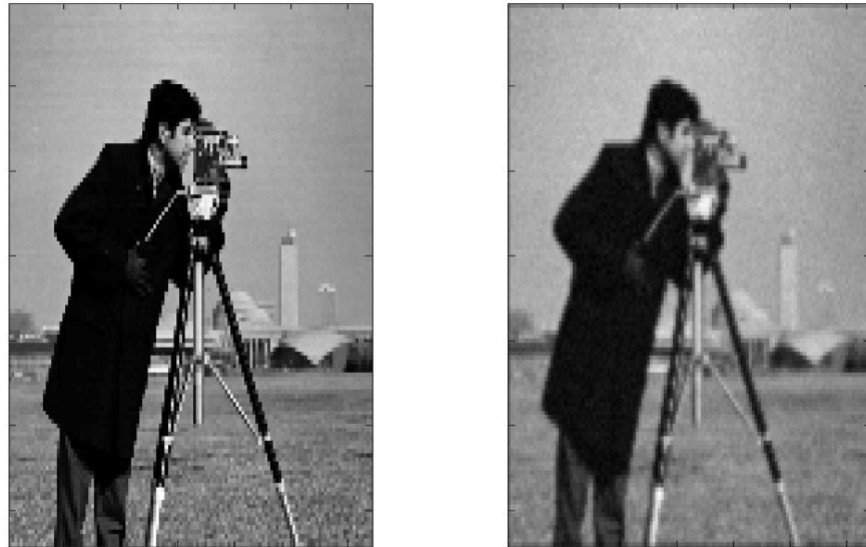


Figure 8-48: Original and blurred cameraman images. The effect of observation noise $v(m, n)$ has not been neglected.

8.18.1. Objectives of Computer Experiment 18

The objectives of this computer simulation experiment can be listed as follows:

- To verify the applicability of the proposed 2-D hybrid lattice algorithm to the simultaneous image and blur parameter identification problem.
- To test the validity of the algorithm for real images when observation noise $v(m, n)$ is not negligible.

8.18.2. Results of Computer Experiment 18

Table 8.40: Original and identified values of blur PSF: Cameraman image, $v(m, n)$ not neglected.

b_{ij}	Original Values	Identified Values
$b_{-1,1}$	0.05	0.0627
$b_{0,1}$	0.11	0.1371
$b_{1,1}$	0.05	0.0316
$b_{-1,0}$	0.11	0.1191
$b_{0,0}$	0.36	0.0115
$b_{1,0}$	0.11	0.1456
$b_{-1,-1}$	0.05	0.0609
$b_{0,-1}$	0.11	0.1346
$b_{1,-1}$	0.05	0.0627

Table 8.41: Original and identified image AR coefficients: Cameraman image, $v(m, n)$ not neglected.

$a_{k,l}$	Original Values	Identified Values
$a_{1,0}$	0.6350	0.4987
$a_{1,1}$	-0.3462	-0.0443
$a_{0,1}$	0.5604	0.3869
$a_{1,-1}$	0.1373	0.1472

9. CONCLUDING REMARKS

We have proposed a new algorithm for identifying the parameters of an unknown 2-D ARMA (M, N) LSI system where M and N can take arbitrary values different from each other. We have called this algorithm as “2-D hybrid lattice” since it incorporates both two-channel and single channel lattice stages in an interleaved manner.

The advantages of our proposed algorithm can be listed as follows:

- The order of AR and MA polynomials of the ARMA system, namely M and N , can take arbitrary values different from each other.
- The parameter b_0 , which is generally assumed to be “1” is estimated for the cases $M \geq N$ and $M < N$.
- This hybrid lattice algorithm inherits the modular structure of the conventional lattices provided that the ordering of the 2D data points is appropriately chosen.
- The new formulation derived for the ARMA parameter estimates, $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$, takes the estimated parameter $\hat{b}_0$ and the forward prediction error vectors tap weights of both channels into account, thus resulting in a more reliable parameter identification. The proofs of the new formulations proposed have been given in Appendices.
- The AR modeled system input parameters and ARMA modeled system output parameters can be identified simultaneously, provided that the input AR model order is N , while the output ARMA model order is (M, N) .
- The proposed 2-D hybrid lattice algorithm can be applied to the simultaneous identification of image and blur parameters since it was proved that blurred observed image could be represented by an ARMA model.

The major disadvantage coming with our proposed method is that since we use a signal $u(k_1, k_2)$ which is the difference between $y(k_1, k_2)$ and $x(k_1, k_2)$, the data points of both the input and output signals in the prediction region should have the same ordering at least for the filter orders extending to the smaller of the (M, N) pair.

Another disadvantage is that we assume that the orders M, N are known, there is no order determination procedure included.

In order to show the efficiency of the proposed method, some computer simulations have been carried out covering the prediction support regions in FQP and ASHP for different M and N values. The parameter identification results obtained have been shown in related tables. In order to make a comparison, power spectrums and contour plots obtained using the original and identified parameters have been given for each of the simulations. The Itakura-Saito distance measure, which indicates the similarity between the original and identified power spectrums, L_1 , L_2 and L_∞ vector norms have been used as performance measures.

It is easy to judge by looking at the identification results listed in tables, at the power spectrum and contour plots and the Itakura-Saito distance measure that, the obtained results are quite satisfactory and they improve with the increasing data sizes. All of the computer simulations have been run for 100 independent Monte-Carlo simulations and the mean values have been given as the parameter identification results. Since these mean values are quite close to the true values, especially for the cases when $M=N$, it is possible to say that these parameter identification results form unbiased estimates.

We can list the following items for future work.

- A proof about the stability and optimality of the algorithm can be included.
- A simultaneous order determination procedure can be included.
- The properties of the estimates can be examined in terms of biasedness, efficiency and Fisher consistency, etc.
- In terms of adaptive modeling and identification, this research can be extended to cover the adaptive ARMA modeling and joint-process estimation.
- The hybrid lattice algorithm can be rearranged to use with 1-D signals, especially in speech processing applications.

10. REFERENCES

- [1] **Perry, F.A., 1980.** Parametric Modeling of Linear and Nonlinear Systems, *Ph.D. Dissertation*, Naval Postgraduate School, Monterey, California.
- [2] **Perry, F.A. and Parker, S.R., 1984.** Transition Formulas for Zero-Pole Modeling, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-32(1)**, 178-180.
- [3] **Lee, D.T.L., Friedlander, B. and Morf, M., 1982.** Recursive Ladder Algorithms for ARMA modeling, *IEEE Transactions on Automatic Control*, **27(4)**, 753-764.
- [4] **Moses, R.L., Cadzow, J.A. and Beex, A.A.L., 1985.** A Recursive Procedure for ARMA Modeling, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-33(4)**, 1188-1196.
- [5] **Grenier, Y., 1983.** Time-dependent ARMA Modeling of Nonstationary Signals, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-31(4)**, 899-911.
- [6] **Grenier, Y., 1986.** Nonstationary ARMA Models via Simultaneous AR and MA Estimation, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Tokyo, Japan, April 7-11, pp.2339-2342.
- [7] **Levinson, N., 1946.** The Wiener RMS (root mean square) Error Criterion in Filter Design and Prediction, *J. Math. Phys.*, **25(1)**, 261-278.
- [8] **Burg, J.P., 1975.** Maximum Entropy Spectral Analysis, *Ph.D. Dissertation*, Stanford University, Stanford, CA.
- [9] **Karlsson, E. and Hayes, M.H., 1986.** ARMA Modeling of Time-Varying Systems with Lattice Filters, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Tokyo, Japan, 7-11 April, pp. 2335-2338.
- [10] **Karlsson, E. and Hayes, M.H., 1987.** Least Squares ARMA Modeling of Linear Time-Varying Systems: Lattice Filter Structures and Fast RLS

Algorithms, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-35(7)**, 994-1014.

- [11] **Karlsson, E. and Hayes, M.H.**, 1987. Performance Analysis of New Least Squares ARMA Lattice Modeling Algorithms, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, TX USA, 6-9 April, pp. 984-987.
- [12] **Ortigueira, M.D.**, 1988. A Three Step Algorithm for ARMA Modeling, *Signal Processing*, **15**, 23-30.
- [13] **Rayala, J.D. and Reddy, D.C.**, 1988. A New Technique for ARMA System Modeling, *IEEE International Symposium on Circuits and Systems*, Espoo, Finland, 7-9 June, pp.1349-1351.
- [14] **Swami, A., Giannakis, G.B. and Mendel, J.M.**, 1989. A Unified Approach to Modeling Multichannel ARMA Processes, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Glasgow, UK, 23-26 May, pp.2182-2185.
- [15] **Swami, A., Giannakis, G.B. and Shamsunder, S.**, 1994. Multichannel ARMA Processes, *IEEE Transactions on Signal Processing*, **42(4)**, 898-913.
- [16] **Chakraborty, M. and Prasad, S.**, 1993. Multivariate ARMA Modeling by Scalar Algorithms, *IEEE Transactions on Signal Processing*, **41(4)**, 1692-1697.
- [17] **Chakraborty, M. and Prasad, S.**, 1994. Multichannel ARMA Modeling by Least Squares Circular Lattice Filtering, *IEEE Transactions on Signal Processing*, **42(9)**, 2304-2318.
- [18] **Swami, A., Giannakis, G.B. and Mendel, J.M.**, 1989. Computation of Cumulants of ARMA processes, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Glasgow, UK, 23-26 May, pp. 2318-2321.
- [19] **Therrien, C.W. and May, G.L.**, 1991. Comparison of ARMA Modeling Methods in the Time Domain, *Proceedings of the 34th Midwest Symposium on Circuits and Systems*, Monterey, CA USA, 14-17 May, pp.443-446.
- [20] **Prasad, S. and Joshi, S.D.**, 1992. A new recursive pseudo least squares algorithm for ARMA filtering and modeling. I, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-40(11)**, 2766-2774.

- [21] **Prasad, S. and Joshi, S.D.**, 1992. A new recursive pseudo least squares algorithm for ARMA filtering and modeling. II, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-40(11)**, 2775-2783.
- [22] **Min, B.J. and Un, C.K.**, 1994. A Fast Transversal Filtering Algorithm for Pole-Zero Modeling, *Signal Processing*, **37**, 61-72.
- [23] **Therrien, C.W. and Velasco, C.H.**, 1995. An Iterative Prony Method for ARMA Signal Modeling, *IEEE Transactions on Signal Processing*, **43(1)**, 358-361.
- [24] **Walker, A.M.**, 1962. Large Sample Estimation of Parameters for Autoregressive Processes with Moving Average Residuals, *Biometrika*, **49**, 117-131.
- [25] **Solo, V.**, 1983. Finite Data Lattice Algorithms for Instrumental Variable Recursions, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-31(5)**, 1202-1210.
- [26] **Bruzzzone, S. and Kaveh, M.**, 1984. Information Tradeoffs in Using the Sample Autocorrelation Function in ARMA Parameter Estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-32(4)**, 701-715.
- [27] **Friedlander, B. and Porat, B.** 1984. A Spectral Matching Technique for ARMA Parameter Estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-32(2)**, 338-343.
- [28] **Porat, B. and Friedlander, B.**, 1986. Bounds on the Accuracy of Gaussian ARMA Parameter Estimation Methods Based on Sample Covariances, *IEEE Transactions on Automatic Control*, **AC-31(6)**, 579-582.
- [29] **Li, S. and Dickinson, B.W.**, 1986. An Efficient Method to Compute Consistent Estimates of the AR Parameters of an ARMA Model, *IEEE Transactions on Automatic Control*, **AC-31(3)**, 275-278.
- [30] **Tsay, R.S. and Tiao, G.C.**, 1984. Consistent Estimates of Autoregressive Parameters and Extended Sample Autocorrelation Function for Stationary and Nonstationary ARMA Models, *Journal of the American Statistical Association*, **79 (385)**, 84-96.
- [31] **Li, S. and Dickinson, B.W.**, 1988. Application of the Lattice Filter to Robust Estimation of AR and ARMA Models, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **36(4)**, 502-512.

- [32] **Moura, J.M.F and Ribeiro, M.I.**, 1988. Linear Estimation of ARMA Processes, *Fourth Annual ASSP Workshop on Spectrum Estimation and Modeling*, Minneapolis, Minnesota, 3-5 August, pp.229-234.
- [33] **Stoica, P., Friedlander, B. and Söderström, T.**, 1988. A High-Order Yule-Walker Method for Estimation of the AR Parameters of an ARMA Model, *Systems & Control Letters*, **11**, 9-105.
- [34] **Vural, F.Y., Wawrzynski, D. and Çetin, A.E.**, 1989. The Generalized *H*-Lattice Filter and Estimation of the ARMA Models, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37(1)**, 147-151.
- [35] **Sharma, P.L.**, 1989. Estimation of ARMA Models Using Generalized *H*-Lattice Filters, *Proceedings of the 32nd Midwest Symposium on Circuits and Systems*, Champaign, IL USA, 14-16 August, pp.1131-1133.
- [36] **Kwan, H.K. and Lui, Y.C.**, 1989. Normalized ARMA Levinson Algorithm, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Portland, OR, 8-11 May, pp.234-237.
- [37] **Zerubia, J. and Alengrin, G.**, 1991. Estimation of ARMA (*p*, *q*) Parameters, *Signal Processing*, **22**, 53-60.
- [38] **Alengrin, G. and Zerubia, J.**, 1987. A method to Estimate the parameters of an ARMA model, *IEEE Transactions on Automatic Control*, **32(12)**, 1113-1117.
- [39] **Alengrin, G. and Zerubia, J.**, 1988. Estimation of ARMA parameters and Kalman Filtering, *8th IFAC / IFORS Symposium*, Beijing, PRChina, 27-31 August.
- [40] **Chakraborty, M. and Prasad, S.**, 1994. Computation of a Useful Cramer-Rao Bound for Multichannel ARMA Parameter Estimation, *IEEE Transactions on Signal Processing*, **42(2)**, 466-469.
- [41] **Sabiti, J.K.**, 1996. A Fast Estimation Method for ARMA Processes, *Automatica*, **32(2)**, 235-239.
- [42] **Stoica, P., Mari, J. and McKelvey, T.**, 1999. Vector ARMA Estimation: An enhanced Subspace Approach, *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, AZ USA, 7-10 December, pp.3665-3670.

- [43] **Mari, J., Stoica, P. and McKelvey, T.,** 2000. Vector ARMA Estimation: A Reliable Subspace Approach, *IEEE Transactions on Signal Processing*, **48(7)**, 2092–2104.
- [44] **Chon, K.H. and Lu, S.,** 2001. A New Algorithm for Autoregression Moving Average Model Parameter Estimation Using Group Method of Data Handling, *Annals of Biomedical Engineering*, **29(1)**, 92-98.
- [45] **Lu, S., Ju, K.H. and Chon, K.H.,** 2001. A New Algorithm for Linear and Nonlinear ARMA Model Parameter Estimation Using Affine Geometry, *IEEE Transactions on Biomedical Engineering*, **48(10)**, 1116-1124.
- [46] **Box, G.E.P and Jenkins, G.M.,** 1970. Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco, CA.
- [47] **Akaike, H.,** 1973. Maximum Likelihood Identification of Gaussian Autoregressive Moving Average Models, *Biometrika*, **60**, 255-265.
- [48] **Graupe, D., Krause, D.J. and Moore, J.B.,** 1975. Identification of Autoregressive Moving-Average Parameters of Time Series, *IEEE Transactions on Automatic Control*, **AC-20(1)**, 104-107.
- [49] **Brotherton, T.W. and Caines, P.E.,** 1978. ARMA System Identification via the Cholesky Least Squares Method, *IEEE Transactions Automatic Control*, **AC-23(4)**, 698-702.
- [50] **Benveniste, A. and Chaure, C.,** 1981. AR and ARMA Identification Algorithms of Levinson Type: An Innovations Approach, *IEEE Transactions on Automatic Control*, **AC-26(6)**, 1243-1261.
- [51] **Parker, S.R. and Perry, F.A.,** 1981. A discrete ARMA Model for Nonlinear System Identification, *IEEE Transactions on Circuits and Systems*, **CAS-28(3)**, 224-233.
- [52] **Mayne, D.Q. and Firoozan, F.,** 1982. Linear Identification of ARMA Processes, *Automatica*, **18(4)**, 461-466.
- [53] **Hannan, E.J. and Kavalieris, L.,** 1983. Linear Estimation of ARMA Processes, *Automatica*, **19(4)**, 447-448.
- [54] **Lim, Y.C. and Parker, S.R.,** 1986. On the Identification of Systems from Data Measurements Using ARMA Lattice Models, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-34(4)**, 824-828.

- [55] **Alengrin, G., Bucy, R.S., Moura, J., Pages, J. and Ribeiro, I.**, 1987. ARMA Identification, *Journal of Optimization Theory and Applications*, **55(3)**, 345-357.
- [56] **Strobach, P.**, 1988. Recursive Covariance Ladder Algorithms for ARMA System Identification, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **36(4)**, 560-580.
- [57] **Miyanaga Y., Tochinaj, K., Nagaj, N and Nagata, K.**, 1990. Adaptive Identification of a Modified ARMA Lattice Model, *IEEE Transactions on Circuits and Systems*, **37(9)**, 1140-1144.
- [58] **Ribeiro, M.I. and Moura, J.M.F.**, 1991. LD^2 -ARMA Identification Algorithm, *IEEE Transactions on Signal Processing*, **39(8)**, 1822-1835.
- [59] **Du, X.C., Brella, A. and Longchamp, R.**, 1992. Fast Algorithm of Chandrasekhar Type for ARMA Model Identification, *Automatica*, **28(5)**, 927-934.
- [60] **Haseyama, M., Nagai, N. and Miki, N.**, 1992. Frequency-Weighting Model Identification with an Adaptive ARMA Lattice Filter, *IEEE International Conference on Systems Engineering*, Kobe, Japan, 17-19 September, pp.543-546.
- [61] **Liang, G.; Wilkes, D.M.**, 1992. The Recursive Linear Identification Method for ARMA Model Estimation, *IEEE Southeastcon '92 Proceedings*, Birmingham, AL, USA, 12-15 April, pp.677-682.
- [62] **Chen, B.S., Chen J.M. and Shern, S.C.**, 1994. An ARMA Robust System Identification Using a Generalized l_p Norm Estimation Algorithm, *IEEE Transactions on Signal Processing*, **42(5)**, 1063-1073.
- [63] **Monin, A. and Salut, G.**, 1994. Exact ARMA Lattice Predictors from Autocorrelation Functions, *IEEE Transactions on Signal Processing*, **42(4)**, 877-886.
- [64] **Monin, A. and Salut, G.**, 1996. ARMA Lattice Identification: A New Hereditary Algorithm, *IEEE Transactions on Signal Processing*, **44(2)**, 360-370.
- [65] **Zhang, Y. and Asada, H.H.**, 2002. Blind Multi-channel Identification of ARMA Systems: A Least-Squares Approach, *Proceedings of the American Control Conference*, Alaska, USA, 8-10 May, pp.2035-2036.

- [66] **Efe, M.Ö, Kaynak, O. and Wilamowski, B.M.,** 2002. An On-line and Robust Identification Scheme for Time-Varying ARMA Processes, *IEEE 28th Annual Conference of the Industrial Electronics Society*, Sevilla, Spain, 5-8 November, pp.1990-1994.
- [67] **Papakos, V. and Fassois, S.D.,** 2003. Multichannel Identification of Aircraft Skeleton Structures Under Unobservable Excitation: A Vector AR/ARMA Framework, *Mechanical Systems and Signal Processing*, **17(6)**, 1271-1290.
- [68] **Kaveh, M.,** 1979. High Resolution Spectral Estimation for Noisy Signals, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-27(3)**, 286-287.
- [69] **Bruzzone, S. and Kaveh, M.,** 1980. On Some Suboptimum ARMA Spectral Estimators, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-28(6)**, 753-755.
- [70] **Kay, S.M.,** 1980. A New ARMA Spectral Estimator, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-28(5)**, 585-588.
- [71] **Cadzow, J.A.,** 1980. High Performance Spectral Estimation: A new ARMA Method, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-28(5)**, 524-529.
- [72] **Cadzow, J.A.,** 1982. Spectral Estimation: An Over-determined Rational Model Equation Approach, *Proceedings of IEEE*, **70**, 907-938.
- [73] **Friedlander, B.,** 1982. Recursive Lattice Forms for Spectral Estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-30(6)**, 920-930.
- [74] **Friedlander, B.,** 1983. Instrumental Variable Methods for ARMA Spectral Estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-31(2)**, 404-415.
- [75] **Friedlander, B. and Porat, B.,** 1984. The Modified Yule-Walker Method of ARMA Spectral Estimation, *IEEE Transactions on Aerospace and Electronic Systems*, **AES-20(2)**, 158-173.
- [76] **Talkhan, E.A., Hassan, A.F. and Amin, N.A.,** 1983. An ARMA Model for Power Spectrum Estimation, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Boston, MA USA, April 14-16, pp.1080-1083.

- [77] **Ogino, K.**, 1983. A Fast Algorithm for Unmodified ARMA Spectrum Estimation, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Boston, MA USA, April 14-16, pp.1106-1109.
- [78] **Ortigueira, M.D. and Tribolet, J.M.**, 1983. On the Double Levinson Recursion Formulation of ARMA Spectral Estimation, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Boston, MA USA, 14-16 April, pp.1076-1079.
- [79] **Martinelli, G. and Orlandi, G.**, 1984. ARMA Spectrum Estimation by Extended Lattice, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, San Diego, CA USA, March 19-21, pp.539-541.
- [80] **Lagunas-Hernández M.A., SantaMaría-Perez, M.E. and Figueiras-Vidal A.R.**, 1984. ARMA Model Maximum Entropy Power Spectral Estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-32(5)**, 984-990.
- [81] **Zhang, X.D. and Takeda, H.**, 1987. An Approach to Time Series Analysis and ARMA Spectral Estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-35(9)**, 1303-1313.
- [82] **Ribeiro, M.I. and Moura, J.M.F.**, 1987. Dual Algorithm for ARMA Spectrum Estimation, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, TX USA, 6-9 April, pp.2081-2084.
- [83] **Mandal, S., Witz, J.A. and Lyons, G.J.**, 1992. Reduced Order ARMA Spectral Estimation of Ocean Waves, *Applied Ocean Research*, **14**, 303-312.
- [84] **Haseyama, M., Nagai, N. and Miki, N.**, 1993. An Adaptive ARMA Four-Line Lattice Filter for Spectral Estimation with Frequency Weighting, *IEEE Transactions on Signal Processing*, **41(6)**, 2193-2207.
- [85] **Kaderli, A. and Kayhan, A.S.**, 2000. Spectral Estimation of ARMA Processes Using the Evolutionary Cepstrum, *IEEE Signal Processing Letters*, **7(9)**, 259-261.
- [86] **Kaderli, A. and Kayhan, A.S.**, 2002. Spectral Estimation of Nonstationary ARMA Processes Using ARMA Cepstrum Recursion, *IEEE Signal Processing Letters*, **9(4)**, 130-132.

- [87] **Scharf, L.L. and Luby, J.C.**, 1979. Statistical Design of ARMA Filters, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Washington, D.C. USA, April 2-4, pp.818-821.
- [88] **Scharf, L.L. and Luby, J.C.**, 1979. Statistical Design of Autoregressive-Moving Average Digital Filters, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-27(3)**, 240-247.
- [89] **Yegnanarayana, B.**, 1981. Design of ARMA Digital Filters by Pole-Zero Decomposition, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-29(3)**, 433-439.
- [90] **Lim, Y.C. and Parker, S.R.**, 1984. On the Synthesis of Lattice Parameter Digital Filters, *IEEE Transactions on Circuits and Systems*, **CAS-31(7)**, 593-601.
- [91] **Lim, Y.C.**, 1984. On the Synthesis of IIR Digital Filters Derived from Single Channel AR Lattice Network, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-32(4)**, 741-749.
- [92] **Rao, S.K. and Kailath, T.**, 1984. Orthogonal Digital Filters for VLSI Implementation, *IEEE Transactions on Circuits and Systems*, **CAS-31(11)**, 933-945.
- [93] **Miyanaga, Y., Nagai, N. and Miki, N.**, 1987. ARMA Digital Lattice Filter Based on a New Criterion, *IEEE Transactions on Circuits and Systems*, **CAS-34(6)**, 617-628.
- [94] **Mennecke, D.W. and Tummala M.**, 1988. Adaptive ARMA Lattice Based on Generalized Mullis-Roberts Criterion, *Twenty-Second Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA USA, October 31-November 2, pp.328-332.
- [95] **Sellami, L. and Newcomb, R.**, 1996. Synthesis of ARMA Filters by Real Lossless Digital Lattices, *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, **43(5)**, 379-386.
- [96] **Haseyama, M. and Kitajima, H.**, 1997. Inherent Matrix Identities on ARMA Lattice Filter Realization Algorithm and Their Application, *IEEE Transactions on Signal Processing*, **45(9)**, 2395-2398.
- [97] **Kwan, H.K.**, 2002. ARMA Lattice Digital Filter Design, *45th Midwest Symposium on Circuits and Systems*, Tulsa, OK USA, 4-7 August, pp. III-489-III-492.

- [98] **Kwan, H.K.**, 2003. Minimal ARMA Lattice Digital Filter Realization, *Proceedings of the International Symposium on Circuits and Systems*, Bangkok, Thailand, 25-28 May, pp. IV-305-IV-308.
- [99] **Itakura, F. and Saito, S.**, 1971. Digital Filtering Techniques for Speech Analysis and Synthesis, *Proceedings of the 7th International Congress on Acoustics*, Budapest, August, pp.261-264.
- [100] **Friedlander, B. and Maitra, S.**, 1981. Speech Deconvolution by Recursive ARMA Lattice Filters, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Atlanta, GA USA, 30 March-1 April, pp. 343-346.
- [101] **Miyanaga Y., Gozen, S. and Ohtsuki, N.**, 2000. A Robust Speech Analysis in Speech Recognition, *Proceedings of the 5th International Conference on Signal Processing*, Beijing, China, 21-25 August, pp.706-709.
- [102] **Kwan, H.K. and Li, T.X.**, 2000. ARMA Lattice Modeling for Isolated Word Speech Recognition, *Proceedings of the 43rd Midwest Symposium on Circuits and Systems*, Lansing, MI USA, 8-11 August, pp.1186-1190.
- [103] **Kwan, H.K. and Wang, M.**, 2001. Pitch-Excited ARMA Lattice Model for Speech Synthesis, *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, BC Canada, 26-28 August, pp.659-662.
- [104] **Söderström, T.**, 1974. Convergence Properties of the Generalized Least Squares Identification Method, *Automatica*, **10**, 617-626.
- [105] **Söderström, T.**, 1974. On the Generalized Least Squares Method. Counter Examples to General Convergence, *Automatica*, **10**, 681-683.
- [106] **Strobach, P.**, 1986. Pure Order Recursive Least-Squares Ladder Algorithms, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-34(4)**, 880-897.
- [107] **Akaike, H.**, 1974. A New Look at the Statistical Model Identification, *IEEE Transactions on Automatic Control*, **AC-19(6)**, 716-723.
- [108] **Mullis, C.T. and Roberts, R.A.**, 1976. The Use of Second-Order Information in the Approximation of Discrete-Time Linear Systems, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-24(6)**, 226-238.

- [109] **Gray, A.H. Jr. and Markel, J.D.**, 1973. Digital Lattice and Ladder Filter Synthesis, *IEEE Transactions on Audio Electroacoustics*, **AU-21(6)**, 491-500.
- [110] **Lee, D.T.L. and Morf, M.**, 1980. Recursive Square-Root Ladder Estimation Algorithms, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Tulsa, OK USA, 10-12 April, pp.1005-1017.
- [111] **Marzetta, T.L.**, 1979. The Design of 2-D Recursive Filters in the 2-D Reflection Coefficient Domain, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Washington DC, 2-4 April, pp. 32-35.
- [112] **Marzetta, T.L.**, 1980. Two-dimensional Linear Prediction: Autocorrelation Arrays, Minimum-Phase Prediction Error Filters, and Reflection Coefficient Arrays, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-28(6)**, 725-733.
- [113] **Therrien, C.W.**, 1981. Relation between 2-D and Multichannel Linear Prediction, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-29**, 454-456.
- [114] **Parker, S.R. and Kayran, A.H.**, 1984. Lattice Parameter Autoregressive Modeling of Two-Dimensional Fields-Part I: The Quarter Plane Case, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-32**, 872-885.
- [115] **Lev-Ari, H. and Parker, S.R.**, 1985. Lattice Filter Modeling of Two-Dimensional Fields, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Tampa, FL, 26-29 March, pp. 1317-1320.
- [116] **Lenk, P.J. and Parker, S.R.**, 1986. Schur and Levinson Recursions for Two-Dimensional Data Fields, *Proceedings of IEEE International Symposium on Circuits and Systems*, San Jose, CA, May, pp.92-95.
- [117] **Zhao, P.Y. and Yu, D.R.**, 1988. 2-D Noncausal AR Lattice Modeling, *Twenty-Second Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA USA, October 31-November 2, pp.767-770.

- [118] **Yildiz, N.**, 1994. The Calculation of the 2-D Lattice Parameters from Short Data Records, *M.Sc.Thesis*, I.T.U. Institute of Science and Technology, Istanbul.
- [119] **Kayran, A.H.**, 1996. Two-Dimensional Orthogonal Lattice Structures for Autoregressive Modeling of Random Fields, *IEEE Transactions on Signal Processing*, **44(4)**, 963-978.
- [120] **Kayran, A.H.**, 1996. 2D Schur Algorithm Using A New Generator Matrix, *Electronics Letters*, **32(22)**, 2044-2045.
- [121] **Kayran, A.H. and Parker, S.R.**, 1997. Optimum Quarter-Plane Autoregressive Modeling of 2-D Fields Using Four-Field Lattice Approach, *IEEE Transactions on Signal Processing*, **45(9)**, 2363-2373.
- [122] **Nakachi, T., Yamashita, K. and Hamada, N.**, 1997. Asymmetric Half-Plane Lattice Modeling Based on 2-D Levinson Algorithm, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, **44(10)**, 865 -868.
- [123] **Kayran, A.H., Küçük, Ü. and Parker, S.R.**, 1998. Two-Dimensional Schur Algorithm, *Multidimensional Systems and Signal Processing*, **9**, 7-37.
- [124] **Marple, S.L. Jr.**, 2000. Two-Dimensional Lattice Linear Prediction Parameter Estimation Method and Fast Algorithm, *IEEE Signal Processing Letters*, **7(6)**, 164-168.
- [125] **Kayran, A.H. and Erer, I.**, 2004. Optimum Asymmetric Half-plane Autoregressive Lattice Parameter Modeling of 2D Fields, *IEEE Transactions on Signal Processing*, **52(3)**, 807-819.
- [126] **Lev-Ari, H. and Parker, S.R.**, 1986. Stable and Efficient 2-D Lattice Filters, *Proceedings of IEEE Symposium on Circuits and Systems*, San Jose, CA, 1-2 May, pp. 659-698.
- [127] **Kwan, H.K. and Liu, Y.C.**, 1989. Lattice Implementation of Two-Dimensional Recursive Digital Filters, *IEEE Transactions on Circuits and Systems*, **36(3)**, 383-386.
- [128] **Qunshan, G. and Xiyu, D.**, 1990. 2-D Lattice Structure and Explicit Formulas for the Design of 2-D Digital Fan Filters, *IEEE International Symposium on Circuits and Systems*, New Orleans, LA USA, 1-3 May, pp.3019-3022.

- [129] **Kayran, A.H.**, 1990. Design of 2-D Recursive Filters with Asymmetric Half-Plane Lattice Modeling, *IEE Proceedings*, **137**, 427-438.
- [130] **Ertüzün, A., Kayran, A.H. and Panayirci, E.**, 1990. Entropy Relations of 2-D Lattice Filters, *Proceedings of Bilcon Conference on Control and Signal Processing*, Ankara, Turkey, 25 July, pp.1337-1343.
- [131] **Ertüzün, A., Kayran, A.H. and Panayirci, E.**, 1992. An Improved 2-D Lattice Filter and Its Entropy Relations, *Signal Processing*, **28**, 1-24.
- [132] **Ertüzün, A., Kayran, A.H. and Panayirci, E.**, 1995. Further Improved 2-D Lattice Filter Structure Employing Missing Reflection Coefficients, *Circuits Systems Signal Processing*, **14(4)**, 473-494.
- [133] **Ertüzün, A., Kayran, A.H. and Panayirci, E.**, 1990. Stable Quarter-Plane 2-D Lattice Filters, *Electronics Letters*, **26(12)**, 806-807.
- [134] **Mahmoud, E., Mashali, S., Elcherif, Y. and Talkhan, E.**, 1991. Lattice Realization of Decomposed 2-D Filters, *IEEE International Conference on Systems Engineering*, Dayton, OH USA, 1-3 August, pp.382-385.
- [135] **Kawamata, M., Kawakami, E. and Higuchi, T.**, 1993. Realization of Lattice-Form Separable-Denominator 2D Adaptive Filters, *IEEE International Symposium on Circuits and Systems*, Chicago, IL USA, 3-6 May, pp.295-298.
- [136] **Kuduvalli, G.R. and Rangayyan R.M.**, 1993. An Algorithm for Direct Computation of 2-D Linear Prediction Coefficients, *IEEE Transactions on Signal Processing*, **41(2)**, 996-1000.
- [137] **Kayran, A.H. and Eksioglu, E.**, 2000. 2D FIR Wiener Filter Realization Using Orthogonal Lattice Structures, *Electronics Letters*, **36(12)**, 1078-1079.
- [138] **Antoniou, G.E.**, 2001. 2-D Lattice Discrete Filters: Minimal Delay and State Space Realization, *IEEE Signal Processing Letters*, **8(1)**, 23-25.
- [139] **Schnauffer, B.A. and Jenkins, W.K.**, 1991. Development of a New Adaptive Two-Dimensional Joint Process Estimator, *IEEE International Symposium on Circuits and Systems*, Singapore, 11-14 June, pp.536-539.
- [140] **Nakachi, T., Yamashita, K. and Hamada, N.**, 1996. A Two-Dimensional Adaptive Joint-Process Estimator Using New Lattice Form, *IEEE*

International Symposium on Circuits and Systems, Atlanta, GA USA, 12-15 May, pp.33 –36.

- [141] **Yamashita, K., Kinjyo, T., Asato, K. and Miyagi, H.**, 1998. A 2-D Adaptive Joint-Process IIR Filter with Generalized Lattice Structure, *IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA USA, 11-14 October, pp.4545-4549.
- [142] **Moro, H., Watanabe, T., Taguchi, A. and Hamada, N.**, 1989. On the Adaptive Algorithm and its Convergent Rate Improvement of 2-D Lattice Filter, *European Conference on Circuit Theory and Design*, Brighton UK, 5-8 September, pp.430-434.
- [143] **Liu, X., Baylou, P. and Najim, M.**, 1992. A New 2D Fast Lattice RLS Algorithm, *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, San Francisco, CA USA, 23-26 March, pp.329-332.
- [144] **Liu, X. and Najim, M.**, 1996. A Two-Dimensional Fast Lattice Recursive Least Squares Algorithm, *IEEE Transactions on Signal Processing*, **44(10)**, 2557-2567.
- [145] **Cadzow, J.A. and Ogino, K.**, 1981. Two-Dimensional Spectral Estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-29(3)**, 396-401.
- [146] **Kayran, A.H., Parker, S.R. and Klich, D.J.**, 1984. Two-Dimensional Spectral Estimation with Autoregressive Lattice Parameters, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, San Diego, CA USA, March 19-21, pp.138-141.
- [147] **Ranganath, S. and Jain, A.K.**, 1985. Two-Dimensional Linear Prediction Models-Part I: Spectral Factorization and Realization, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-33(1)**, 280-299.
- [148] **Kimura, H. and Honoki, Y.**, 1987. A Hybrid Approach to High Resolution Two-Dimensional Spectrum Analysis, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-35(7)**, 1024-1036.
- [149] **Therrien, C.W. and El-Shaer, H.T.**, 1989. A Direct Algorithm for Computing 2-D AR Spectrum Estimates, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37**, 1795-1797.

- [150] **Therrien, C.W. and El-Shaer, H.T.**, 1989. Multi-channel 2-D AR Spectrum Estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37**, 1798-1800.
- [151] **McGuffin, B.F. and Liu, B.**, 1989. An Efficient Algorithm for Two-Dimensional Autoregressive Spectrum Estimation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37(1)**, 106-117.
- [152] **Kayran, A.H.**, 1996. 2D Maximum Likelihood Spectrum Estimation Using Orthogonal 2D Lattice Filters, *Electronic Letters*, **32(15)**, 1339-1340.
- [153] **Alata, O., Baylou, P. and Najim, M.**, 1997. A New 2-D Spectrum Estimate Using Multichannel AR Approach of 2-D Fast RLS Algorithms, *Proceedings of International Conference on Image Processing*, Santa Barbara, CA USA, 26-29 October, pp.442-445.
- [154] **Rouquette, S., Alata, O., Najim, M. and Therrien, C.W.**, 1999. 2-D High Resolution Spectral Estimation Based on Multiple Regions of Support, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, AZ USA, 15-19 March, pp.1589-1592.
- [155] **Jakobsson, A., Marple, S.L. Jr. and Stoica, P.**, 1999. Two-Dimensional Capon Spectrum Analysis, *Department of Information Technology Uppsala University Technical Report*, **1999-007**, Uppsala, Sweden.
- [156] **Jakobsson, A., Marple, S.L. Jr. and Stoica, P.**, 2000. Computationally Efficient Two-Dimensional Capon Spectrum Analysis, *IEEE Transactions on Signal Processing*, **48(9)**, 2651-2661.
- [157] **Poehler, P. and Choi, J.**, 1983. Linear Predictive Coding of Imagery for Data Compression Applications, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Boston, MA, 14-16 April, pp. 1240-1243.
- [158] **Maragos, P.A., Schafer, R.W. and Mersereau, R.M.**, 1984. Two-Dimensional Linear Prediction and Its Application to Adaptive Predictive Coding of Images, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-32(6)**, 1213-1229.
- [159] **Kwan, H.K.**, 1986. Image Data Compression Using 2-D Lattice Predictors, *Electronics Letters*, **20**, 994-995.
- [160] **El-Shaer, H.T. and Therrien, C.W.**, 1988. Application of Multi-channel 2D Linear Prediction to Color Image Coding, *Proceedings of*

International Conference on Acoustics, Speech and Signal Processing, New York, NY USA, 11-14 April, pp.796-799.

- [161] **Onuk, N.T. and Ertüzün, A.**, 1994. Image Compression Using 2-D Lattice Filters, *Proceedings of the 7th Mediterranean Electrotechnical Conference*, Antalya, Turkey, 12-14 April, pp.227-230.
- [162] **Kwan, H.K. and Chan C.H.**, 2001. Noncausal Predictive Lattice Model for Image Compression, *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, 2-4 May, pp. 170-173.
- [163] **Ives, R.V.**, 2000. Lossless Compression of SAR Imagery Using a Multiple-Pass Gradient Adaptive Lattice Filter, *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA USA, 29 October-1 November, pp.229-232.
- [164] **Ives, R.V.**, 2001. Near-Lossless Compression of SAR Imagery Using Gradient Adaptive Lattice Filters, *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA USA, 4-7 November, pp.663-666.
- [165] **Tekalp, A.M., Kaufman, H and Woods, J.W.**, 1986. Identification of Image and Blur Parameters for the Restoration of Noncausal Blurs, *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-34(4), 963-972.
- [166] **Youlal, H., Janati-I, M. and Najim, M.**, 1992. Two-Dimensional Joint Process Lattice for Adaptive Restoration of Images, *IEEE Transactions on Image Processing*, 1, 366-378.
- [167] **Kayran, A.H.**, 1998. Improved 2D Joint Process Lattice for Adaptive Restoration of Images, *Electronics Letters*, 34(4), 350-351.
- [168] **Meylani, R., Ertüzün, A. and Erçil, A.**, 1996. Texture Defect Detection Using the Adaptive Two-Dimensional Lattice Filter, *Proceedings of International Conference on Image Processing*, Lausanne, Switzerland, 16-19 September, pp. 165-168.
- [169] **Hendekli, C.M. and Ertüzün, A.**, 1999. Two-Dimensional Delta-Domain Lattice Filter and Its Application to Adaptive Image Enhancement, *IEEE Transactions on Signal Processing*, 47(1), 266-273.

- [170] **Hendekli, C.M. and Ertüzün, A.**, 2001. Design of a Multichannel Two-Dimensional Delta-Domain Lattice Filter for Noise Removal, *IEEE Transactions on Signal Processing*, **49(7)**, 1581-1593.
- [171] **Gupta, I.J.**, 1994. High Resolution Radar Imaging Using 2-D Linear Prediction, *IEEE Transactions on Antennas and Propagation*, **42(1)**, 31-37.
- [172] **Sari, N., Sari, F., Paker, S., Özden, M.T. and Kayran, A.H.**, 2000. Radar Imaging Using 2-D Autoregressive Lattice Modeling (in Turkish), *Proceedings of the 8th National Signal Processing and Application Symposium*, Belek, Antalya, Turkey, June, pp.
- [173] **Erer, I., Kartal, M. and Kayran, A.H.**, 2001. 2-D Data Extrapolation for High Resolution Radar Imaging Using Autoregressive Lattice Modeling, *IEE Proceedings, Radar, Sonar and Navigation*, **148(5)**, 277-283.
- [174] **Erer, I., Kartal, M. and Kayran, A.H.**, 2001. Superresolution ISAR Imaging by 2-D Complex Asymmetric Half-plane Lattice Predictors, *IEEE International Geoscience and Remote Sensing Symposium*, Sydney, NSW Australia, 9-13 July, pp.2268-2270.
- [175] **Ffrench P.A., Zeidler J.R. and Ku, W.H.**, 1994. An Improved 2-D Adaptive Lattice Filtering Algorithm and Its Application to Detection of Small Objects in Correlated Clutter, *IEEE International Conference on Image Processing*, Austin, TX USA, 13-16 November, pp.765-769.
- [176] **Ffrench P.A., Zeidler J.R. and Ku, W.H.**, 1997. Enhanced Detectability of Small Objects in Correlated Clutter Using an Improved 2-D Adaptive Lattice Algorithm, *IEEE Transactions on Image Processing*, **6(3)**, 383-397.
- [177] **Kashyap, R.L.**, 1984. Characterization and Estimation of Two-Dimensional ARMA Models, *IEEE Transactions on Information Theory*, **IT-30(5)**, 736-745.
- [178] **Kwan, H.K and Liu, Y.C.**, 1989. Normalized 2-D ARMA Lattice Parameter Modeling Algorithm, *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, BC Canada, 1-2 June, pp.372-375.

- [179] **Zhang, X.D. and Cheng J.**, 1991. High Resolution Two-dimensional ARMA Spectral Estimation, *IEEE Transactions on Signal Processing*, **39(3)**, 765-770.
- [180] **Mikhael, W.B. and Yu, H.**, 1991. Two-Dimensional Least-square ARMA Modeling in the Transform Domain, *Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA USA, 4-6 November, pp.650-653.
- [181] **Chung, Y.S. and Kanefsky, M.**, 1992. On 2-D Recursive LMS Algorithms Using ARMA Prediction for ADPCM Encoding of Images, *IEEE Transactions on Image Processing*, **1(3)**, 416-422.
- [182] **Ghosh, S.M. and Mikhael, W.B.**, 1994. Two-dimensional Optimal Algorithms for Image Compression Using ARMA Predictors, *IEEE International Symposium on Circuits and Systems*, London, UK, 30 May-2 June, pp.613-616.
- [183] **Mikhael, W.B. and Yu, H.**, 1994. A Linear Approach for Two-Dimensional, Frequency Domain, Least Square, Signal and System Modeling, *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, **41(12)**, 786-795.
- [184] **Zhang, Q., Roman, J.R., Davis, D.W., and Mikhael, W.B.**, 1997. A Generalized Two-Dimensional, Frequency Domain Least Square Algorithm for ARMA System Modeling, *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, Sacramento, CA USA, 3-6 August, pp.965-968.
- [185] **Zhang, Q., Roman, J.R.; Davis, D.W., Mikhael, W.B.**, 1999. Modeling Performance of the Two Dimensional Frequency Domain Least Squares Algorithm in Airborne Surveillance Radar Applications, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Orlando, FL USA, 30 May-2 June, pp. III 239-III 242.
- [186] **Nijim, Y.W., Stearns, S.D., and Mikhael, W.B.**, 1997. Linear ARMA Predictors for the Lossless Compression of Two-Dimensional Signals, *Digital Signal Processing*, **7(2)**, 120-126.
- [187] **Kizilkaya , A., Kayran, A.H.**, 2003. 2-D ARMA Model Parameter Estimation Using ARMA Cepstrum Recursion, *IJCI Proceedings of International Conference on Signal Processing*, Çanakkale, Türkiye, 24-26 September, pp.266-270.

- [188] **Eksioglu, E.M. and Kayran, A.H.**, 2005. Lattice-Ladder Structure for 2D ARMA Filters, *EEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, PA USA, 18-23 March, pp.(IV-101)-(IV-104).
- [189] **Kayran, A.H.**, 1996. 2D ARMA Lattice Modeling Using Two-Channel AR Lattice Approach, *Electronics Letters*, **32(16)**, 1434-1435.
- [190] **Sari, N. and Kayran, A.H.**, 2004. Two-Dimensional ARMA Parameter Identification with Two-Channel AR Lattice Approach (in Turkish), *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, Kusadasi, Turkey, 28-30 April, pp.192-195.
- [191] **Sari, N. and Kayran, A.H.**, 2005. 2D ARMA Parameter Identification Using A Hybrid Lattice Design, *IEEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, PA USA, 18-23 March, pp.(IV-109)-(IV-112).
- [192] **Sari, N., Sari, F. and Kayran, A.H.**, 2005. Simultaneous Identification of Image and Blur Parameters with the 2-D Hybrid Lattice Structure (in Turkish), *Proceedings of the IEEE 13th Signal Processing and Communications Applications Conference*, Kayseri, Turkey, 16-18 May.
- [193] **Strand, O.N.**, 1977. Multichannel Complex Maximum Entropy (autoregressive) Spectral Analysis, *IEEE Transactions on Automatic Control*, **AC-22**, 634-640.
- [194] **Itakura, F. and Saito, S.**, 1970. A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies, *Electronic Comm. Japan.*, **53-A**, 36-43.
- [195] **El-Jaroudi, A. and Makhoul, J.**, 1991. Discrete All-Pole Modeling, *IEEE Transactions on Signal Processing*, **39(2)**, 411-423.
- [196] **McAulay, R.J.**, 1984. Maximum Likelihood Spectral Estimation and Its Application to Narrow-Band Speech Coding, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-32(2)**, 243-251.
- [197] **Woods, J.W.**, 1972. Two-dimensional Discrete Markovian Fields, *IEEE Transactions on Information Theory*, **IT-18(2)**, 232-240.
- [198] **Andrews, H.C. and Hunt, B.R.**, 1977. Digital Image Restoration Englewood Cliffs, NJ: Prentice-Hall.

- [199] **Woods, J.W. and Ingle, V.K.**, 1981. Kalman Filtering In Two-Dimensions- Further Results, *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-29(2)**, 188-197.
- [200] **Manolakis, D.G, Ingle, V.N and Kogon, S.M.**, 2000. Statistical and Adaptive Signal Processing, McGraw-Hill.

APPENDIX-A: DERIVATION OF EQUATIONS (4.8a)- (4.8b)

We rewrite here the model equation describing the 2-D ARMA system to be identified.

$$y(k_1, k_2) = b_0 x(k_1, k_2) + \sum_{n=1}^N b_n x((k_1, k_2) - n) - \sum_{m=1}^M a_m y((k_1, k_2) - m) \quad (\text{A.1})$$

The prediction error fields of the last prediction orders for the case $M = N$ can be defined as follows.

$$f_{u_0}^M(k_1, k_2) \triangleq u(k_1, k_2) - \sum_{m=1}^M a_m u((k_1, k_2) - m) - \sum_{n=1}^N b_n x((k_1, k_2) - n) \quad (\text{A.2})$$

$$f_{x_0}^N(k_1, k_2) \triangleq x(k_1, k_2) - \sum_{n=1}^N g_n x((k_1, k_2) - n) - \sum_{m=1}^M q_m u((k_1, k_2) - m) \quad (\text{A.3})$$

If we substitute $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$ in the above equation, we obtain the following:

$$f_{u_0}^M(k_1, k_2) \triangleq y(k_1, k_2) - x(k_1, k_2) - \sum_{m=1}^M a_m y((k_1, k_2) - m) - \sum_{n=1}^N (b_n - a_n) x((k_1, k_2) - n) \quad (\text{A.4})$$

$$f_{x_0}^N(k_1, k_2) \triangleq x(k_1, k_2) - \sum_{n=1}^N (g_n - q_n) x((k_1, k_2) - n) - \sum_{m=1}^M q_m y((k_1, k_2) - m) \quad (\text{A.5})$$

Here, Equation (A.4) represents the estimation of the output data field $y(k_1, k_2)$ with an estimation error $f_{u_0}^M(k_1, k_2)$ while (A.5) represents the estimation of the input data field $x(k_1, k_2)$ with an estimation error $f_{x_0}^N(k_1, k_2)$. Therefore we can write the following equations for $\hat{y}(k_1, k_2)$ and $\hat{x}(k_1, k_2)$.

$$\hat{y}(k_1, k_2) = f_{u_0}^M(k_1, k_2) + x(k_1, k_2) + \sum_{n=1}^N (b_n - a_n) x((k_1, k_2) - n) + \sum_{m=1}^M a_m y((k_1, k_2) - m) \quad (\text{A.6})$$

$$\hat{x}(k_1, k_2) = f_{x_0}^N(k_1, k_2) + \sum_{n=1}^N (g_n - q_n) x((k_1, k_2) - n) + \sum_{m=1}^M q_m y((k_1, k_2) - m) \quad (\text{A.7})$$

Now we replace $x(k_1, k_2)$ with $\hat{x}(k_1, k_2)$ of Equation (A.7) in Equation (A.1) and in Equation (A.6) respectively and obtain the following equations for $\hat{y}(k_1, k_2)$.

$$\hat{y}(k_1, k_2) = b_0 f_{x_0}^N(k_1, k_2) + \sum_{n=1}^N (b_0(g_n - q_n) + b_n)x((k_1, k_2) - n) + \sum_{m=1}^M (b_0 q_m - a_m)y((k_1, k_2) - m) \quad (\text{A.8})$$

$$\begin{aligned} \hat{y}(k_1, k_2) = f_{u_0}^M(k_1, k_2) + f_{x_0}^N(k_1, k_2) + \sum_{n=1}^N (g_n - q_n + b_n - a_n)x((k_1, k_2) - n) \\ + \sum_{m=1}^M (q_m + a_m)y((k_1, k_2) - m) \end{aligned} \quad (\text{A.9})$$

Now we compare Equation (A.8) and Equation (A.9) to obtain the following relations for the AR and MA coefficients of the ARMA model.

$$b_0 q_m - a_m = q_m + a_m \quad (\text{A.10})$$

$$\hat{a}_m = -a_m - (1 - b_0)q_m \quad \text{for } m = 1 \dots M \quad (\text{A.11})$$

$$b_0(g_n - q_n) + b_n = g_n - q_n + b_n - a_n \quad (\text{A.12})$$

$$\hat{b}_n = (b_n - a_n) + (1 - b_0)(g_n - q_n) \quad \text{for } n = 1 \dots N \quad (\text{A.13})$$

When we look at Equation (A.4), we observe that $-a_m$ values are the M th order forward prediction error vector tap weights of the first channel corresponding to the coefficients of $y(k_1, k_2)$ and have been given as the vector $[\mathbf{a}_{u_0}^{(M)}(2) \dots \mathbf{a}_{u_0}^{(M)}(M+1)]^T$ in Equation (4.8a). As easily seen from Equation (A.5), $-q_m$ values are the M th order forward prediction error tap weights of the second channel corresponding to the coefficients of $y(k_1, k_2)$ and have been given as the vector $[\mathbf{a}_{x_0}^{(N)}(2) \dots \mathbf{a}_{x_0}^{(N)}(M+1)]^T$ in Equation (4.8a). Therefore we can re-write Equation (A.11) to be an exact match and proof of Equation (4.8a) as follows:

$$\hat{a}_m = \mathbf{a}_{u_0}^{(M)}(m+1) + (1 - \hat{b}_0)\mathbf{a}_{x_0}^{(N)}(m+1) \quad \text{for } m = 1 \dots M \quad (\text{A.14})$$

It is apparent from Equation (A.5) that $(g_n - q_n)$ term is the N th order forward prediction error vector tap weights of the second channel corresponding to the coefficients of $x(k_1, k_2)$ and has been given as the vector

$\left[\mathbf{a}_{x_0}^{(N)}(M+3) \quad \dots \quad \mathbf{a}_{x_0}^{(N)}(2M+2) \right]^T$ in Equation (4.8b). Furthermore, by looking at the Equation (A.4), we find that $(b_n - a_n)$ term is the M th order forward prediction error tap weights of the first channel corresponding to the coefficients of $x(k_1, k_2)$ and has been defined as the vector $\left[\mathbf{a}_{u_0}^{(M)}(M+3) \quad \dots \quad \mathbf{a}_{u_0}^{(M)}(2M+2) \right]^T$ in Equation (4.8b). Therefore we can re-write Equation (A.13) to be an exact match and proof of Equation (4.8b) as follows.

$$\hat{b}_n = \mathbf{a}_{u_0}^{(M)}(M+n+2) + (1 - \hat{b}_0) \mathbf{a}_{x_0}^{(N)}(N+n+2) \quad \text{for } n = 1 \dots N \quad (\text{A.15})$$

APPENDIX B: DERIVATION OF EQUATIONS (4.12a)- (4.12b)

The prediction error fields of the last prediction orders for the case $M > N$ can be defined as follows.

$$f_{u_0}^N(k_1, k_2) \triangleq u(k_1, k_2) - \sum_{m=1}^N a'_m u((k_1, k_2) - m) - \sum_{n=1}^N b'_n x((k_1, k_2) - n) \quad (\text{B.1})$$

$$f_{x_0}^N(k_1, k_2) \triangleq x(k_1, k_2) - \sum_{n=1}^N g'_n x((k_1, k_2) - n) - \sum_{m=1}^N q'_m u((k_1, k_2) - m) \quad (\text{B.2})$$

If we substitute $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$ in the above equation, we obtain the following:

$$f_{u_0}^N(k_1, k_2) \triangleq y(k_1, k_2) - x(k_1, k_2) - \sum_{m=1}^N a'_m y((k_1, k_2) - m) - \sum_{n=1}^M (b'_n - a'_n) x((k_1, k_2) - n) \quad (\text{B.3})$$

$$f_{x_0}^N(k_1, k_2) \triangleq x(k_1, k_2) - \sum_{n=1}^N (g'_n - q'_n) x((k_1, k_2) - n) - \sum_{m=1}^N q'_m y((k_1, k_2) - m) \quad (\text{B.4})$$

Here, Equation (B.3) represents the estimation of the output data field $y(k_1, k_2)$ with an estimation error $f_{u_0}^N(k_1, k_2)$ while (B.4) represents the estimation of the input data field $x(k_1, k_2)$ with an estimation error $f_{x_0}^N(k_1, k_2)$. Therefore we can write the following equations for $\hat{y}(k_1, k_2)$ and $\hat{x}(k_1, k_2)$.

$$\hat{x}(k_1, k_2) = f_{x_0}^N(k_1, k_2) + \sum_{n=1}^N (g'_n - q'_n) x((k_1, k_2) - n) + \sum_{m=1}^N q'_m y((k_1, k_2) - m) \quad (\text{B.5})$$

We can write the forward prediction error field of the first channel for the last prediction order ($r = M; p = M$) as follows:

$$f_{y_0}^M(k_1, k_2) = y(k_1, k_2) - \sum_{m=1}^M a_m y((k_1, k_2) - m) - x(k_1, k_2) - \sum_{n=1}^N b_n x((k_1, k_2) - n) \quad (\text{B.6})$$

In Equation (B.6), if we replace $x(k_1, k_2)$ with $\hat{x}(k_1, k_2)$ given in Equation (B.5), we obtain the following:

$$f_{y_0}^M(k_1, k_2) = y(k_1, k_2) - f_{x_0}^N(k_1, k_2) - \sum_{n=1}^N (g'_n - q'_n + b_n) x((k_1, k_2) - n) - \sum_{m=1}^N (a_m + q'_m) y((k_1, k_2) - m) - \sum_{m=N+1}^M a_m y((k_1, k_2) - m) \quad (\text{B.7})$$

$$\hat{y}(k_1, k_2) = f_{y_0}^M(k_1, k_2) + f_{x_0}^N(k_1, k_2) + \sum_{n=1}^N (g'_n - q'_n + b_n) x((k_1, k_2) - n) + \sum_{m=1}^N (a_m + q'_m) y((k_1, k_2) - m) + \sum_{m=N+1}^M a_m y((k_1, k_2) - m) \quad (\text{B.8})$$

In order to make a comparison or analogy between the original and estimated coefficients, in Equation (A.1), we replace $x(k_1, k_2)$ with $\hat{x}(k_1, k_2)$ given in Equation (B.5) and we get the following relation.

$$y(k_1, k_2) = b_0 f_{x_0}^N(k_1, k_2) + \sum_{n=1}^N (b_0 (g'_n - q'_n) + b_n) x((k_1, k_2) - n) - \sum_{m=1}^N (a_m + b_0 q'_m) y((k_1, k_2) - m) + \sum_{m=N+1}^M a_m y((k_1, k_2) - m) \quad (\text{B.9})$$

Now we compare Equation (B.8) and Equation (B.9) to obtain the following relations for the AR and MA coefficients of the ARMA model.

$$(g'_n - q'_n + b_n) = (b_0 (g'_n - q'_n) + b_n) \quad (\text{B.10})$$

$$(a_m + q'_m) = -(a_m + b_0 q'_m) \quad (\text{B.11})$$

$$\hat{a}_m = -a_m - (1 - b_0) q'_m \quad \text{for } m = 1 \dots N \quad (\text{B.12a})$$

$$\hat{a}_m = -a_m \quad \text{for } m = N + 1 \dots M \quad (\text{B.12b})$$

$$\hat{b}_n = b_n + (1 - b_0)(g'_n - q'_n) \quad \text{for } n = 1 \dots N \quad (\text{B.13})$$

When we look at Equation (B.6), we observe that $-a_m$ values are the M th order forward prediction error vector tap weights of the first channel corresponding to the coefficients of $y(k_1, k_2)$ and have been given as the vector $[\mathbf{a}_{u_0}^{(M)}(2) \dots \mathbf{a}_{u_0}^{(M)}(M+1)]^T$ in Equation (4.12a). As easily seen from Equation (B.4), $-q'_m$ values are the N th order forward prediction error tap weights of

the second channel corresponding to the coefficients of $y(k_1, k_2)$ and have been given as the vector $\left[\mathbf{a}_{x_0}^{(N)}(2) \quad \dots \quad \mathbf{a}_{x_0}^{(N)}(N+1) \right]^T$ in Equation (4.12a). Therefore we can re-write Equation (B.12a) to be an exact match and proof of Equation (4.12a) for $m=1 \dots N$ as follows:

$$\hat{a}_m = \mathbf{a}_{u_0}^{(M)}(m+1) + (1 - \hat{b}_0) \mathbf{a}_{x_0}^{(N)}(m+1) \quad \text{for } m = 1 \dots N \quad (\text{B.14a})$$

It is apparent from Equation (B.7) that, for $m=N+1 \dots M$, the AR parameter estimates equal to the $-a_m$ values. Therefore we can re-write Equation (B.12b) to be an exact match and proof of Equation (4.12a) for $m=N+1 \dots M$ as follows:

$$\hat{a}_m = \mathbf{a}_{u_0}^{(M)}(m+1) \quad \text{for } m = N+1 \dots M \quad (\text{B.14b})$$

As for the MA parameter estimates, it is easily seen from Equation (B.6) that b_n term is the N th order forward prediction error vector tap weights of the first channel corresponding to the coefficients of $x(k_1, k_2)$ and has been given as the vector $\left[\mathbf{a}_{u_0}^{(M)}(M+3) \quad \dots \quad \mathbf{a}_{u_0}^{(M)}(M+N+2) \right]^T$ in Equation (4.12b). Furthermore, by looking at the Equation (B.4), we find that $(g'_n - q'_n)$ term is the N th order forward prediction error tap weights of the second channel corresponding to the coefficients of $x(k_1, k_2)$ and has been given as the vector $\left[\mathbf{a}_{x_0}^{(N)}(M+3) \quad \dots \quad \mathbf{a}_{x_0}^{(N)}(M+N+2) \right]^T$ in Equation (4.12b). Therefore we can re-write Equation (B.13) to be an exact match and proof of Equation (5.13b)

$$\hat{b}_n = \mathbf{a}_{u_0}^{(M)}(M+n+2) + (1 - \hat{b}_0) \mathbf{a}_{x_0}^{(N)}(M+n+2) \quad \text{for } n = 1 \dots N \quad (\text{B.15})$$

APPENDIX C: DERIVATION OF EQUATIONS (4.17a)- (4.17b)

The prediction error fields of the last prediction orders for the case $M < N$ can be defined as follows.

$$f_{u_0}^M(k_1, k_2) \triangleq u(k_1, k_2) - \sum_{m=1}^M a'_m u((k_1, k_2) - m) - \sum_{m=1}^M b'_m y((k_1, k_2) - m) \quad (C.1)$$

$$f_{y_0}^N(k_1, k_2) \triangleq y(k_1, k_2) - \sum_{m=1}^M g'_m y((k_1, k_2) - m) - \sum_{m=1}^M q'_m u((k_1, k_2) - m) \quad (C.2)$$

If we substitute $u(k_1, k_2) = y(k_1, k_2) - x(k_1, k_2)$ in the above equation, we obtain the following:

$$f_{u_0}^M(k_1, k_2) \triangleq y(k_1, k_2) - x(k_1, k_2) - \sum_{m=1}^M (a'_m + b'_m) y((k_1, k_2) - m) + \sum_{m=1}^M a'_m x((k_1, k_2) - m) \quad (C.3)$$

$$f_{y_0}^M(k_1, k_2) \triangleq y(k_1, k_2) - \sum_{m=1}^M (g'_m + q'_m) y((k_1, k_2) - m) + \sum_{m=1}^M q'_m x((k_1, k_2) - m) \quad (C.4)$$

Here, Equation (C.3) represents the estimation of the output data field $y(k_1, k_2)$ with an estimation error $f_{u_0}^M(k_1, k_2)$ while (C.4) represents the estimation of the input data field $x(k_1, k_2)$ with an estimation error $f_{y_0}^M(k_1, k_2)$. Therefore we can write the following equations for $\hat{y}(k_1, k_2)$ and $\hat{x}(k_1, k_2)$.

$$\hat{y}(k_1, k_2) = f_{y_0}^M(k_1, k_2) + \sum_{m=1}^M (g'_m - q'_m) y((k_1, k_2) - m) - \sum_{m=1}^M q'_m x((k_1, k_2) - m) \quad (C.5)$$

$$\hat{x}(k_1, k_2) = -f_{u_0}^M(k_1, k_2) + y(k_1, k_2) - \sum_{m=1}^M (a'_m + b'_m) y((k_1, k_2) - m) + \sum_{m=1}^M a'_m x((k_1, k_2) - m) \quad (C.6)$$

In Equation (C.6), if we replace $y(k_1, k_2)$ with $\hat{y}(k_1, k_2)$ given in Equation (C.5), we obtain the following:

$$\hat{x}(k_1, k_2) = f_{y_0}^M(k_1, k_2) - f_{u_0}^M(k_1, k_2) + \sum_{m=1}^M ((g'_m + q'_m) - (a'_m + b'_m))y((k_1, k_2) - m) - \sum_{m=1}^M (q'_m - a'_m)x((k_1, k_2) - m) \quad (C.7)$$

We can write the prediction error field for the last prediction order ($r = N$; $p = N$) as follows:

$$f_{y_0}^N(k_1, k_2) = y(k_1, k_2) - \sum_{m=1}^M a_m y((k_1, k_2) - m) - x(k_1, k_2) - \sum_{n=1}^N b_n x((k_1, k_2) - n) \quad (C.8)$$

In Equation (C.8), if we replace $x(k_1, k_2)$ with $\hat{x}(k_1, k_2)$ given in Equation (C.7), we obtain the following:

$$f_{y_0}^N(k_1, k_2) = f_{u_0}^M(k_1, k_2) - f_{y_0}^M(k_1, k_2) + y(k_1, k_2) - \sum_{m=1}^M (a_m + ((g'_m + q'_m) - (a'_m + b'_m))y((k_1, k_2) - m) - \sum_{n=1}^M (b_n - (q'_n - a'_n))x((k_1, k_2) - n) - \sum_{n=M+1}^N b_n x((k_1, k_2) - n) \quad (C.9)$$

$$\hat{y}(k_1, k_2) = f_{y_0}^N(k_1, k_2) - f_{u_0}^M(k_1, k_2) + f_{y_0}^M(k_1, k_2) + \sum_{m=1}^M (a_m + ((g'_m + q'_m) - (a'_m + b'_m)))y((k_1, k_2) - m) + \sum_{n=1}^M (b_n - (q'_n - a'_n))x((k_1, k_2) - n) + \sum_{n=M+1}^N b_n x((k_1, k_2) - n) \quad (C.10)$$

In order to make a comparison or analogy between the original and estimated coefficients, in Equation (A.1), we replace $x(k_1, k_2)$ with $\hat{x}(k_1, k_2)$ given in Equation (C.7) and we get the following relation.

$$\hat{y}(k_1, k_2) = b_0(f_{y_0}^M(k_1, k_2) - f_{u_0}^M(k_1, k_2)) - \sum_{m=1}^M (a_m - b_0((g'_m + q'_m) - (a'_m + b'_m)))y((k_1, k_2) - m) + \sum_{n=1}^M (b_n - b_0(q'_n - a'_n))x((k_1, k_2) - n) + \sum_{n=M+1}^N b_n x((k_1, k_2) - n) \quad (C.11)$$

Now we compare Equation (C.10) and Equation (C.11) to obtain the following relations for the AR and MA coefficients of the ARMA model.

$$a_m + ((g'_m + q'_m) - (a'_m + b'_m)) = -(a_m - b_0((g'_m + q'_m) - (a'_m + b'_m))) \quad (C.12)$$

$$\hat{a}_m = -a_m + (1-b_0)((a'_m + b'_m) - (g'_m + q'_m)) \quad \text{for } m=1 \dots M \quad (\text{C.13})$$

$$(b_n - (q'_n - a'_n)) = (b_n - b_0)(q'_n - a'_n) \quad (\text{C.14})$$

$$\hat{b}_n = b_n + (1-b_0)(a'_n - q'_n) \quad \text{for } n=1 \dots M \quad (\text{C.15a})$$

$$\hat{b}_n = b_n \quad \text{for } n=M+1 \dots N \quad (\text{C.15b})$$

When we look at Equation (C.8), we observe that $-a_m$ values are the N th order forward prediction error vector tap weights of the first channel corresponding to the coefficients of $y(k_1, k_2)$ and have been given as the vector $[\mathbf{a}_{u_0}^{(N)}(2) \dots \mathbf{a}_{u_0}^{(N)}(M+1)]^T$ in Equation (4.17a). $(a'_m + b'_m)$ term in (C.3) is the vector $[\mathbf{a}_{y_0}^{(M)}(2) \dots \mathbf{a}_{y_0}^{(M)}(M+1)]^T$ defined in Equation (4.17a). $(g'_m + q'_m)$ term in (C.4) is the vector $[\mathbf{a}_{u_0}^{(M)}(2) \dots \mathbf{a}_{u_0}^{(M)}(M+1)]^T$ defined in Equation (4.17a). Therefore we can rewrite Equation (C.13) to be an exact match and proof of Equation (4.17a) as follows:

$$\hat{a}_m = \mathbf{a}_{u_0}^{(N)}(m+1) + (1-\hat{b}_0)(\mathbf{a}_{y_0}^{(M)}(m+1) - \mathbf{a}_{u_0}^{(M)}(m+1)) \quad \text{for } m=1 \dots M \quad (\text{C.16})$$

As for the MA parameter estimates, it is easily seen from Equation (C.8) that b_n term is the N th order forward prediction error vector tap weights of the first channel corresponding to the coefficients of $x(k_1, k_2)$ and has been defined as $-\mathbf{a}_{u_0}^N(M+3) \dots \mathbf{a}_{u_0}^N(2M+2)]^T$ in Equation (4.17b). Furthermore, by looking at the Equation (C.3), we find that a'_n term is the M th order forward prediction error tap weights of the first channel corresponding to the coefficients of $x(k_1, k_2)$ and has been defined as the vector $[\mathbf{a}_{u_0}^M(M+3) \dots \mathbf{a}_{u_0}^M(2M+2)]^T$ in Equation (4.17b). As for the parameter q'_n , Equation (C.4) shows that it is the M th order forward prediction error tap weights of the second channel corresponding to the coefficients of $x(k_1, k_2)$ and has been defined as the vector $[\mathbf{a}_{y_0}^M(M+3) \dots \mathbf{a}_{y_0}^M(2M+2)]^T$ in Equation (4.17b). Therefore we can re-write Equation (C.15a) to be an exact match and proof of Equation (4.17b) for $n=1 \dots M$ as follows.

$$\hat{b}_n = -\mathbf{a}_{u_0}^N(M+n+2) + (1-\hat{b}_0)(\mathbf{a}_{u_0}^M(M+n+2) - \mathbf{a}_{y_0}^M(M+n+2)) \quad \text{for } n=1 \dots M \quad (\text{C.17a})$$

It is apparent from Equation (C.9) that, for $n=M+1\dots N$, the MA parameter estimates equal to the $-b_m$ values, therefore we can rewrite Equation (C.15b) to be an exact match and proof of Equation (4.17b) for $n=M+1\dots N$ as follows.

$$\hat{b}_n = -\mathbf{a}_{u_0}^N (M + n + 3) \quad \text{for } n = M + 1 \dots N \quad (\text{C.17b})$$

CURRICULUM VITAE

Nursen Yildiz SARI was born in Erzincan in 1970. She got the Bachelor of Science degree from Yildiz University Kocaeli Engineering Faculty, Electronics and Communication Engineering Department in 1991. At the same year she was accepted to the Master of Science program in the Electronics and Communication Department of Istanbul Technical University Science and Technology Institute and graduated in 1994. In 1994 Fall semester, she started her Ph.D. education at the same university and department. Currently, she is working as senior researcher in TUBITAK Marmara Research Center, Information Technologies Research Institute in Gebze, Kocaeli.