



LIBRARY
Michigan State
University

This is to certify that the
dissertation entitled

**RADAR TARGET DISCRIMINATION
USING NEURAL NETWORKS**

presented by

CHANG-YING TSAI

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Electrical Eng.

A handwritten signature in black ink, appearing to be "H. H. H.", written over a horizontal line.

Major professor

Date 6/29/95

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
Feb 24 1996	_____	_____
Feb 24 1996	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

MSU is An Affirmative Action/Equal Opportunity Institution

c:\circ\data\due.pm3-p.1

**RADAR TARGET DISCRIMINATION USING NEURAL
NETWORKS**

By

Tsai, Chang-Ying

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1995

ABSTRACT

RADAR TARGET DISCRIMINATION USING NEURAL NETWORKS

By

Tsai, Chang-Ying

This study uses several different memory-based neural networks to discriminate radar targets based on their early-time, aspect-dependent response, and demonstrates that target discrimination can be accomplished in a high-noise environment with great reliability. The difficulty of locating the beginning response point in practice prompts the use of FFT frequency spectrum magnitudes as aspect process patterns since a time shift is implicated in the phase of the spectrum. The effects of analog data and bipolar data with different quantization levels on network performances are investigated. Especially promising is the Recurrent Correlation Accumulation Adaptive Memory-Generalized Inverse (RCAAM-GI) cascade neural network. This network uses a dynamic memory structure to accumulate the converging information and has a stability criterion to allow us to define the final stable state. It can be considered as a real-time adaptive learning network with contamination observability and flexible decision strategy. From the simulation results, the network demonstrates computation space efficiency, and high noise tolerance.

To My Family in Taiwan

ACKNOWLEDGMENTS

I would like to express my sincerest thanks to Dr. K. M. Chen, Dr. E. J. Rothwell and Dr. D. P. Nyquist. Your support, concern, advice and consideration are deeply appreciated. I also thank Dr. Byron Drachman for participating as a member of my guidance committee. I am indebted to Rober Bebermeyer for his esteemed and accurate measurements.

I also want to thank Dr. J. E. Ross, III and Dr. Ponniah Ilavarasan for their encouragement, help and friendship. Finally, I would like to thank my colleague, Dr. Chi-wei Chang, and his lovely family for their hospitality relieving a traveller from loneliness.

To my family expression is too savage. Keeping in heart is forever.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	x
CHAPTER 1	
Introduction	1
CHAPTER 2	
Measurements and Data Preprocessing	6
2.1 Introduction	6
2.2 Measurements	6
CHAPTER 3	
Multi-Layer Feedforward with Error Backpropagation and Generalized Inverse Networks	19
3.1 Introduction	19
3.2 Multi-layer Feedforward with Error Backpropagation Networks	19
3.2.1 Gradient Descent Rule	22
3.2.2 ML/BP Learning Mechanism	23
3.3 Gaussian Noise Generation	26
3.4 ML/BP Network Trainings and Performances for Radar Target Discrimination	27
3.4.1 ML/BP Without Hidden Layer	31
3.4.2 ML/BP With One Hidden Layer	41
3.5 Generalized Inverse Network	48
3.5.1 Generalized Inverse Network Performance Using Bipolar Data ..	52
CHAPTER 4	
Recurrent Correlation Associative Memories	58
4.1 Introduction	58
4.2 Hopfield Network and Bidirectional Associative Memory (BAM)	59

4.2.1 Simulations and Results	66
4.3 High order Correlation Associative Memory (HCAM) and Exponential Correlation Associative Memory (ECAM)	71
4.3.1 Simulations and Results	76
4.4 RCAM-GI Cascade Networks	82
4.4.1 Simulations and Results	85
 CHAPTER 5	
Recurrent Correlation Accumulation Adaptive Memories	91
5.1 Introduction	91
5.2 Recurrent Correlation Accumulation Adaptive Memories (RCAAM)	91
5.3 Performance of the Recurrent Correlation Accumulation Adaptive Memory with dynamic input (RCAAM/di)	99
5.3.1 RCAAM/di performances with respect to stability criterion effect	100
5.3.2 RCAAM/di performances with respect to initial order effect . . .	122
5.3.3 RCAAM/di performances for the same averaged order group . .	141
5.4 Performance of the Recurrent Correlation Accumulation Adaptive Memory with fixed input (RCAAM/fi)	158
5.5 Performance of the Recurrent Correlation Accumulation Adaptive Memory with analog input and digital output (RCAAM/ad)	178
5.6 Comparisons of Correlation-based Discrimination Resolutions Between Analog and Bipolar Patterns	181
5.7 Performances of RCAAM-GI Cascade Networks	201
5.7.1 Converging Efficiency Comparisons Between the RCAAM and the RCAAM-GI Cascade Network	202
5.7.2 Noise Tolerance Comparisons Between the RCAAM and RCAAM-GI Cascade Network	203
 CHAPTER 6	
Target Discrimination using Neural Network with Spectrum Magnitude Response	224
6.1 Introduction	224
6.2 Spectrum Magnitude Process and Normalization	226
6.3 Manipulation of Spectrum Magnitude Data Before Network Process	238
6.4 Comparisons of Correlation-based Discrimination Resolutions for Different Data Formats	244
6.5 Network Simulations with Spectrum Magnitude Response	261
6.6 Network Comparisons	277
 CHAPTER 7	
Conclusions	294

BIBLIOGRAPHY	299
--------------------	-----

I

I

I

I

I

I

I

I

I

I

I

I

LIST OF TABLES

Table 3.1 Code assignment for 7 quantization levels coded by three bipolar bits.	53
Table 5.1 Comparisons of normalized target correlation gains subject to an assumed active threshold of 30% between analog data and encoded bipolar data.	198
Table 5.2 Comparisons of normalized target correlation gains subject to an assumed active threshold of 20% between analog data and encoded bipolar data.	199
Table 5.3 Comparisons of normalized target correlation gains subject to an assumed active threshold of 10% between analog data and encoded bipolar data.	200
Table 6.1 Code assignment of 5 quantization levels encoded by three bipolar bits. . .	229
Table 6.2 Estimated normalized target correlation gains subject to a active threshold of 40% for different spectral process data forms.	255
Table 6.3 Estimated normalized target correlation gains subject to a active threshold of 50% for different spectral process data forms.	257
Table 6.4 Estimated normalized target correlation gains subject to a active threshold of 60% for different spectral process data forms.	258
Table 6.5 Comparisons of network discrimination iterations among different data forms by testing networks with contaminated stored patterns.	259
Table 6.6 Architecture summary of the recurrent correlation associative networks used.	288
Table 6.7 Network architectures and performances summary for time domain target discrimination with 7 quantization levels encoded by 3 bits.	291
Table 6.8 Network architectures and performances summary for spectrum magnitude target discrimination with 5 quantization levels encoded by 3 bits.	292

Table 6.9 Network architectures and performances summary for spectrum magnitude target discrimination with 7 quantization levels encoded by 3 bits.	293
---	-----

Fig

Fig

Fig

Fig

Fig

Fig

Fig

Fig

Fig

Fig

Fig

LIST OF FIGURES

Figure 2.1 Frequency domain transient measurement system	7
Figure 2.2 Block diagram for the anechoic chamber measurement system	8
Figure 2.3 Time gated backscattered responses of B52, at aspect angle 0° , after 8192-point IFFT.	15
Figure 2.4 Normalized time response of B52, at aspect angle 0° , serves as a network analog pattern.	16
Figure 2.5 68 100-sample aspect time responses, truncated from 68 820-sample time-gated IFFT responses, for 4 scale aircraft models.	17
Figure 2.6 68 normalized 100-sample aspect time responses serving as aspect stored patterns for time domain networks.	18
Figure 3.1 Structure of a multi-layer feedforward with error backpropagation network	21
Figure 3.2 (a) An artificial neuron unit model. The O_{li} 's are outputs from previous layer H_i , θ_j is the bias for neuron j and W_{ij} 's are interconnecting weights between layer H_i and the neuron j of layer H_j . (b) Activation function, $g(x)$, used for bipolar process.	21
Figure 3.3 100 time responses of B52 used for network process at azimuthal aspect 0° . Solid line denotes responses without noise, while the dash-dot denotes the one with added noises of SNR= 10 dB.	28
Figure 3.4 100 time responses of B52 used for network process at azimuthal aspect 0° . Solid line denotes responses without noise, while the dash-dot denotes the one with added noises of SNR= 0 dB.	29
Figure 3.5 100 time responses of B52 used for network process at azimuthal aspect 0° . Solid line denotes responses without noise, while the dash-dot denotes the one with added noises of SNR= -3 dB.	30

Figure 3.6 Activation functions, $G_{\beta}(x)$, with different activation parameters. Solid line denotes the one with activation parameter $\beta=1$, while the dash line denotes the one with parameter $\beta=3$	32
Figure 3.7 The first derivatives, $d[G_{\beta}(x)]/dx$, of activation function with different activation parameters. Solid line denotes the one with activation parameter $\beta=1$, while the dash line denotes the one with parameter $\beta=3$	33
Figure 3.8 Performance of ML/BP network w/o hidden layer and w/o extra trainings vs. SNR(dB). (a). Network performance for the 17 training patterns of target B58. (b). Overall performance for all 68 training patterns.	37
Figure 3.9 Performance of ML/BP network w/o hidden layer and with extra trainings vs. SNR(dB). (a). Network performance for the 17 training patterns of target B58. (b). Overall performance for all 68 training patterns.	38
Figure 3.10 The generalization performance of ML/BP network w/o hidden layer and w/o extra trainings vs. SNR(dB). (a). Network generalization performance for the 16 untrained patterns of target B58. (b). Overall generalization performance for the 64 untrained patterns.	39
Figure 3.11 The generalization performance of ML/BP network w/o hidden layer and with extra trainings vs. SNR(dB). (a). Network generalization performance for the 16 untrained patterns of target B58. (b). Overall generalization performance for the 64 untrained patterns.	40
Figure 3.12 Normalized derivatives of error functions of three different orders w.r.t. weight W_{jk} , $-dE(W)/dW_{jk}$. The desired output is assumed to be 1 and an activation parameter $\beta=2$ is used for output neurons. The dash-dot line denotes the error function with order of 6, the solid line denotes the one with order of 4, and the dash line shows the one with order of 2.	43
Figure 3.13 Normalized derivatives of error function w.r.t. weight W_{jk} , $-dE(W)/dW_{jk}$, with three different activation parameters. The desired output is assumed to be 1 and the error function with order of 4 is used here. The dash-dot line denotes the activation function with parameter $\beta=3$, the solid line denotes the one with $\beta=2$, and the dash line shows the one with $\beta=1$	44
Figure 3.14 Performance of ML/BP network with one hidden layer vs. SNR(dB).	

(a). Network performance for the 17 training patterns of target B58.	
(b). Overall performance for all 68 training patterns.	46
Figure 3.15 The generalization performance of ML/BP network with one hidden layer vs. SNR(dB).	
(a). Network generalization performance for the 16 untrained patterns of target B58.	
(b). Overall generalization performance for the 64 untrained patterns.	47
Figure 3.16 Generalized Inverse (GI) network performance vs. SNR(dB).	
(a). Network performance for the 17 training patterns of target B58.	
(b). Overall performance for all 68 training patterns.	56
Figure 3.17 The generalization performance of Generalized Inverse (GI) network vs. SNR(dB).	
(a). Network generalization performance for the 16 untrained patterns of target B58.	
(b). Overall generalization performance for the 64 untrained patterns.	57
Figure 4.1 BAM network performances vs. SNR(dB) for the 68 stored patterns.	
(a). Pattern recognition performance.	
(b). Target discrimination performance by using target group code.	69
Figure 4.2 BAM generalization performance by using target group code for the 64 unstored patterns.	
	70
Figure 4.3 HCAM (order 3) network performances vs. SNR(dB) for the 68 stored patterns.	
(a). Pattern recognition performance.	
(b). Target discrimination performance.	77
Figure 4.4 HCAM (order 3) generalization performance for the 64 unstored patterns. .	
	78
Figure 4.5 HCAM (order 5) network performances vs. SNR(dB) for the 68 stored patterns.	
(a). Pattern recognition performance.	
(b). Target discrimination performance.	80
Figure 4.6 HCAM (order 5) generalization performance for the 64 unstored patterns. .	
	81
Figure 4.7 ECAM network performances vs. SNR(dB) for the 68 stored patterns.	
(a). Pattern recognition performance.	
(b). Target discrimination performance.	83
Figure 4.8 ECAM generalization performance for the 64 unstored patterns.	
	84
Figure 4.9 HCAM (order 3)-GI cascade network performances vs. SNR(dB).	

(a). Target discrimination performance for the 68 stored patterns.	
(b). Generalization performance for the 64 unstored patterns.	88
Figure 4.10 HCAM (order 5)-GI cascade network performances vs. SNR(dB).	
(a). Target discrimination performance for the 68 stored patterns.	
(b). Generalization performance for the 64 unstored patterns.	89
Figure 4.11 ECAM-GI cascade network performances vs. SNR(dB).	
(a). Target discrimination performance for the 68 stored patterns.	
(b). Generalization performance for the 64 unstored patterns.	90
Figure 5.1 Unrecognized discrimination of the RCAAM/di with an initial order of 2 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns.	
104	
Figure 5.2 Correct and wrong pattern discrimination performances of the RCAAM/di with an initial order of 2 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.	
(a) Correct pattern discrimination performances vs. stability criterion.	
(b) Wrong pattern discrimination performances vs. stability criterion.	105
Figure 5.3 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 2 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs stability criterion.	
(b) Wrong target discrimination performances vs. stability criterion.	106
Figure 5.4 Unrecognized discrimination of the RCAAM/di with an initial order of 2 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns.	
107	
Figure 5.5 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 2 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs stability criterion.	
(b) Wrong target discrimination performances vs. stability criterion.	108
Figure 5.6 Unrecognized discrimination of the RCAAM/di with an initial order of 3 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns.	
111	
Figure 5.7 Correct and wrong pattern discrimination performances of the RCAAM/di with an initial order of 3 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.	
(a) Correct pattern discrimination performances vs. stability criterion.	
(b) Wrong pattern discrimination performances vs. stability criterion.	112

Figure 5.8 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 3 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs stability criterion.	
(b) Wrong target discrimination performances vs. stability criterion.	113
Figure 5.9 Unrecognized discrimination of the RCAAM/di with an initial order of 3 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns. . .	114
Figure 5.10 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 3 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs stability criterion.	
(b) Wrong target discrimination performances vs. stability criterion.	115
Figure 5.11 Unrecognized discrimination of the RCAAM/di with an initial order of 6 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns.	117
Figure 5.12 Correct and wrong pattern discrimination performances of the RCAAM/di with an initial order of 6 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.	
(a) Correct pattern discrimination performances vs. stability criterion.	
(b) Wrong pattern discrimination performances vs. stability criterion.	118
Figure 5.13 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 6 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs stability criterion.	
(b) Wrong target discrimination performances vs. stability criterion.	119
Figure 5.14 Unrecognized discrimination of the RCAAM/di with an initial order of 6 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns. . .	120
Figure 5.15 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 6 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs stability criterion.	
(b) Wrong target discrimination performances vs. stability criterion.	121
Figure 5.16 Unrecognized discrimination of the RCAAM/di with a stability criterion of 2 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 68 stored patterns.	124

Fig

Fig

Fig

Fig

Fig

Fig

Fig

Fig

Fig

Figure 5.17 Correct and wrong pattern discrimination performances of the RCAAM/di with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for 68 stored patterns.

- (a) Correct pattern discrimination performances vs. initial order.
- (b) Wrong pattern discrimination performances vs. initial order. 125

Figure 5.18 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 68 stored patterns belonging to 4 targets..

- (a) Correct target discrimination performances vs initial order.
- (b) Wrong target discrimination performances vs. initial order. 126

Figure 5.19 Unrecognized discrimination of the RCAAM/di with a stability criterion of 2 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 64 unstored patterns. . 127

Figure 5.20 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 64 unstored patterns belonging to 4 targets..

- (a) Correct target discrimination performances vs initial order.
- (b) Wrong target discrimination performances vs. initial order. 128

Figure 5.21 Unrecognized discrimination of the RCAAM/di with a stability criterion of 5 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 68 stored patterns. . . 131

Figure 5.22 Correct and wrong pattern discrimination performances of the RCAAM/di with a stability criterion of 5 under 40, 14 and 0 dB SNR vs. initial order for 68 stored patterns.

- (a) Correct pattern discrimination performances vs. initial order.
- (b) Wrong pattern discrimination performances vs. initial order. 132

Figure 5.23 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 5 under 40, 14 and 0 dB SNR vs. initial order for the 68 stored patterns belonging to 4 targets..

- (a) Correct target discrimination performances vs initial order.
- (b) Wrong target discrimination performances vs. initial order. 133

Figure 5.24 Unrecognized discrimination of the RCAAM/di with a stability criterion of 5 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 64 unstored patterns. . 134

Figure 5.25 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 5 under 40, 14 and 0 dB SNR vs. initial order for the 64 unstored patterns belonging to 4 targets..

- (a) Correct target discrimination performances vs initial order.

F

Fi

Fig

Fig

Fig

Fig

Fig

Fig

Fig

Figure

Figure

a

(b) Wrong target discrimination performances vs. initial order.	135
Figure 5.26 Unrecognized discrimination of the RCAAM/di with a stability criterion of 10 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 68 stored patterns. . . .	136
Figure 5.27 Correct and wrong pattern discrimination performances of the RCAAM/di with a stability criterion of 10 under 40, 14 and 0 dB SNR vs. initial order for 68 stored patterns.	
(a) Correct pattern discrimination performances vs. initial order.	
(b) Wrong pattern discrimination performances vs. initial order.	137
Figure 5.28 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 10 under 40, 14 and 0 dB SNR vs. initial order for the 68 stored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs initial order.	
(b) Wrong target discrimination performances vs. initial order.	138
Figure 5.29 Unrecognized discrimination of the RCAAM/di with a stability criterion of 10 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 64 unstored patterns. . .	139
Figure 5.30 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 10 under 40, 14 and 0 dB SNR vs. initial order for the 64 unstored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs initial order.	
(b) Wrong target discrimination performances vs. initial order.	140
Figure 5.31 Averaged orders required for target discrimination of the RCAAM/di with contaminated stored pattern inputs vs. stability criterion.	142
Figure 5.32 Averaged orders required for target discrimination of the RCAAM/di with contaminated unstored pattern inputs vs. stability criterion.	143
Figure 5.33 Orders required for stability criterion of the RCAAM/di with an average order between 9 and 10 vs. SNR.	145
Figure 5.34 Performances of the RCAAM/di with an average order between 9 and 10 vs. SNR for 68 stored patterns.	
(a) Correct target discriminations vs. SNR for 68 stored patterns.	
(b) Unrecognized and wrong target discriminations vs. SNR for 68 stored patterns.	146
Figure 5.35 Orders required for unstored pattern discriminations of the RCAAM/di with an average order of stored patterns between 9 and 10 vs. SNR.	147

Figure 5.36 Performances of the RCAAM/di with an average order of stored patterns between 9 and 10 vs. SNR for 64 unstored patterns.	
(a) Correct target discriminations vs. SNR for 64 unstored patterns.	
(b) Unrecognized and wrong target discriminations vs. SNR for 64 unstored patterns.	148
Figure 5.37 Orders required for stability criterion of the RCAAM/di with an average order between 10 and 11 vs. SNR.	149
Figure 5.38 Performances of the RCAAM/di with an average order between 10 and 11 vs. SNR for 68 stored patterns.	
(a) Correct target discriminations vs. SNR for 68 stored patterns.	
(b) Unrecognized and wrong target discriminations vs. SNR for 68 stored patterns.	150
Figure 5.39 Orders required for unstored pattern discriminations of the RCAAM/di with an average order of stored patterns between 10 and 11 vs. SNR.	151
Figure 5.40 Performances of the RCAAM/di with an average order of stored patterns between 10 and 11 vs. SNR for 64 unstored patterns.	
(a) Correct target discriminations vs. SNR for 64 unstored patterns.	
(b) Unrecognized and wrong target discriminations vs. SNR for 64 unstored patterns.	152
Figure 5.41 Orders required for stability criterion of the RCAAM/di with an average order between 11 and 12 vs. SNR.	154
Figure 5.42 Performances of the RCAAM/di with an average order between 11 and 12 vs. SNR for 68 stored patterns.	
(a) Correct target discriminations vs. SNR for 68 stored patterns.	
(b) Unrecognized and wrong target discriminations vs. SNR for 68 stored patterns.	155
Figure 5.43 Orders required for unstored pattern discriminations of the RCAAM/di with an average order of stored patterns between 11 and 12 vs. SNR.	156
Figure 5.44 Performances of the RCAAM/di with an average order of stored patterns between 11 and 12 vs. SNR for 64 unstored patterns.	
(a) Correct target discriminations vs. SNR for 64 unstored patterns.	
(b) Unrecognized and wrong target discriminations vs. SNR for 64 unstored patterns.	157
Figure 5.45 Orders required for stability criterion of the RCAAM/di with an average order between 13 and 14 vs. SNR.	159

Figure 5.46	Performances of the RCAAM/di with an average order between 13 and 14 vs. SNR for 68 stored patterns.	
	(a) Correct target discriminations vs. SNR for 68 stored patterns.	
	(b) Unrecognized and wrong target discriminations vs. SNR for 68 stored patterns.	160
Figure 5.47	Orders required for unstored pattern discriminations of the RCAAM/di with an average order of stored patterns between 13 and 14 vs. SNR.	161
Figure 5.48	Performances of the RCAAM/di with an average order of stored patterns between 13 and 14 vs. SNR for 64 unstored patterns.	
	(a) Correct target discriminations vs. SNR for 64 unstored patterns.	
	(b) Unrecognized and wrong target discriminations vs. SNR for 64 unstored patterns.	162
Figure 5.49	Unrecognized discrimination of the RCAAM/fi with an initial order of 1 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns.	165
Figure 5.50	Correct and wrong pattern discrimination performances of the RCAAM/fi with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.	
	(a) Correct pattern discrimination performances vs. stability criterion.	
	(b) Wrong pattern discrimination performances vs. stability criterion.	166
Figure 5.51	Correct and wrong target discrimination performances of the RCAAM/fi with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..	
	(a) Correct target discrimination performances vs stability criterion.	
	(b) Wrong target discrimination performances vs. stability criterion.	167
Figure 5.52	Unrecognized discrimination of the RCAAM/fi with an initial order of 1 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns.	168
Figure 5.53	Correct and wrong target discrimination performances of the RCAAM/fi with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..	
	(a) Correct target discrimination performances vs stability criterion.	
	(b) Wrong target discrimination performances vs. stability criterion.	169
Figure 5.54	Unrecognized discrimination of the RCAAM/fi with a stability criterion of 2 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 68 stored patterns.	172
Figure 5.55	Correct and wrong pattern discrimination performances of the RCAAM/fi with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for 68 stored	

patterns.

(a) Correct pattern discrimination performances vs. initial order.

(b) Wrong pattern discrimination performances vs. initial order. 173

Figure 5.56 Correct and wrong target discrimination performances of the RCAAM/fi with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order. 174

Figure 5.57 Unrecognized discrimination of the RCAAM/fi with a stability criterion of 2 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 64 unstored patterns. . . 175

Figure 5.58 Correct and wrong target discrimination performances of the RCAAM/fi with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order. 176

Figure 5.59 Averaged orders required for target discrimination of the RCAAM/fi with contaminated stored pattern inputs vs. stability criterion. 177

Figure 5.60 Unrecognized discrimination of the RCAAM/ad with an initial order of 1 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns. . . . 182

Figure 5.61 Correct and wrong pattern discrimination performances of the RCAAM/ad with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.

(a) Correct pattern discrimination performances vs. stability criterion.

(b) Wrong pattern discrimination performances vs. stability criterion. 183

Figure 5.62 Correct and wrong target discrimination performances of the RCAAM/ad with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs stability criterion.

(b) Wrong target discrimination performances vs. stability criterion. 184

Figure 5.63 Unrecognized discrimination of the RCAAM/ad with an initial order of 1 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns. . . 185

Figure 5.64 Correct and wrong target discrimination performances of the RCAAM/ad with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs stability criterion.	
(b) Wrong target discrimination performances vs. stability criterion.	186
Figure 5.65 Correct and wrong target discrimination performances of the RCAAM/ad with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 68 stored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs initial order.	
(b) Wrong target discrimination performances vs. initial order.	187
Figure 5.66 Correct and wrong target discrimination performances of the RCAAM/ad with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 64 unstored patterns belonging to 4 targets..	
(a) Correct target discrimination performances vs initial order.	
(b) Wrong target discrimination performances vs. initial order.	188
Figure 5.67 Averaged orders required for target discrimination of the RCAAM/ad with contaminated stored pattern inputs vs. stability criterion.	189
Figure 5.68 Orders required for the discriminations of RCAAM/fi1 vs. SNR for contaminated stored patterns.	194
Figure 5.69 Orders required for the discriminations of RCAAM/ad1 vs. SNR for contaminated stored patterns.	195
Figure 5.70 Unrecognized discriminations of the RCAAM/di2 and RCAAM/di22-GI vs. initial order for 40 dB and 0 dB SNR.	204
Figure 5.71 Unrecognized discriminations of the RCAAM/fi1 and RCAAM/fi12-GI vs. initial order for 40 dB and 0 dB SNR.	205
Figure 5.72 Unrecognized discriminations of the RCAAM/ad1 and RCAAM/ad12-GI vs. initial order for 40 dB and 0 dB SNR.	206
Figure 5.73 Performance comparisons of the RCAAM/di's with an initial order of 2 and RCAAM/di22-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.	
(a) Correct target discrimination vs. SNR for the 68 stored patterns.	
(b) Wrong target discrimination vs. SNR for the 68 stored patterns.	208
Figure 5.74 Unrecognized discriminations and discrimination orders of the RCAAM/di's with an initial order of 2 and RCAAM/di22-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.	
(a) Unrecognized discriminations vs. SNR for the 68 stored patterns.	

(b) Orders required for discriminations vs. SNR for the 68 stored patterns. 209

Figure 5.75 Performance comparisons of the RCAAM/di's with an initial order of 2 and RCAAM/di22-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

- (a) Correct target discriminations vs. SNR for the 64 unstored patterns.
- (b) Wrong target discriminations vs. SNR for the 64 unstored patterns. 210

Figure 5.76 Unrecognized discriminations and discrimination orders of the RCAAM/di's with an initial order of 2 and RCAAM/di22-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

- (a) Unrecognized discriminations vs. SNR for the 64 unstored patterns.
- (b) Orders required for discriminations vs. SNR for the 64 unstored patterns. 211

Figure 5.77 Performance comparisons of the RCAAM/fi's with an initial order of 1 and RCAAM/fi12-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.

- (a) Correct target discriminations vs. SNR for the 68 stored patterns.
- (b) Wrong target discriminations vs. SNR for the 68 stored patterns. 213

Figure 5.78 Unrecognized discriminations and discrimination orders of the RCAAM/fi's with an initial order of 1 and RCAAM/di12-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.

- (a) Unrecognized discriminations vs. SNR for the 68 stored patterns.
- (b) Orders required for discriminations vs. SNR for the 68 stored patterns. 214

Figure 5.79 Performance comparisons of the RCAAM/fi's with an initial order of 1 and RCAAM/fi12-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

- (a) Correct target discriminations vs. SNR for the 64 unstored patterns.
- (b) Wrong target discriminations vs. SNR for the 64 unstored patterns. 215

Figure 5.80 Unrecognized discriminations and discrimination orders of the RCAAM/fi's with an initial order of 1 and RCAAM/fi12-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

- (a) Unrecognized discriminations vs. SNR for the 64 unstored patterns.
- (b) Orders required for discriminations vs. SNR for the 64 unstored patterns. 216

Figure 5.81 Performance comparisons of the RCAAM/ad's with an initial order of 1 and RCAAM/ad12-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.

- (a) Correct target discriminations vs. SNR for the 68 stored patterns.
- (b) Wrong target discriminations vs. SNR for the 68 stored patterns. 220

Figure 5.82 Unrecognized discriminations and discrimination orders of the RCAAM/ad's with an initial order of 1 and RCAAM/di12-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.	
(a) Unrecognized discriminations vs. SNR for the 68 stored patterns.	
(b) Orders required for discriminations vs. SNR for the 68 stored patterns.	221
Figure 5.83 Performance comparisons of the RCAAM/ad's with an initial order of 1 and RCAAM/ad12-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.	
(a) Correct target discriminations vs. SNR for the 64 unstored patterns.	
(b) Wrong target discriminations vs. SNR for the 64 unstored patterns.	222
Figure 5.84 Unrecognized discriminations and discrimination orders of the RCAAM/ad's with an initial order of 1 and RCAAM/ad12-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.	
(a) Unrecognized discriminations vs. SNR for the 64 unstored patterns.	
(b) Orders required for discriminations vs. SNR for the 64 unstored patterns.	223
Figure 6.1 Two aspect (5.4° and 9°) stored patterns with an apparent time-shift resulted from an inappropriate beginning response determination occurred in time domain simulations.	
.	225
Figure 6.2 68 aspect FFT spectrum magnitude patterns, with normalized energy of 1, used for spectrum process network trainings/storage. Spectrum process networks simulate 4 targets, each target has 17 trained/stored aspect spectral patterns. The effective frequency band is 1-7 GHz.	228
Figure 6.3 FFT spectrum magnitude patterns of two time responses from target F14 with respective aspect angle of 5.4° and 9°.	236
Figure 6.4 Normalized FFT spectrum magnitude patterns of two time responses from target F14 with respective aspect angle of 5.4° and 9°.	237
Figure 6.5 820 time responses of target B52 at aspect angle 0° used as measured responses by a time domain radar. The solid line shows the true responses, while the dotted line denotes the contaminated signal with a SNR of 0 dB.	240
Figure 6.6 1358 spectrum magnitudes transformed from the previous two time responses by using a 1358-point DFT. The solid line shows the true responses, while the dotted line denotes the contaminated signal with a SNR of 0 dB. The evaluation of 0 dB is based on the 820 time responses.	241

Figure 6.7 100 time responses truncated from the previous 820 responses used as the time domain network process aspect pattern. The solid line shows the true responses, while the dash-dot line denotes the contaminated signal with a SNR of 0 dB. Here the evaluation of 0 dB is based on the 100 time responses.	242
Figure 6.8 100 spectrum magnitudes truncated from the previous 1358 spectral responses used as the spectrum network process pattern. The solid line shows the true responses, while the dash-dot line denotes the contaminated signal with a SNR of 0 dB. The evaluation of 0 dB is based on the 820 time responses.	243
Figure 6.9 Correlation gain population distribution for 68 stored time domain patterns with different data forms.	247
Figure 6.10 Correlation gain population distribution, only considering gain scale, for 68 stored time domain patterns with different data forms.	248
Figure 6.11 Correlation gain population distribution for 68 stored spectral process patterns with different data forms.	252
Figure 6.12 Correlation gain population distributions, only considering gain scale, for 68 stored spectral process patterns with different data forms.	253
Figure 6.13 Spectrum magnitude GI network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.	263
Figure 6.14 Spectrum magnitude GI network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.	264
Figure 6.15 Spectrum magnitude HCAM and HCAM-GI cascade network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.	265
Figure 6.16 Spectrum magnitude HCAM and HCAM-GI cascade network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.	266
Figure 6.17 Spectrum magnitude ECAM and ECAM-GI cascade network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.	267
Figure 6.18 Spectrum magnitude ECAM and ECAM-GI cascade network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.	268

Figure 6.19 Spectrum magnitude RCAAM/di22 and RCAAM/di22-GI cascade network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.	271
Figure 6.20 Spectrum magnitude RCAAM/di22 and RCAAM/di22-GI cascade network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.	272
Figure 6.21 Spectrum magnitude RCAAM/fi12 and RCAAM/fi12-GI cascade network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.	273
Figure 6.22 Spectrum magnitude RCAAM/fi12 and RCAAM/fi12-GI cascade network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.	274
Figure 6.23 Spectrum magnitude RCAAM/ad12 and RCAAM/ad12-GI cascade network performances with bipolar output forms coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.	275
Figure 6.24 Spectrum magnitude RCAAM/ad12 and RCAAM/ad12-GI cascade network generalization performances with bipolar output forms coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.	276
Figure 6.25 Performance Comparisons of different time domain networks (7-level bipolar data, if used) vs. SNR for 68 contaminated stored patterns.	280
Figure 6.26 Generalization performance Comparisons of different time domain networks (7-level bipolar data, if used) vs. SNR for 64 contaminated unstored patterns. ...	281
Figure 6.27 Performance Comparisons of different spectrum magnitude process networks (5-level bipolar data, if used) vs. SNR for 68 contaminated stored patterns. ...	282
Figure 6.28 Generalization performance Comparisons of different spectrum magnitude process networks (5-level bipolar data, if used) vs. SNR for 64 contaminated unstored patterns.	283
Figure 6.29 Performance Comparisons of different spectrum magnitude process networks (7-level bipolar data, if used) vs. SNR for 68 contaminated stored patterns. ...	284
Figure 6.30 Generalization performance Comparisons of different spectrum magnitude	

process networks (7-level bipolar data, if used) vs. SNR for 64 contaminated unstored patterns. 285

CHAPTER 1

Introduction

Many interesting schemes have been proposed for radar target discrimination [1-16], Particularly fascinating are those which use the transient response of the target. These include methods based on the aspect-independent late-time response [1-9], time domain imaging techniques [10-12], correlation [13-15] and wavelet transforms [16]. Many of these schemes use only the early-time specular target response, or the late-time resonant portion. Of those techniques that use the entire waveform, problems arise in the amount of computer storage required and the time needed to process the measured response of an unknown target. Most estimation procedures, from mean-squared linear estimation to nonlinear regression, require a rough mathematical model expressing the relationship between input and output, therefore they are model-based estimations. The radar target discrimination based on the early-time transient scattered field response can't be reached from a model-based method since an accurate mathematical aspect-dependent scattering transform formula is unavailable in the practical situation. Then the task may be devoted to the model estimation by using adaptive estimation procedures. However, an adaptive estimation requires prior information about the model order, number of estimated parameters, and the rough function type, and becomes very difficult for high orders under noisy situations. Hence this kind of model estimation is almost impossible.

Speaking of the pattern classification, the stochastic pattern recognition algorithms [17]-[20] can be divided into two categories : density/distribution estimation and pattern

clustering. The parameter estimations (discriminant function, Maximum Likelihood and Bayesian estimations) require the density model for each scattered response, while the clustering algorithms (supervised/unsupervised/differential competitive-learning or c-means algorithm) require consistent and clear pattern grouping. Unfortunately, the early-time radar target scattered responses usually show that the dependence on aspect is higher than the dependence on target in some aspect ranges, thus the target clustering among aspect responses is ambiguous and inconsistent.

A neural network's structure learns the given (or experienced) input-output associations through a synapse-interconnected black box, thus the neural learning is from experience and is model-free. A model-free estimation doesn't require to estimate the mathematical shape of the presented task, and, in contrast, the neural learning occurring in the black-box will implicitly represent the experienced input-output associations in its own way. Although mathematical models can analyze a task in detail, human takes actions typically by instinct (or neural learning) instead of sequential mathematical calculations. For example, nobody picks up a cup on a desk by using the complicated mathematical algorithm that a robot manipulator [21] undertakes. A child can finish the task without difficulty, while an adult may proceed in a more efficient and graceful way or in motion.

Compared to conventional methods, the artificial neural networks are quite suitable to solve the complicated systems for which deterministic methods are incapable or inefficient. Consider a practical problem : Can an unprofessional driver or deterministic mathematical control system continuously and smoothly back up, from an arbitrary initial position, a long trailer (cap+trail) along a given line in real-time ? In this problem, a deterministic model can't

consistently and effectively proceed with task since each action will accumulatively affect the future actions and errors, and then the trailer will become out of control if some jackknife angles occur. The neural networks can be applied to those fields within which stochastic and deterministic methods may suffer difficulties. Recently, neural networks have been used to perform target discrimination with reasonable storage requirements and rapid processing times [22, 23]. Much of this effort was based on simple back-propagation networks. This thesis will examine more sophisticated networks and demonstrate that target discrimination can be accomplished in a high-noise environment with great reliability. We will concentrate our research on the noise tolerance of various neural networks and the processing efficiency.

The transient scattered field response of a radar target is aspect sensitive, but for an interrogating pulse of a given bandwidth, a discretization of aspect angle can be found for which changes are gradual from angle to angle. Therefore, we can store some specified aspect responses as the reference patterns for each target, design a neural network to memorize the association among reference patterns and expect the network will correctly converge to some reference pattern when it is triggered at the input by a pattern that is sufficiently close to one of the reference patterns.

Using the Michigan State University anechoic chamber, an HP-8720B network analyzer is used to perform stepped frequency measurements of four scale-aircraft models, B52(1:72), B58(1:48), F14(1:48) and TR1(1:48). The targets are measured from 0° to 28.8° with an azimuthal aspect increment of 0.9° , resulting in 33 aspect measurements for each target. The frequency response spectra are calibrated using a 14" sphere as in [24] and taken into the time domain using the inverse fast Fourier Transform (IFFT). We then select 17 time-

domain responses from 0° to 28.8° with aspect increment 1.8° for each target as training/stored patterns, and 16 responses as untrained/unstored network generality test patterns from 0.9° to 27.9° with the same increment. Therefore, every untrained/unstored test pattern resides at the middle of two training/stored patterns.

We have used both time-response patterns and FFT spectrum-magnitude patterns to simulate target discrimination for each neural network. In time-response processing, the beginning response time has been assumed known, and we extract the next 100 response points as the aspect prototype from the assumed beginning point. Based on this assumed segment, noise is later added to test network noise tolerances. The difficulty of locating the beginning response point in practice prompts the use of FFT frequency spectrum magnitudes as aspect process patterns since a time shift is implicated in the phase of the spectrum.

In binomial input simulations, we quantize each time response using 7 numerical levels and encode these 7 levels using 3 bits. For spectrum magnitude processes, we simulate bipolar data formats with two quantization levels, 5 quantization levels and 7 quantization levels encoded by 3 bits. More levels are used for the time responses, since they exhibit a much wider oscillation range than the spectral magnitudes. 5 levels can be used to express response variations for the spectral magnitudes, since they have a small dynamic oscillation range.

In chapter 2, the target-aspect scattering measurement procedure is given and the data preprocessing described. We then present the theory for several different neural networks for target discrimination. In chapter 3, the Multi-Layer Feedforward and Error-BackPropagation (ML/BP) network and its backpropagation learning algorithm are presented. The Generalized Inverse (GI) algorithm and its iterative network learning procedure are also given in chapter

3. In chapter 4, we discuss Recurrent Correlation Associative Memories (RCAM), and analyze the High Order Recurrent Correlation Associative Memory (HCAM) and Exponential Correlation Associative Memory (ECAM). The advantage of the RCAM-GI cascade network is investigated.

In chapter 5, we propose a new network structure, the Recurrent Correlation Accumulation Adaptive Memory (RCAAM), which uses a dynamic memory structure to accumulate the converging information and has a stable criterion to allow spurious states to either stay as unknowns or converge to one of the stored patterns. We describe it as a real-time adaptive learning network with contamination observability and flexible decision strategy. The RCAAM performs discrimination equally well as the ECAM, always outperforms the HCAM with low orders, and requires much less processing space than the ECAM. An estimate formula for target crosscorrelation gain is developed, and a correlation-based discrimination resolution comparison between bipolar data and analog data is undertaken. In chapter 6, we introduce the FFT spectrum magnitude process to relieve the time-shift inconsistency occurring in the time responses. A modified process is constructed for analog spectrum process networks. We summarize and compare neural networks used in this paper to some popular ones, and also compare the estimated target discrimination resolutions among all data formats used in the time domain and spectrum magnitude processes. We briefly discuss implementation complexities and conclude the thesis in chapter 7.

CHAPTER 2

Measurements and Data Preprocessing

2.1 Introduction

To determine the transient backscattered field response of a practical radar target the Michigan State University anechoic chamber and a HP-8720B vector network analyzer are used. Four scale aircraft models, B52(1:72), B58(1:48), F14(1:48) and TR1(1:48), are used to perform stepped frequency measurements. The measurement and calibration processes will be given in the next section. Then the backscattered time responses obtained via the inverse Fourier transform from the measured spectral responses and the data preprocess for time domain networks are conducted in section 2.3.

2.2 Measurements [24]

Since an impulse response can help us to understand the target's structure, a synthesized short duration pulse, an approximation to an impulse, is used in the anechoic chamber measurements. A desired temporal impulse is difficult to consistently generate in the time domain, thus we measure spectral responses and then synthesize temporal responses. From Fourier frequency spectrum analysis and synthesis, we know an impulse in the time domain has an uniform frequency spectrum covering the whole frequency band. Therefore, a vector network analyzer capable of measuring both spectral magnitude and phase can allow us to synthesize a short duration pulse through wide frequency sweeping. The frequency domain measurement system we use is shown in **Figure 2.1**. The HP-8720B network analyzer is programmed to generate sweeping frequency waves to emulate a temporal

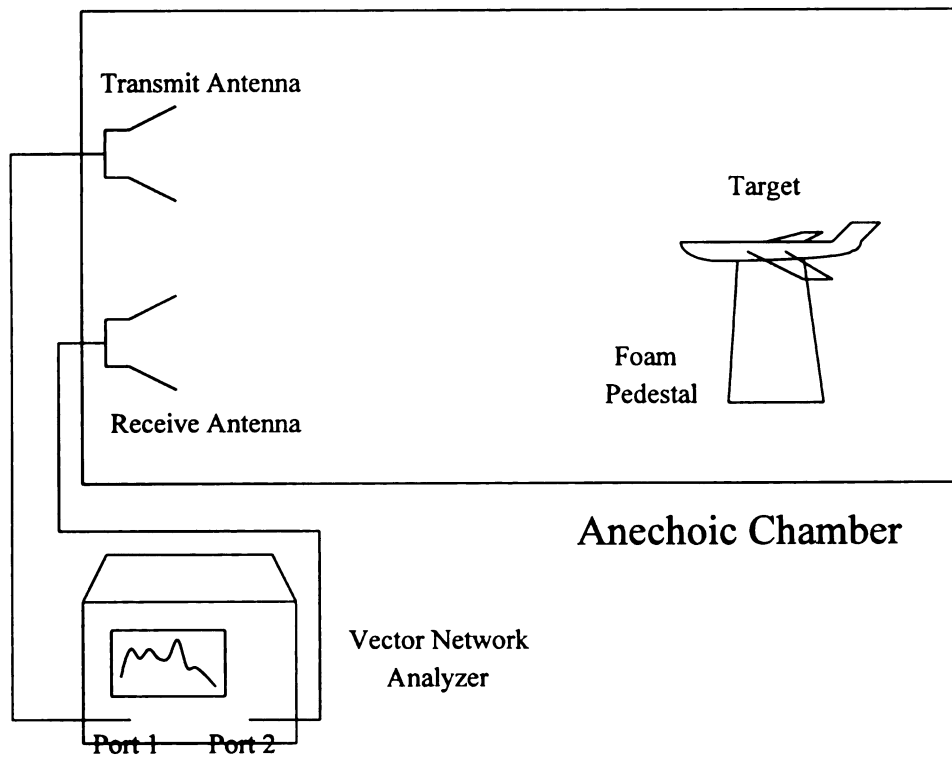


Figure 2.1 Frequency domain transient measurement system

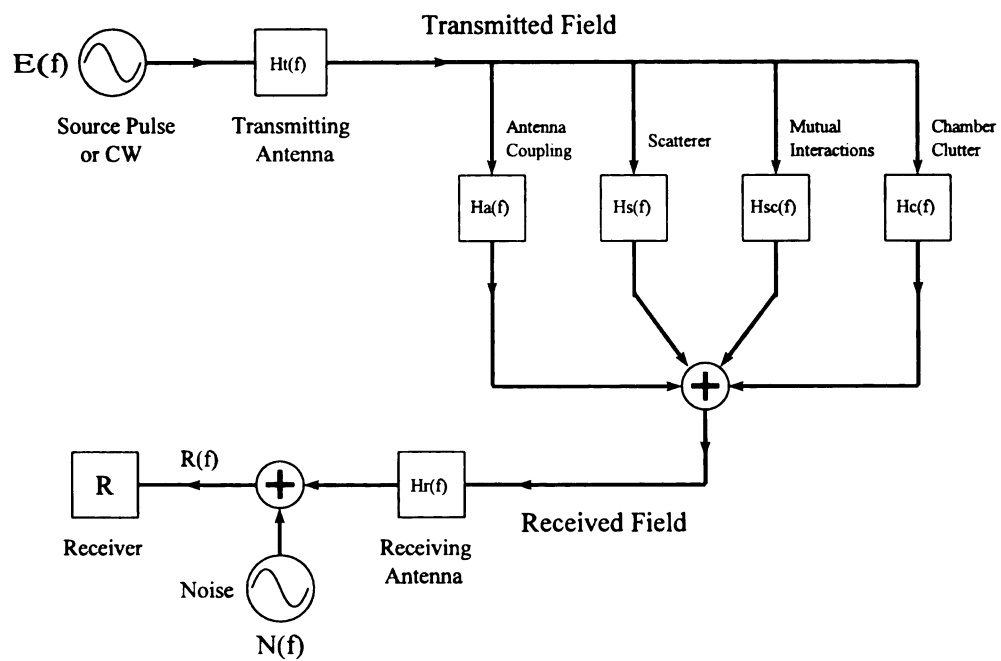


Figure 2.2 Block diagram for the anechoic chamber measurement system

impulse.

Since the measurement is conducted in an anechoic chamber, a system calibration is required for accurate results. First the block diagram for the measurement system is presented in **Figure 2.2**. $H_a(f)$ denotes the transfer function of the direct coupling from the transmit antenna to the receive antenna, while $H_c(f)$ represents the transfer function of the coupling from the transmit antenna to the receive antenna via the anechoic chamber, antenna supports, and target mount. $H_r(f)$ and $H_t(f)$ are the transfer functions of the receive and transmit antennas from the transmission line into the free-field environment, while $E(f)$ represents the spectral content of the pulse or CW source.

The background environment includes the path $H_a(f)$ and $H_c(f)$. Thus we can model the background measurement by

$$\begin{aligned} R^b(f) &= E(f) \cdot H_t(f) \cdot [H_a(f) + H_c(f)] \cdot H_r(f) + N^b(f) \\ &= S(f) \cdot [H_a(f) + H_c(f)] \cdot N^b(f) \end{aligned} \quad (1)$$

where $S(f)$ is the system transfer function $S(f) = H_r(f) \cdot H_t(f) \cdot E(f)$ and $N^b(f)$ is background random noise. If we measure some target, t , then we have

$$R^{t,b}(f) = S(f) \cdot [H_a(f) + H_c(f) + H_s^t(f) + H_{sc}^t(f)] \cdot N^{t,b}(f) \quad (2)$$

where $H^t(f)$ is the transfer function of the target, $H_{sc}^t(f)$ is the transfer function of the multi-interaction of the target with the anechoic chamber and $N^{t,b}(f)$ is the random noise spectrum. Since $H_{sc}^t(f)$ doesn't exist until the target is placed, $H_{sc}^t(f)$ is a causal term. Therefore, we have

$$R^t(f) = R^{t,b}(f) - R^b(f) = S(f) \cdot [H_s^t(f) + H_{sc}^t(f)] \cdot N^t(f) \quad (3)$$

where $N^l(f) = N^{l+b}(f) - N^b(f)$.

From the above equations, we can't solve the desired target scattering response, $Hs^l(f)$, even ignoring the multi-interaction term $Hsc^l(f)$, because the system transfer function is still not available. Thus a calibration measurement is required to find the system transfer function $S(f)$. In the calibration procedure, we need to use an object which has a known transfer function, $Hs^c(f)$. Then the calibration measurement gives

$$R^{c+b}(f) = S(f) \cdot [H_a(f) + H_c(f) + H_s^c(f) + H_{sc}^c(f)] + N^{c+b}(f) \quad (4)$$

where $Hs^c(f)$ is the transfer function of the calibration object with known transfer function, $Hsc^c(f)$ is the transfer function of the multi-interaction between the calibration and the anechoic chamber, and $N^{c+b}(f)$ is the random noise spectrum. Again, the multi-interaction term $Hsc^c(f)$ is causal. We have used a 14" sphere for calibration, since sphere scattering has the well known solution given by the Mie theory. The frequency response of a sphere can be attained in both magnitude and phase over a wide frequency band. And the sphere response is aspect-independent and thus we can avoid another calibration on aspect angle. Then we have

$$R^c(f) = R^{c+b}(f) - R^b(f) = S(f) \cdot [H_s^c(f) + H_{sc}^c(f)] + N^c(f) \quad (5)$$

where $N^c(f) = N^{c+b}(f) - N^b(f)$.

Since $Hsc^c(f)$ represents the multi-path interaction between the calibration object and the chamber, the first temporal response resulting from $Hsc^c(f)$ will apparently lag behind the ending temporal response of $Hs^c(f)$. Therefore, a time gating window can be used to eliminate the multi-path interaction portion of temporal response of $R^c(f)$. First the time response of the

calibration measurement can be calculated by

$$r^c(t) = \mathcal{F}^{-1} \{ R^c(f) \} \quad (6)$$

Since the temporal responses resulting from multi-path interaction between the calibration object and the chamber are sufficiently delayed beyond the end of the calibration object response, a time gate window function, $w(t)$, can be used to exclude the multi-path interaction portion as follows :

$$r^{cw} = r^c(t) \cdot w(t) \quad (7)$$

If $\mathcal{F}^{-1} \{ S(f) H_s^c(f) \}$ is time limited and $w(t)$ is well designed, then the $r^{cw}(t)$ will be the time response corresponding to spectral response, $S(f) H_s^c(f)$, that is

$$R^{cw}(f) = \mathcal{F} \{ r^{cw}(t) \} = S(f) \cdot H_s^c(f) \quad (8)$$

Since $H_s^c(f)$ is known and

$$R^{cw}(f) = \mathcal{F} \{ r^{cw}(t) \} = \mathcal{F} \{ w(t) \cdot \mathcal{F}^{-1} \{ R^c(f) \} \} \quad (9)$$

can be calculated, $S(f)$ can be represented as follows :

$$S(f) = \frac{R^{cw}(f)}{H_s^c(f)} \quad (10)$$

After the system transfer function is found from the calibration procedure, then

$R^l(f) = S(f) \cdot [H_s^l(f) + H_{sc}^l(f)] + N^l(f)$ will give

$$H_s^l(f) + H_{sc}^l(f) = \frac{R^l(f)}{S(f)} \quad (11)$$

if noise is ignored. Therefore,

$$h_s'(t) + h_{sc}'(t) = \mathcal{F}^{-1}\{H_s'(f) + H_{sc}'(f)\} = \mathcal{F}^{-1}\left\{\frac{R'(f)}{S(f)}\right\} \quad (12)$$

If the target impulse response is approximately time limited, and the multi-path interaction is delayed beyond the end of the target impulse response, then a time gate window, $w'(t)$, can be used to isolate the target impulse response as follows :

$$h_s'(t) = [h_s'(t) + h_{sc}'(t)] \cdot w'(t) \quad (13)$$

In our measurements, the effective operating band of the antennas is 1-7 GHz, and the sweep frequenc step size is set to 0.01 GHz. Therefore, we attain 601 measured spectral responses, including magnitude and phase, from 1 to 7 GHz. Since the antenna operating frequency band is restricted to 1-7 GHz in our measurement, we use a weighting function, the Gaussian Modulated Cosine, to window the effective measurement band and eliminate band edge discontinuities.

2.3 Data Preprocessing

To have a small time increment in the temporal responses after the inverse Fourier Transform, the spectral responses are expanded to 4096 samples by attaching (4096-601) zeros to the measured and weighted spectral responses. Therefore, there are 99 zeros artificially assigned to the band 0.01 GHz to 0.99 GHz and 3396 zeros are artificially assigned to the band 7.01 GHz to 40.96 GHz. Since a real time waveform has a complex conjugate Fourier spectrum, the 4096 spectral response points are expanded to 8192 samples by using their conjugate partners. Then we use a 8192-point Inverse Fast Fourier Transform (IFFT) to transfer the 8192 spectral response points to 8192 temporal response points for each aspect

measurement. After the IFFT, the temporal responses have a duration of 100 ns and a sampling spacing of 0.01221 ns. As analyzed above, the responses from multi-path interaction between target and chamber are apparently delayed beyond the direct backscattered responses from target, so we use a time gate window to extract the effective 820 temporal response points. **Figure 2.3** shows the 820 backscattered response points after the IFFT for the B52 target at an aspect angle of 0° .

The scale aircraft models are around 40 cm long, and thus the early-time backscattered response duration is the wave propagation time traveling twice aircraft length in the wave propagation direction, $40 \times 2 / 30 \text{ ns} = 2.67 \text{ ns}$ if the aspect angle is 0° . This estimated early-time backscattered response duration for a 40 cm scale aircraft model will cover about 219 temporal response points. Since we like to use 100 time samples as the network analog pattern, we double the time sample spacing so that the new time sample spacing is 0.02442 ns and the response duration time of the 100 new time samples is 2.442 ns. With 0° aspect angle the 100 new temporal responses cover about 3/4 of the B52's early-time responses, while they can cover the whole TR1 response.

Since the beginning time of a target response is difficult to locate in practice, we design a simple detection algorithm to find the beginning of a response. When the beginning response time is decided, we pick 100 time response points, starting from the detected beginning of the response, as the network aspect prototype. With aspect angle changing and noise contamination, our simple detection isn't consistent among aspect angles. However, we still use these truncated patterns, with inconsistency for some aspect ranges, as stored aspect response prototypes to emulate the practical situation. An example of the inconsistencies in

our time domain stored aspect patterns will be given in chapter 6, Target Discrimination using Neural Network with Spectrum Magnitude Response. To systematically process target discrimination the 100 time response points are normalized to an energy of 1. **Figure 2.4** shows the 100 normalized time response points acting as a network analog pattern for the B52 target at an aspect angle of 0° .

In this thesis, each target is measured at aspect angles from 0° to 28.8° with an azimuthal increment of 0.9° . Therefore each target has 33 aspect measurements. We then select 17 aspect responses from 0° to 28.8° with aspect increment 1.8° for each target as network training/stored aspect patterns, and 16 aspect responses as untrained/unstored network generalization test patterns from 0.9° to 27.9° with the same increment. Therefore, every untrained/unstored test pattern resides at the middle of two training/stored aspect patterns. **Figure 2.5** shows the 68 100-sample aspect time responses, truncated from 68 820-sample time-gated IFFT responses, for 4 scale aircraft models. **Figure 2.6** presents the 68 normalized aspect time responses serving as the 68 aspect stored patterns for time domain networks. The details for bipolar coding schemes and data processing of spectrum magnitude network will be discussed in later chapters.

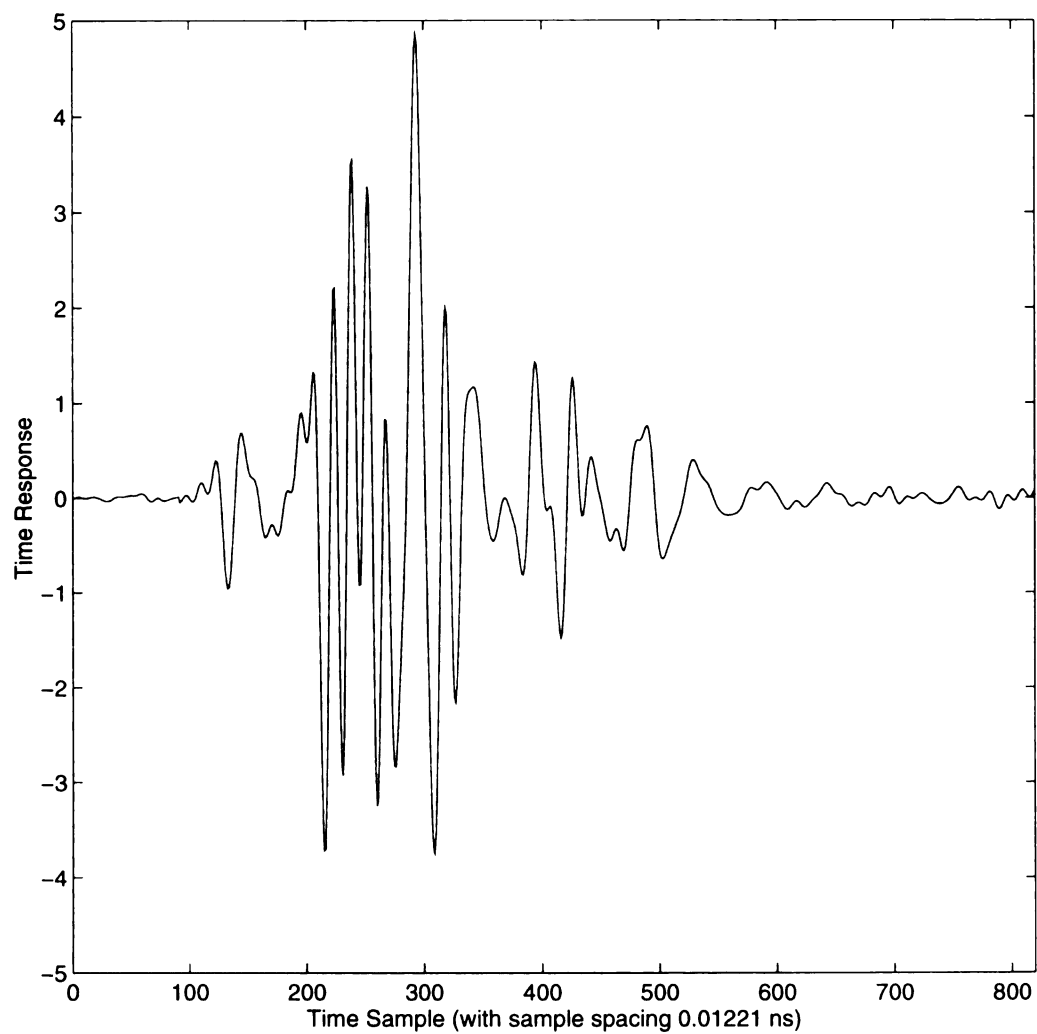


Figure 2.3 Time gated backscattered responses of B52, at aspect angle 0° , after 8192-point IFFT.

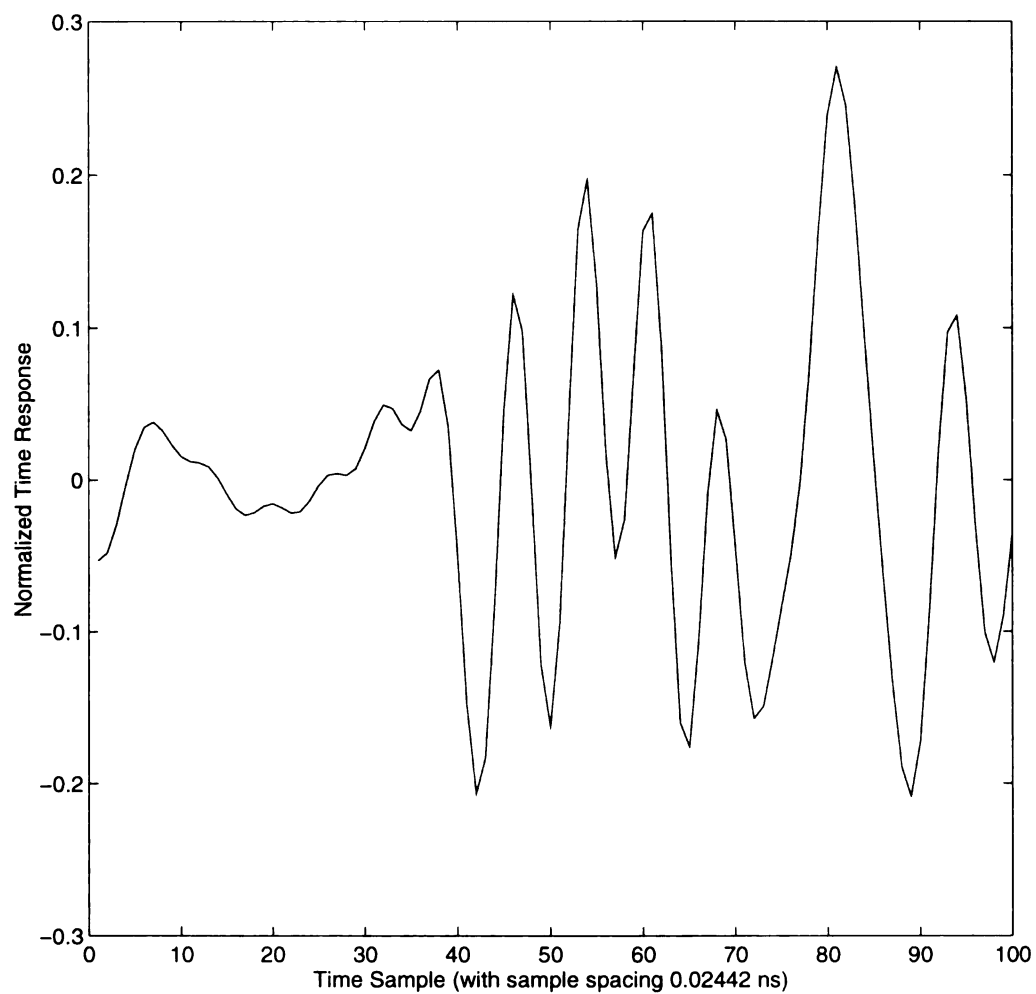


Figure 2.4 Normalized time response of B52, at aspect angle 0° , serves as a network analog pattern.

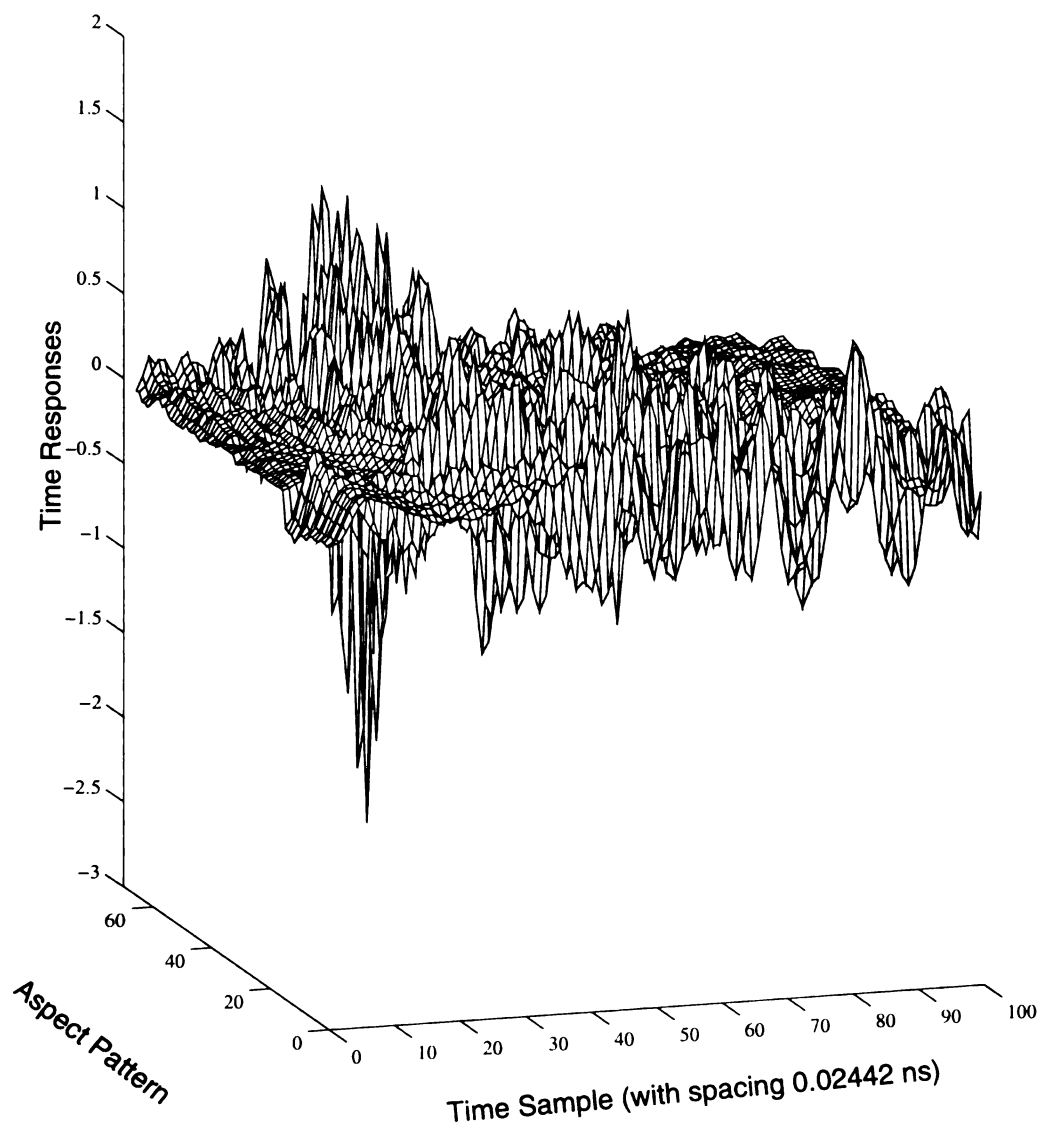


Figure 2.5 68 100-sample aspect time responses, truncated from 68 820-sample time-gated IFFT responses, for 4 scale aircraft models.

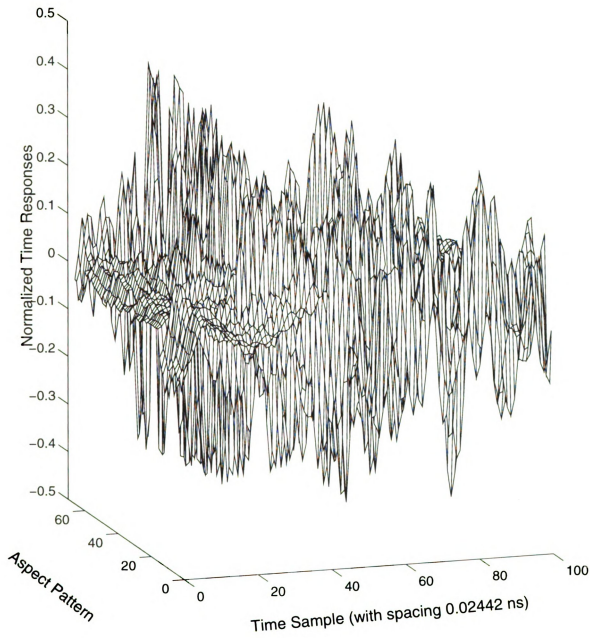


Figure 2.6 68 normalized 100-sample aspect time responses serving as aspect stored patterns for time domain networks.

CHAPTER 3

Multi-Layer Feedforward with Error Backpropagation and Generalized Inverse Networks

3.1 Introduction

In this chapter, we will discuss two training based artificial neural networks, Multi-layer Feedforward with Error Backpropagation (ML/BP) networks and Generalized Inverse (GI) network. The limitations of the Perceptron were pointed out by Minsky and Papert [25] in 1969. The perceptron learning can poorly behave with nonseparable data and it will stop learning upon the point where a critical solution is reached. Therefore, the perceptron learning will only barely work. Widrow and Hoff [26] developed a model for ADALINE (ADaptive LINear Elements), an important variation of perceptron learning. This model uses Widrow-Hoff rule or LMS (least mean square) algorithm to eliminate the oscillation caused by nonseparable learning examples and approximate to the solution with least mean square. However, it is not necessary the optimal one.

3.2 Multi-layer Feedforward with Error Backpropagation Networks

The Multi-layer feedforward with Error-Backpropagation (ML/BP) networks are most well known, widely applied, effective and flexible artificial neural networks. The backpropagation learning ideas were rediscovered independently by different workers (Werbos [27]; Parker [28]; Rumelhart, Hinton, and Williams [29]). Strictly speaking the backpropagation is a learning algorithm, not a type of network. Currently the backpropagation is the most important and most widely used algorithm for doing hard

connectionist learning.

Essentially the ML/BP network involves two phases. The first phase, the forward phase, occurs when the input is presented and propagated forward through the network to attain an value for each processing element. Then all current outputs are compared with the desired outputs, and the differences, or errors, are computed. In the second phase, the backward phase, the recurring difference computation from the first phase is now performed in a backward direction.

Generally the ML/BP net is composed of a hierarchy of processing units, organized in a series of two or more mutually exclusive sets of artificial neuron layers. Essentially this feedforward network uses a refinement of the Widrow-Hoff technique, which calculates the difference between real outputs and the desired outputs. The weights are changed in proportion to the error times the input. Therefore, we require inputs, outputs and the desired outputs all at any learning neuron. However, this is hard to do with hidden units, since you don't know how much responsibility a particular hidden unit should take.

To solve this problem, the neuron learnings are run backward, so you can tell how strongly a particular neuron is connected. Each hidden learning neuron's error is a weighted sum of the errors in the successive layer. Therefore, the forward phase is used to estimate the error, then the backward phase is introduced to modify weights based on backpropagation algorithm so that the error is decreased.

Figure 3.1 illustrates a typical ML/BP network structure. The U represents input layer with n inputs, while $H_1, \dots, H_I$ and H_J represent hidden layers with I neuron units in H_1 and J neuron units in H_J respectively. And the O_K denotes the output layer with K outputs.

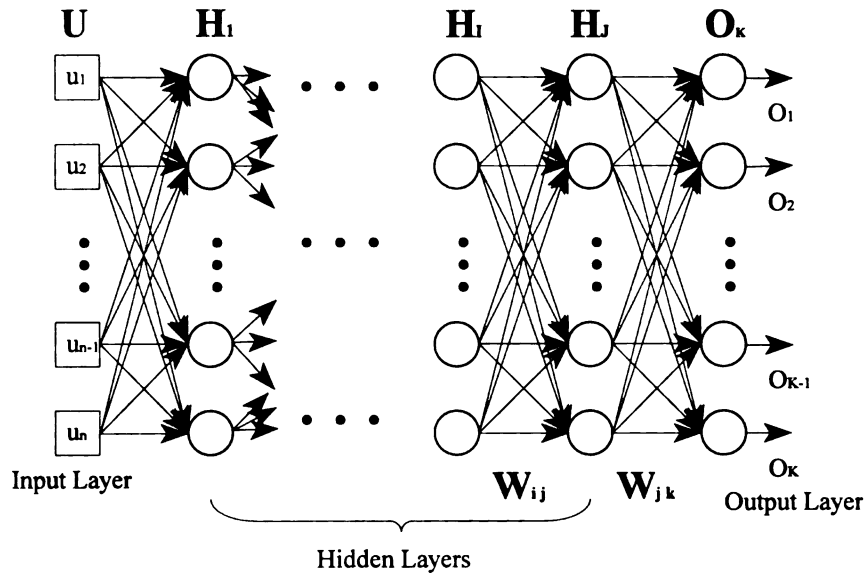


Figure 3.1 Structure of a multi-layer feedforward with error backpropagation network

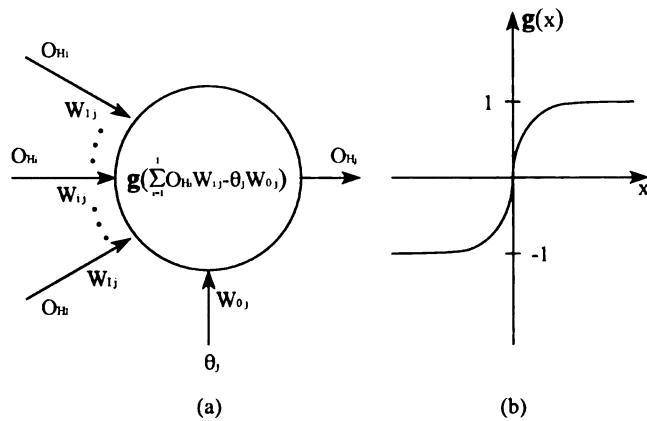


Figure 3.2 (a) An artificial neuron unit model. The O_{Hi} 's are outputs from previous layer H_1 , θ_j is the bias for neuron j and W_{ij} 's are interconnecting weights between layer H_1 and the neuron j of layer H_j . (b) Activation function, $g(x)$, used for bipolar process.

W_{ij} represents interconnecting weights between the hidden layers H_i and H_j , while W_{jk} denotes interconnecting weights between the hidden layer H_j and output layer O_k . In **Figure 3.2**, (a) shows an artificial neuron unit model used in **Figure 3.1**, and a nonlinear activation function used in an artificial neuron for bipolar process is illustrated in (b).

3.2.1 Gradient Descent Rule

Gradient descent rule is an important part in most connectionist learning algorithms, especially mean squared error and backpropagation. It is a mathematical approach to minimizing the error between actual and desired outputs. The weights are modified by an amount proportional to the first derivative of the error function with respect to the weight. The Widrow-Hoff (or Delta) Rule is one example of a gradient descent rule.

Suppose we have a network weight $W = (w_1, \dots, w_n)$ for a single cell problem, and it produces a differentiable measure of error $E(W)$. Since the error function is differentiable, we can compute its derivative (or gradient) vector

$$\nabla E = \left(\frac{\partial E}{\partial w_1} \dots \frac{\partial E}{\partial w_n} \right) \quad (1)$$

at any point of weight space. This gradient vector gives the direction in the weight space that has the maximum increase of the error when an infinitesimally small weight change is made in that direction. Therefore the vector $-\nabla E(W)$ will minimize error for a sufficiently small step. This introduces a learning algorithm with which weights, W , will continue error convergence by evaluating $-\nabla E(W)$ and taking small steps in that direction until some convergent criterion is reached. So the gradient descent learning rule can be formulated as follows with a small positive learning rate η :

$$W_{Updated} = W - \eta \nabla E(W) \quad (2)$$

3.2.2 ML/BP Learning Mechanism

The ML/BP network is a supervised net, so it needs a training (or example) set including input pattern set and the corresponding desired output set to begin learnings. The network architecture learns to adapt to the training set by using the errors backpropagated layer by layer to adjust its interconnection weights. As we can observe from **Figure 3.1** that the network interconnects any two adjacent layers by synapses and each layer has its artificial neurons, but there are no interconnection synapses between any two neurons in the same layer. The learnings adjust synapse weightings connecting any two neurons of any two adjacent layers in backward direction. The network iteratively continues its learnings pattern by pattern until the error criterion is reached, thus a well trained network will satisfy the given training pattern set with error less than the required criterion.

Suppose a ML/BP network has output layer O_k with K neurons, the first backward hidden layer H_j with J neurons and the second backward hidden layer H_i with I neurons. Let T_k^r denotes the k^{th} target (or desired) output bit for the training pattern r , W_{jk} denotes the synapse weighting connecting the j^{th} neuron of hidden layer H_j to the k^{th} neuron of the output layer O_k , and W_{ij} denotes the synapse weighting connecting the i^{th} neuron of hidden layer H_i to the j^{th} neuron of hidden layer H_j . Then the network has

$$\begin{aligned} O_k^r &= G_{\beta} \left(\sum_{j=1}^J W_{jk} H_j^r \right) \\ H_j^r &= G_{\beta} \left(\sum_{i=1}^I W_{ij} H_i^r \right) \end{aligned} \quad (3)$$

1.
the
the

Compare:
through the
Sm
200,000

An activation shown in **Figure 3.2** is used here for bipolar $[-1 \ 1]$ process,

$$G_{\beta}(v) = \frac{2}{1 + \exp(-\beta v)} - 1 \quad (4)$$

and this kind of nonlinear sigmoid function is frequently used to emulate the bi-state for a biological neuron, activated or inactive.

Then we define an error function for training pattern $\mathbf{r}$ as follow,

$$E^r = \sum_{k=1}^K E_k^r = \frac{1}{2} \sum_{k=1}^K (T_k^r - O_k^r)^2 \quad (5)$$

Now, we have calculated the error between real network output, $\mathbf{O}^r$, and target output, $\mathbf{T}^r$, for the training pattern $\mathbf{r}$. After the error is available, the gradient descent rule [20][29][30] for the output unit $\mathbf{k}$ with respect to weight W_{jk} is given by

$$\begin{aligned} \Delta W_{jk} &= -\eta \frac{\partial E^r}{\partial W_{jk}} = -\eta (T_k^r - O_k^r) \left[-\frac{\partial O_k^r}{\partial W_{jk}} \right] \\ &= \eta H_j^r \left[G'_{\beta} \left(\sum_{j=1}^J W_{jk} H_j^r \right) (T_k^r - O_k^r) \right] = \eta H_j^r \delta_k^r \end{aligned} \quad (6)$$

where

$$\delta_k^r = G'_{\beta} \left(\sum_{j=1}^J W_{jk} H_j^r \right) (T_k^r - O_k^r) \quad (7)$$

Compared to the Widrow-Hoff rule, δ_k^r , can be regarded as the error backpropagated through the nonlinear activation from output unit $\mathbf{k}$.

Since all neuron j 's, $j=1 \dots J$, in hidden layer $\mathbf{H}_j$ are connected to the output neuron $\mathbf{k}$ in our model, all the hidden neurons in layer $\mathbf{H}_j$ should be responsible for error occurred at

the output unit k. The backpropagation algorithm proposes that the error occurred on any output unit k will be broadcasted to neurons of previous layer which are connected to the unit k. Therefore the gradient descent update rule for the interconnecting weight W_{ij} between hidden layers H_i and H_j is given by

$$\Delta W_{ij} = -\eta \frac{\partial E^r}{\partial W_{ij}} = -\eta \frac{\partial H_j^r}{\partial W_{ij}} \frac{\partial E^r}{\partial H_j^r} \quad (8)$$

where

$$\frac{\partial H_j^r}{\partial W_{ij}} = G'_\beta \left(\sum_{i=1}^I W_{ij} H_i^r \right) H_i^r \quad (9)$$

$$\frac{\partial E^r}{\partial H_j^r} = \sum_{k=1}^K (T_k^r - O_k^r) \frac{\partial O_k^r}{\partial H_j^r} = \sum_{k=1}^K (T_k^r - O_k^r) G'_\beta \left(\sum_{j=1}^J W_{jk} H_j^r \right) W_{jk} = \left(\sum_{k=1}^K W_{jk} \delta_k^r \right) \quad (10)$$

Therefore we have

$$\begin{aligned} \Delta W_{ij} &= \eta H_i^r G'_\beta \left(\sum_{i=1}^I W_{ij} H_i^r \right) \left(\sum_{k=1}^K W_{jk} \delta_k^r \right) \\ &= \eta H_i^r G'_\beta(\tilde{H}_j^r) \tilde{E}_j^r = \eta H_i^r \delta_j^r \end{aligned} \quad (11)$$

where

$$\begin{aligned} H_j^r &= G_\beta(\tilde{H}_j^r) \\ \tilde{E}_j^r &= \sum_{k=1}^K W_{jk} \delta_k^r \end{aligned} \quad (12)$$

$$\delta_j^r = G'_\beta(\tilde{H}_j^r) \left(\sum_{k=1}^K W_{jk} \delta_k^r \right) \quad (13)$$

In equation (11), $\tilde{E}_j^r$ denotes the errors backpropagated to neuron j of hidden layer H_j ,

through interconnecting synapse weights from all output neurons. Since there are no apparent desired outputs to calculate error for hidden layer, this backpropagated error from all neurons of previous layer can be treated as a measure of current error for this hidden layer. Again δ_j^r represents the error backpropagated through the nonlinear activation function.

3.3 Gaussian Noise Generation

In previous chapter we have 33 normalized azimuthal aspect responses of 100 analog sample points from 0° to 28.8° with aspect increment 0.9° for each target. The 17 of them from 0° to 28.8° with aspect increment 1.8° are selected as training pattern set, while the rest 16 of them are kept untrained from 0.9° to 27.9° with the same increment. In this thesis, we simulate all neural networks by using software package, the Matlab 4.0. To test network performances, we add noises to test inputs before putting them to network. In order to have a basis for comparing network performances, we define a Signal-To-Noise (SNR) in dB. Suppose we have a continuous input pattern $s = \{s_1 \ s_2 \ \dots \ s_{100}\}$, then we calculate its average signal power by

$$\bar{s}^2 = \frac{1}{100} \sum_{m=1}^{100} s_m^2 \quad (15)$$

If the 100 added noises, $N = (N_1 \ N_2 \ \dots \ N_{100})$, have average noise power

$$\bar{\sigma}^2 = \frac{1}{100} \sum_{m=1}^{100} E(N_m^2) \quad (16)$$

, then we define the SNR and SNR in dB as follows

$$\begin{aligned}
 SNR &= \frac{\bar{s}^2}{\bar{\sigma}^2} \\
 SNR (dB) &= 10 \log_{10} \frac{\bar{s}^2}{\bar{\sigma}^2}
 \end{aligned} \tag{17}$$

And we simulate a noise set with $SNR=snr$ by using Matlab's random noise generator to generate a Gaussian noise set, $N(0, \sigma_G)$, with 0 mean and σ_G variance given by

$$\sigma_G = \sqrt{\frac{\bar{s}^2}{snr}} \tag{18}$$

Ten noisy circumstances with SNR's of [40, 30, 20, 14, 10, 6, 3, 0, -3, -6] dB have been simulated for each network to test network noise tolerances. The solid line of **Figure 3.3** shows the 100 time responses of B52 used for network process at azimuthal aspect 0°, while dash-dot line denotes the one with added noises of 10 dB. A noisy response with 0 dB is shown in **Figure 3.4**, while one with -3 dB is shown in **Figure 3.5**. After noises are added to the test signal, the noisy signal is normalized again before proceeding it. Network performance under each case is simulated 10 times and then average is calculated and used as resultant performances.

3.4 ML/BP Network Trainings and Performances for Radar Target Discrimination

In this section, we implement two ML/BP nets, one without hidden layer and the other with one hidden layer, and train the two networks by using 17 time-domain azimuthal aspect responses for each target. There are four targets used for network trainings, so total 68 time domain azimuthal aspect responses are used for network training. Here we use continuous

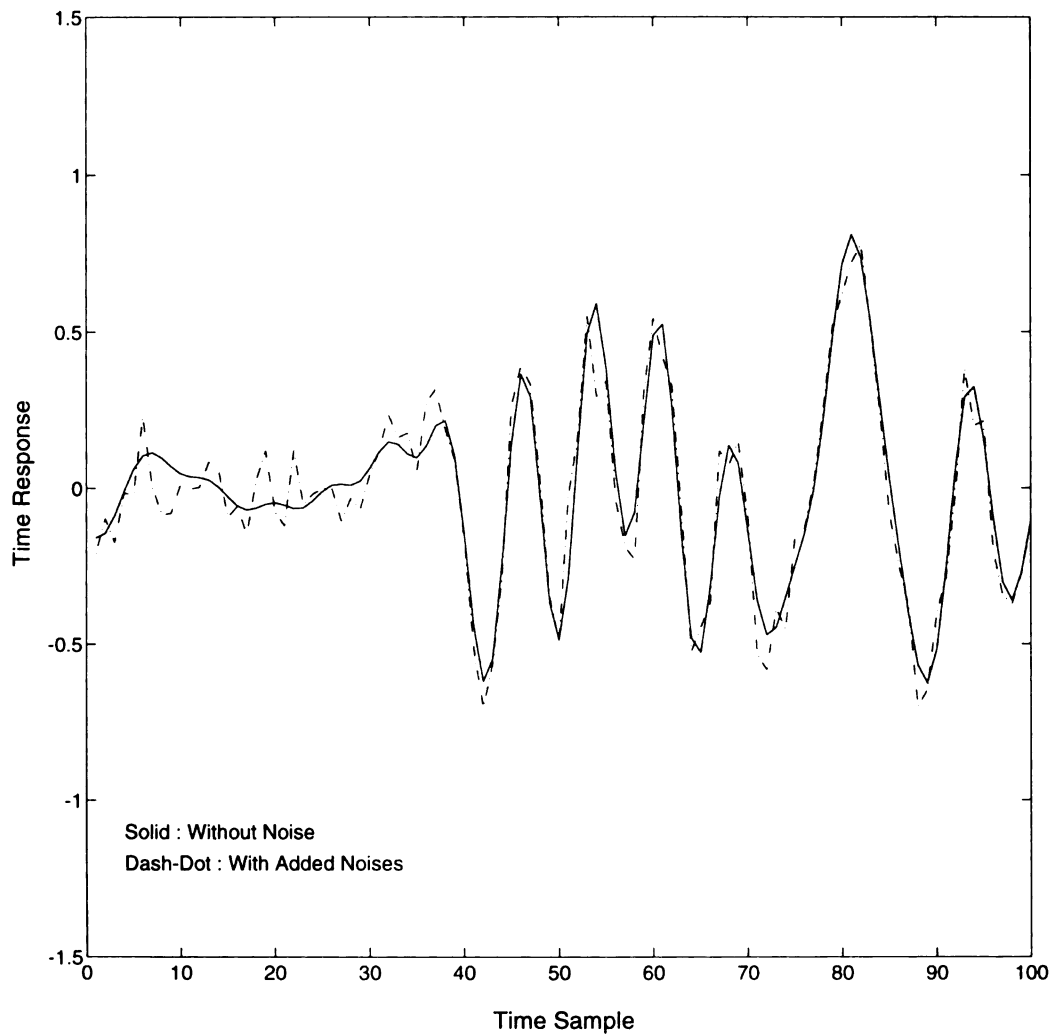


Figure 3.3 100 time responses of B52 used for network process at azimuthal aspect 0° . Solid line denotes responses without noise, while the dash-dot denotes the one with added noises of SNR= 10 dB.

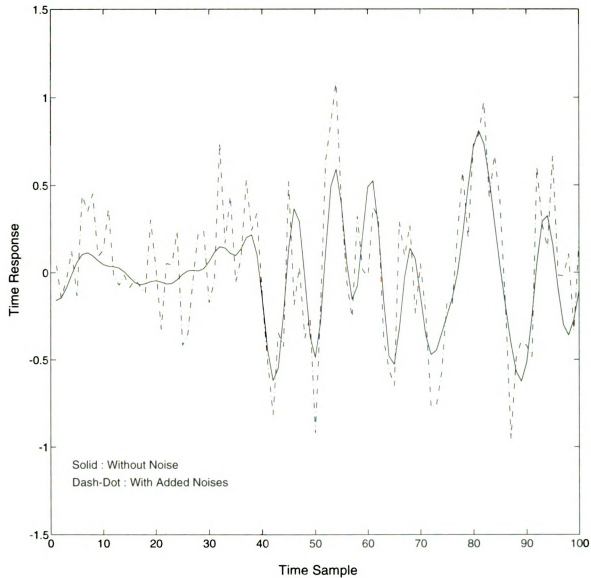


Figure 3.4 100 time responses of B52 used for network process at azimuthal aspect 0° . Solid line denotes responses without noise, while the dash-dot denotes the one with added noises of SNR= 0 dB.

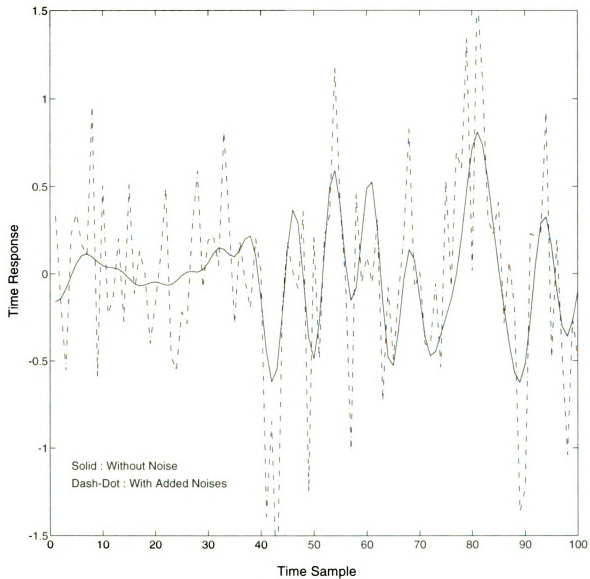


Figure 3.5 100 time responses of B52 used for network process at azimuthal aspect 0° . Solid line denotes responses without noise, while the dash-dot denotes the one with added noises of SNR= -3 dB.

inputs but bipolar outputs for network process. Four output patterns are used to serve as desired target codes. They are : [1 -1 -1 -1] for B52, [-1 1 -1 -1] for B58, [-1 -1 1 -1] for F14 and [-1 -1 -1 1] for TR1. Then we test the networks' performances by adding Gaussian noises of 10 different SNR's to the training patterns, and test networks' generalization capability by adding noises to 16 untrained azimuthal aspect responses for each target.

3.4.1 ML/BP Without Hidden Layer

There is no hidden layer for this network. Since we use continuous input values and bipolar output form, the network has 100 inputs in its input layer and four units at the output layer. The network is trained by using the 68 training pattern set for four targets to learn the artificial synaptical interconnections between input and output layers. It is equivalent to say a 100 x 4 matrix is designed to learn network synaptic weights. We randomly initialize network weights by Gaussian function with norm 1, and then times initial weights by 0.2, since the activation function will saturate if the input value to the function is large. **Figure 3.6** shows two activation functions, $G_{\beta}(x)$, with different activation parameters, while their first derivatives (or slopes), $dG_{\beta}(x)/dx$, are shown in **Figure 3.7**, solid line for $\beta=1$ and dash one for $\beta=3$. For example, if the nonlinear sigmoid function has activation parameter $\beta=2$ and an input value 2, then the activation function will have output $G_2(2)=0.964$ and slope $G_2'(x)|_{x=2} = 0.0707$. The gradient descent learning rule working on backpropagated error will become idle with a derivative near 0 even though the error is as large as -2 or 2. Therefore, the initial weights are required to be small enough to have weighted sums unsaturated to neurons of the next layer if the learning in this layer is expected in the beginning. We train training patterns in random order to avoid being trapped in local minima. The momentum has

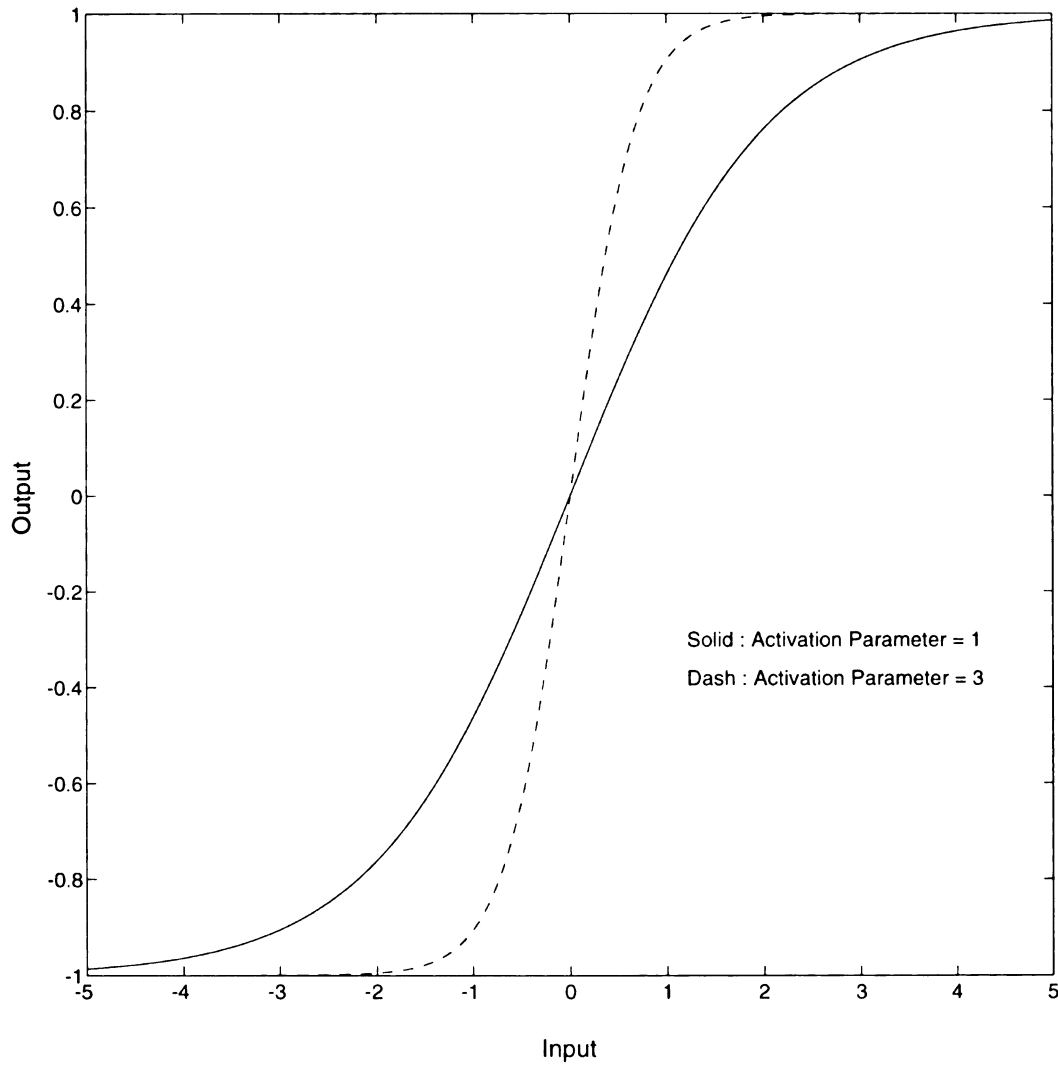


Figure 3.6 Activation functions, $G_\beta(x)$, with different activation parameters. Solid line denotes the one with activation parameter $\beta=1$, while the dash line denotes the one with parameter $\beta=3$

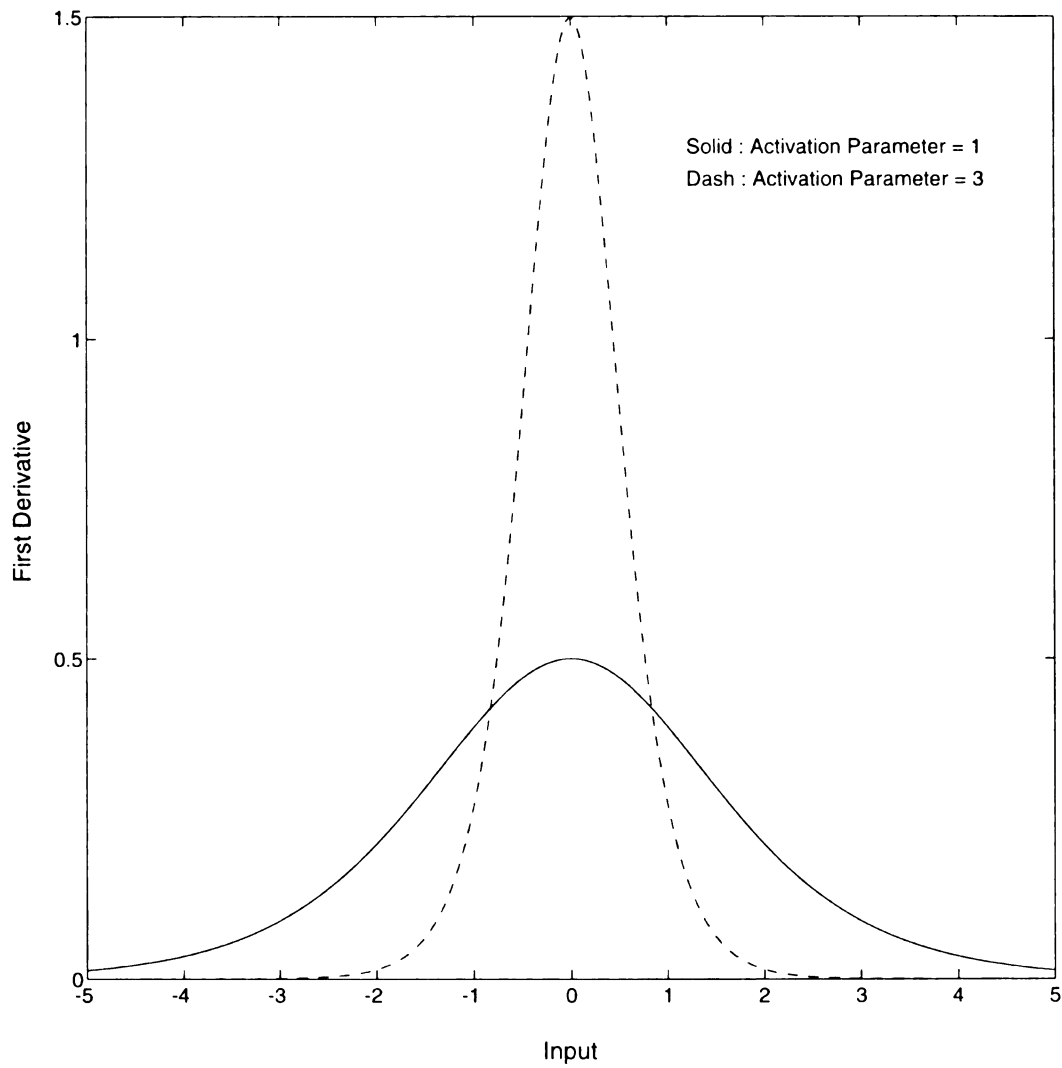


Figure 3.7 The first derivatives, $d[G_{\beta}(x)]/dx$, of activation function with different activation parameters. Solid line denotes the one with activation parameter $\beta=1$, while the dash line denotes the one with parameter $\beta=3$.

also been used to increase learning efficiency and reduce the chance of being trapped in local minima.

The learnings between patterns are competitive. For example, if there is a training pattern, p , has 1 as the k^{th} desired output bit, then the learning for this specific training pattern is expected to have

$$G_{\beta} \left(\sum_{n=1}^{100} u_n W_{nk} \right) \rightarrow 1 \quad (19)$$

At the same time the weights, W_{nk} , are adjusted by another 67 training patterns to satisfy their respective desired output. Therefore, if learnings of most training patterns adjust the weights in the opposite direction to the training pattern p , then the iterative learning epochs may bring the k^{th} output of training pattern p far away from its desired one, say -1, before those dominant learning patterns converge to their respective desired outputs within correct saturation regions (or convergent criterions). To avoid this phenomenon to occur, we train network by initially using a larger activation parameter, then using a smaller one after most training patterns converge.

For example, if the k^{th} output neuron initially use an activation parameter 1 and have input value -2 for the training pattern p , i.e.

$$\sum_{n=1}^{100} u_n^p W_{nk} = -2 \quad (20)$$

then the network currently has $G_1(-2)=-0.7616$ instead of the desired one, 1, at the k^{th} output corresponding to the training pattern p . According to gradient descent rule, the network

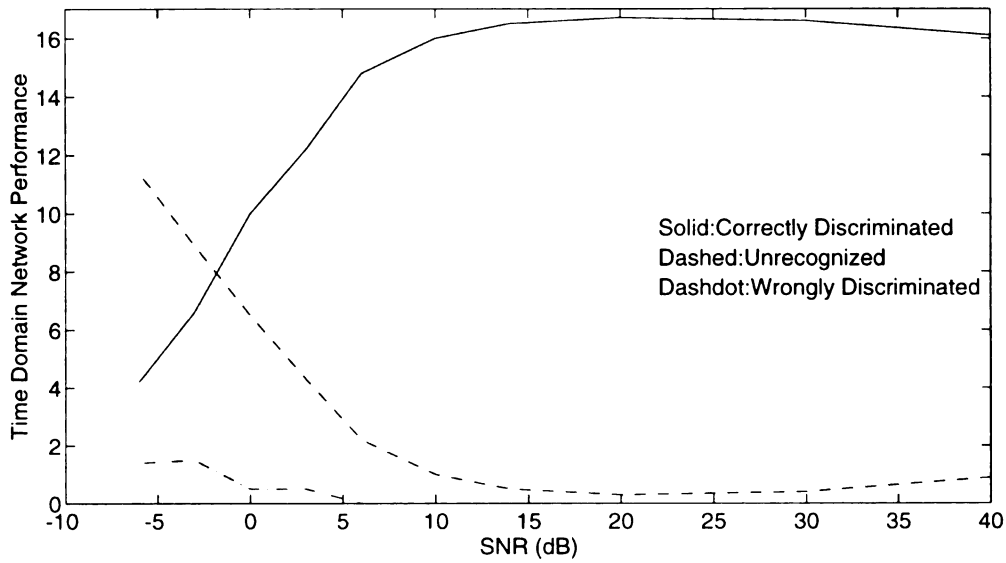
learning would have reduced the error $1 - (-0.7616)$ to a smaller value. But the dominant learnings coming from the other 67 training patterns may adjust the learning weights for output unit k , W_{nk} , in a direction opposite to the pattern p , and then force the input to output neuron k for pattern p to deeply enter wrong area, more negative than -2 , instead of approaching toward the positive side. From equation(4) or **Figure 3.7**, we can find the first derivative of activation function with parameter $\beta=1$, $G_\beta'(x)$, larger than 0.051 for inputs within interval $[-3.6 \ 3.6]$. It means dominant learnings won't converge until their respective inputs to output neurons have a distance larger than 3.6 from origin. The adjusted weights might have pushed the input of output neuron k for the training pattern p deeply into a more negative area, less than -4 , after dominant learnings converge. Therefore the successive learnings for the training pattern p have no chance to pull its wrong output back to the correct side. Since, according to the gradient descent learning rule, a learning on the weights connected to output neuron k for the training pattern p has $\Delta W_{nk} = u_n G_\beta'(x) (T_k^p - O_k^p)$ with $G_2'(x)|_{x=-4} = 0.0013 \ll 1$, the learning is idle even though its error, $(T_k^p - O_k^p) = 1 - G_2(-4) = 1.9993$, is much larger than those convergent ones. Therefore we initially use a parameter β around 2 or 3 to start trainings, and then use a smaller one to ensure continuous learnings for those few nonconvergent training patterns after most training patterns converge.

The trainings consist of two sections. The first training part uses hard limiter to determine if all training patterns converge to their respective desired outputs. The hard limiter function is a simple Signum (**Sign**) function, and it has

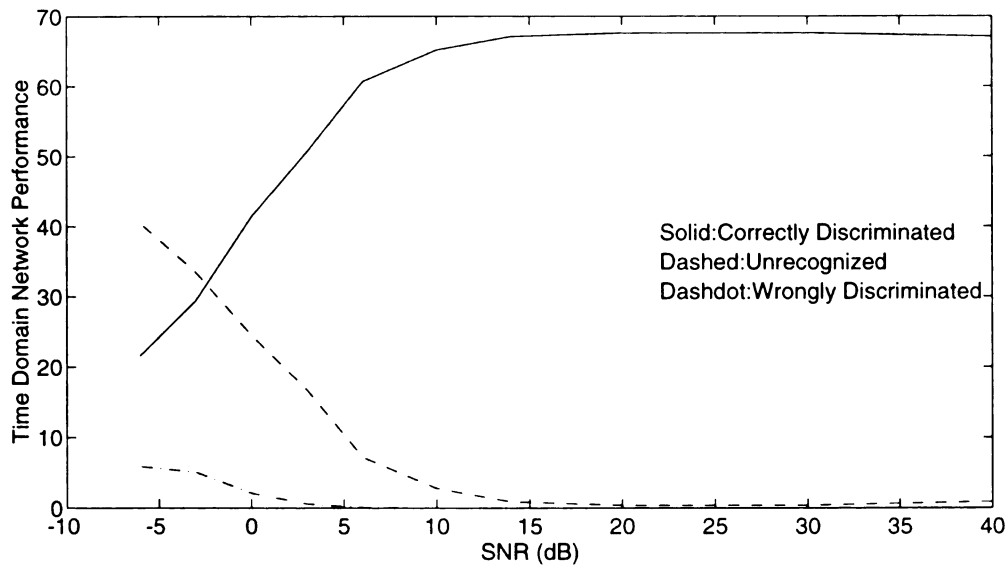
$$\text{Sign}(v) = \begin{cases} 1, & v > 0 \\ -1, & v < 0 \end{cases} \quad (21)$$

for bipolar process. When this convergent criterion is reached, all training patterns under no contaminations can be correctly discriminated by the network with a Sign threshold function. However, it doesn't indicate tolerance to contaminations. For example, it may occur that the last convergent training pattern, which has a desired value 1 for output neuron k , converges its input value to 0.001 and stop. This critical convergent weight won't correctly serve for the pattern with little distortion resulting in weighted sum (or input) to neuron k shifting toward negative side with an amount larger than 0.001. Therefore, the extra training is required to converge all training patterns far away from the origin and then increase network tolerances. However, it may only affect the last few convergent training patterns. The first training section takes 164 epochs to converge, and then another 1012 extra training epochs are followed.

Figure 3.8 (a) shows the performance of the ML/BP network without hidden layer and without extra trainings for the 17 training patterns of target B58, while **Figure 3.8** (b) shows the overall network performance for all 68 training patterns. **Figure 3.9** (a) presents the network performance with extra trainings for the 17 training patterns of target B58, while **Figure 3.9** (b) shows the overall network performance for all 68 training patterns. It is apparent the extra training makes the network have better tolerance for the 17 training patterns of target B58. Another interesting phenomenon is that the network prefer unknown to wrong discrimination. The network generalization capability is tested by using the other 64 untrained patterns for four targets. **Figure 3.10** (a) shows the network generalization performance without extra trainings for the 16 untrained patterns of target B58, while **Figure 3.10** (b) shows the overall network generalization performance for the 64 trained patterns.



(a)

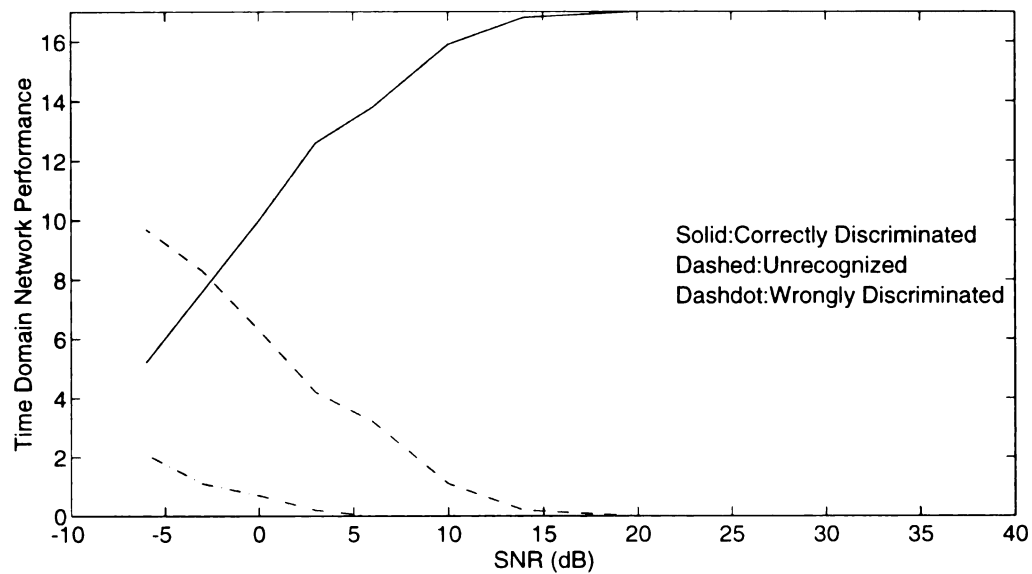


(b)

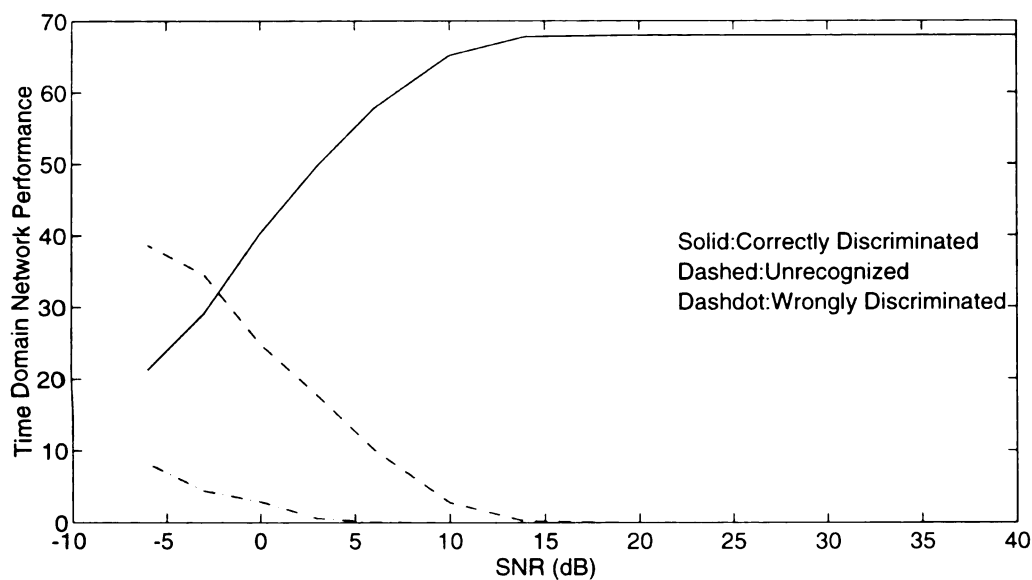
Figure 3.8 Performance of ML/BP network w/o hidden layer and w/o extra trainings vs. SNR(dB).

(a). Network performance for the 17 training patterns of target B58.

(b). Overall performance for all 68 training patterns.



(a)

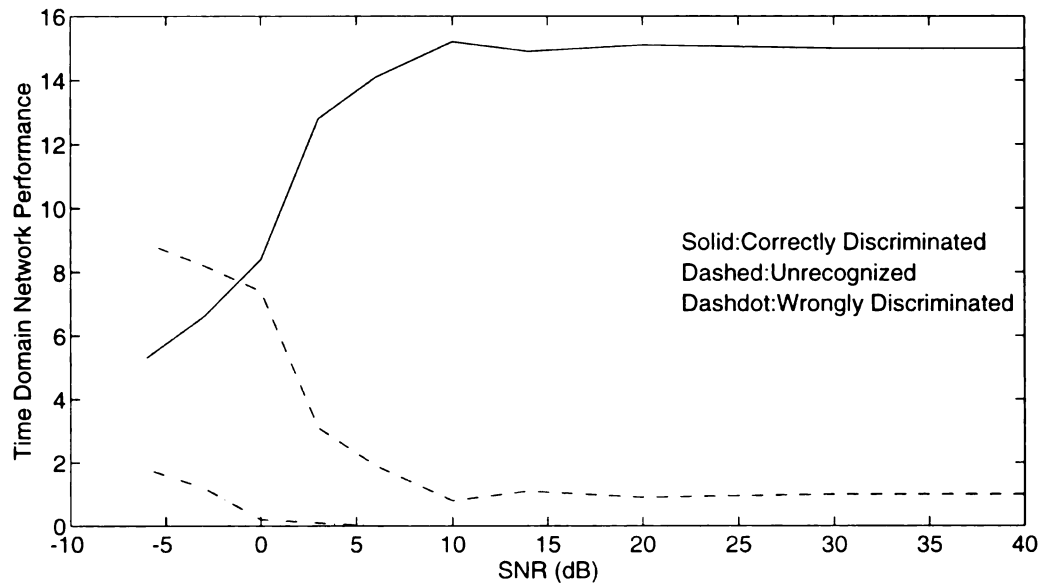


(b)

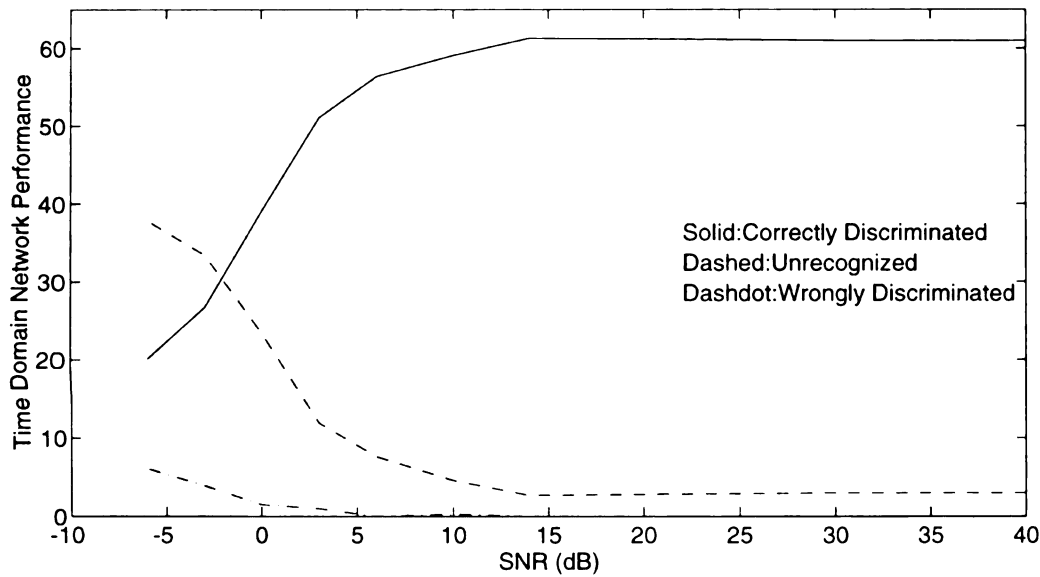
Figure 3.9 Performance of ML/BP network w/o hidden layer and with extra trainings vs. SNR(dB).

(a). Network performance for the 17 training patterns of target B58.

(b). Overall performance for all 68 training patterns.



(a)

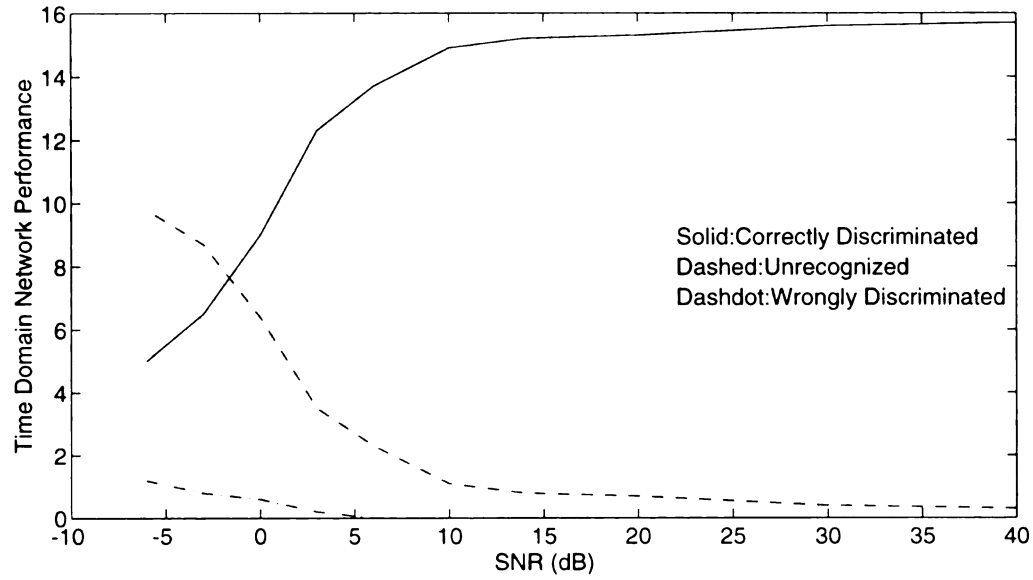


(b)

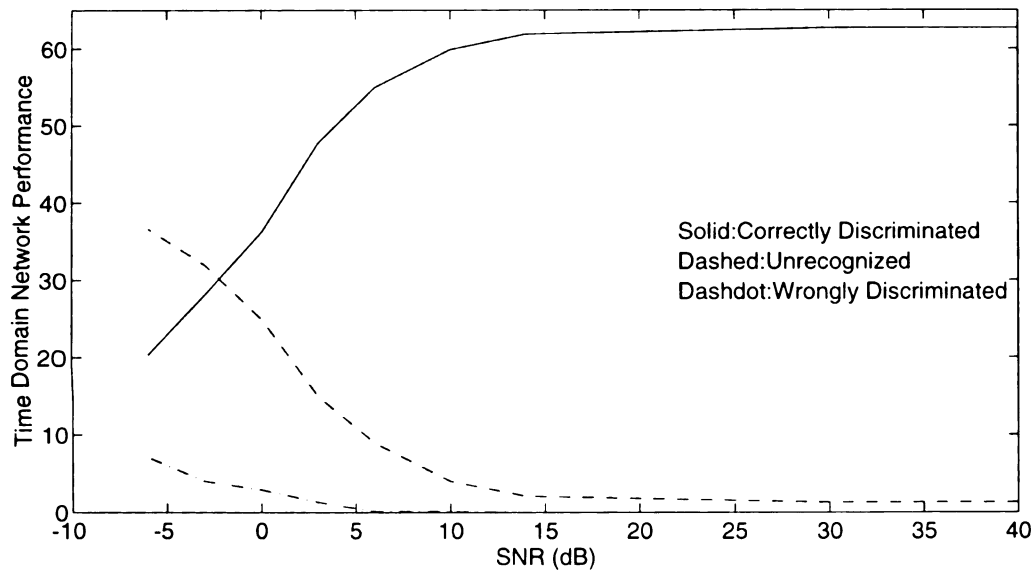
Figure 3.10 The generalization performance of ML/BP network w/o hidden layer and w/o extra trainings vs. SNR(dB).

(a). Network generalization performance for the 16 untrained patterns of target B58.

(b). Overall generalization performance for the 64 untrained patterns.



(a)



(b)

Figure 3.11 The generalization performance of ML/BP network w/o hidden layer and with extra trainings vs. SNR(dB).

(a). Network generalization performance for the 16 untrained patterns of target B58.

(b). Overall generalization performance for the 64 untrained patterns.

Figure 3.11 (a) presents the network generalization performance with extra trainings for the 16 untrained patterns of target B58, while **Figure 3.11** (b) shows the overall network generalization performance for the 64 trained patterns. Again the network with extra training has larger generalization tolerance for the 16 untrained patterns of target B58.

3.4.2 ML/BP With One Hidden Layer

We add a hidden layer with 25 neurons to the previous ML/BP net to see if the network noise tolerance and generalization ability will increase. This network trainings take much more time and manipulation to learn all the training patterns without error, since the hidden layer has ambiguous desired outputs. The competitive learnings usually occur between interconnection weights W_{nj} 's with activation parameters β_j on the hidden layer H_j and W_{jk} 's with β_k on the output layer O_k . If the activation parameter β_j for neurons in the hidden layer is larger than the one for the output layer, then the learnings for W_{nj} are greatly reduced and the W_{jk} will dominate the network learning. Referring to **Figure 3.6** and **Figure 3.7**, the larger activation parameter will make the hidden layer learning region (transition region of activation function) narrower and then most inputs to hidden neurons will enter the saturation area. Therefore the backpropagation learning will be inhibited by a derivative about 0. It means the outputs at hidden layer have mostly been either 1 or -1 when the learning on weights between the output layer and the adjacent hidden layer only proceeds for few iterations. The network thus degenerate to one without a hidden layer, and the W_{nj} works as a transform function only transferring inputs to the dimension of the hidden layer.

To make hidden layers useful, the activation parameter for hidden neurons should not be too large with respect to the neuron input scale. The hidden layer play an important role

which acts as a new representation of inputs. This new representation of inputs will help to solve linearly nonseparable problems of which the perceptrons, ADALINE and network without hidden layers can't asymptotically converge to a solution. So the learnings occurred on W_{nj} between the input and hidden layers tend to find new representations for all the training patterns which can make network outputs converge to their respective desired outputs.

In the previous section, we found that if an input of neuron is beyond the transition region of the neuron activation function, then the learning for this input is nearly idle even though the network output has a large error. This phenomenon is not expected, since it prohibits those training patterns resident in wrong saturation region from learning. Although gradually releasing the activation function's slope may help to pull back those nonconvergent ones, it is only effective for the inputs not too far away from the transition area. One may need to reinitiate network weights and retrain the network, if the nonconvergent training patterns have been pushed deeply into saturation area of the wrong side after those dominant training patterns converge.

So we design a new error function with reenforced order of error to overcome the idle learning caused by a derivative near 0. The ML/BP net defines an error function as the sum of squared errors shown in equation(5). From the gradient descent rule, equation (6), the weight learning is proportional to the derivatives of error function and activation function respectively. Therefore, we can define an error function with high order, say 4, then its first derivative still has an order of 3 and will overcome the derivative of activation function in its saturation area. It means that a reenforced order of error function will increase the width of

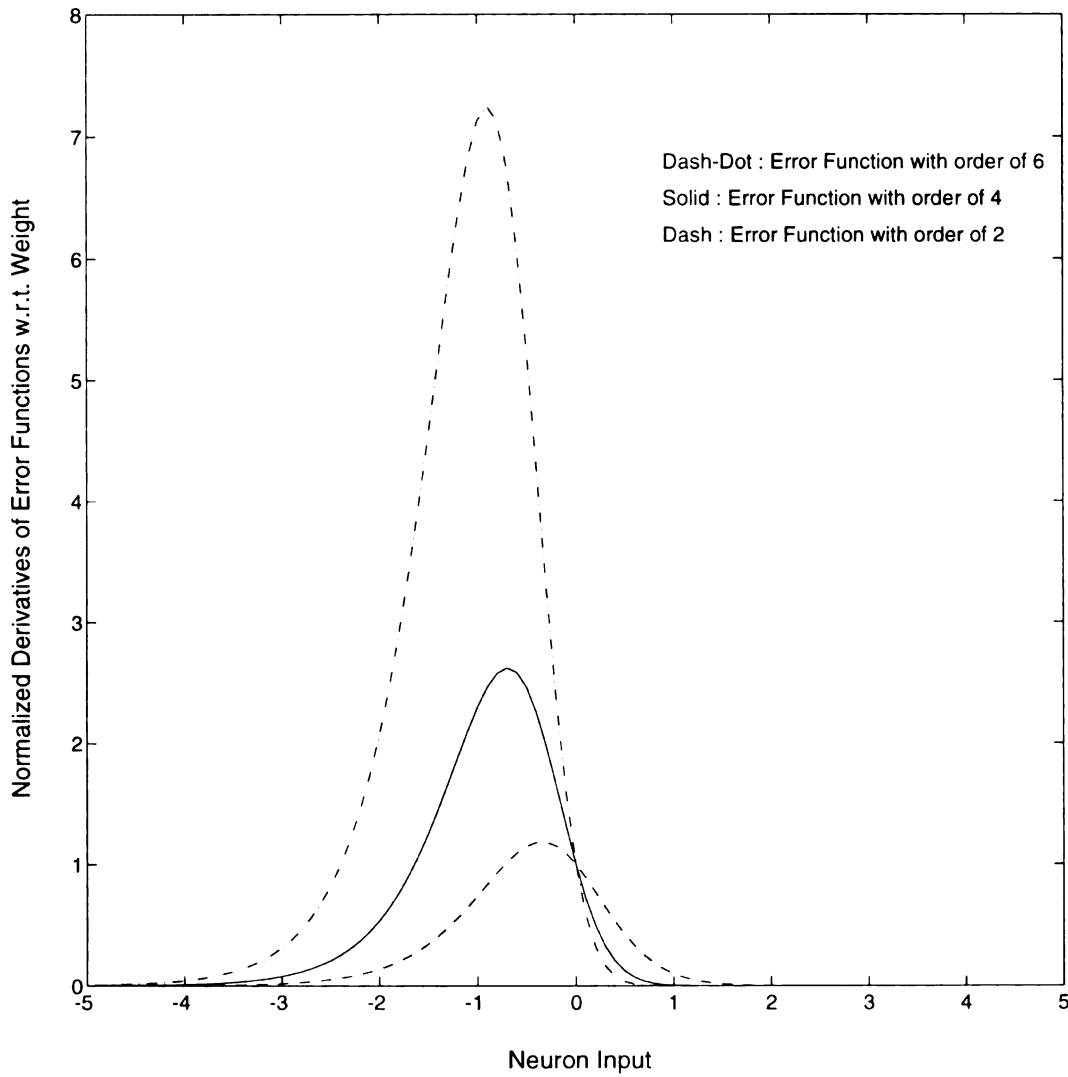


Figure 3.12 Normalized derivatives of error functions of three different orders w.r.t. weight W_{jk} , $-dE(W)/dW_{jk}$. The desired output is assumed to be 1 and an activation parameter $\beta=2$ is used for output neurons. The dash-dot line denotes the error function with order of 6, the solid line denotes the one with order of 4, and the dash line shows the one with order of 2.

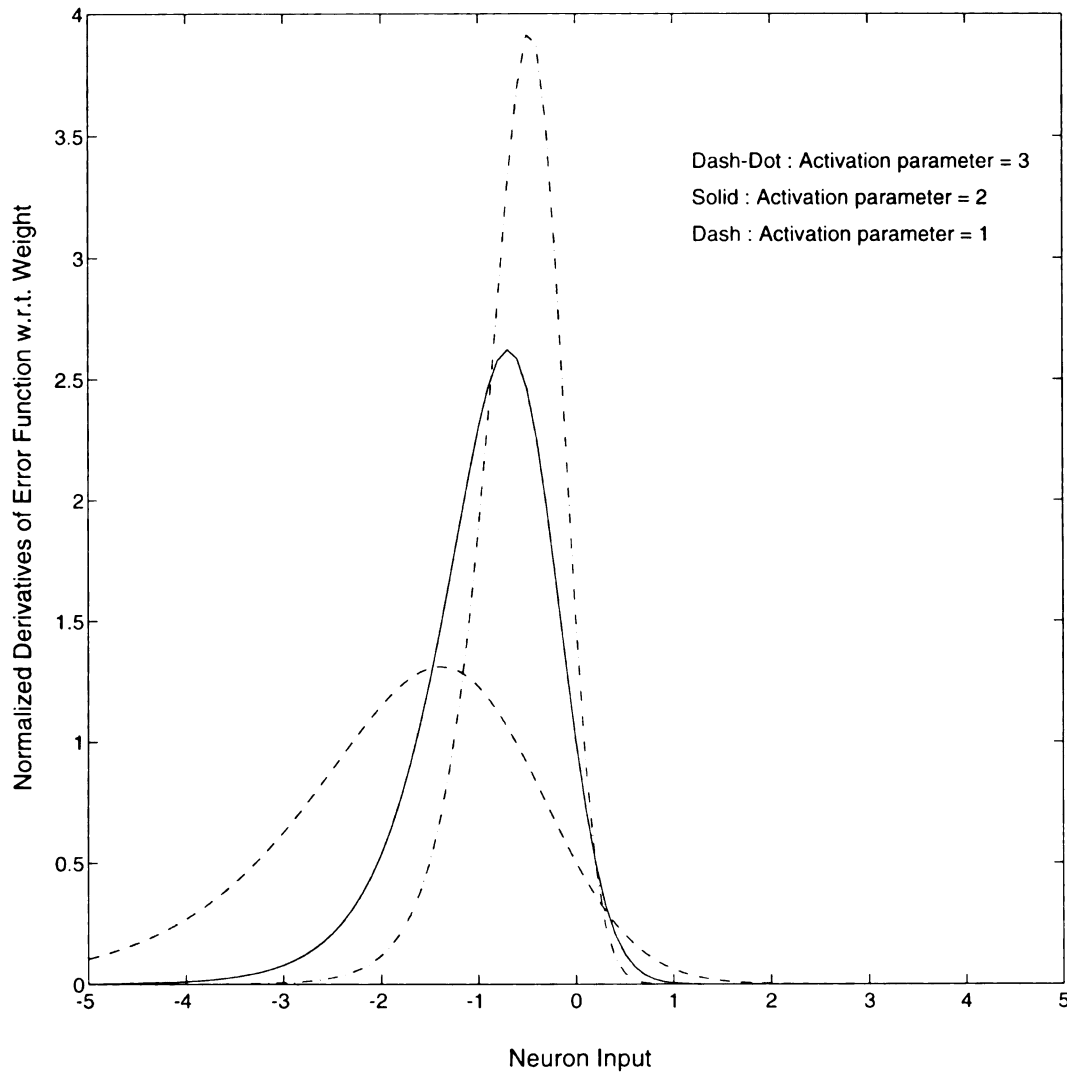


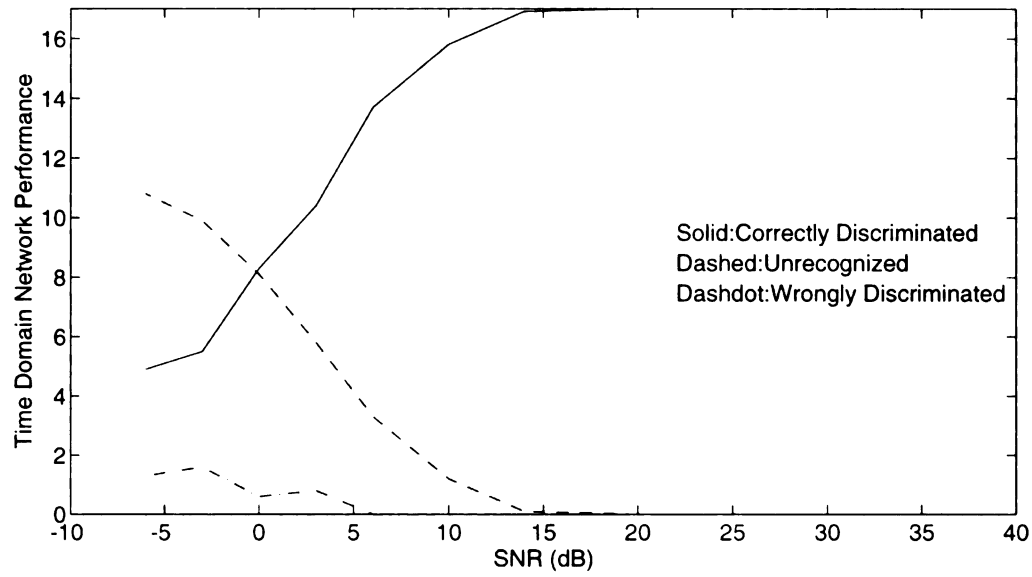
Figure 3.13 Normalized derivatives of error function w.r.t. weight W_{jk} , $-dE(W)/dW_{jk}$, with three different activation parameters. The desired output is assumed to be 1 and the error function with order of 4 is used here. The dash-dot line denotes the activation function with parameter $\beta=3$, the solid line denotes the one with $\beta=2$, and the dash line shows the one with $\beta=1$.

learning region, and then make the idle ones with derivative order of one possible to be pulled back from saturation area of the wrong side.

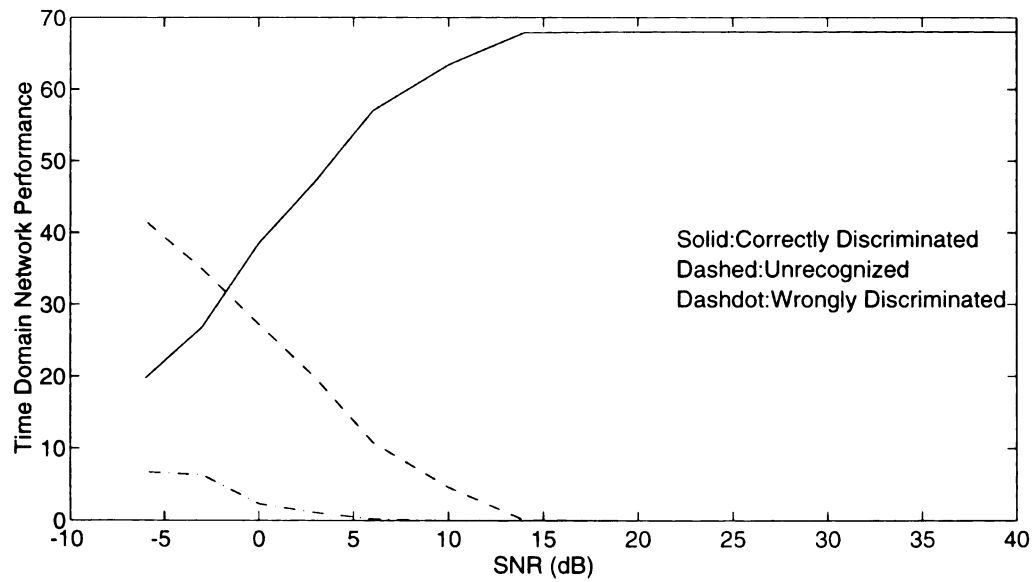
Suppose the desired output corresponding to the training pattern p is 1 without losing generalization, then **Figure 3.12** shows normalized derivatives of error functions of three different reinforced orders with respect to the learning weight W_{jk} , $-dE(W)/dW_{jk}$. The neuron activation parameter is assumed 2 for those three cases. The dash-dot line denotes the one with a reinforcement order of 6, the solid line denotes the one with order of 4, and the dash line shows the one with order of 2. **Figure 3.13** presents normalized derivatives of error functions with respect to learning weight W_{jk} , $-dE(W)/dW_{jk}$, with three different activation parameters. The reinforcement order of error function is assumed 4 for those three cases. The dash-dot line denotes the one with activation parameter $\beta=3$, the solid line denotes the one with $\beta=2$, and the dash line shows the one with $\beta=1$. Both figures are normalized by the respective reinforcement orders of error functions and network input u_n .

Two interesting characteristics observed by comparing **Figure 3.12** and **Figure 3.13** are the order reinforcement of error function increases the width of learning area and shifts it toward deeper saturation area, while the enlargement of activation parameter shrinks the learning area and shifts it toward the transition region.

The network trainings take 624 epochs to converge with a hard limiter, and then another 414 extra training epochs are followed. Without the order reinforcement of error function, the training can't escape from deep saturation area and is reinitiated a couple of times. By monitoring network trainings, being trapped or idle phenomena usually occurs with inputs to neuron much far away from the transition region. For a training pattern with desired



(a)



(b)

Figure 3.14 Performance of ML/BP network with one hidden layer vs. SNR(dB).

(a). Network performance for the 17 training patterns of target B58.

(b). Overall performance for all 68 training patterns.

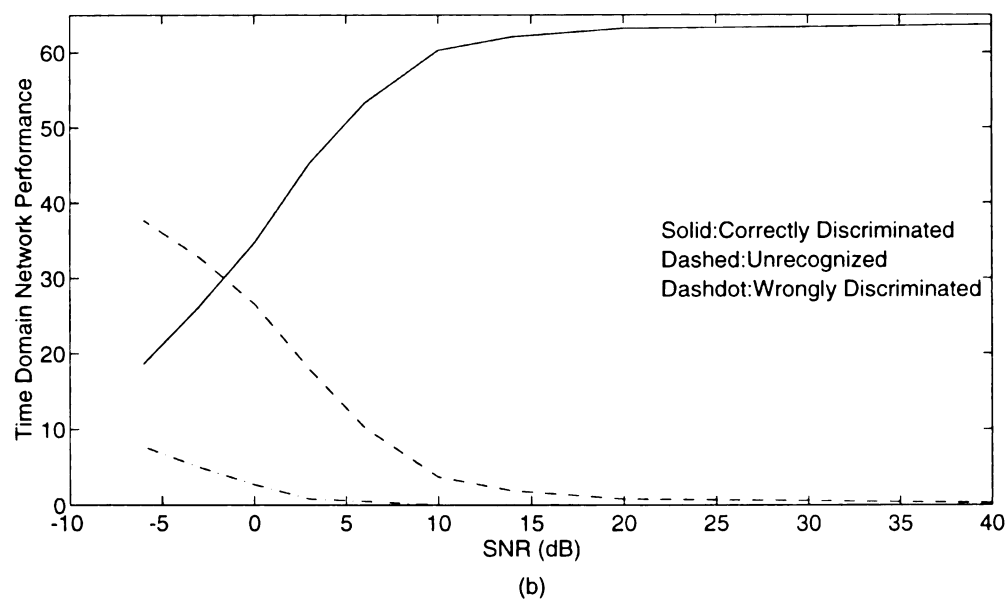
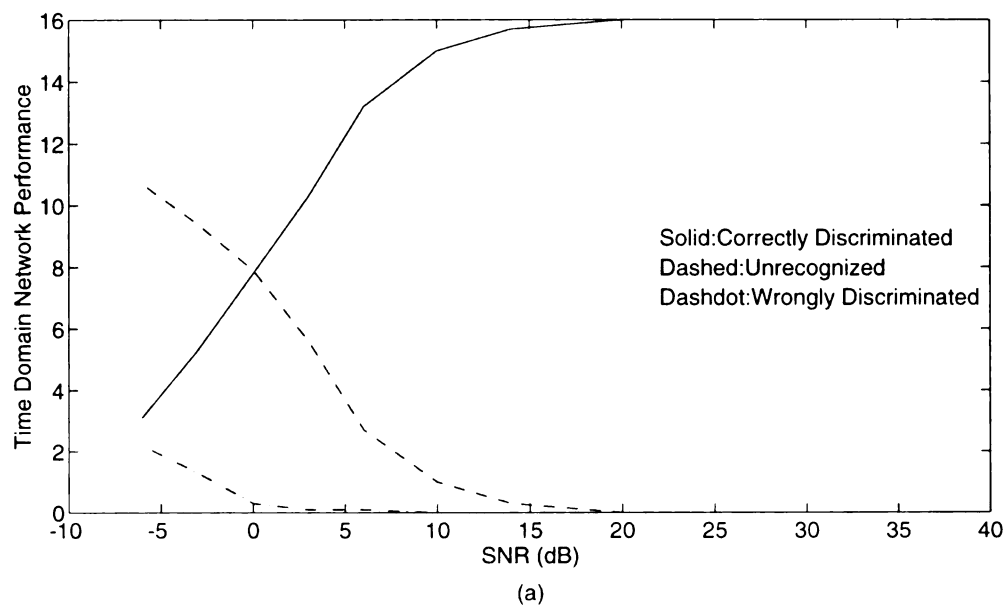


Figure 3.15 The generalization performance of ML/BP network with one hidden layer vs. SNR(dB).

- (a). Network generalization performance for the 16 untrained patterns of target B58.
 (b). Overall generalization performance for the 64 untrained patterns.

output value of 1, an finally nonconvergent input value of -5 to an output neuron with activation parameter $\beta=2$, i.e. $G_2(-5)=-0.9999$ and $G_2'(x)|_{x=-5} \approx 1.82 \cdot 10^{-4}$, is a common idle case. In our simulations, the reenforced order of error function did help pulling those severely wrong ones back.

Figure 3.14 (a) shows the performance of the ML/BP network with one hidden layer for the 17 training patterns of target B58, while **Figure 3.14** (b) presents the overall performance for all 68 training patterns. **Figure 3.15** (a) shows the generalization performance of ML/BP network with one hidden layer for the 16 untrained patterns of target B58, while **Figure 3.15** (b) presents the overall generalization performance for the 64 untrained patterns of four targets. From simulation results, the ML/BP with one hidden layer doesn't perform better than the one without hidden layer. The ambiguous desired outputs of the hidden layer degrade the capability of hidden layers, while being linearly separable makes the ML/BP with hidden layers have no apparent advantage over the one without hidden layers.

3.5 Generalized Inverse Network

The radar target discrimination may be represented as a set of associative equations, and then a matrix equation. The number of response samples for each aspect pattern is the variable number for the associative equation set and the total available aspect patterns is the number of associative equations. However, there are two characteristics which may hinder the conventional matrix operation. One is that the aspect response doesn't apparently change along aspect increment in some specific aspect range. The other is that the responses of the same aspect belonging to different targets may be quite close. This will make direct matrix

approach difficult to deal with singularity, especially for quantization data. An exact numerical solution, which has no generalization capability, is not what we expect, since it won't work under contamination. Neural network learning is good in releasing the singularity stress, and offers tolerance or/and generalization. Then it suggests us to build a hybrid network which is initially constructed by the associative equation set and then learns to converge to a solution.

Assume we have p aspect patterns to be stored in memory, $\mathbf{X} = [\mathbf{X}^1 \mathbf{X}^2 \dots \mathbf{X}^p]$ where $\mathbf{X}^i$ is an m -dimension column vector, i.e. $\mathbf{X}^i = [\mathbf{X}_1^i \mathbf{X}_2^i \dots \mathbf{X}_m^i]^T$. Suppose $\mathbf{Y} = [\mathbf{Y}^1 \mathbf{Y}^2 \dots \mathbf{Y}^p]$ are the associative code patterns corresponding to $\mathbf{X}$ where $\mathbf{Y}^i$ is an n -dimension column vector, i.e. $\mathbf{Y}^i = [\mathbf{Y}_1^i \mathbf{Y}_2^i \dots \mathbf{Y}_n^i]^T$. Then we can write an equation representing the above association as

$$\mathbf{W}\mathbf{X} = \mathbf{Y} \quad (22)$$

where $\mathbf{W}$ is an n by m matrix. For our application, $\mathbf{Y}^i$ is the target associated to the stored aspect pattern $\mathbf{X}^i$. Typically, $\mathbf{X}$ is not a square matrix and $\mathbf{m} > \mathbf{p}$ is assumed. Thus a direct Generalized Inverse matrix computation can be used to solve the interconnection weight matrix $\mathbf{W}$ [20][31][32] as

$$\mathbf{W} = \mathbf{Y}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T = \mathbf{Y}\mathbf{X}^+ \quad (23)$$

where $\mathbf{X}^+$ is the generalized (or pseudo) inverse of $\mathbf{X}$. The generalized inverse $\mathbf{X}^+$ exists only if $\mathbf{m} > \mathbf{p}$, but the direct computation of the generalized inverse becomes difficult or impractical if the dimension of $\mathbf{m}$ or $\mathbf{p}$ is too large. In addition, problems may occur when two adjacent aspect angle target responses are very close. This is not unusual for target aspect responses.

After we quantize them into binomial code form used in our network process, they may have same quantization sequence codes causing the generalized inverse computation to become instable or singular. Therefore, instead of direct computation of the generalized inverse, we can use iterative trainings based on the gradient descent algorithm to gradually approach and then approximate a solution of the interconnection matrix $\mathbf{W}$. We iteratively train the network weight $\mathbf{W}$ and expect the network output to each trained pattern $\mathbf{X}^i$, $\mathbf{W}\mathbf{X}^i$, will retrieve its associative pattern $\mathbf{Y}^i$.

First, we can construct an error (or cost) function $\mathbf{J}(\mathbf{W})$ for the network training as

$$J(\mathbf{W}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{W}\mathbf{X}\|^2 = \frac{1}{2} \sum_{i=1}^P \sum_{j=1}^n [Y_j^i - (\mathbf{W}\mathbf{X}^i)_j]^2 \quad (24)$$

where $\|\cdot\|$ denotes the Euclidian L_2 norm. To minimize the error function $\mathbf{J}(\mathbf{W})$, the gradient descent learning rule [18][31][32] can be used

$$\mathbf{W}[k+1] = \mathbf{W}[k] - \eta \frac{\partial J(\mathbf{W}[k])}{\partial \mathbf{W}[k]} = \mathbf{W}[k] + \eta (\mathbf{Y} - \mathbf{W}[k]\mathbf{X})\mathbf{X}^T \quad (25)$$

where $\mathbf{W}[\mathbf{k}]$ is network weight matrix at learning epoch $\mathbf{k}$, and η is the "learning rate" with $0 < \eta < 1$. Since

$$\eta (\mathbf{Y} - \mathbf{W}[k]\mathbf{X})\mathbf{X}^T = \sum_{i=1}^P \eta (\mathbf{Y}^i - \mathbf{W}[k]\mathbf{X}^i)\mathbf{X}^{iT} \quad (26)$$

, above training algorithm is a batch mode learning by which each learning pattern adjusts the interconnection weights $\mathbf{W}$ without considering the adjustments done by the other learning patterns at the same learning epoch. This batch mode learning may cause over adjustments

(or learnings) problems, since the gradient descent algorithm uses the same network weight to calculate respective errors for all training patterns and then adjust the network weight at the same time. But the basis of gradient descent rule can effectively reduce the error only for the present training pattern with current weight and doesn't promise that the current weight adjustment will also benefit the other training patterns.

The learning algorithm can be modified to asynchronously update for each training pattern, $\mathbf{X}^i$, and its associative pattern components, $\mathbf{Y}_j^i$. Therefore, the current training pattern adjusts the weight $\mathbf{W}$ updated by the previous learning pattern. The asynchronous update rule gives

$$\mathbf{W}_j[k+1] = \mathbf{W}_j[k] + \eta (Y_j^{(i)} - \mathbf{W}_j[k] X^{(i)}) X^{(i)T} \quad (27)$$

where $\mathbf{W}_j[\mathbf{k}]$ denotes the j^{th} row of weight matrix $\mathbf{W}$ at update iteration $\mathbf{k}$ and $\mathbf{Y}_j^{(i)}$ is the j^{th} component of the desired pattern $\mathbf{Y}^i$ associated with $\mathbf{X}^i$. To train each pattern and component fairly, and avoid being trapped in a local minimum, the training pattern i is randomly selected.

In our application to radar target discrimination, the output is expected to clearly present the discriminated target type. Therefore, continuous value form is not appropriate for output representation. The binomial output form also increase the noise tolerance, therefore a nonlinear threshold (activation) function is introduced to the network output stage. Then we have

$$G_{\beta}(WX) = Y \quad (28)$$

and

$$G(v)_{\beta}' = \frac{\beta}{2} (1 + G_{\beta}(v))(1 - G_{\beta}(v)) \quad (29)$$

from equation (4). The asynchronous update rule for pattern $\mathbf{X}^i$ and its associative pattern component $\mathbf{Y}_j^i$ is then given by

$$W_j[k+1] = W_j[k] + \eta G(v)_{\beta|W_j[k]X^{(i)}} (Y_j^{(i)} - W_j[k]X^{(i)})X^{(i)T} \quad (30)$$

where $W_j[k]X^{(i)}$ is the input to the output neuron j and $(Y_j^{(i)} - W_j[k]X^{(i)})$ is the error occurred on the output of neuron j for the current training pattern $X^{(i)}$.

The equation (28) has P associative equations and each has m variables. The weight matrix $\mathbf{W}$ maps P m -dimension vectors to P n -dimension vectors. Therefore, for $m > P$, there are multiple exact solutions. If weight matrix $\mathbf{W}$ is randomly initialized, then the $\mathbf{W}$ may converge to some specific solution that has high noise tolerance for some trained patterns, but low noise tolerance for others. To avoid the solutions being biased by some learning patterns and also to speed up trainings, we may initiate the weight matrix by $\mathbf{W} = \mathbf{YX}^T$. That is, we use the correlation recording matrix or Hopfield memory with nonzero diagonals. And this initial way has great possibility to approximate to the generalized inverse solution $\mathbf{YX}^+$ if $\mathbf{X}^+$ exists. Therefore, the **GI** network will become a multi-layer feedforward with error backpropagation network without a hidden layer, if the network weight $\mathbf{W}$ is not initialized by the correlation recording matrix $\mathbf{YX}^T$. So the **GI** network can be referred as a hybrid network composed of correlation associative memory and learning based backpropagation network. And then we may also say the **GI** network is a single layer net with initial weight matrix $\mathbf{W} = \mathbf{YX}^T$.

3.5.1 Generalized Inverse Network Performance Using Bipolar Data

Since the **GI** network will be cascaded to a Recurrent Correlation Associative

Memory(RCAM) to serve as a target code decoder and a refiner of spurious stable states, bipolar data are adopted for this network in both input and output. The advantages from using binomial form will be discussed in the next chapter.

Similar to ML/BP nets, the **GI** network has 17 training azimuthal aspect patterns for each target, while the other 16 aspect patterns of every target are left untrained to test network generalization after training. Thus, we have total of 68 training aspect patterns and 64 untrained patterns for four targets. From the previous chapter, we know that each time domain aspect pattern contains 100 analog samples. First, we quantize each analog sample by 7 quantization levels for all patterns, then use 3 bits to encode the 7 quantization levels. The 3-bit code assignment for 7 quantization levels is shown below in Table 3.1.

Bit1Bit2 Bit3	-1 -1	-1 1	1 1	1 -1
-1	1	6	5	4
1	2	7		3

Table 3.1 Code assignment for 7 quantization levels coded by three bipolar bits.

The code resulted from 7 levels coded by 3 bits is not exactly a linear code, with which two nearer levels have two similar codes. But if we properly assign the code-to-level

mapping as shown in Table 3.1, the codes have its linear characteristic for the adjacent levels. It means that if the noise contamination range is smaller than ± 1.5 level intervals, then the code linearity will still statistically functions. Therefore the network discrimination won't be affected by the coding scheme. If the noise contamination is severe, then the linearity may not work for every response level. Under the severe noise condition, some wrong or unrecognized discriminations probably result from this nonlinear coding scheme.

For example, if the true response is in level 3 (represented by 1 -1 1), and the contaminated range is in ± 1.5 level intervals, then the contaminated signal will be in either level 2 (represented by -1 -1 1) or level 4 (represented by 1 -1 -1), and these two codes (-1 -1 1 and 1 -1 -1) are still the most similar codes to the true one (1 -1 1) than the others. If the number of quantization levels is fixed, then the input resolution will be the same for different coding schemes. The more digits we use, the more linearity the code mapping has, but more process space is required. Therefore it becomes a trade-off problem, either save process space with less code bits, then process quickly but poorly under severe noises; or expend more memory with more code bits, then process slowly but satisfactorily under severe contamination. This problem can be solved by processing the analog (numerical) valued responses instead of binomial codes for digital computer simulation. But the problem will still remains the same for hardware realization.

The training for **GI** network only takes several epochs to converge to desired target codes with hard limiter and then 120 extra training epochs are followed. **Figure 3.16 (a)** shows the **GI** network performance for the 17 training patterns of target B58, while **Figure 3.16 (b)** presents the network overall performance for all 68 training patterns. **Figure 3.17**

(a) shows the **GI** network generalization performance for the 16 untrained patterns of target B58, while **Figure 3.17** (b) presents the network overall generalization performance for the 64 untrained patterns.

As expected, the **GI** network shares an interesting phenomenon with the analog process ML/BP networks. That is the **GI** network also prefers unknown to wrong discriminations. This interesting characteristic allows us to put more confidence in the target type discriminated by **GI** network. Since the wrong discrimination percentage is quite low, it is reasonable to think that the **GI** network cascading to an Recurrent Correlation Associative Memory (RCAM) will only increase the correct discrimination rate and do no harm to those correctly associated outputs presented by the RCAM. Therefore the **GI** network can serve as a refiner of spurious stable states resulted from the cascaded RCAM. This issue will be appreciated in the following chapters. The **GI** has better performances than the analog ML/BP nets under fair noise conditions, but performs worse under severe contamination. The nonlinear interruptions described earlier may be used to explain that deficiency. Since the ML/BP nets use continuous values for network inputs, the linearity is always kept even under serious contamination.

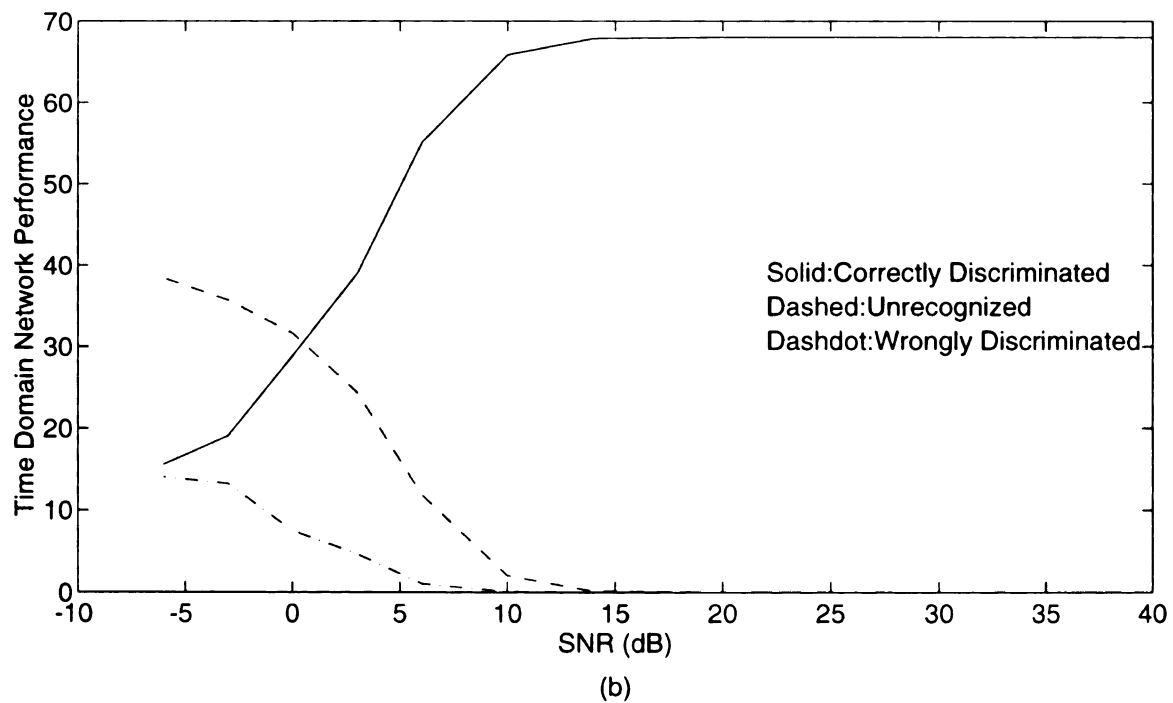
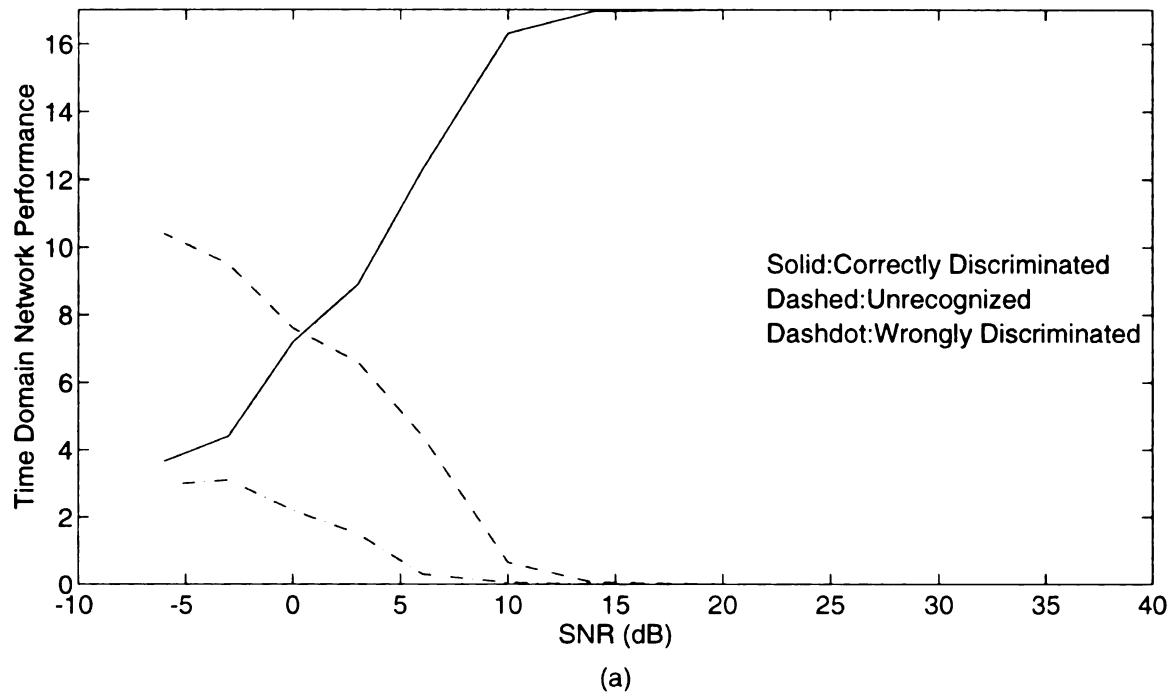


Figure 3.16 Generalized Inverse (GI) network performance vs. SNR(dB).

(a). Network performance for the 17 training patterns of target B58.

(b). Overall performance for all 68 training patterns.

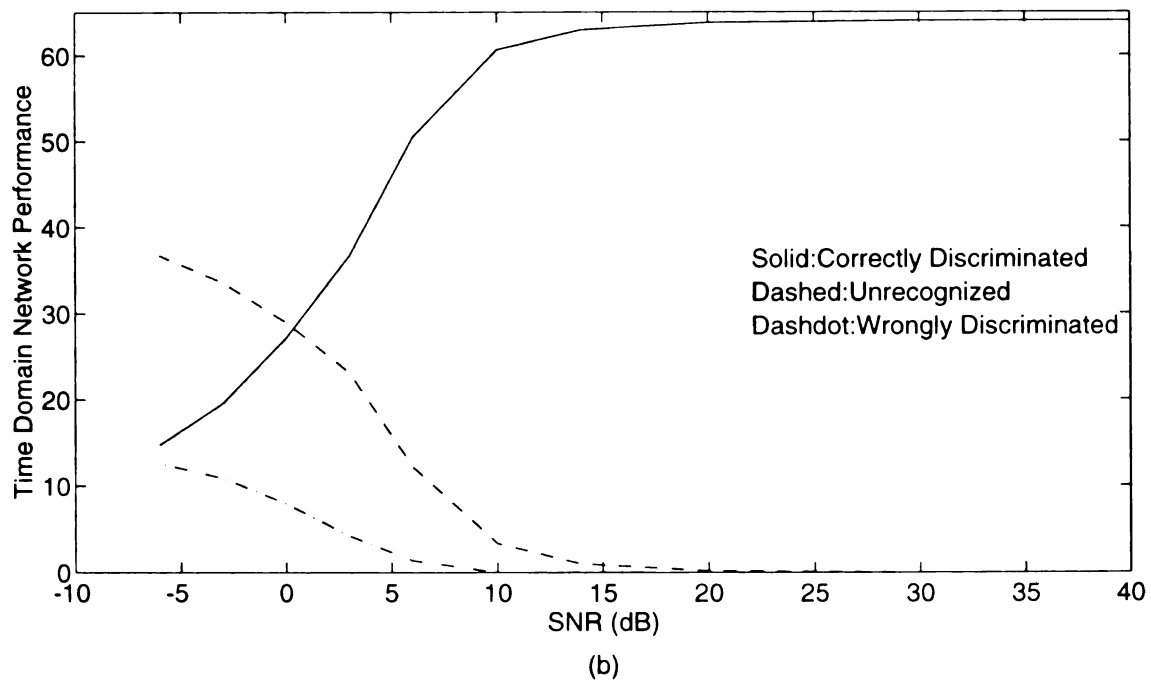
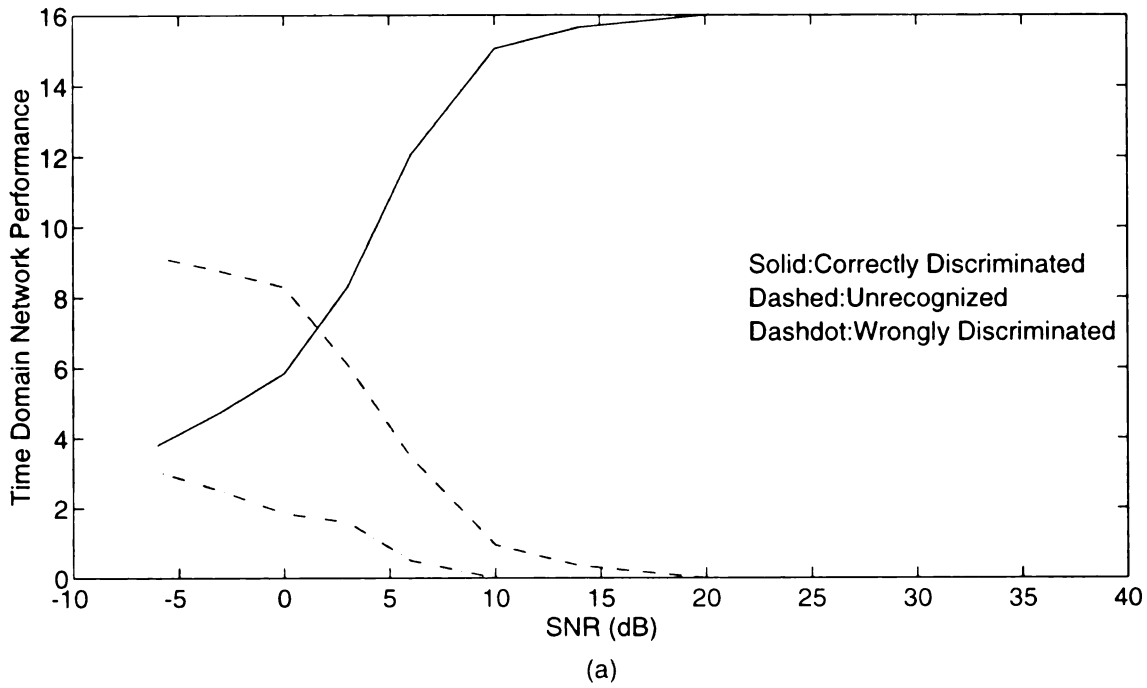


Figure 3.17 The generalization performance of Generalized Inverse (GI) network vs. SNR(dB).

- (a). Network generalization performance for the 16 untrained patterns of target B58.
 (b). Overall generalization performance for the 64 untrained patterns.

CHAPTER 4

Recurrent Correlation Associative Memories

4.1 Introduction

In this chapter we will investigate some Recurrent Correlation Associative Memories (RCAM) [18][31]~[33][38]~[41]. These networks are designed to retrieve the stored information from incomplete or degraded data, and they are "recursive" systems because the output units are fed back to the inputs at each update. The recurrent correlation associative memory is a correlation-based stored data reconstruction memory. The RCAM retrieves stored data from data themselves unlike the traditional address-based memories which access stored data by their corresponding addresses. Therefore, the RCAM not only saves the resources used to calculate the stored data address by address-based memories, but also directly process data in parallel. They can be used as the efficient content-addressable memories [18] and then pattern discriminator. In this application, all stored patterns are explored in parallel because the data storage and retrieval are based on the pattern itself and not on its address.

The associative memories [31]~[33] can be classified in various ways depending on their nature of the stored associations (autoassociative vs. heteroassociative) and the update mode (synchronous vs. asynchronous). The autoassociative memory is a discrete memory in which the associative pattern of each stored pattern is the pattern itself. Therefore, the output has the same dimension as the input. It employs a single layer of perceptrons and has hard-limiter activations at the output stage. The perceptrons are fully interconnected and the

outputs are directly fed back to the inputs with one iteration delay. And the update can operate in a synchronous (parallel) or asynchronous (random) mode. A heteroassociative memory can be regarded as an extension of the autoassociative memory. There are two pattern sets stored in a heteroassociative memory and each pattern in one set has its associative partner, no more itself, in the other set. Therefore, the heteroassociative memory may have its output dimension different than that of its input. They also can be thought of two separate single-layer feedforward networks with each one's output connected to the other one's input.

4.2 Hopfield Network and Bidirectional Associative Memory (BAM)

The learning mechanisms occurred in correlation-based associative memories can be widely explained by the Hebb's rule [34]. The Hebb's rule suggests when a neuron i repeatedly participates in activating another neuron j , then the efficiency of neuron i in activating neuron j is increased and the efficiency should be decreased in the opposite case. Or, in the view of synaptic connection weight, it can be quantitatively expressed by that the synaptic weight w_{ij} should be enhanced if neurons i and j much often have the same activity state, and should be inhibited if neurons i and j much often have the opposite activity state. Therefore, the Hebb's rule proposed the meaning of correlation and synaptic strength between two connected neurons.

Suppose neurons have binomial states, 0 or 1 for binary form and -1 or 1 for bipolar form. For a given input, if the neuron i has state S_i and the neuron j has state S_j , then the Hebb's rule indicates that the synaptic weight should be adjusted to respond to the training input by $\Delta w_{ij} = \Delta w_{ji} = r S_i S_j$ where r is a positive learning rate. Assuming that we want p

associative pattern pairs $\{(\mathbf{x}^q, \mathbf{y}^q) | q=1, 2, \dots, P\}$ stored in memory, where $\mathbf{x}^q$ is an m -dimension column vector, and $\mathbf{y}^q$ is an n -dimension column vector associated to $\mathbf{x}^q$, so that $\mathbf{X}=[\mathbf{x}^1 \mathbf{x}^2 \dots \mathbf{x}^P]$ and $\mathbf{Y}=[\mathbf{y}^1 \mathbf{y}^2 \dots \mathbf{y}^P]$. The associative memory is an autoassociative memory if $\mathbf{x}^q=\mathbf{y}^q$ for all P patterns, otherwise it is a heteroassociative memory.

The Hopfield [35][36] network is an autoassociative memory, so the recorded associative patterns have $\mathbf{x}^q=\mathbf{y}^q$ for all P patterns. Suppose the network has binary data form and w_{ij} denotes the interconnection weight between input neuron i and output neuron j , then a Hopfield memory can be constructed by

$$w_{ij} = \begin{cases} \sum_{q=1}^P (2x_i^q - 1)(2x_j^q - 1) & , i \neq j \\ 0 & , i = j \end{cases} \quad (1)$$

Except w_{ii} , the synaptic weight between neurons i and j is increased if neurons i and j have the same binary state, and is decreased if two neurons have opposite active state. Therefore, the Hopfield memory recording algorithm is in accordance with the Hebbian rule. Since it is autoassociative, the interconnection weight has $w_{ij}=w_{ji}$. In binary form, the relationship between one state, b , and its complementary state, b' , can be written by $b'=(1-b)$ or $b=(1-b')$. Then we have $[2(1-x_i)-1][2(1-x_j)-1]=(2x_i-1)(2x_j-1)$, so the above Hopfield recording scheme also stores the complementary partners of the desired stored patterns. For binary form, given an input state $s[k]$, the Hopfield network will have the next state $s[k+1]$, by

$$s_j[k+1] = \begin{cases} 1 & , \sum_{i=1}^m w_{ij} s_i[k] > 0 \\ 0 & , \sum_{i=1}^m w_{ij} s_i[k] \leq 0 \end{cases} \quad (2)$$

In equation (1), the auto-feedback synaptic weight for each neuron is set to 0 and then the weight matrix will have 0 diagonals. This setting will prevent the network from blindly amplifying the j state of an arbitrary input by the number of stored patterns, when the network intends to determine output activity for neuron j . Without the diagonal annulling, a Hopfield network with a large number of stored patterns will become insensitive to variable inputs and is inclined to leave an arbitrary input unchanged. Setting the diagonals to 0 can be physically expressed as that the neuron itself is not permitted to get involved in determining its next activity state. The storage capacity of the Hopfield memory has been empirically found to scale about $0.15n$ [35] and theoretically to scale less than $n/(2\log n)$ [30][37]. The latter suggests that for large n , the ratio of stored patterns to pattern dimension P/n required for correct recall approaches zero, thus it reflects the poor capacity for the Hopfield memory. Since the activation threshold is nonlinear, the Hopfield net is a nonlinear dynamic network. A nonlinear system is generally difficult to analyze its characteristics. The Russian mathematician Liapunov devised a stability test based on energylike functions. For a network W and a state s , the energy function is defined as

$$E(s) = -\frac{1}{2} s^T W s \quad (3)$$

or rewritten as

$$\begin{aligned} E(s) &= -\frac{1}{2} \sum_{j=1}^m s_j \sum_{i=1}^m w_{ij} s_i \\ &= -\frac{1}{2} \sum_{j=1}^m \sum_{i=1}^m w_{ij} s_i s_j \end{aligned} \quad (4)$$

Now we would like to know how the energy change from state to state by synchronous update, that is [20]

$$\frac{\partial E(s)}{\partial s} = -W_s \quad (5)$$

or by asynchronous update

$$\frac{\Delta E(s)}{\Delta s_j} = -\sum_{i=1}^m w_{ij}s_i \quad (6)$$

From equation (6), the network has $\sum_{i=1}^m w_{ij}s_i > 0$ and $\Delta s = 1 > 0$ if s_i changes from 0 to

1, and it has $\sum_{i=1}^m w_{ij}s_i \leq 0$ and $\Delta s = -1 < 0$ if s_i changes from 1 to 0. Therefore, the energy change

$\Delta E(s)$ can't be positive whenever a neuron changes its state. Since the weight matrix W is deterministic for a finite number of stored patterns and stored patterns X is bounded in dimension of m and binary state $[0, 1]$, the energy E is bounded. Therefore, the network W should converge a given state to a local minimum of E . But there may be more than one local minimum or stable state with the same amount in energy, i.e. energy may be kept the same when a state changes. Then the network may have oscillation cycle among the same energy states. Let's consider a simple example here. Suppose we need the network to store a binary pattern $x = (x_1 \ x_2 \ x_3 \ x_4)^T = (1 \ 0 \ 1 \ 0)^T$, then, according the equation (1), the network weight matrix has $w_{ij} = (2x_i - 1)(2x_j - 1)$ for $i \neq j$ and $w_{ij} = 0$ for $i = j$. And we know the network interconnection weights are symmetrical, so only 6 weights are required to be calculated.

They are $w_{12}=(2x_1-1)(2x_2-1)=(2-1)(0-1)=-1=w_{21}$, $w_{13}=1=w_{31}$, $w_{14}=-1=w_{41}$, $w_{23}=-1=w_{32}$, $w_{24}=1=w_{42}$ and $w_{34}=-1=w_{43}$.

Now the Hopfield network has

$$W_{HF} = \begin{bmatrix} 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ -1 & 1 & -1 & 0 \end{bmatrix} \quad (7)$$

Let's see if the stored state is stable. Assume the next state is s , then we have $s=g(Wx)$ where $g()$ is the binary activation function and the Wx is given by

$$Wx = \begin{bmatrix} 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ -1 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 1 \\ -2 \end{bmatrix} \quad (8)$$

Then the network has next state $s=g[(1 \ -2 \ 1 \ -2)^T]=(1 \ 0 \ 1 \ 0)^T=x$, thus the stored state x is a stable state. Let's examine the state changing trace and its corresponding energy change by considering two initial states $s^1=(1 \ 0 \ 0 \ 1)^T$ and $s^2=(0 \ 0 \ 0 \ 1)^T$. For the initial state s^1 , the network has the next state $s^1(1)=g(Ws^1)=g[(-1 \ 0 \ 0 \ -1)^T]=(0 \ 0 \ 0 \ 0)^T$, and then the next state $s^1(2)=g[Ws^1(1)]=g[(0 \ 0 \ 0 \ 0)^T]=(0 \ 0 \ 0 \ 0)^T=s^1(1)$. The final stable state is $(0 \ 0 \ 0 \ 0)^T$ which is not a desired stored state. From equation (3), the energy for the only stored state $x=(1 \ 0 \ 1 \ 0)$ is $E(x)=-0.5x^TWx=-1$. The initial state s^1 has energy $E(s^1)=-0.5(s^1)^TWs^1=1$, and then the stable state $s^1(1)=s^1(2)$ has energy $E[s^1(1)]=0$. In this example the initial state truly converges to a lower energy stable state. Now for the initial state s^2 , the network has the next state

$s^2(1)=g(Ws^2)=g[(-1 \ 1 \ -1 \ 0)^T]=(0 \ 1 \ 0 \ 0)^T$, and then the next state $s^2(2)=g[Ws^2(1)]=g[(-1 \ 0 \ -1 \ 1)^T]=(0 \ 0 \ 0 \ 1)^T=s^2$. Therefore, an oscillation cycle, $s^2 \rightarrow s^2(1) \rightarrow s^2 \rightarrow s^2(1) \dots$, occurs. The state s^2 has energy $E(s^2)=0$ and the state $s^2(1)$ also has the same energy $E[s^2(1)]=0$, therefore the states oscillate between these two local minima, which share the same energy.

Another typical associative network we use to compare with Recurrent Correlation Accumulation Adaptive Memories (RCAAM), discussed in next chapter, is Bidirectional Associative Memory (BAM) [38][39]. BAM is a heteroassociative memory, i.e. $x^q \star y^q$, and the retrieval operations are sequentially proceeded in alternative directions until two associative stable states are reached at two sides of the network. Suppose the bipolar data form is used, then, with the above assumed associative stored pattern pairs $\{(x^q, y^q) | q=1, 2, \dots, P\}$ and dimensions (m for column vector x^q and n for column vector y^q), a BAM can be constructed as follows :

$$w_{ij} = \sum_{q=1}^P x_i^q y_j^q, \quad i=1, \dots, m \text{ and } j=1, \dots, n \quad (9)$$

or

$$W = \sum_{q=1}^P x^q y^{qT} = XY^T \quad (10)$$

where $(y^q)^T$ is the transpose of y^q , $X=[x^1 \ x^2 \ \dots \ x^P]$ and $Y=[y^1 \ y^2 \ \dots \ y^P]$. Therefore, the BAM has a m by n weight matrix W and then it can have either an m -dimension input or an n -dimension input. Again the BAM weight implementation algorithm is a Hebbian learning rule based correlation matrix. Given an input state $u(0)$ with dimension of m , the network has an

n-dimension next state $v(1)=g[W^T u(0)]$, then an m-dimension next state $u(2)=g[Wv(1)]$, then an n-dimension state $v(3)=g[W^T u(2)]$, then $v(3) \Rightarrow u(4) \Rightarrow v(5) \dots$, until $u(k) \Rightarrow v(k+1) \Rightarrow u(k+2) \Rightarrow u(k)$. Then the $u(k)$ and $v(k+1)$ are the retrieved associative patterns by the BAM corresponding to the given input $u(0)$. Similarly, the input of the BAM can be an n-dimension vector $v(0)$ and the successive updates are the same as above.

The similar energy function for the BAM system of associative state (x,y) is defined by

$$E(x,y) = -x^T W y \quad (11)$$

Then, we have the energy change with respect to state change in asynchronous update,

$$\begin{aligned} \frac{\Delta E}{\Delta x_i} &= - \sum_{j=1}^n W_{ij} y_j \\ \frac{\Delta E}{\Delta y_j} &= - \sum_{i=1}^m x_i W_{ij} \end{aligned} \quad (12)$$

or in synchronous update,

$$\begin{aligned} \Delta E(\Delta x, y) &= -\Delta x^T W y = - \sum_{i=1}^m \Delta x_i W_{ij} y_j \\ \Delta E(x, \Delta y) &= -x^T W \Delta y = - \sum_{j=1}^n x_i W_{ij} \Delta y_j \end{aligned} \quad (13)$$

where W_i is the i th row of W and W_j is the j th column of W . Thus, the energy decreases whenever a state changes either in asynchronous or synchronous update mode. Since, for a finite number of stored states, the BAM weight matrix W and bipolar state $\{-1, 1\}$ are bounded, the energy is bounded. From equations (12) and (13), for an arbitrary given initial state (x,y) , the BAM will converge the initial state to a stable state (x_f, y_f) and thus the BAM

has no oscillation phenomenon, which occurred in Hopfield net.

Let's roughly estimate the BAM's storage capacity. Suppose the input is one of stored patterns with m-dimension x^r , then the next n-dimension output before bipolar activation threshold is

$$\begin{aligned} x^{rT} W &= (x^{rT} x^r) y^r + \sum_{q=1, q \neq r}^P (x^{rT} x^q) y^q \\ &= m y^r + \sum_{q=1, q \neq r}^P (x^{rT} x^q) y^q \end{aligned} \quad (14)$$

Since the y^r is the n-dimension stored pattern associated to the stored pattern x^r , the BAM is expected to retrieve pattern y^r if presented the pattern x^r . From the above equation, the associative pattern y^r is amplified by the dimension of stored pattern set X and the crosscorrelation gains between x^r and x^q with $r \neq q$ can be regarded as the noise amplification gains. If the numbers of stored patterns, P, is greater than the dimension of X, m, then the noise terms may possibly prevail over the expected associative pattern. Similarly, the unreliable retrieval may possibly occur if the presented input is y^r with n-dimension and P is greater than n. Thus the BAM storage capacity should be limited by $P < \min(m, n)$.

4.2.1 Simulations and Results

In Hopfield network and BAM simulations, we use the same four targets as used in the previous chapter, and each target has 17 aspect patterns stored in network and another 16 aspect patterns unstored to test network generalization. The scheme of 3-bit encoding 7 quantization levels is again used here, therefore, each stored pattern in Hopfield net and one stored pattern set of BAM has 300 bits. Here the network presents two performances, pattern

recognition and target discrimination, for stored pattern tests, and only target discrimination for unstored pattern test, i.e. generalization performance. The pattern recognition performance shows how well the network can recognize the contaminated stored patterns, while the target discrimination performance presents how well the network can classify a contaminated pattern into its own target. All networks are simulated 10 times and each stored/unstored pattern is tested under 10 different SNR's, [40, 30, 20, 14, 10, 6, 3, 0, -3, -6] dB, in each simulation, and then the average is used to present the network performance.

The autoassociative Hopfield net always brings inputs to some undefined states and leaves nothing discriminated. This network couldn't do anything for our radar target discrimination application. From the previous storage capacity formula for the Hopfield net $n/(2\log n)$, the Hopfield net can store less than 26 300-bit patterns due to $300/(2\log 300)=26.3$. Compared to the simulation results, we can realize that the radar target scattered responses are far away from random signals, therefore the storage capacity derived from random signals is almost useless for our application. Since the BAM is a heteroassociative memory, the network should have another pattern set, which will act as network output codes, stored to associate with the lab-measured aspect response set. Therefore, we need to design another pattern code set to associate with the one we used. The BAM also has poor performances when another stored pattern set uses the orthogonal codes, i.e. the bit i of pattern i is set to 1 and the other bits are set to -1's. Since the BAM requires bidirectional retrieval, another code set should have distinct codes, i.e. any two codes can not be identical. It means two associative stored pattern sets should have one-to-one mapping and won't allow multiple-to-one mapping. This is a disadvantage, since the one-to-one mapping prohibits the BAM from

clustering or target grouping application. However, our purpose is at least that the network can recognize its own target if given an aspect pattern. With above disadvantage constraint, we intend to partition an output code into two sections, one for target identification purpose and the other for BAM bidirectional operation. We design a set of 7-bit heteroassociative codes corresponding to the 68 300-bit aspect pattern codes. The first two bits of the 7-bit codes are designed as a target group code for four different targets, i.e. $[-1 \ -1]$ for B52, $[-1 \ 1]$ for B58, $[1 \ -1]$ for F14 and $[1 \ 1]$ for TR1, and then the next five bits are encoded to represent 17 azimuthal responses of each target.

Then we alter the BAM process strategy by using target group code in its heteroassociative partners, and we only discriminate the target group code portion of the final stable state and ignore the rest of output code. This effort has greatly improved its correct recognition rate and also reduced the wrong rates in simulations. **Figure 4.1** (a) shows the BAM pattern recognition performance for 68 contaminated stored patterns, while **Figure 4.1** (b) presents the BAM target discrimination performance. The pattern recognition uses all 7 bits to discriminate stored patterns, and it has worse performance. The target discrimination only uses the target group codes, first two bits, to discriminate a given pattern, so the BAM will correctly discriminate an input pattern or wrongly discriminate and leaves none unknown. Compared to correct pattern recognition rate, the correct rate of target discrimination raises about 60% at 40 dB SNR and 55% at 0 dB SNR. **Figure 4.2** shows the BAM generalization performance by testing 64 contaminated unstored patterns. The correct target discrimination rate of BAM generalization performance is 83% at 40 dB.

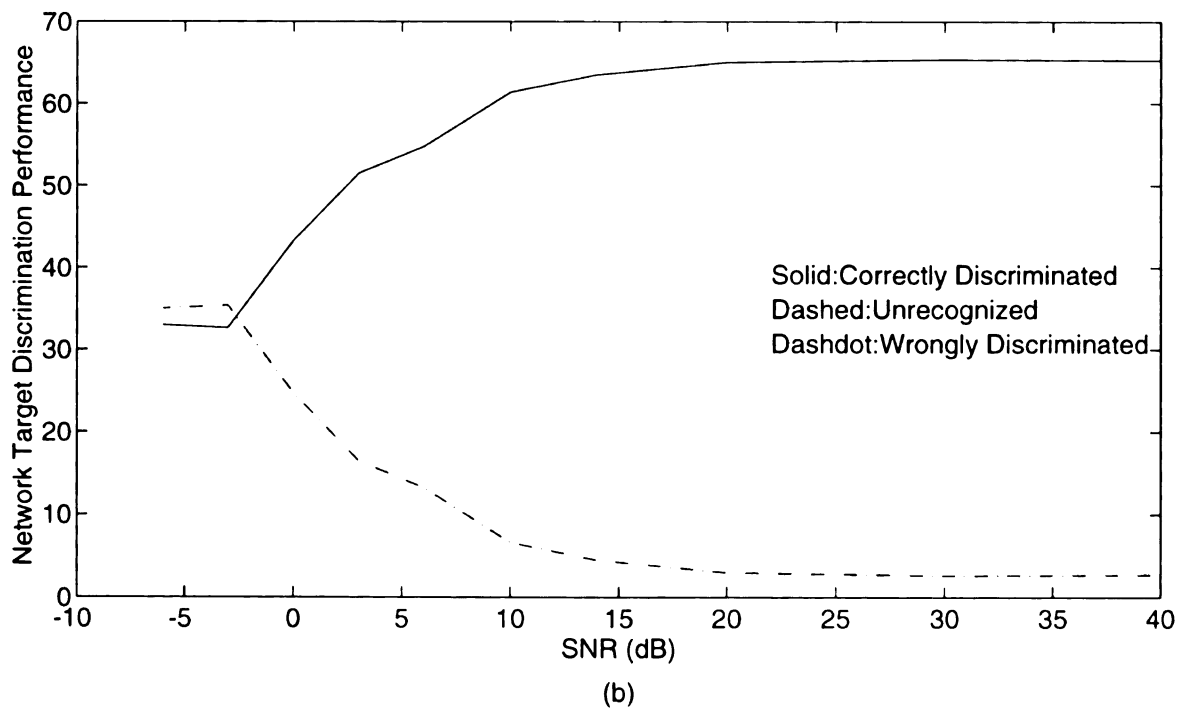
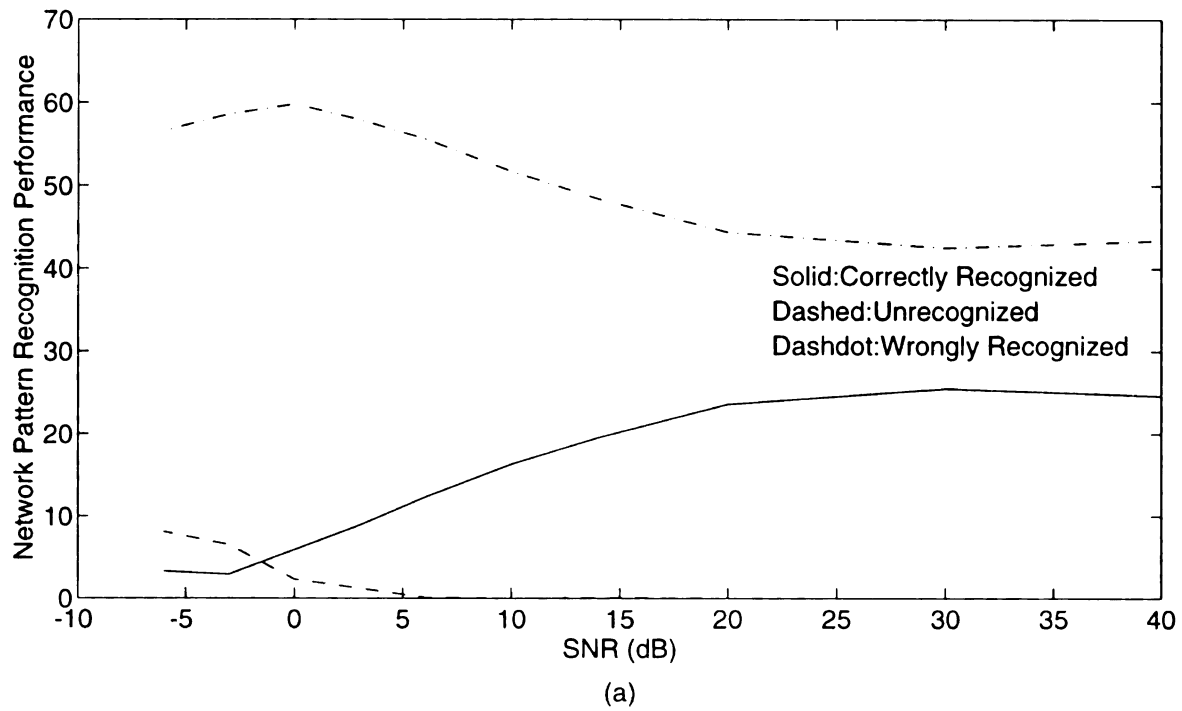


Figure 4.1 BAM network performances vs. SNR(dB) for the 68 stored patterns.
 (a). Pattern recognition performance.
 (b). Target discrimination performance by using target group code.

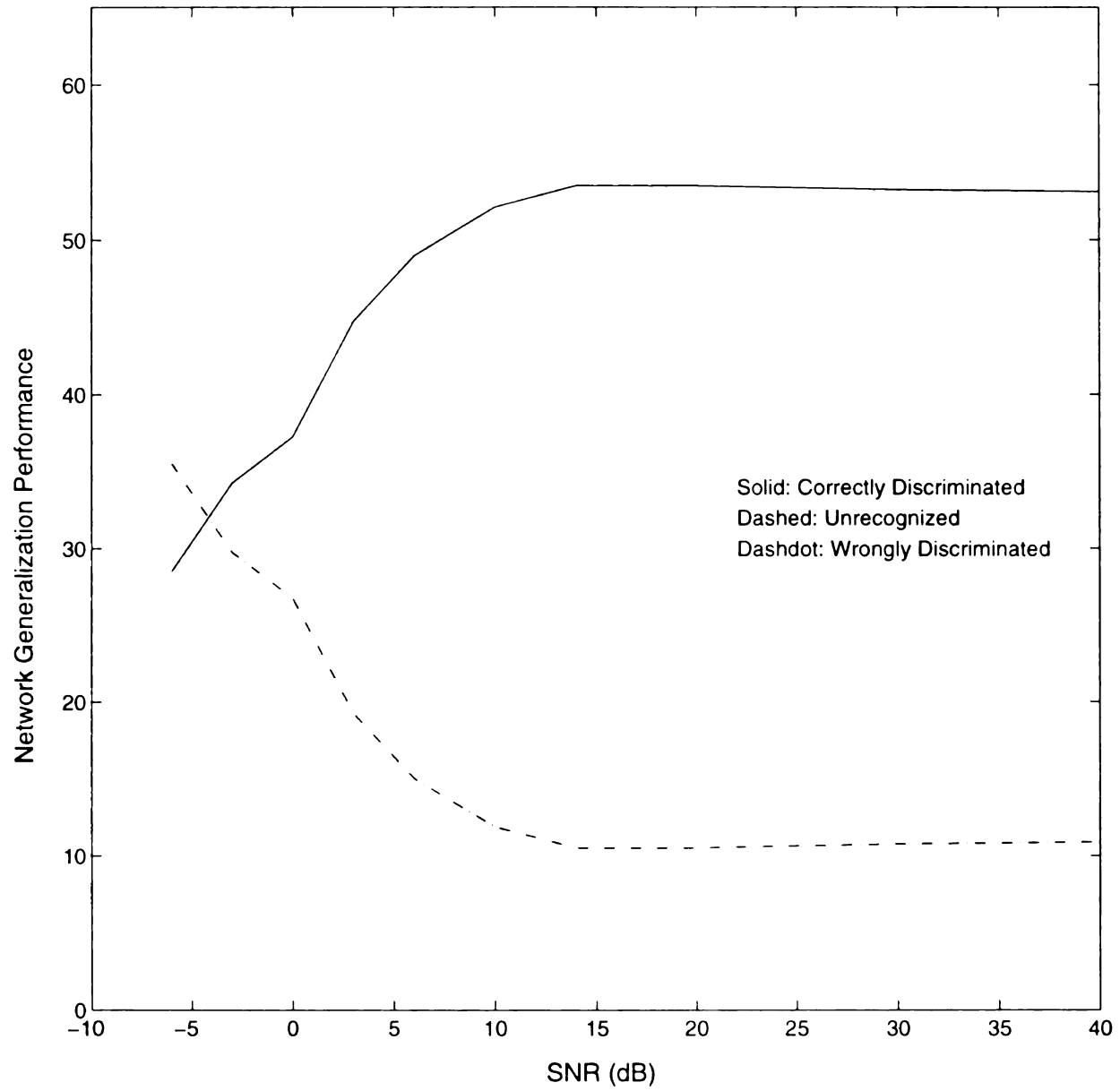


Figure 4.2 BAM generalization performance by using target group code for the 64 unstored patterns.

4.3 High order Correlation Associative Memory (HCAM) and Exponential Correlation Associative Memory (ECAM)

In the last section, we showed that the Hopfield net couldn't do anything for our application, while the BAM needed a well redesigned heteroassociative code set. Furthermore, the one-to-one mapping makes the coding assignment less flexible and also interferes the bidirectional retrieval consistence, especially for two distinct stored patterns with two close associative codes. Thus it will degrade the network performance and makes the BAM highly sensitive to coding. Both Hopfield net and BAM mechanisms give low discrimination resolution, therefore the stored states usually confuse together at each updated state.

The Recurrent Correlation Associative Memories (RCAM) are designed to recall the associative pattern y^i by using recurrent correlation operations, if given an input u which is sufficiently close to x^i . This type of neural network has application in our radar target discrimination. In this section, we will discuss High order Correlation Associative Memory (HCAM) and Exponential Correlation Associative Memory (ECAM) with autoassociative stored patterns, i.e. $x^i = y^i$ for all P stored patterns. Since the correlation of two normalized or bipolar signals is a measure of how close two signals are, the RCAM can be applied to discriminate patterns based on this property.

If x^i has binomial (binary or bipolar) components and s is an input or the current state with dimension m , then we can write the evolutionary behavior [40][41] for an RCAM by

$$s' = G \left\{ \left(\sum_{i=1}^P f_i(s^T x^{(i)}) x^{(i)} \right) \right\} \quad (15)$$

where $\mathbf{s}'$ is the next network state, $\mathbf{f}_i$ is a weighting function and $\mathbf{G}$ is a threshold (or activation) function. In bipolar processing, the Signum (**Sign**) function,

$$\text{Sign}(v) = \begin{cases} 1, & v > 0 \\ -1, & v < 0 \end{cases} \quad (16)$$

is used for the threshold function $\mathbf{G}$. We can see that the Hopfield network is a special case of the RCAM with weighting function $f(c)=c$ and degenerated diagonals. An RCAM with the above evolution equation is asymptotically stable [9] in both synchronous and asynchronous update modes if its weighting function is continuous and monotone nondecreasing. The network first computes the correlations between the given state $\mathbf{s}$ and each stored pattern, then processes each correlation by weighting function $\mathbf{f}_i$ to obtain the weighted correlation gain, and then multiplies each stored pattern by its weighted correlation gain. Finally, the network adds every amplified pattern together, and then manipulates the sum by the **Sign** threshold function to have the network output (next state). Since the correlation between two binomial vectors can be regarded as one similarity measurement, we can classify a given input into its stored prototype by appropriately using correlation gains. Generally, the larger a correlation value two vectors have, the closer they are. So the weighting function should be strictly increasing to assure the viability of the correlation-based retrieval algorithm.

For a Hopfield net with nonzero diagonals, the network has next state $\mathbf{s}'$

$$\mathbf{s}' = \text{Sign} \left\{ \sum_{i=1}^p (\mathbf{s}^T \mathbf{x}^{(i)}) \mathbf{x}^{(i)} \right\} \quad (17)$$

The network only considers a 1-dimension relationship between the given state and the

stored vector x^i . Since the next state is generated by the sum of each amplified pattern, similar to the above BAM analysis, the weighted sum can be separated into two terms, one is the associative pattern amplified by its correlation gain with the input s and the sum of the others weighted patterns is a noisy term. Only the weighted associative pattern prevails over the noisy term during recurrent nonlinear updates, then the correct recall can be attained. Therefore, with a fair number of stored patterns, a pattern with a slightly larger correlation gain may be distorted by the sum of the others, and then it won't dominate the next state after the addition of all amplified patterns. One important reason is that the discrimination resolution for one order correlation is not sufficiently high enough to survive a stored pattern input through the sum of weighted patterns. If we compare the relationship between $x_j^i x_k^i$ and $s_j s_k$, where $j=1, \dots, m$ and $k=1, \dots, m$, then we can use more information to emphasize the correlation between these two vectors x^i and s before adding them to provide the next state. The one-dimension model (Hopfield net) compares two vector strings bit by bit to compute the number of identical bits, while the two-dimensional model constructs the individual autocorrelation plane for each stored pattern and the given input, i.e. $x^i(x^i)^T$ and $s(s)^T$, and then compares the input autocorrelation 2-d plane to each individual stored pattern 2-d plane to find the closest autocorrelation plane structure among all stored patterns. The autocorrelation plane not only offers the similarity information about linear position but also investigates the similarity of crosscorrelations among bit positions between the given state and all stored states. So it's clear that the 2-d model uses m times the information of the 1-d model to discriminate patterns.

The network by taking advantage of high dimensional autocorrelation hyperplane is

called the High Order Correlation Associative Memory (HCAM). A HCAM [40][41] has the weighting function

$$f(c) = (c + T_{os})^r \quad (18)$$

where $r > 1$. T_{os} is some offset value designed to avoid amplifying negative correlation gains for even r . If $(c + T_{os})$ is positive or r is a positive odd integer, the weighting function f is strictly increasing, as required for correct retrieves.

Another RCAM used for our target discrimination simulations is the Exponential Correlation Associative Memory (ECAM) which has the weighting function

$$f(c) = b^c \quad (19)$$

where $b > 1$. Again, this weighting function f is strictly increasing. RCAM's with a continuous and strictly increasing weighting function f are asymptotically stable in both synchronous and asynchronous update. This means that network recurrent operations will drive a given input state to some stable state, therefore ensuring no oscillation cycle during the recurrent convergent iterations. Thus the RCAM will converge to either one of the stored patterns or some spurious (unknown) state when triggered at the input by a given pattern.

A HCAM needs a predetermined fixed order r to proceed, since the sequential convergence will continue until the stable state is reached as soon as an input is presented to the network. A network with low order r can't retrieve correct associative patterns, while a high order r wastes computation space. When the ECAM exponentially amplifies the correlation gains, producing excellent discrimination resolution, it also exponentially expands the network computation space. It is not possible to physically realize an ECAM for

processing patterns with large dimension. For example, in our simulations each bipolar stored pattern has 300 bits, so the maximum weighted correlation gain is 2^{300} for the ECAM with the weighting function $f(c)=2^c$, i.e. $b=2$. Therefore, a huge processing space is required to fulfill the hardware realization in chip design scale. When a given input is closer to one of the stored patterns, the ECAM requires a much larger scale space for processing, and the larger scale computation needs a longer time for processing. This phenomenon somehow seems to be illogical and unreasonable.

The storage capacity of ECAM is not a deterministic scale or a definite meaning for our application. Since radar target response signals are far away from random signals, the estimated storage capacity based on random data and a lot of impractical probability assumptions couldn't be a valuable operation guide to us. But for reference purpose we still give it here. Consider an N -bit input pattern with r ($r \equiv \rho N$ and $0 \leq \rho < 1/2$) bits away from the nearest stored pattern. Then the definition of storage capacity is, as the pattern's dimension (or number of bits) $N \rightarrow \infty$, defined as what is the greatest rate of growth of $M(N)$ so that after one iteration the bit-error probability (the probability that a bit in the next state is different from the corresponding bit in the nearest stored pattern) is less than $(4\pi T)^{-1/2}e^{-T}$, where T is a fixed and large number.

Suppose an ECAM is loaded with [41]

$$M(N) = \begin{cases} [a^4/(4T)] 2^{N(1-\mathcal{H}(\rho'))} + 1 & \text{if } \rho' \geq (1+a^{-2})^{-1} \\ [a^4/(4T)] 2^N [(1+a^{-2})a^{2\rho'}]^{-N} + 1 & \text{if } \rho' < (1+a^{-2})^{-1} \end{cases} \quad (20)$$

N -bit memory patterns, where $\rho' = \rho + (1/N)$, $0 \leq \rho < 1/2$, and $\mathcal{H}(\rho') = -x \log_2 x - (1-x) \log_2$

$(1-x)$ is the binary information entropy of p' . If the current state pattern $\mathbf{x}$ is ρN bits away from the nearest memory pattern, then as $N \Rightarrow \infty$, the bit-error probability ($\mathbf{P}_e$) is less than $(4\pi T)^{-1/2} e^{-T}$ where T is a fixed and large number. The proof of the above capacity is trivial and is omitted here. According to the above formula, the ECAM has a storage capacity that scales exponentially with the number of bits, N , in memory patterns, however the dynamic range required of exponentiation hardware implementation also grows exponentially with N . Therefore, it is almost impossible to realize in hardware for processing patterns with large dimensions.

4.3.1 Simulations and Results

First of all, we simulate the HCAM with an order of 3 and find that quite a few stored patterns can not be discriminated. Then the HCAM with an order to 5 is simulated and the performance is greatly improved. **Figure 4.3** (a) shows the HCAM (order 3) pattern recognition performance for 68 contaminated stored patterns, while **Figure 4.3** (b) presents the HCAM (order 3) target discrimination performance. The target discrimination rate is the same as the pattern recognition rate under low contaminations, while the target discrimination is 9.1% greater than the pattern recognition rate at 0 dB SNR. Compared to Hopfield net, the HCAM's performances definitely prevail although both networks are recurrent autoassociative nets. Compared to the BAM's pattern recognition performance, the HCAM of order 3 is much better than the BAM.

Since the HCAM is an autoassociative net and the stored aspect pattern codes are encoded from the real lab-measured responses, the stored aspect patterns did not have a designed distinct target code for every 17 aspect patterns of each target. On the other hand,

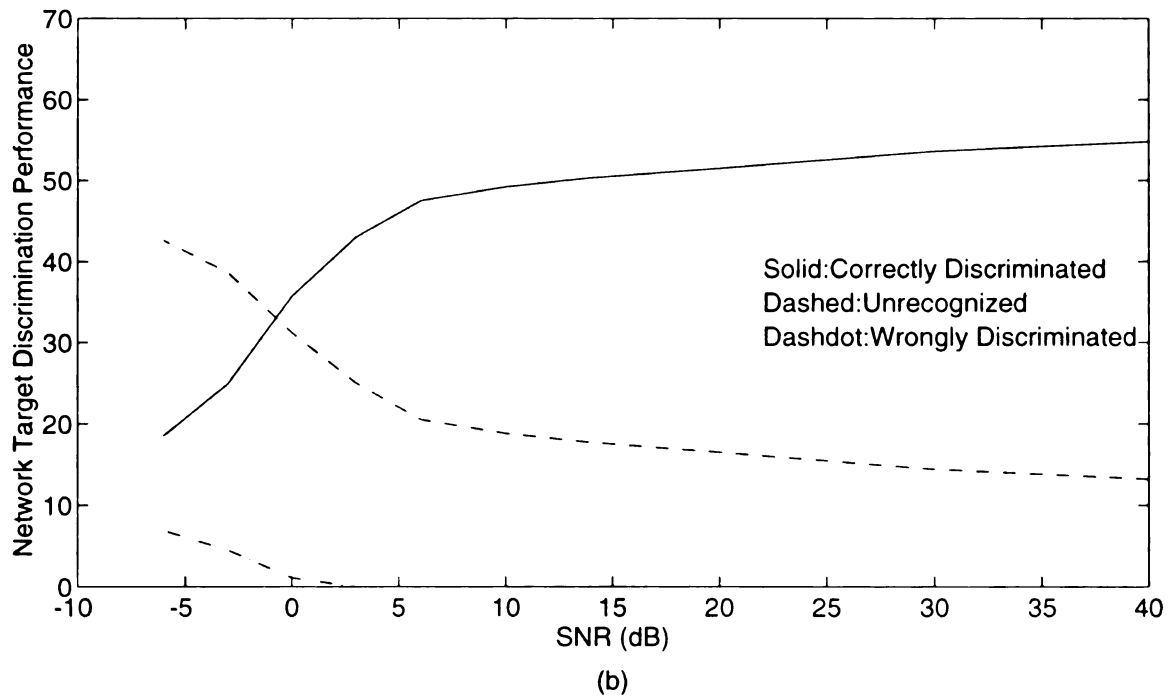
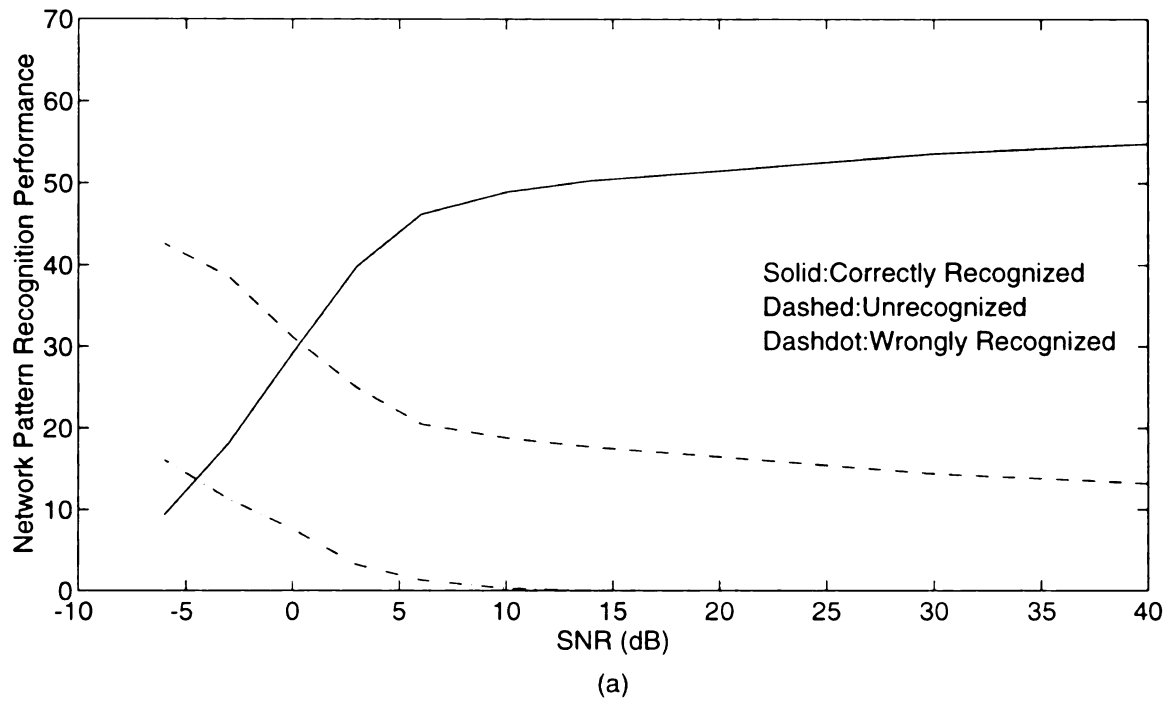


Figure 4.3 HCAM (order 3) network performances vs. SNR(dB) for the 68 stored patterns.

- (a). Pattern recognition performance.
 (b). Target discrimination performance.

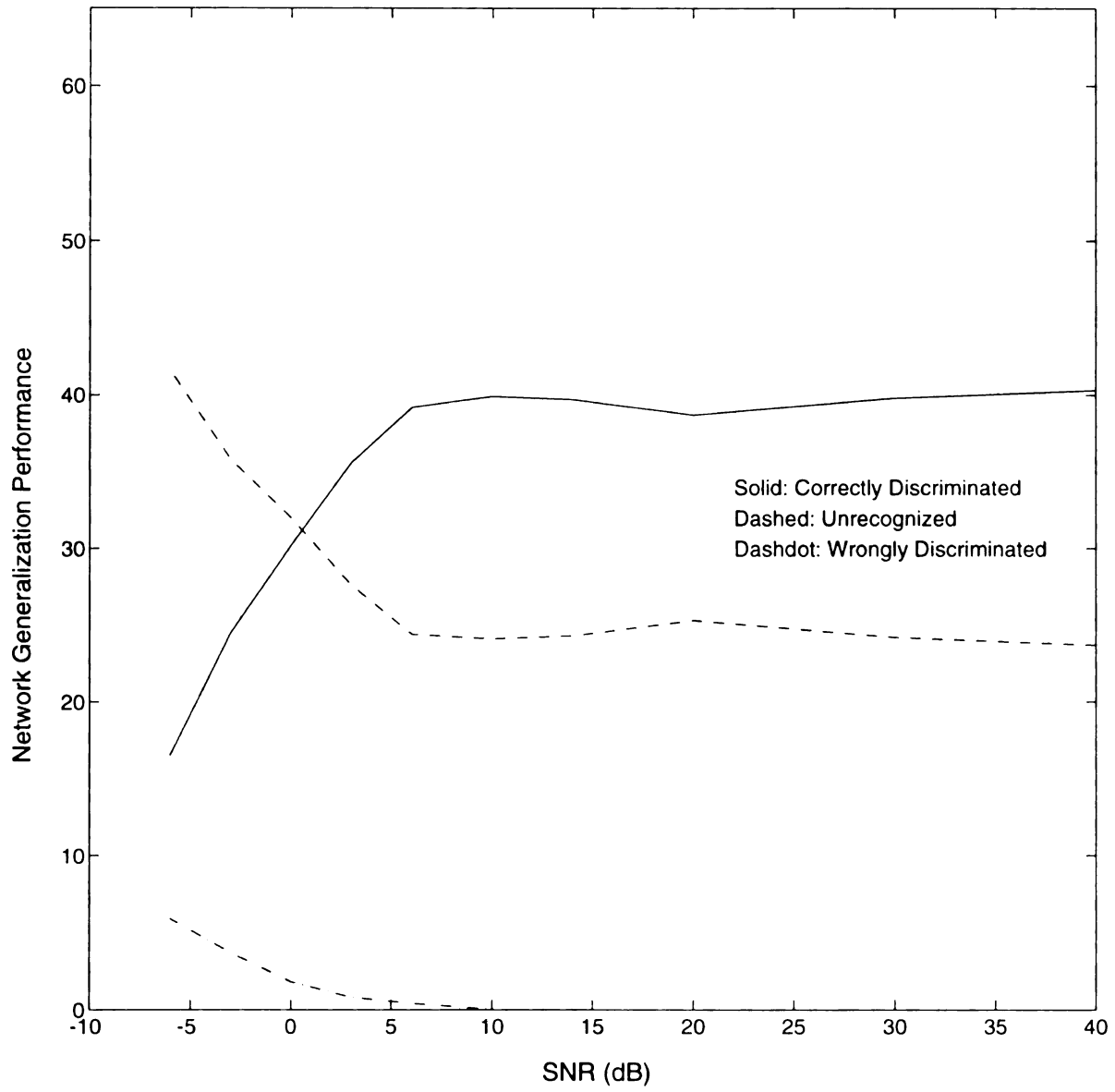


Figure 4.4 HCAM (order 3) generalization performance for the 64 unstored patterns.

we gave the BAM a postdesigned code set, target code + dummy code, with prior known information about the aspect-target relationship to associate with the real world measured aspect patterns. Therefore, comparison between the BAM's target discrimination performance and the HCAM's is not appropriate. The target discrimination comparison between the BAM with target group code and the HCAM-GI cascade network, discussed later, will be meaningful. **Figure 4.4** presents the HCAM (order 3) generalization performance for 64 contaminated unstored patterns and the result shows that the unknowns increase. The HCAM performances for both stored and unstored patterns show that the HCAM prefer to leave an contaminated pattern unrecognized rather than to classify it into a wrong pattern or target. **Figure 4.5 (a)** shows the pattern recognition performance of the HCAM with an order of 5 for 68 contaminated stored patterns, while **Figure 4.5 (b)** presents the target discrimination performance. Compared to the one with an order of 3, HCAM with an order of 5 has greatly reduced its unrecognized rates and then increases the correct discrimination rates. Under serious contaminations, some distorted aspect patterns converge to a wrong aspect pattern of its own target, and then this only decreases the correct pattern recognition performance but won't degrade the target discrimination. **Figure 4.6** shows the generalization performance of the HCAM with an order of 5 for 64 contaminated unstored patterns. Compared to the one of order 3, the HCAM of order 5 has also greatly increased the generalization performance.

Figure 4.7 (a) shows the ECAM pattern recognition performance for 68 contaminated stored patterns, while **Figure 4.7 (b)** presents the ECAM target discrimination performance. The ECAM performances for both pattern and target discriminations are better than the ones of the HCAM with an order of 5. It is apparent that the ECAM almost leaves no unknown

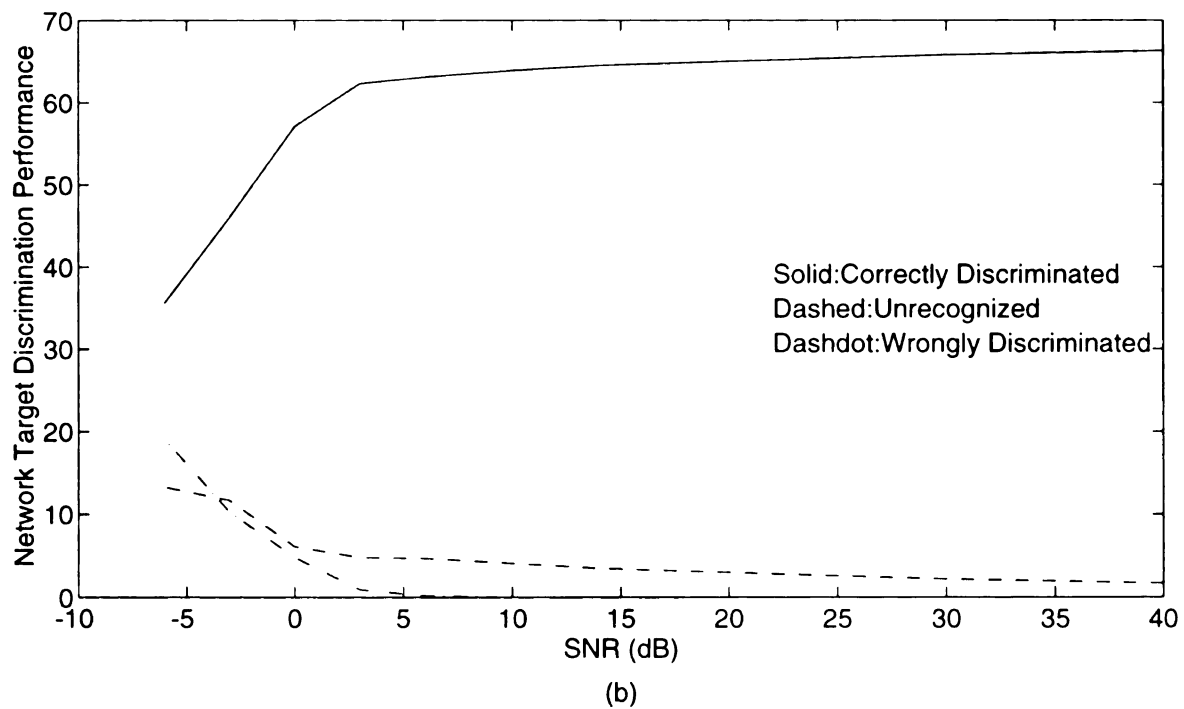
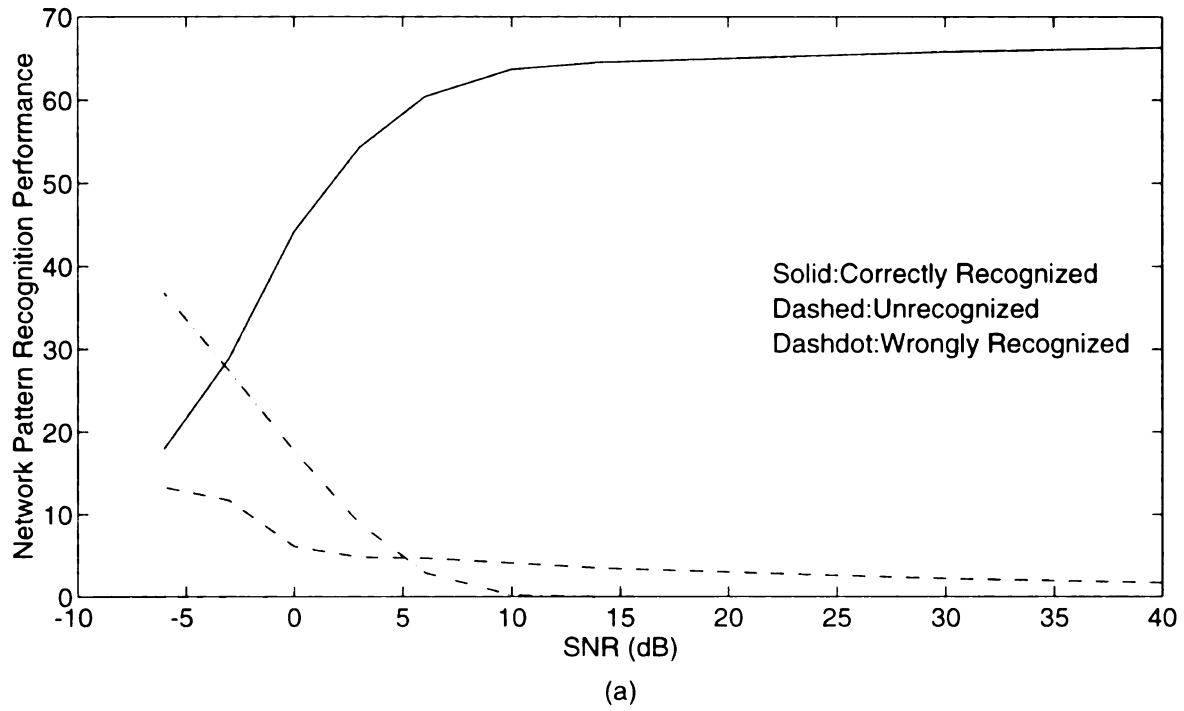


Figure 4.5 HCAM (order 5) network performances vs. SNR(dB) for the 68 stored patterns.

- (a). Pattern recognition performance.
 (b). Target discrimination performance.

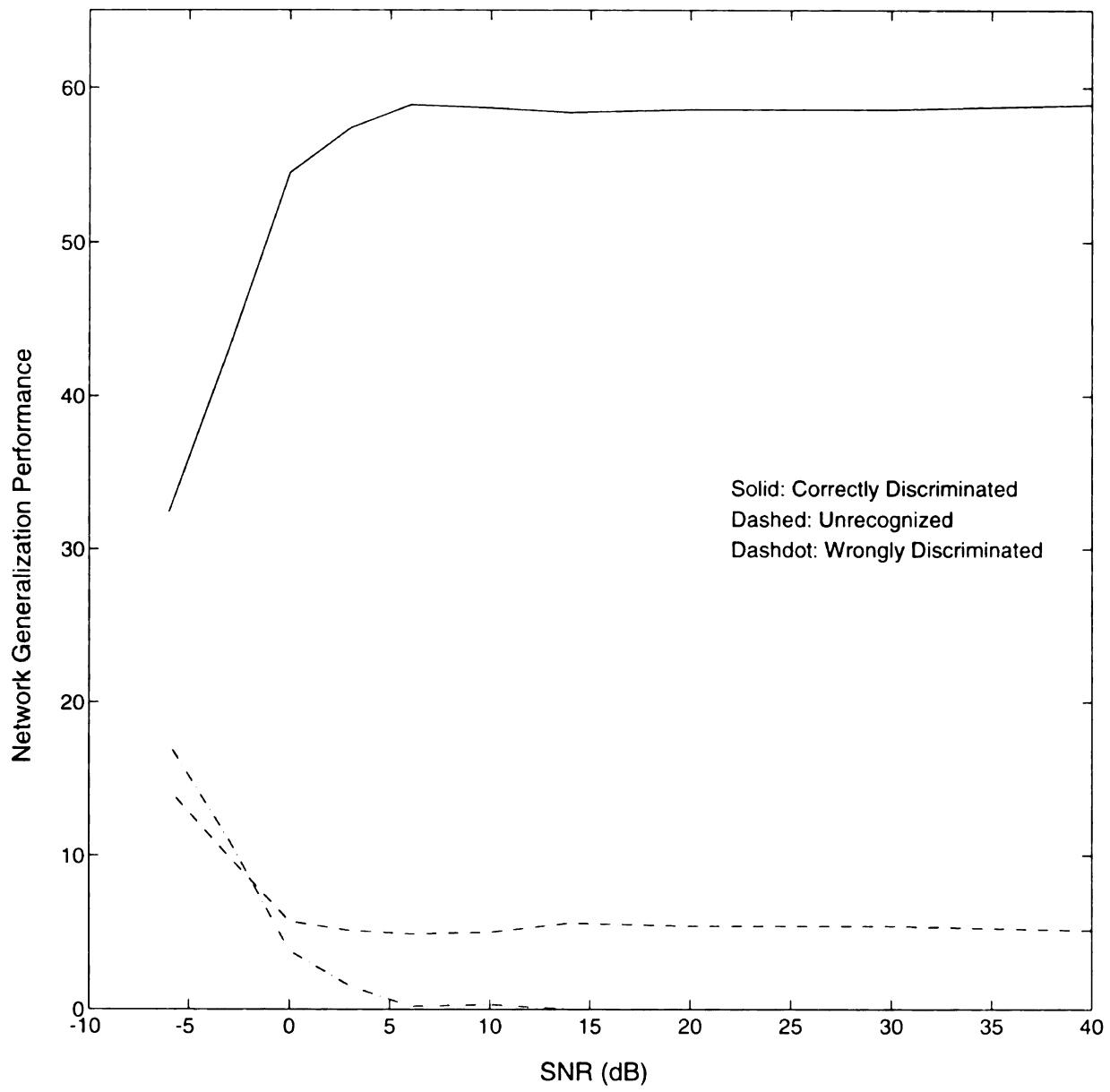


Figure 4.6 HCAM (order 5) generalization performance for the 64 unstored patterns.

behind even under severe distortions. Since the ECAM exponentially amplifies correlation gains and then multiplies each stored patterns by its individual gain, the resolution space is extremely huge. Thus the ECAM will still blindly classify a nonsensically ambiguous input or even an arbitrary noise into one of the stored patterns by calculating the exponentially exaggerated correlations between input and each individual stored pattern. This characteristic is good for light or fair contaminations but not reliable for severe distortions. Compared to the HCAM, the ECAM converges all unknowns to correct patterns or targets for contamination less than 3 dB SNR, and divides unknowns into correct and wrong discrimination for distortion greater than 3 dB SNR. At -6 dB SNR, the risk becomes higher than one half. The wrong discrimination rates dramatically increase for severe distortions, since the coding scheme is not linear when the contamination range is statistically greater than 1.5 quantization levels, analyzed in the previous chapter. But the ECAM's converging everything is the network's own characteristic and is not subjected to the coding scheme. **Figure 4.8** shows the ECAM generalization performance for 64 contaminated unstored patterns. Compared to HCAM, it also has excellent performance for fairly contaminated unstored patterns. As predicted, the ECAM converges everything and leaves no unknown for contaminated unstored patterns.

4.4 RCAM-GI Cascade Networks

From the previous section, the HCAM of order 5 still left quite a few unknowns behind even for light contaminations. The computation space will be increased, although the network performance will be improved, if the HCAM uses a larger order. The larger order HCAM will waste resource under light contaminations. From the last chapter, we know that

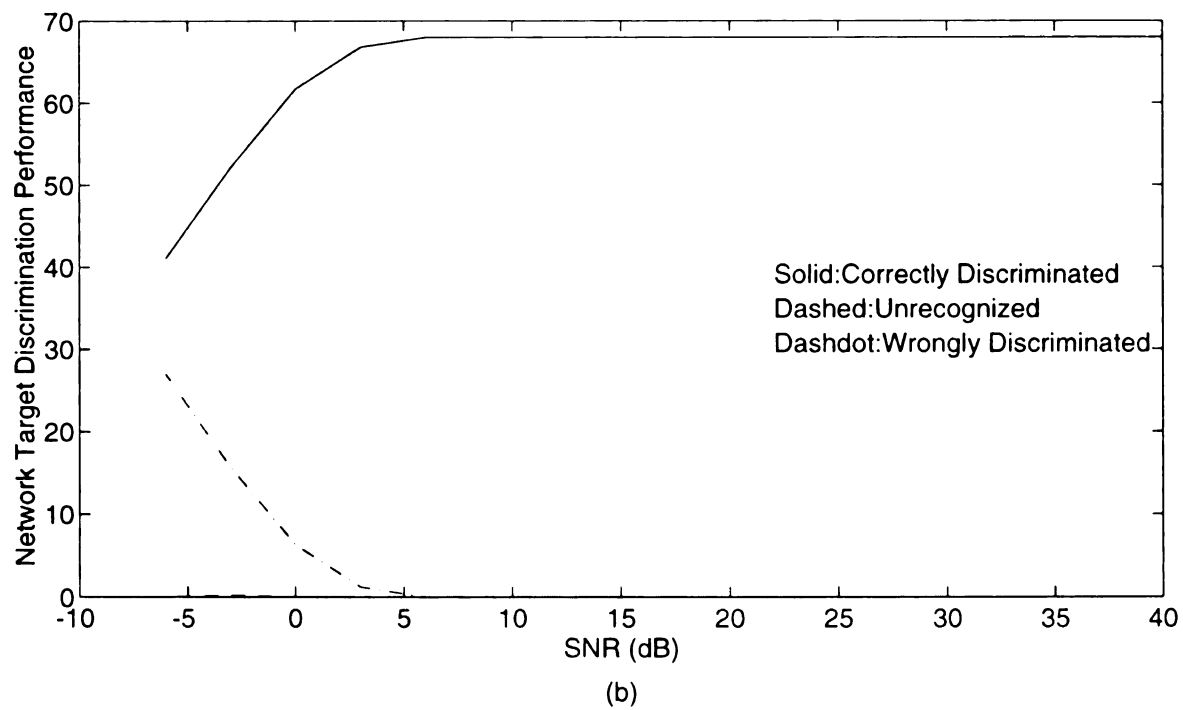
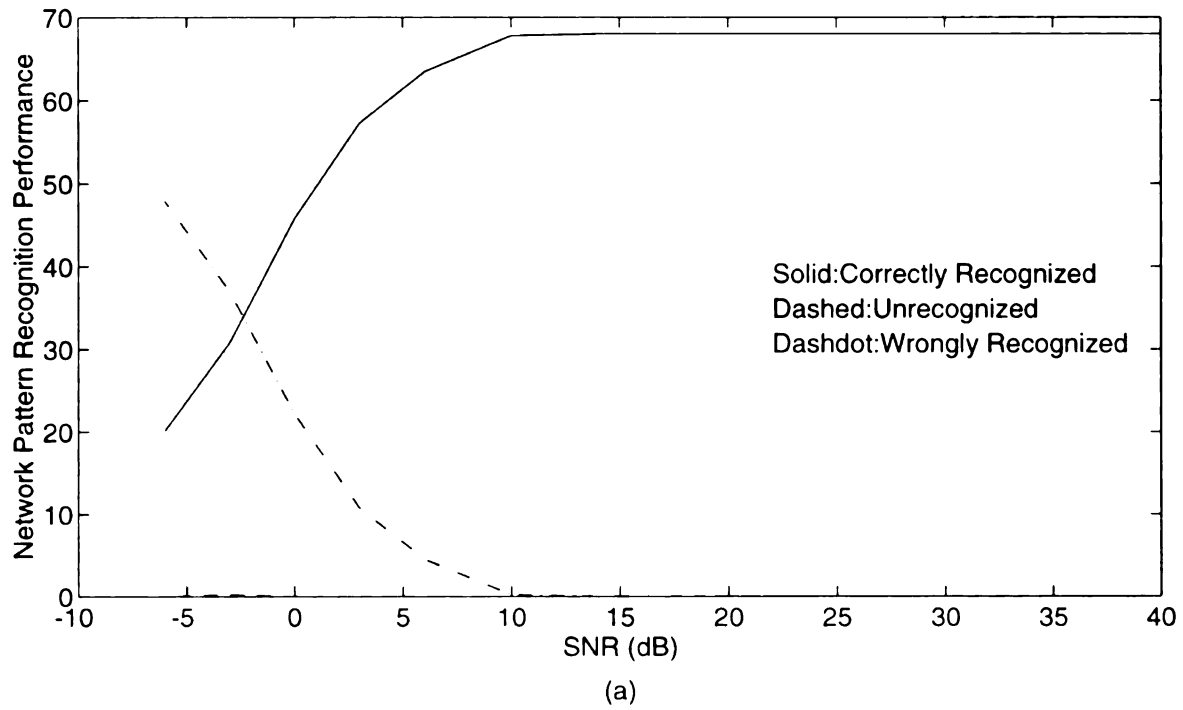


Figure 4.7 ECAM network performances vs. SNR(dB) for the 68 stored patterns.
 (a). Pattern recognition performance.
 (b). Target discrimination performance.

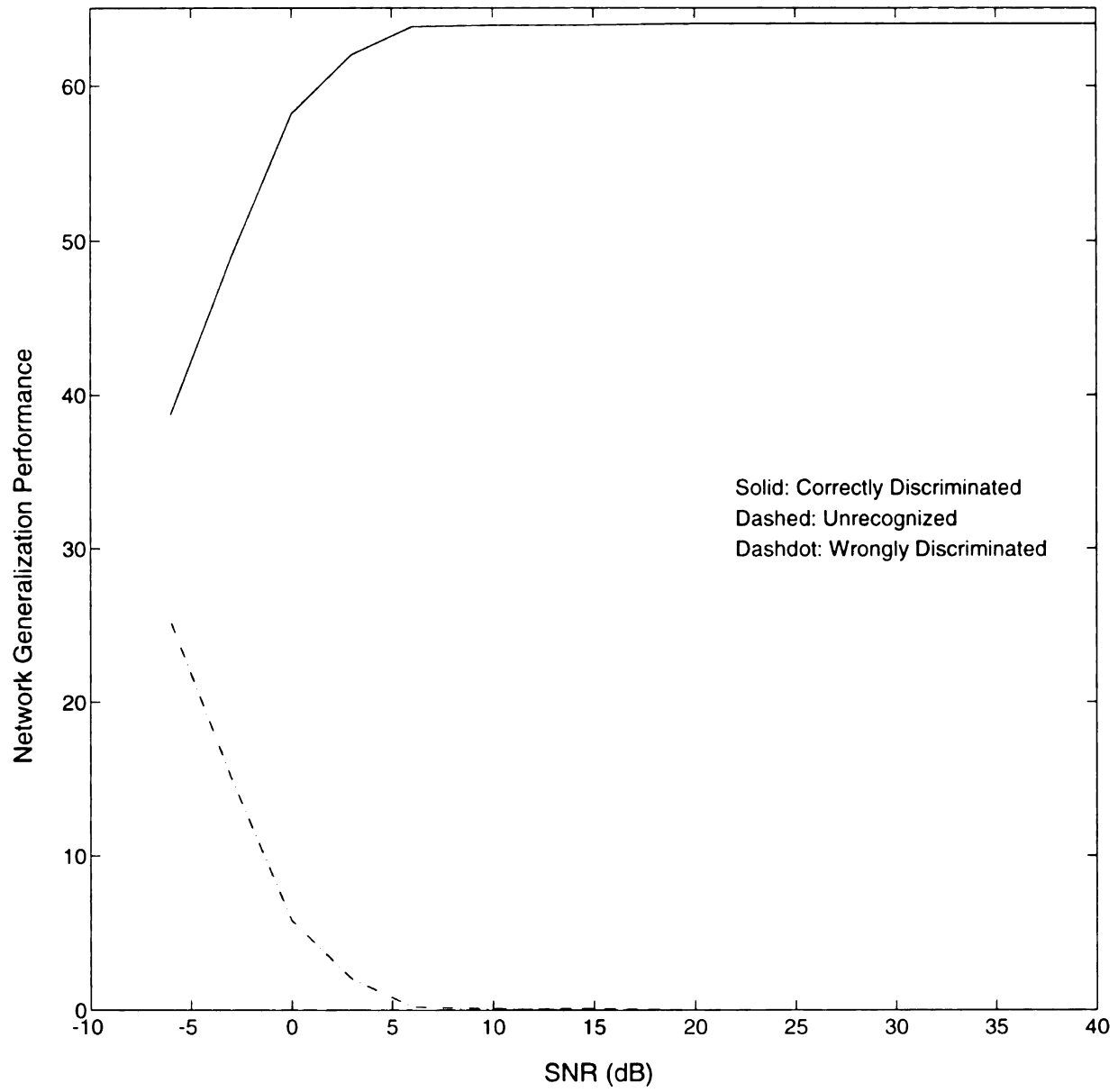


Figure 4.8 ECAM generalization performance for the 64 unstored patterns.

the Generalized Inverse (GI) network can converge most unknowns to correct target under fair contaminations, and prefers to leave an ambiguous state unrecognized rather than to wrongly discriminate it. Another advantage of GI network is that the GI network requires a small implementation memory. And our purpose is to discriminate target by using variable aspect inputs. Therefore, it is reasonable to expect that an RCAM-GI cascade network will save much memory and converge more unknowns.

We have constructed the GI network by initializing a correlation memory and then learning to converge all desired associative pattern pairs. In our GI net, the desired associative patterns have multiple-to-one mapping, thus multiple aspect responses map to a same target. The GI net initially has correlation-based record memory and then learn to enlarge the attractive basin realms of each stored pattern. Then it will converge to the target with which the basin is associated, if a given input falls into one of those attractive basins. By using this learned attractive potential, we can use a fair order HCAM to save space and leave unknowns behind. Then the cascading GI net will converge those attracted by its basins and still leave ambiguous ones unrecognized.

4.4.1 Simulations and Results

Figure 4.9 (a) shows the HCAM (order 3)-GI cascade network performance for 68 contaminated stored patterns, while **Figure 4.9** (b) presents the HCAM (order 3)-GI cascade network generalization performance for 64 contaminated unstored patterns. As expected, compared to **Figure 4.3** (b), the HCAM (order 3)-GI cascade network correctly converges all unknowns left by the HCAM (order 3) for the SNR's greater than 6 dB SNR. For serious distortions, the cascade network converges most unknowns and leaves some of them

unrecognized, and the ratio of correct convergence to wrong convergence is greater than 1. Even compared to the HCAM (order 5), the HCAM (order 3)-GI cascade network still has better performances for the SNR greater than 0 dB. This better performance has rewarded us the desired fulfillment that the HCAM-GI cascade net not only save processing memory but also increases discrimination performances. The cascade network has better generalization performances than the HCAM with order of 3 or 5, except it wrongly discriminates one slightly contaminated unstored pattern.

Figure 4.10 (a) shows the HCAM (order 5)-GI cascade network performance for 68 contaminated stored patterns, while **Figure 4.10** (b) presents the HCAM (order 5)-GI cascade network generalization performance for 64 contaminated unstored patterns. Again, the HCAM (order 5)-GI cascade network has a better performance than the HCAM of order 5. Compared to the HCAM (order 3)-GI cascade net, the HCAM (order 5)-GI cascade network also demonstrates further improved performances, due to the fact that the HCAM of order 5 has better performances than the HCAM of order 3. The HCAM (order 5)-GI cascade network this time doesn't wrongly discriminate any slightly contaminated unstored pattern, and shows a better generalization performance than the HCAM of order 5.

Figure 4.11 (a) shows the ECAM-GI cascade network performance for 68 contaminated stored patterns, while **Figure 4.11** (b) presents the ECAM-GI cascade network generalization performance for 64 contaminated unstored patterns. Except the unknown of ECAM at -3 dB SNR is converged, the ECAM-GI cascade network performance is the same as the one of the ECAM. Since the extremely huge discrimination space of the ECAM leaves no unrecognized states for the cascading GI network to further converge, the ECAM-GI

cascade net goes nowhere to improve performance. Similarly, the ECAM-GI cascade network has the same generalization performance as the ECAM does.

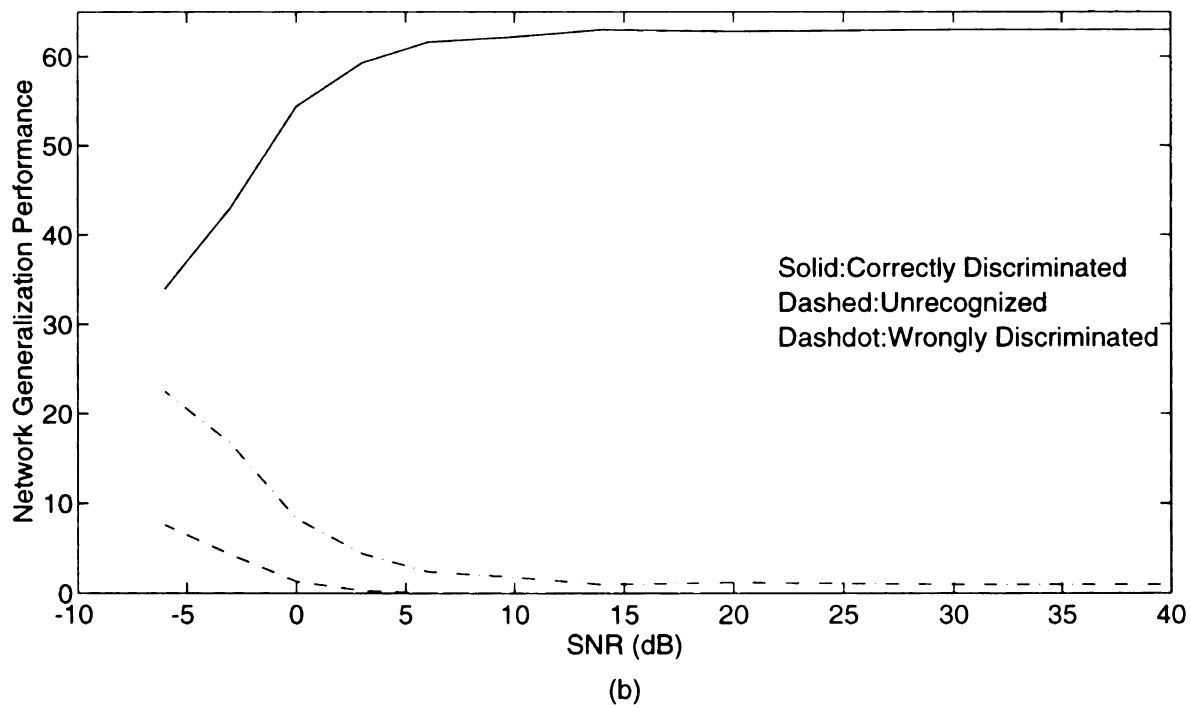
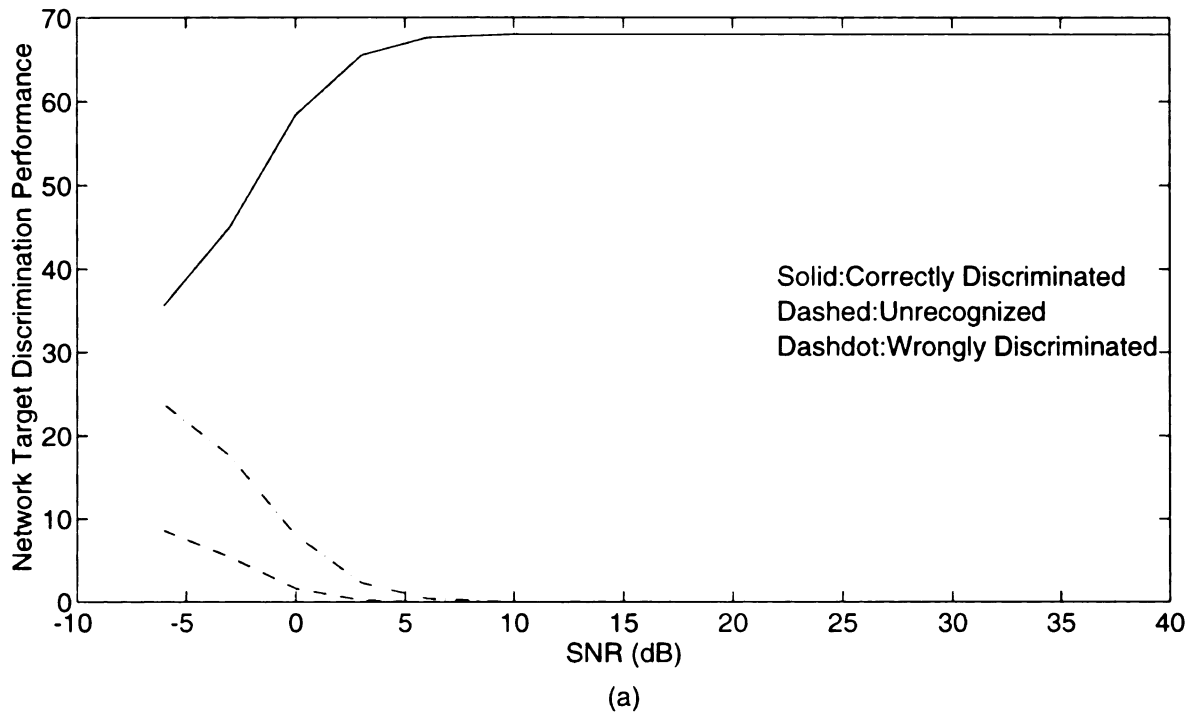


Figure 4.9 HCAM (order 3)-GI cascade network performances vs. SNR(dB).

(a). Target discrimination performance for the 68 stored patterns.

(b). Generalization performance for the 64 unstored patterns.

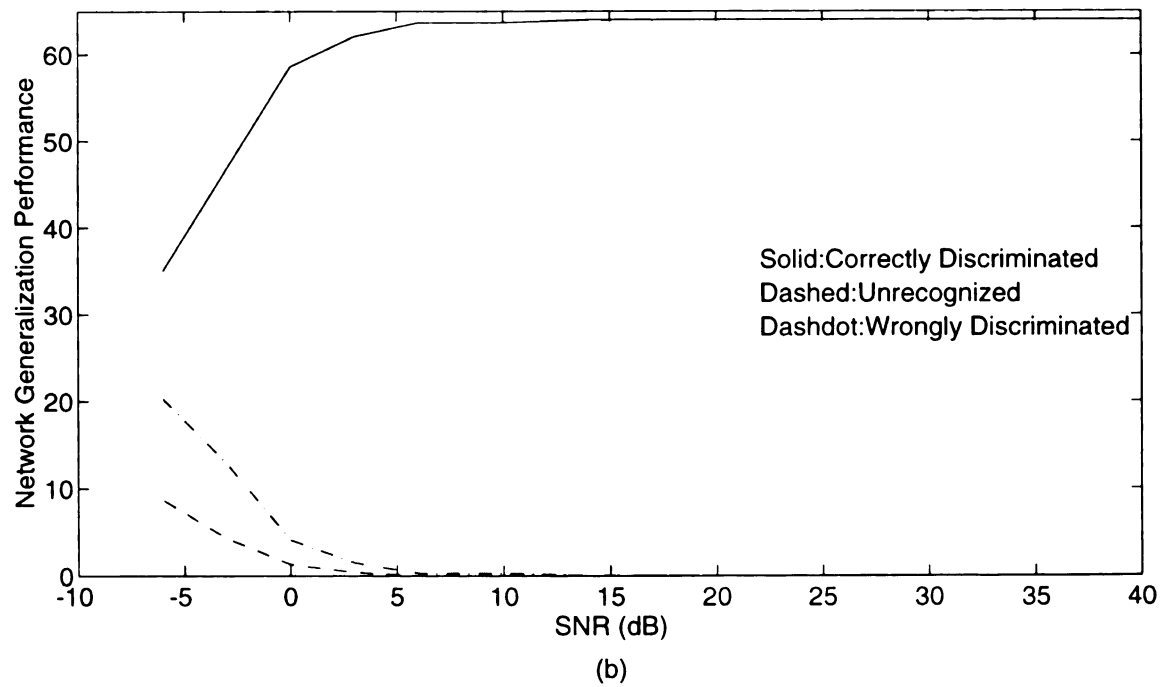
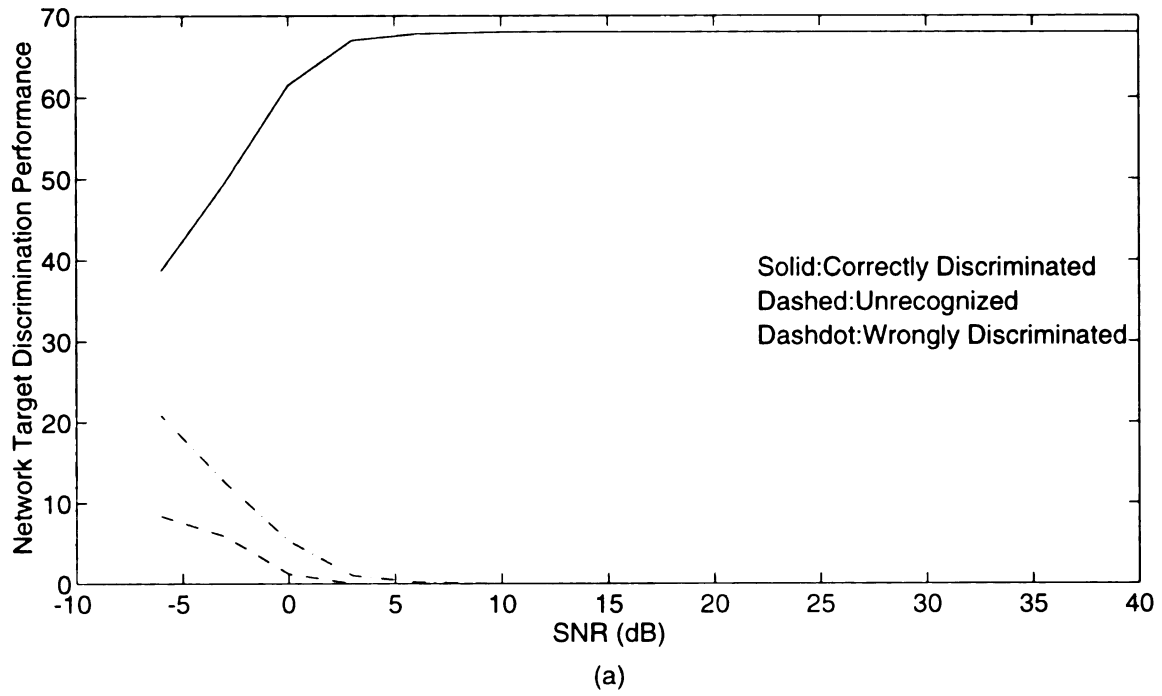


Figure 4.10 HCAM (order 5)-GI cascade network performances vs. SNR(dB).

(a). Target discrimination performance for the 68 stored patterns.

(b). Generalization performance for the 64 unstored patterns.

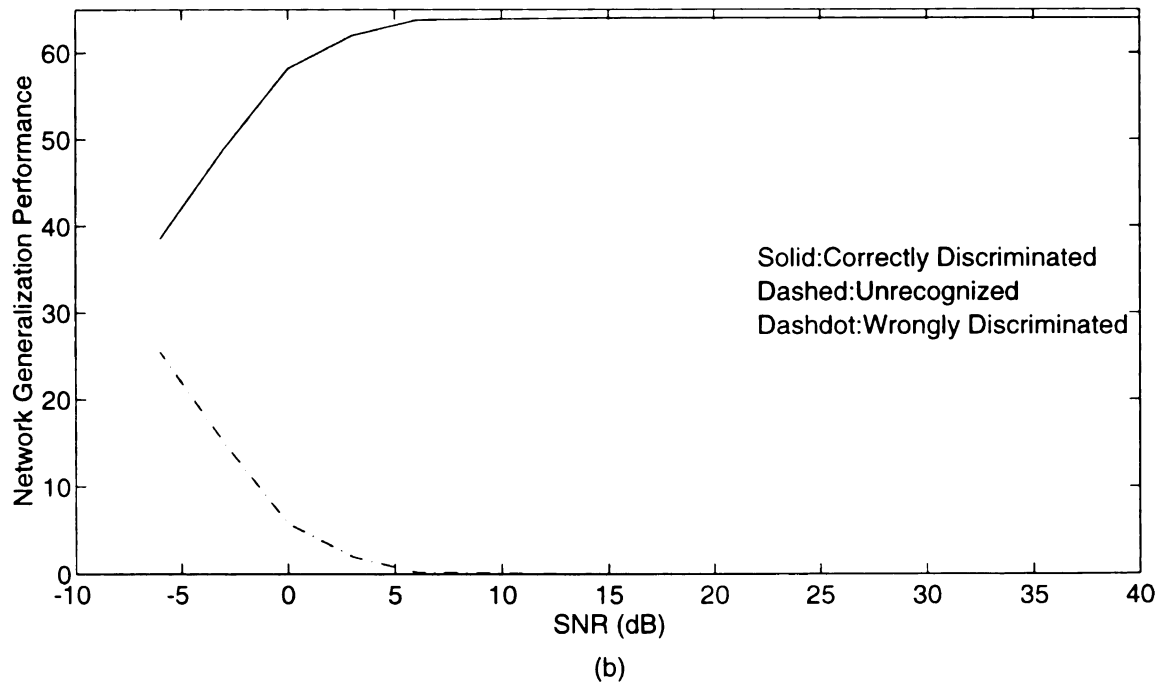
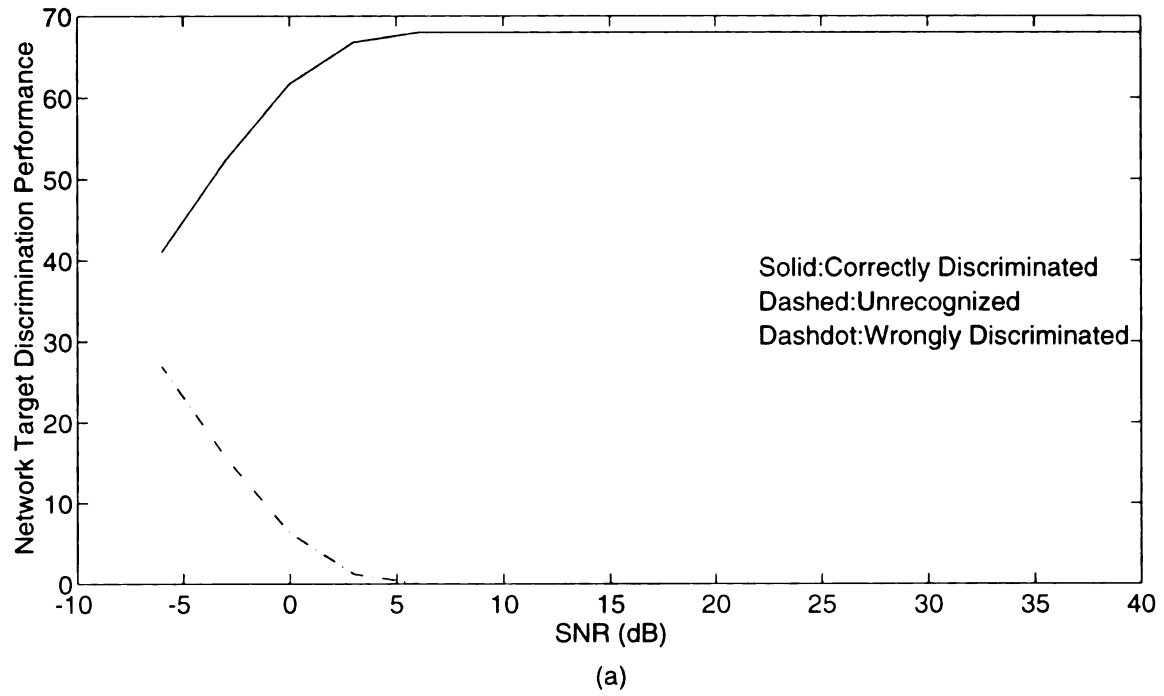


Figure 4.11 ECAM-GI cascade network performances vs. SNR(dB).

(a). Target discrimination performance for the 68 stored patterns.

(b). Generalization performance for the 64 unstored patterns.

CHAPTER 5

Recurrent Correlation Accumulation Adaptive Memories

5.1 Introduction

From the previous Chapter, for a HCAM, we need to guess for what order the network can discriminate well among all possible inputs before processing the input. Some inputs may be easily discriminated, while some may be more difficult. We also know it is nearly impossible to physically realize an ECAM for large dimension patterns, since it exponentially expands network computation space. It is also unreasonable and wasteful that the ECAM blindly reserves a huge memory without considering the necessity of variable inputs. This abuse will give no information about the input contamination when the output appears. If a network can use flexible and sufficient orders of correlation to reach similar performance, then it will be a better choice. Thus we wish to use a dynamic order, dependent on the input, to discriminate among the stored patterns.

5.2 Recurrent Correlation Accumulation Adaptive Memories (RCAAM)

When the recurrent update of RCAM amplifies individual correlations between the initial given input and stored patterns, it also introduces noisy crosscorrelation terms between any two stored patterns i and j with $i \neq j$. Assume we have P patterns $\{\mathbf{x}^i | i=1, 2, \dots, P\}$ stored in memory, where $\mathbf{x}^i$ is an m -dimension column vector, so that $\mathbf{X}=\{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^P\}$. If $\mathbf{x}^i$ has binomial (binary or bipolar) components and $\mathbf{s}$ is an initial input, then the HCAM of order r will have output state $\mathbf{s}''$ after two synchronous updates

$$s'' = \text{Sign} \left\{ \sum_{j=1}^P [x^{(j)T} \text{Sign} \left(\sum_{i=1}^P (x^{(i)T} s) x^{(i)} \right)] x^{(j)} \right\} \quad (1)$$

Therefore it did not purely amplify the correlations between the initial given input and stored patterns, and the nonlinear threshold function **Sign** prohibits the recurrent iterations from linearly accumulating the respective pattern correlation gains, $[(x^{(i)})^T s]^r$, produced by the last iteration. The recurrent feedbacks introduce noisy crosscorrelation terms. If the network can release the nonlinear interferences caused by the threshold function **Sign** and accumulate the previous respective correlation gains for each stored pattern, then the network will function efficiently and stably. No nonlinear interference means there will be linear amplifications on respective (cross) correlation terms generated during the last iteration, and the accumulations of previous recurrent iterations will speed up the order of correlations.

We propose a Recurrent Correlation Accumulation Adaptive Memory (RCAAM) which uses dynamic memory structure to accumulate the correlation information between the input(s) and all stored patterns. Then the network discrimination resolution (ability) to an input will increase as the recurrent iterations increase. Compared to the ECAM and HCAM, the RCAAM uses recurrent accumulative and dynamic structure to gradually converge a given input to some (semi-)stable state(s). We may regard this network as a real time learning network. The network adjusts its real-time learning structure to converge the given input to the nearest stable state associated to the stored patterns as long as the recurrent operations continue. And, unlike Multi-layer Feedforward Error-Backpropagation learning it can avoid being trapped in local minimum states.

Suppose the stored associative pattern pairs are $\{(\xi^i, \zeta^i) | i = 1, \dots, P\}$, where ξ^i is a

column vector with length m and ζ' is a column vector with length n . We intend to implement a neural network that can recall ξ' if given an input sufficiently close to ζ' . If U is an n -dimension column vector input and **init** is a positive initial order, then we can construct the initial RCAAM, M_o , as follows :

$$M_o = \sum_{i=1}^P \xi^{(i)} [(\zeta^{(i)T} U)^{\text{init}} \zeta^{(i)}]^T = \sum_{i=1}^P \xi^{(i)} (w_o^{(i)} \zeta^{(i)})^T \quad (2)$$

where $w_o^{(i)}$ is the dynamic weighting for the i^{th} stored pattern at time 0. If **init**=0, then M_o will degrade to the Hopfield memory with nonzero diagonal. Suppose the stored patterns ξ' have bipolar form, then, with this initial memory matrix, the network output is given by

$$V_o = \text{Sign} \{M_o U\} = \text{Sign} \left\{ \sum_{i=1}^P [(w_o^{(i)} \zeta^{(i)})^T U] \xi^{(i)} \right\} \quad (3)$$

We have developed three versions of the RCAAM based on the recurrent updatings. The first version is RCAAM with fixed input (RCAAM/fi). Version 2 is RCAAM with dynamic input connected to output (RCAAM/di). Version 3 is RCAAM with analog input/digital output (RCAAM/ad). Therefore, RCAAM/fi and RCAAM/di have binomial ζ' and ξ' , while RCAAM/ad has analog ζ' but binomial ξ' . Suppose U_k denotes the network input at time k , and V_k denotes the network output at time k corresponding to the input U_k . Then, the dynamic memory has $U_k=U$ for both RCAAM/fi and RCAAM/ad, and has $U_k=V_{k-1}$ for RCAAM/di. The dynamic memory and the network output at iteration 1 are given by

$$\begin{aligned}
M_1 &= \sum_{i=1}^P \xi^{(i)} \{[(w_o^{(i)} \zeta^{(i)})^T U] \zeta^{(i)T}\} = \sum_{i=1}^P \xi^{(i)} (w_1^{(i)} \zeta^{(i)})^T \\
V_1 &= \text{Sign} \{M_1 U_1\} = \text{Sign} \left\{ \sum_{i=1}^P [(w_1^{(i)} \zeta^{(i)})^T U_1] \xi^{(i)} \right\}
\end{aligned} \tag{4}$$

where the dynamic weighting for the stored pattern ζ^i at iteration 1 has $\mathbf{w}_1^{(i)} = (\mathbf{w}_o^{(i)} \ \zeta^i)^T \mathbf{U}$.

Then the adaptive memory and network output at time $\mathbf{k}$ are given by

$$\begin{aligned}
M_k &= \sum_{i=1}^P \xi^{(i)} \{[(w_{k-1}^{(i)} \zeta^{(i)})^T U_{k-1}] \zeta^{(i)T}\} = \sum_{i=1}^P \xi^{(i)} (w_k^{(i)} \zeta^{(i)})^T \\
V_k &= \text{Sign} \{M_k U_k\} = \text{Sign} \left\{ \sum_{i=1}^P [(w_k^{(i)} \zeta^{(i)})^T U_k] \xi^{(i)} \right\}
\end{aligned} \tag{5}$$

where $U_k = U$ for both RCAAM/fi and RCAAM/ad, $U_k = V_{k-1}$ for RCAAM/di, and $\mathbf{w}_k^{(i)} = (\mathbf{w}_{k-1}^{(i)} \ \zeta^i)^T U_{k-1}$ is the weighting matrix for the stored pattern ζ^i at time $\mathbf{k}$. The algorithm shows the RCAAM has dynamic stored patterns weighted by $\mathbf{w}_k^{(i)}$ at time $\mathbf{k}$, and each pattern weighting indicates the correlation accumulation through iterations between the network recurrent input states and the stored pattern itself. Therefore, the weightings of those stored patterns that are closer to the given input will become larger than the others, as long as the recurrent iterations increase. It is equivalent to say that by real-time adjustment of the individual pattern weightings of the dynamic memory, the recurrent outputs will gradually adapt to the closest stored pattern. Since the pattern weighting adjustments are parallel in each stored pattern vector direction at each iteration, there is no local minimum trap phenomenon in RCAAM. With its dynamical memory structure, the dynamic accumulation will eliminate the oscillation phenomenon which occurs in recurrent Hopfield nets. Although

it may have semi-stable states at which the network stays for a finite number of iterations its adaptive accumulation memory will function to leave those states, if the coding scheme used by the network allows it. Therefore, the network updatings won't be trapped in an oscillation cycle.

For RCAAM/fi and RCAAM/ad, the pattern weighting iteration, $\mathbf{w}_k^{(i)} = (\mathbf{w}_{k-1}^{(i)} \zeta^i)^T \mathbf{U}$, doesn't introduce the nonlinear threshold function **Sign**, and thus crosscorrelation interference terms among stored patterns won't become a problem. Therefore, the correlation accumulation becomes linear. This useful processing structure is possible for RCAAM/fi and RCAAM/ad, but not for the RCAM's. And this advantage still benefits from the dynamically accumulative memory structure. Compared to [39] which trains the associative memory off-line by Linear Programming or Sequential Multiple Training to guarantee the recall of stored patterns, the RCAAM learns on-line and adjusts the stored pattern weightings to converge the given input to the closest stored pattern. The RCAAM doesn't have predetermined order or a fixed memory matrix, so the network is quite flexible and applicable to any kind of distorted inputs. The network only takes a few recurrent iterations to recognize a slightly distorted stored pattern, and requires more recurrent iterations to discriminate a more ambiguous input.

Although the RCAAM/di has its output feedback to input, the previous correlations between recurrent input states and each stored pattern have been sequentially accumulated in memory. Then the accumulative correlation gains will acts as a reinforcement term to keep the next output associated with the last state and then the original input. It reminds the network of the passed converging trace. It is equivalent to saying that the accumulative

correlation gains in RCAAM/di plays the part of adaptive momentum which directs the network toward the converging trace with less confusion and keeps the converging accesses consistent. The momentum will be enhanced, if the stored patterns are close to it. The momentum will be adjusted and then the current one will gradually decay, if the stored patterns are not in accordance with it. The recurrent feedback gives the RCAAM/di great adaptive elasticity and enhances the target group idea. If patterns in a group are consistent and close to the input, then the recurrent output states will be apparently subjected to the group attraction. Therefore, the recurrent feedback makes the RCAAM/di still have a great adaptive elasticity, especially for ones with small initial orders, even under momentum guidance. Then, compared to the RCAAM/fi and the RCAAM/ad, the cooperation of adaptive elasticity and momentum in RCAAM/di will speed up its convergence for fairly or severely contaminated inputs.

This adaptive elasticity is helpful especially for partially inconsistent pattern groups which may have patterns closer to the other groups' than its own group's. For partially inconsistent pattern groups, an input may be a little closer to some special pattern belonging to other groups than its own group patterns. Under this circumstance, the adaptive elasticity will favor the group attraction, since multiple consistent attractions, under low correlation order, will prevail and then iteratively adapt the input to its own group before possible high order correlation manipulation. Typically, the RCAAM/di requires less space than the RCAAM/fi and the RCAAM/ad. Aspect sensitive radar target scattering usually has this partial inconsistency phenomena.

5.2.1 Implementations of the RCAAM

For easy implementation, the RCAAM can be further realized, then the initial memory

$\mathbf{M}_o$ is

$$\begin{aligned}
 M_o &= \sum_{i=1}^P \xi^{(i)} [(\zeta^{(i)T} U)^{init} \zeta^{(i)}]^T \\
 &= \sum_{i=1}^P \xi^{(i)} (w_o^{(i)} \zeta^{(i)})^T \\
 &= \xi \cdot \{ \text{Diag} [(\zeta^T \cdot U) \cdot \wedge (init)] \cdot \zeta^T \} \\
 &= \xi \cdot [\text{Diag} (W_o)] \cdot \zeta^T
 \end{aligned} \tag{6}$$

where $\zeta = [\zeta^{(1)} \zeta^{(2)} \dots \zeta^{(P)}]$, $\xi = [\xi^{(1)} \xi^{(2)} \dots \xi^{(P)}]$, $W_o \equiv [w_o^1 w_o^2 \dots w_o^P]$,

$\mathbf{A} \cdot \wedge^q \equiv [A_1^q A_2^q \dots A_P^q]^T$ and

$$\text{Diag} (A) \equiv \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & 0 & \dots \\ \dots & & & \\ 0 & \dots & 0 & A_P \end{bmatrix}$$

, if $\mathbf{A} = [A_1 A_2 \dots A_P]$. Then the network output is

$$\begin{aligned}
 V_o &= \text{Sign} \{M_o U\} = \text{Sign} \left\{ \sum_{i=1}^P [(w_o^{(i)} \zeta^{(i)})^T U] \xi^{(i)} \right\} \\
 &= \xi \cdot \{ [\text{Diag} (W_o)] \cdot (\zeta^T U) \}
 \end{aligned} \tag{7}$$

Thus the dynamic memory at recurrent iteration time k is

$$\begin{aligned}
 M_k &= \sum_{i=1}^P \xi^{(i)} \{ [(w_{k-1}^{(i)} \zeta^{(i)})^T U_{k-1}] \zeta^{(i)T} \} \\
 &= \xi \cdot [\text{Diag} (W_{k-1}) \cdot \text{Diag} (\zeta^T U_{k-1})] \cdot \zeta^T \\
 &= \sum_{i=1}^P \xi^{(i)} (w_k^{(i)} \zeta^{(i)})^T \\
 &= \xi \cdot [\text{Diag} (W_k)] \cdot \zeta^T
 \end{aligned} \tag{8}$$

where $\mathbf{W}_k \equiv [\mathbf{w}_k^1 \mathbf{w}_k^2 \dots \mathbf{w}_k^P]$ and $\mathbf{w}_k^{(i)} = \mathbf{w}_{k-1}^{(i)}(\zeta^{(i)T} \mathbf{U}_{k-1})$,

and the output is

$$\begin{aligned}
 \mathbf{V}_k &= \text{Sign} \{ \mathbf{M}_k \mathbf{U}_k \} = \text{Sign} \left\{ \sum_{i=1}^P [(\mathbf{w}_k^{(i)} \zeta^{(i)})^T \mathbf{U}_k] \xi^{(i)} \right\} \\
 &= \xi \cdot \{ [\text{Diag}(\mathbf{W}_k) \zeta^T] \cdot \mathbf{U}_k \} \\
 &= \xi \cdot \{ [\text{Diag}(\mathbf{W}_k)] \cdot (\zeta^T \mathbf{U}_k) \}
 \end{aligned} \tag{9}$$

Again, RCAAM/fi and RCAAM/ad have $\mathbf{U}_k = \mathbf{U}$, while RCAAM/di has $\mathbf{U}_k = \mathbf{V}_{k-1}$.

Therefore, the dynamic memories $\mathbf{M}_k$ and the evolution outputs $\mathbf{V}_k$ for RCAAM/fi and RCAAM/ad can be further simplified as

$$\begin{aligned}
 \mathbf{M}_k &= \sum_{i=1}^P \xi^{(i)} \{ [(\mathbf{w}_{k-1}^{(i)} \zeta^{(i)})^T \mathbf{U}] \zeta^{(i)T} \} \\
 &= \xi \cdot [\text{Diag}(\mathbf{W}_{k-1}) \cdot \text{Diag}(\zeta^T \mathbf{U})] \cdot \zeta^T \\
 &= \xi \cdot \{ \text{Diag}[(\zeta^T \mathbf{U}) \wedge (\text{init} + k)] \} \cdot \zeta^T \\
 \mathbf{V}_k &= \text{Sign} \{ \mathbf{M}_k \mathbf{U} \} = \text{Sign} \left\{ \sum_{i=1}^P [(\mathbf{w}_k^{(i)} \zeta^{(i)})^T \mathbf{U}] \xi^{(i)} \right\} \\
 &= \xi \cdot [\text{Diag}(\mathbf{W}_k) \cdot (\zeta^T \cdot \mathbf{U})] \\
 &= \xi \cdot [(\zeta^T \mathbf{U}) \wedge (\text{init} + k + 1)]
 \end{aligned} \tag{10}$$

For heteroassociative memory (ζ^i is not equal to ξ^i), we have 68 stored aspect response patterns belonging to four different targets, so ζ has 68 columns. Suppose the B52 is encoded by $[1 \ -1 \ -1 \ -1]$, the B58 by $[-1 \ 1 \ -1 \ -1]$, the F14 by $[-1 \ -1 \ 1 \ -1]$ and the TR1 by $[-1 \ -1 \ 1 \ 1]$, then $\xi^1 \sim \xi^{17} = [1 \ -1 \ -1 \ -1]^T$, $\xi^{18} \sim \xi^{34} = [-1 \ 1 \ -1 \ -1]^T$, $\xi^{35} \sim \xi^{51} = [-1 \ -1 \ 1 \ -1]^T$ and $\xi^{52} \sim \xi^{68} = [-1 \ -1 \ 1 \ 1]^T$. From the above realization algorithms, the network only requires storing memories for $P=68$ dynamically weighted stored patterns, 4 target group codes, one current input pattern and one output. Since RCAAM has a dynamic memory structure, there may be

semi-stable states at which the network stays until the correlation accumulation is high enough to escape the temporary spurious state and move toward other states. We define the stability criterion **sc** by $V_k = V_{k-1} = \dots = V_{k-sc}$. Therefore, we regard the state V_k , which satisfies $V_k = V_{k-1} = \dots = V_{k-sc}$, as the network discrimination pattern, if the stability criterion **sc** is adopted. This indicates that the RCAAM has a flexible decision strategy for an ambiguous input, allowing us to either leave it unknown or force it to one of the stored patterns. To leave those spurious states as unknown, the stability criterion can be set to a low value (eg., 1 or 2). The stability criterion can be set to a high value (eg., larger than 2), if a definite discrimination is required. This allows convergence to one of the stored patterns. The RCAAM requires only the minimum computation scale space which the RCAM can possibly offer to discriminate an arbitrarily given input. Thus the RCAAM not only needs less processing space than RCAM, but will perform the same or better.

5.3 Performance of the Recurrent Correlation Accumulation Adaptive Memory with dynamic input (RCAAM/di)

In this section, we simulate RCAAM/di with several different initial orders and stability criterions. First we simulate the RCAAM/di performances with respect to stability criterion for three different initial orders, 2, 3 and 6. Then the performances of RCAAM/di with respect to initial order are simulated for three different stability criterions, 2, 5 and 10. Finally, RCAAM/di performances for groups with different average orders are compared. If a RCAAM/di has an initial order **p** and a stability criterion **q**, then we abbreviate this RCAAM/di by **dipq**. The order required for network stability criterion (or discrimination) is $p+k$, if the network takes k recurrent iterations to satisfy its stability criterion. Thus the

average order is calculated by summing the orders required for 10 SNR's and 4 targets, and then taking the average value. There are four average order groups simulated, the first group has an average order between 9 and 10, i.e. $9 < p+k < 10$, the second has $10 < p+k < 11$, the third has $11 < p+k < 12$, and the fourth group has $13 < p+k < 14$. These average order groups are presented here to help us design and implement an RCAAM/di under a limited computation space and specified discrimination capability.

5.3.1 RCAAM/di performances with respect to stability criterion effect

Figure 5.1 shows unrecognized discrimination in percent of RCAAM/di with an initial order of 2 (i.e. d_i2) under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns, while **Figure 5.2** presents its correct pattern discrimination in (a) and wrong pattern discrimination in (b). The definition of pattern discrimination is same as Chapter 4. **Figure 5.3** presents correct target discrimination in (a) and wrong target discrimination in (b). We can see the unrecognized discrimination is decreasing when the stability criterion is increasing. This proves the stability criterion is really effective for RCAAM/di to further converge the unknowns to stored patterns. This is an important advantage over the RCAM networks, such as HCAM and ECAM, which reach their final output state when the same output occurs twice. By using this characteristic, we can observe the contamination of an input signal and then decide whether to accept the network output or disregard it. The higher the stability criterion the network takes, the harder it is to discriminate an input signal. The simulation results show the network requires more iterations to converge unknowns to final stable states under 40 dB than 14 or 0 dB. The network has about 7% patterns unrecognized when a stability criterion of 2 is adopted, and the unknown reduces to 0.5% when a stability criterion

of 10 is used. This means that there are at least (7-0.5)% of contaminated stored patterns staying at semi-stable states when a stability criterion of 2 is used to judge the final state. And these semi-stable states will converge to stored patterns when a stability criterion of 10 is adopted. The simulation results are consistent with the noise contaminations. For example, the noisy stored patterns with a SNR of 0 dB require a stability criterion of 9 to converge unknowns to 0.5%, while the ones with 14 dB need a stability criterion of 4, and the ones with 40 dB only need a stability criterion of 2 to converge all unknowns to stored patterns. From the correct and wrong discrimination performances, both discrimination rates increase when the stability criterion increases. It is expected that the unknowns converged will contribute either to correct or wrong stored patterns. Their relative increasing rates can help us to evaluate how well the RCAAM/di2 (i.e. with an initial order of 2) can benefit from setting a higher stability criterion. Suppose a RCAAM/di2 increases its stability criterion from 2 to 10 under an SNR of 0 dB. Then, from **Figure 5.3**, the correct target discrimination rate increases about 5%, while the wrong one only increases about 1%. This means that the network RCAAM/di2 will gain a correct to wrong target discrimination increase of 5:1 when 6% of unknowns is converged by increasing its stability criterion. This is a good trade. Therefore, setting a higher stability criterion for RCAAM/di2 has a positive and effective improvement on target discrimination rates, especially for higher noise contamination.

Comparing **Figure 5.2** to **Figure 5.3**, we see that the correct pattern discrimination is about 21% less than the correct target discrimination for RCAAM/di2 with an **sc** (stability criterion) of 2 under 0 dB, while the correct pattern discrimination is 24% less than the correct target discrimination for an **sc** of 10. This indicates that about 21% of the

contaminated stored patterns with a SNR of 0 dB will be converged and then recognized as another stored pattern which belongs to the same target as the input one, but resides at a different aspect angle. The stored patterns from the same target thus have different attractions to their own noisy patterns. Some aspect patterns are more attractive while others are less so. The attractive patterns have a larger convergent or attractive basin, within which any state will be converged to the stored pattern, than the less attractive ones. This phenomenon may be caused by the nature of radar target scattering (associated with target geometrical shape) and the coding scheme. This is an applicable characteristic, since our purpose is to discriminate one target from another instead of one aspect angle from another. By using this target group idea, the cascade networks are proposed and are more efficient as expected. In **Figure 5.3**, the correct discrimination increases about 6% under 0 dB by increasing **sc** from 2 to 10. Since the correct pattern discrimination in **Figure 5.2** only increase 3% from increasing **sc**, another 3% increasing in correct target discrimination are contributed from the wrong pattern discrimination. Therefore, this 3% noisy stored patterns with a SNR of 0 dB are wrongly recognized in their aspect angles but correctly discriminated on targets.

Figure 5.4 shows the unrecognized discrimination of RCAAM/di2 vs. stability criterion for unstored pattern inputs, while **Figure 5.5** presents the correct target discrimination in (a) and wrong target discrimination in (b). The unrecognized discrimination rates decreases like the stored pattern inputs case when the stability criterion increases. Both the correct and wrong target discrimination rates rise when the **sc** goes high. A correct to wrong target discrimination increasing ratio of 7:1 can be found for 0 dB from **Figure 5.5**, when the **sc** changes from 2 to 10. Only one eight of unknowns, which can be further

converged from increasing sc , will go to wrong targets. Therefore, the existence of stable criteria also improves the target discrimination performances of RCAAM/di2 for unstored pattern inputs. By comparing **Figure 5.3** to **Figure 5.5**, the correct target discrimination for contaminated stored pattern inputs under 0 dB SNR is less than the correct target discrimination for contaminated unstored pattern inputs, but it doesn't occur for light contamination cases. It should be remembered from previous chapters every unstored test pattern is resident at the middle of two adjacent stored patterns in our simulations. A severely contaminated stored pattern may shift far away from the true stored pattern, while a severely contaminated unstored pattern might have a chance to shift toward either adjacent stored pattern. Therefore, a heavily distorted unstored pattern may get closer to a true stored pattern than a heavily distorted stored pattern. This seemingly unreasonable phenomenon results from the deterministic allocation arrangements around stored and unstored test patterns. It is not expected in practical situation. By definition, a light distorted pattern should be closer to its own clean pattern than the others. Therefore the strange results described above will not happen to light distortion cases.

Figure 5.6 to **Figure 5.10** present another performance set of the RCAAM/di with an initial order of 3 (RCAAM/di3) vs. stability criterion. By comparing the RCAAM/di3 with RCAAM/di2, the unrecognized discriminations of RCAAM/di3 for small criterion are smaller than the ones of RCAAM/di2. The pattern and target discrimination performances of RCAAM/di3 along with stability criterion are flatter than the RCAAM/di2's. For example, under 0 dB SNR, the RCAAM/di3 has an increase of 4.5% in correct target discrimination corresponding to the change of sc from 2 to 10, while the RCAAM/di2 has an increase of

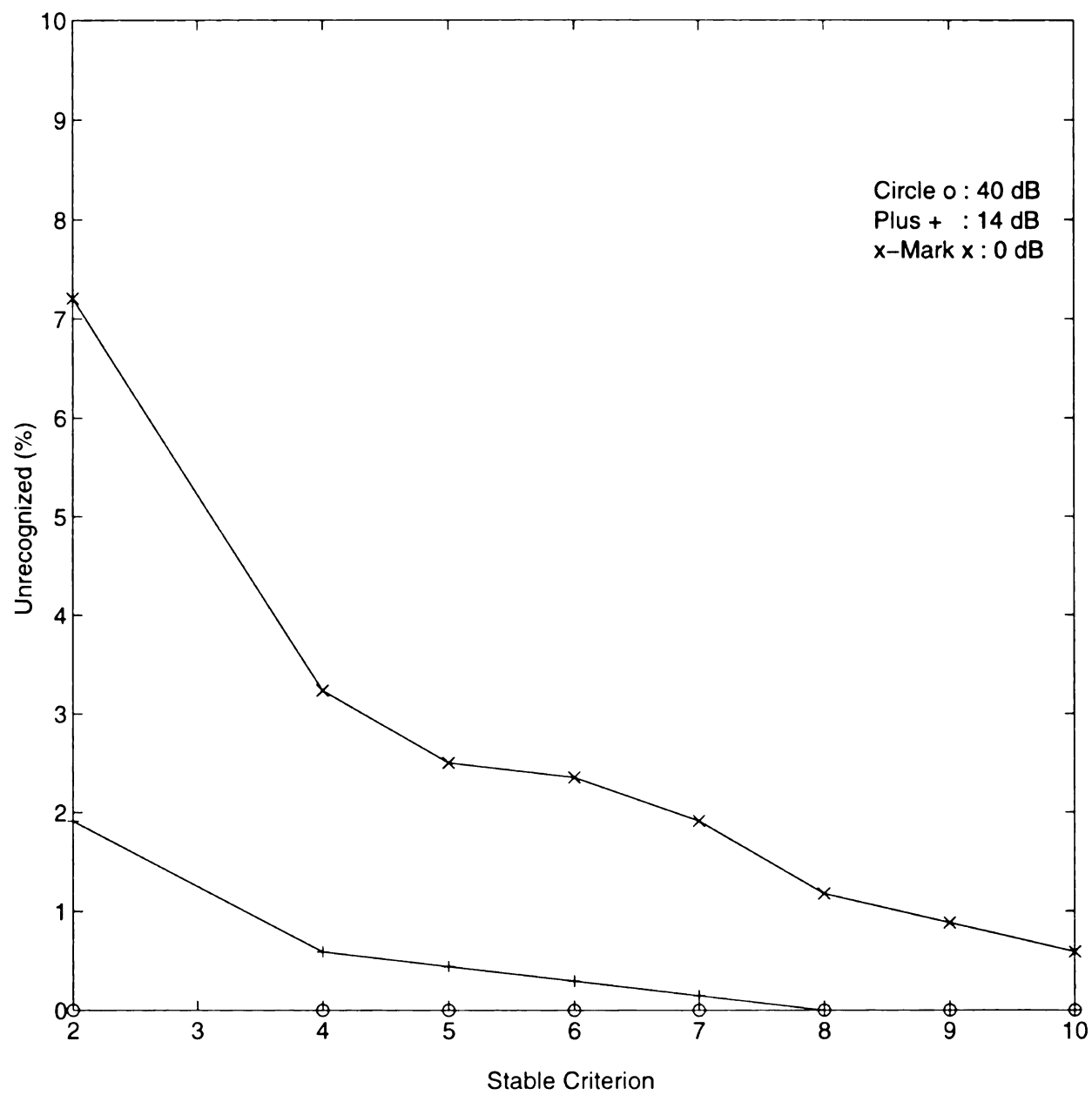


Figure 5.1 Unrecognized discrimination of the RCAAM/di with an initial order of 2 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns.

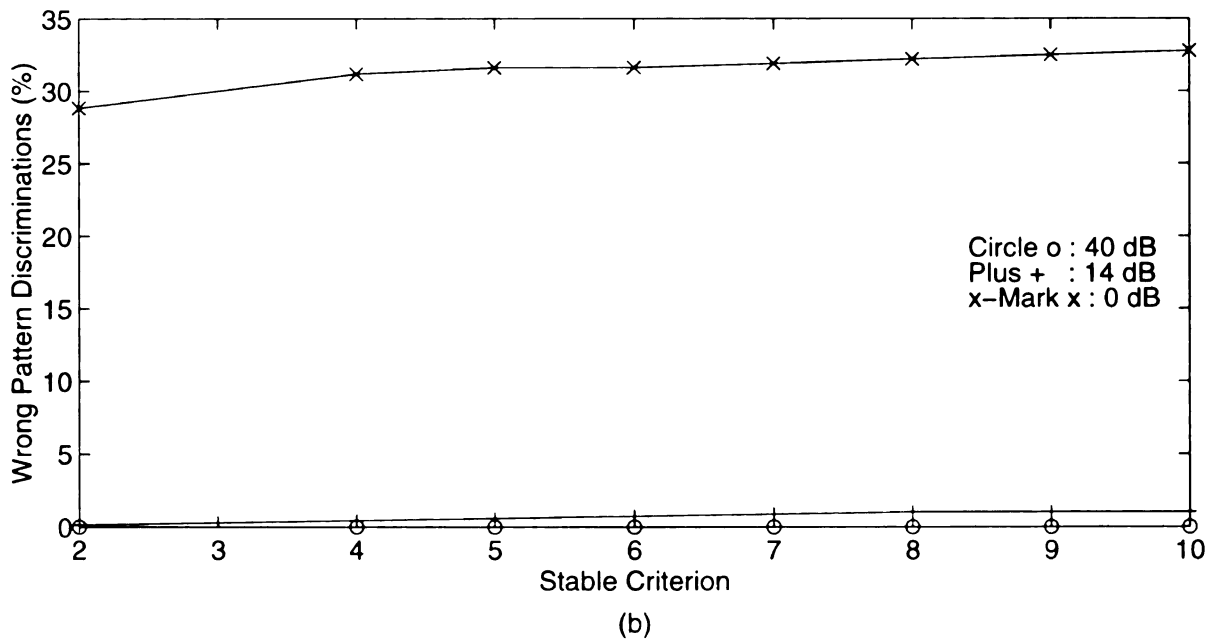
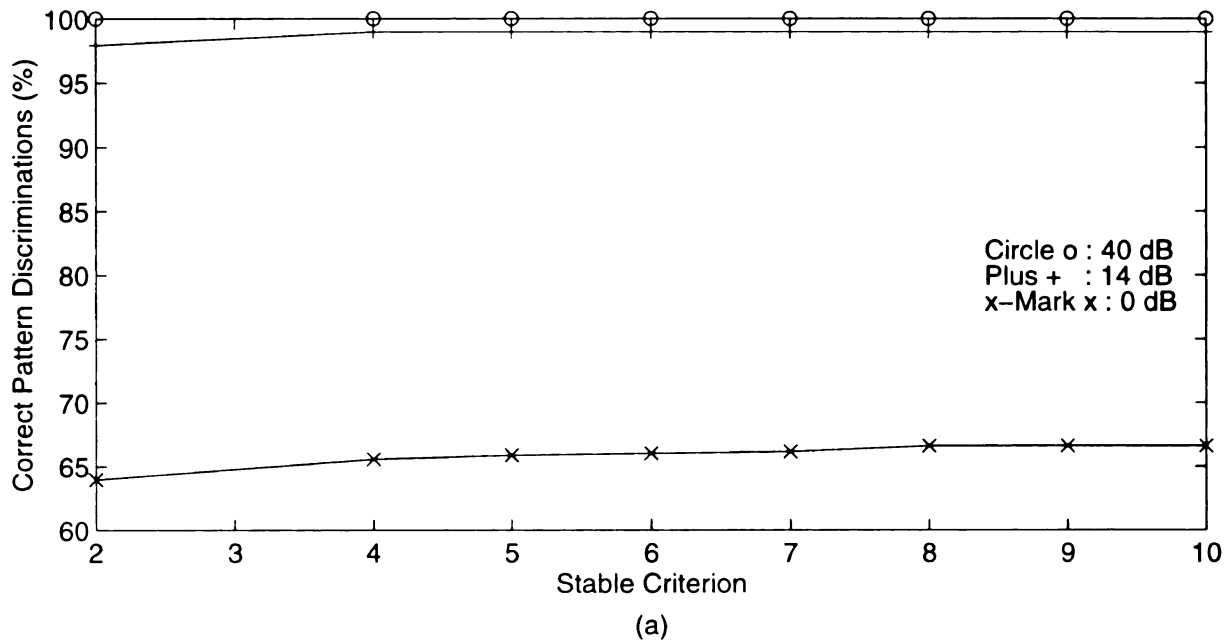


Figure 5.2 Correct and wrong pattern discrimination performances of the RCAAM/di with an initial order of 2 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.

- (a) Correct pattern discrimination performances vs. stability criterion.
 (b) Wrong pattern discrimination performances vs. stability criterion.

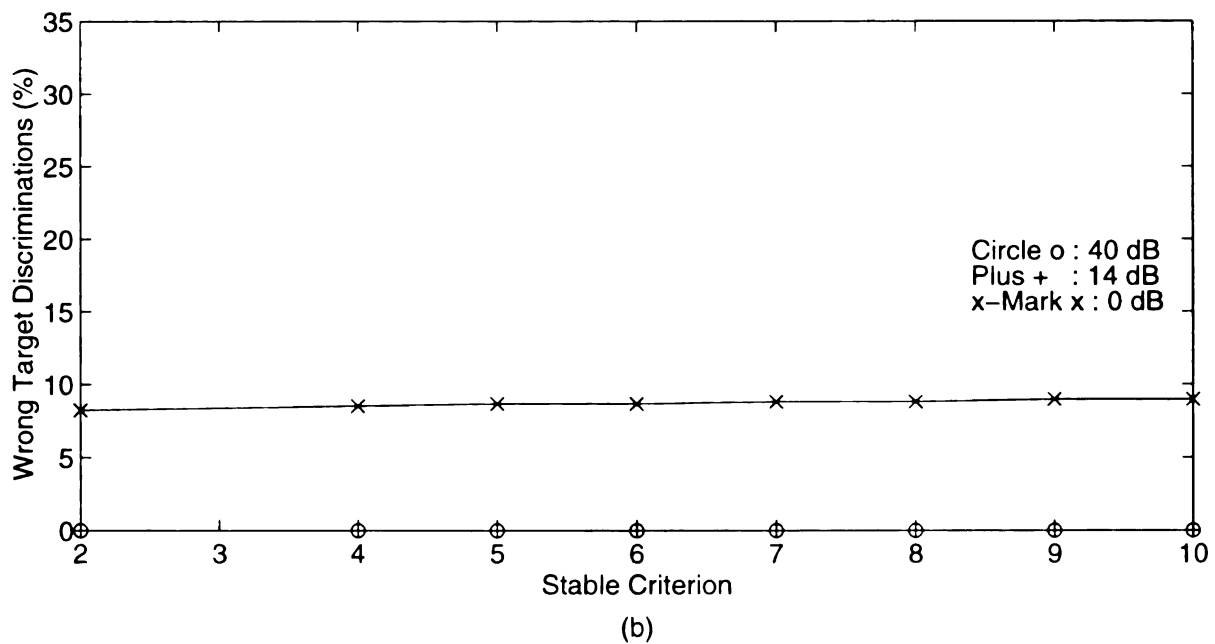
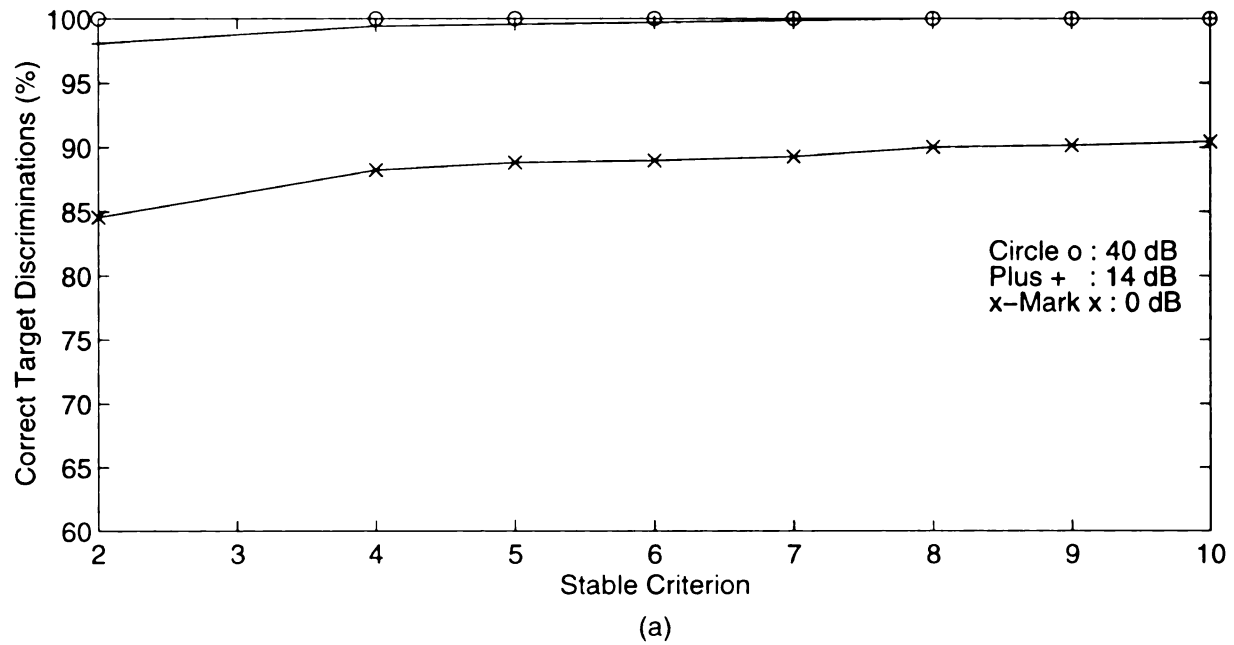


Figure 5.3 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 2 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs. stability criterion.

(b) Wrong target discrimination performances vs. stability criterion.

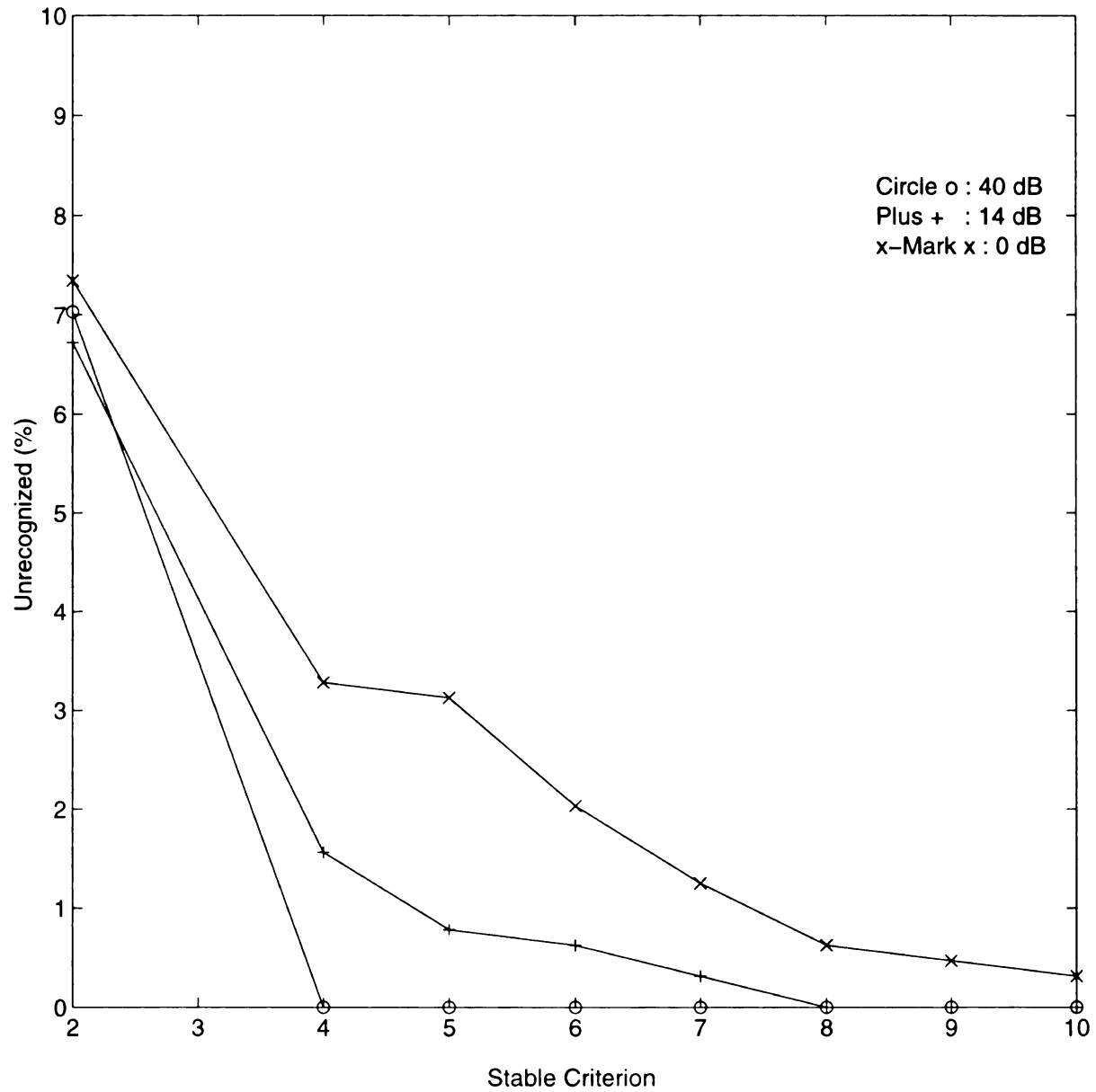


Figure 5.4 Unrecognized discrimination of the RCAAM/di with an initial order of 2 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns.

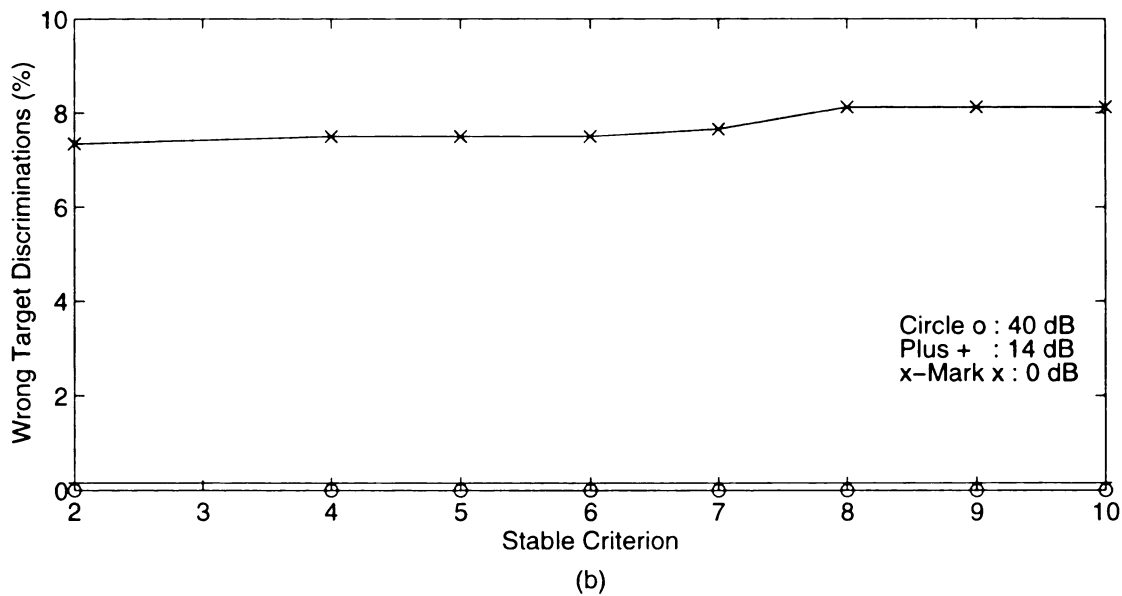
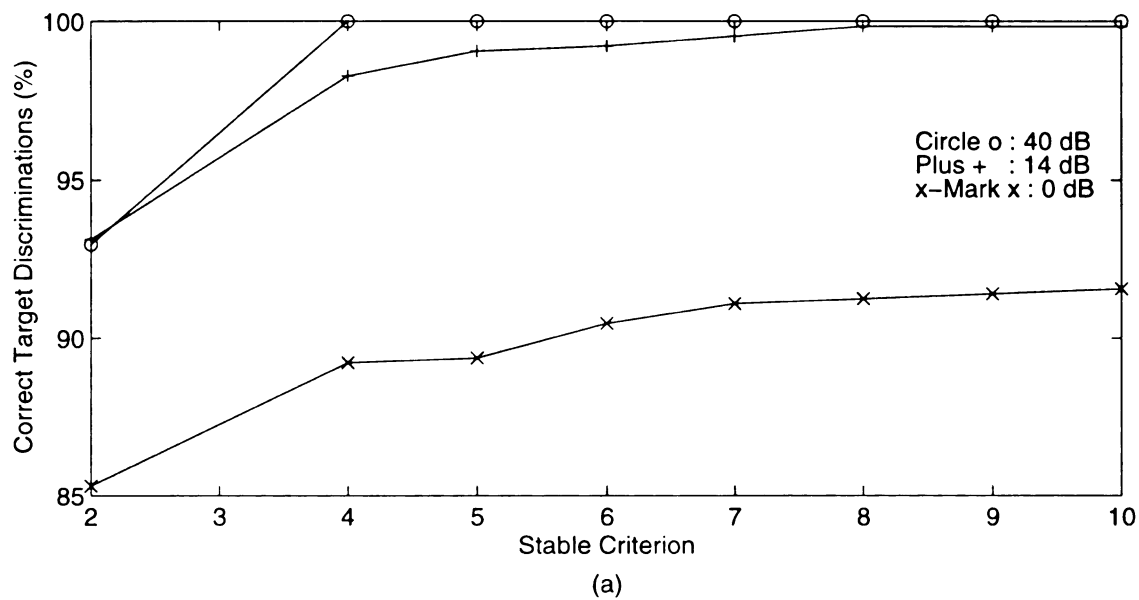


Figure 5.5 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 2 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs stability criterion.

(b) Wrong target discrimination performances vs. stability criterion.

6.5%. The wrong discriminations of RCAAM/di3 nearly haven't changed while its correct target discriminations increase along with **sc**. To further analyze and derive a generalized rule by which the RCAAM/di with different initial orders will be affected with **sc**, a RCAAM/di with an initial order of 6 (di6) is also simulated. **Figure 5.11** to **Figure 5.15** shows the simulation performances of the RCAAM/di with an initial order of 6 vs. **sc**.

Comparing these three RCAAM/di's (di2, di3 and di6) performances, two interesting characteristics can be found.

- (1). The higher the initial order, the flatter the performance curve.
- (2). The flatter the performance curve is, the higher the **sc** needed for converging unstored pattern inputs.

Rule 1 can be interpreted that higher initial order RCAAM/di has less affect with **sc** than the lower initial order one. And rule 2 can be described as the RCAAM/di with an higher initial order needs larger **sc** to converge all unstored pattern inputs. From comparing the unrecognized performances of three RCAAM/di's for both stored and unstored pattern inputs, the di2 has the deepest slopes , the di3 has fair ones, and the di6 has the smoothest ones.

Characteristic 2 can be considered as an interpretation of network adaptive elasticity. For a RCAAM/di with a large initial order, the successive iterations after the first output will have less convergent ability since the high order correlation spanned a large space and then separated a state from each other with a larger distance. This means the first iteration output of the network will be at some state which is far away from any other state. If the first output is at some ambiguous state, then it is hard to turn this deeply trapped state back to some stored state. Therefore, we can say the deeply trapped state has dramatically lost its

convergent elasticity and has a great chance to stick around those ambiguous states for many iterations. Consider the case that a RCAAM/di has stored pattern inputs with a SNR of 14 dB, i.e. the star mark '+'. Then the di2 needs an **sc** of 8 to converge unknowns in **Figure 5.4**, the di3 needs a **sc** of 7 in **Figure 5.9**, and the di6 only requires a **sc** of 3 in **Figure 5.14**. The reverse property will occur by considering the performances with unstored pattern inputs. With an SNR of 40 dB, the di2 requires only an **sc** of 4 to converge unknowns, the di3 needs an **sc** of 6, and the di6 needs an **sc** of 8 to complete the task. Under 14 dB, the di2 requires an **sc** of 8 to converge unknowns, the di3 needs an **sc** of 10, and the di6 needs an **sc** at least as large as 10. There are two observations here. For light contamination stored inputs, the RCAAM/di with a high initial order requires a smaller **sc** to converge unknowns, while a low initial order RCAAM/di needs a larger **sc** to complete the task. For unstored pattern inputs, the RCAAM/di with a high initial order then needs a larger **sc** to converge unknowns if possible, while a low initial order RCAAM/di requires a smaller **sc** to finish the job.

To explain the above seemingly inconsistent simulation performances, let's consider two different factors that affect the convergent speed along with **sc**. First consider an input close to one of the stored patterns. The high order correlation will bring the input much closer to its true stored pattern. Here the high initial order thus speeds up the convergence. This factor explains why the di6 converges light contaminated stored inputs by using a small **sc**, although it has the flattest performance curve. Next consider an input that is not apparently close to one of the stored patterns or ambiguous to stored patterns. This may be caused by a linear combination of a couple of stored patterns or a pattern which is close to one specific stored pattern but a little less close to several stored patterns belonging to a same target.

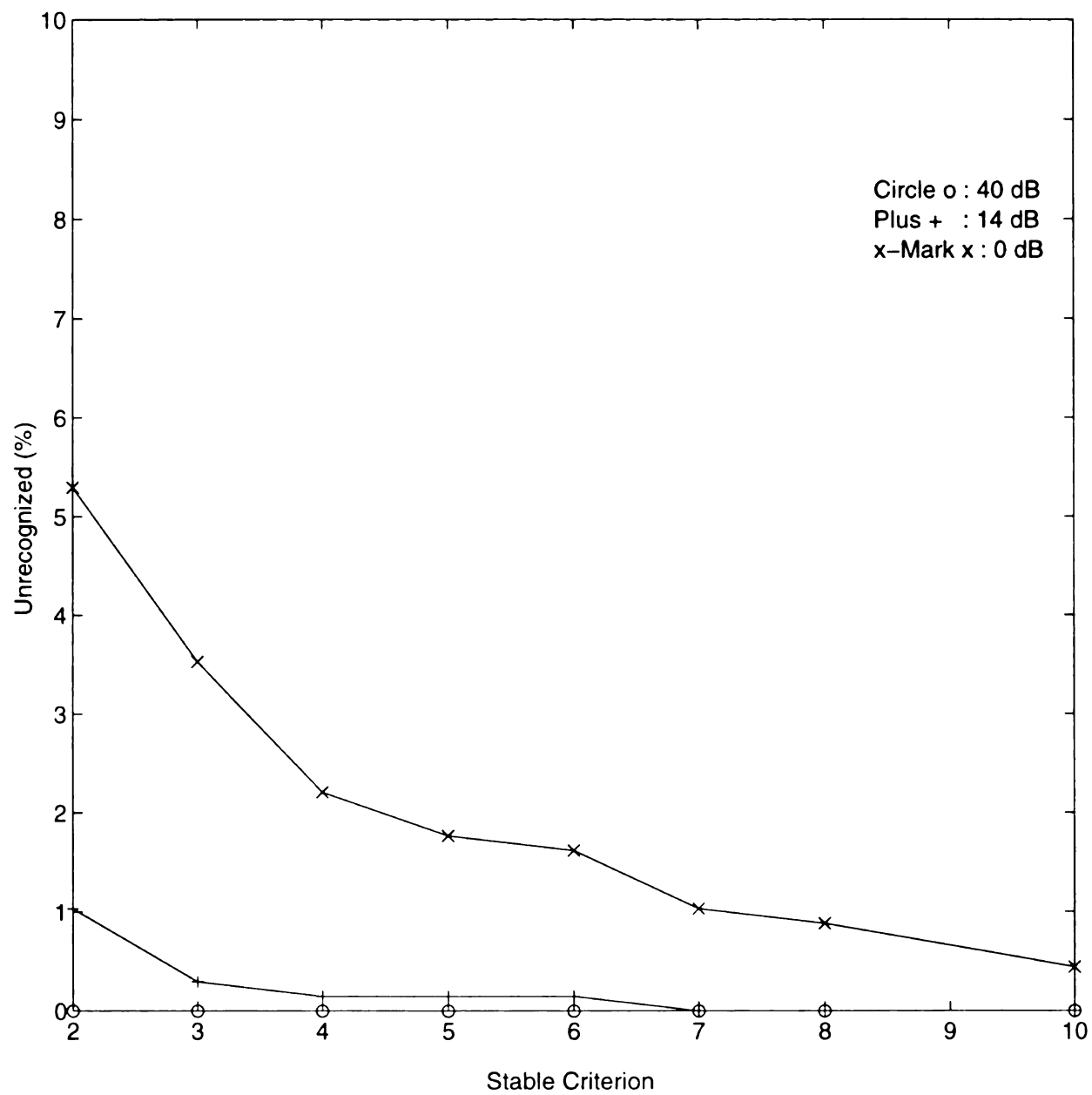


Figure 5.6 Unrecognized discrimination of the RCAAM/di with an initial order of 3 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns.

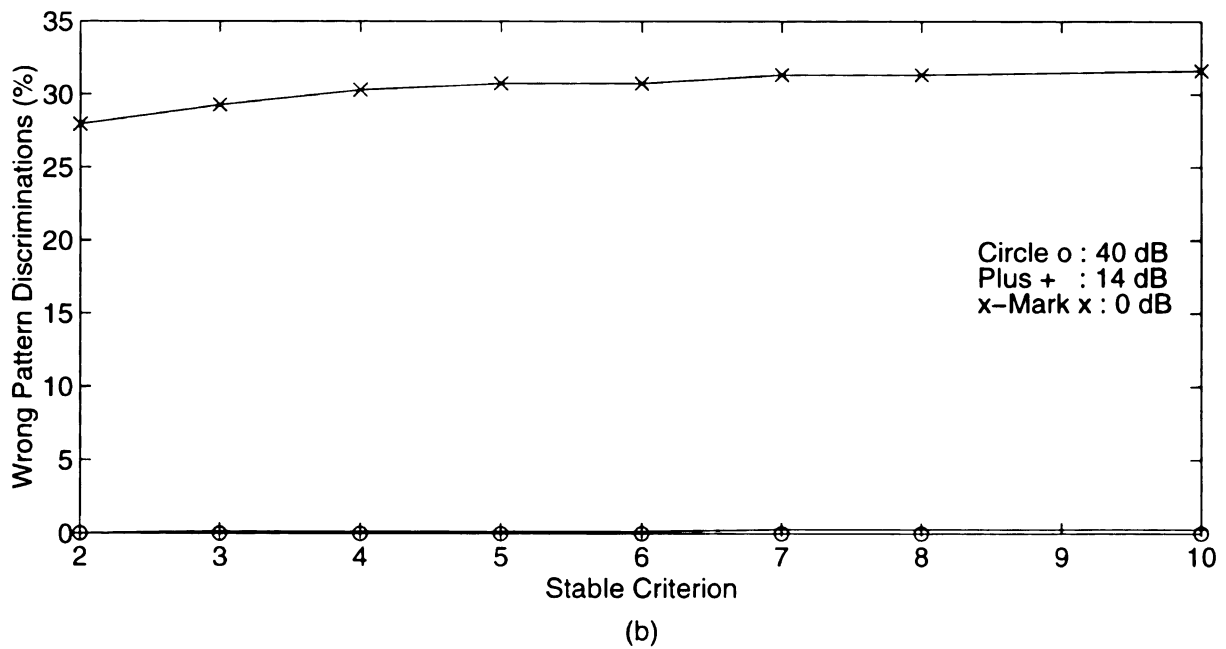
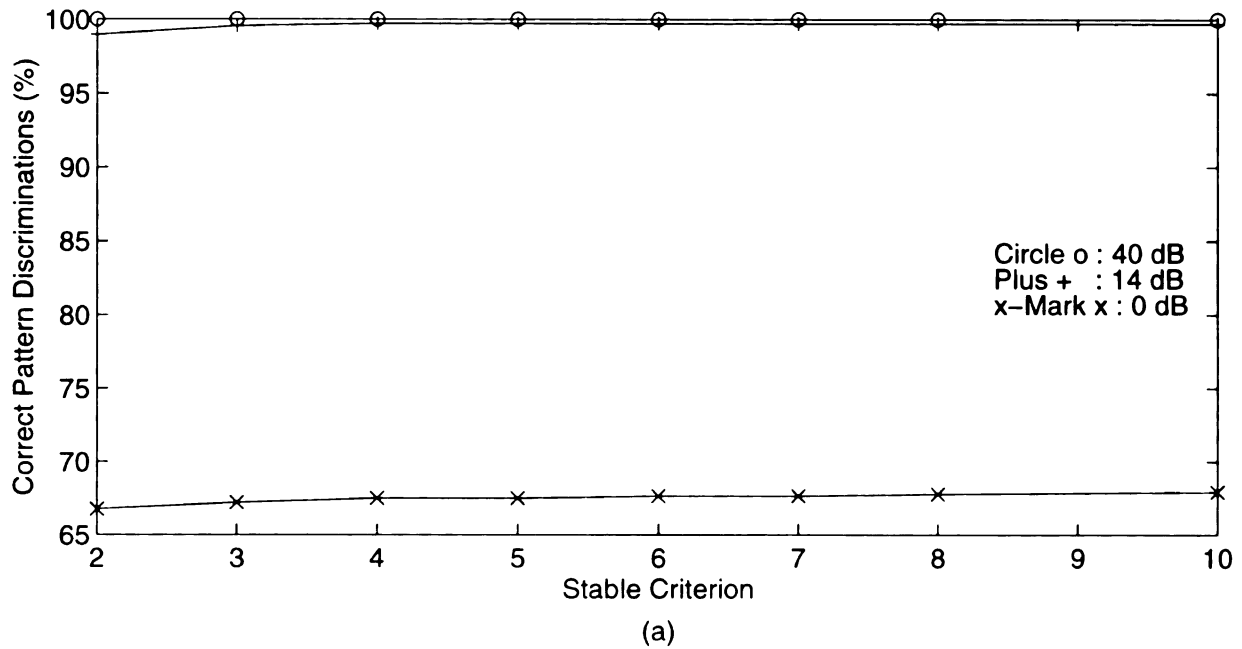


Figure 5.7 Correct and wrong pattern discrimination performances of the RCAAM/di with an initial order of 3 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.

(a) Correct pattern discrimination performances vs. stability criterion.

(b) Wrong pattern discrimination performances vs. stability criterion.

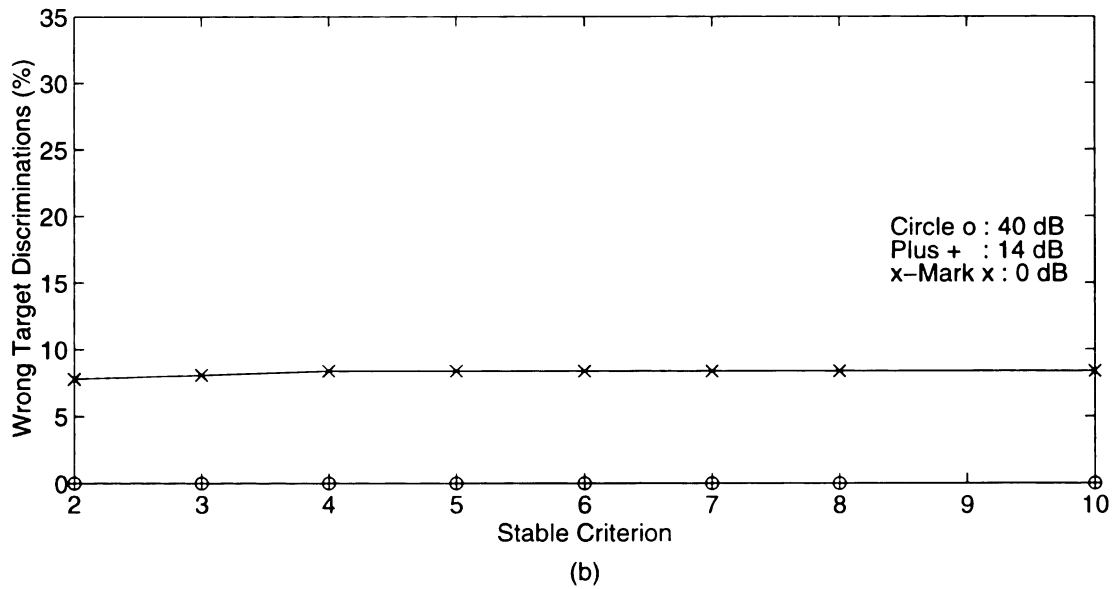
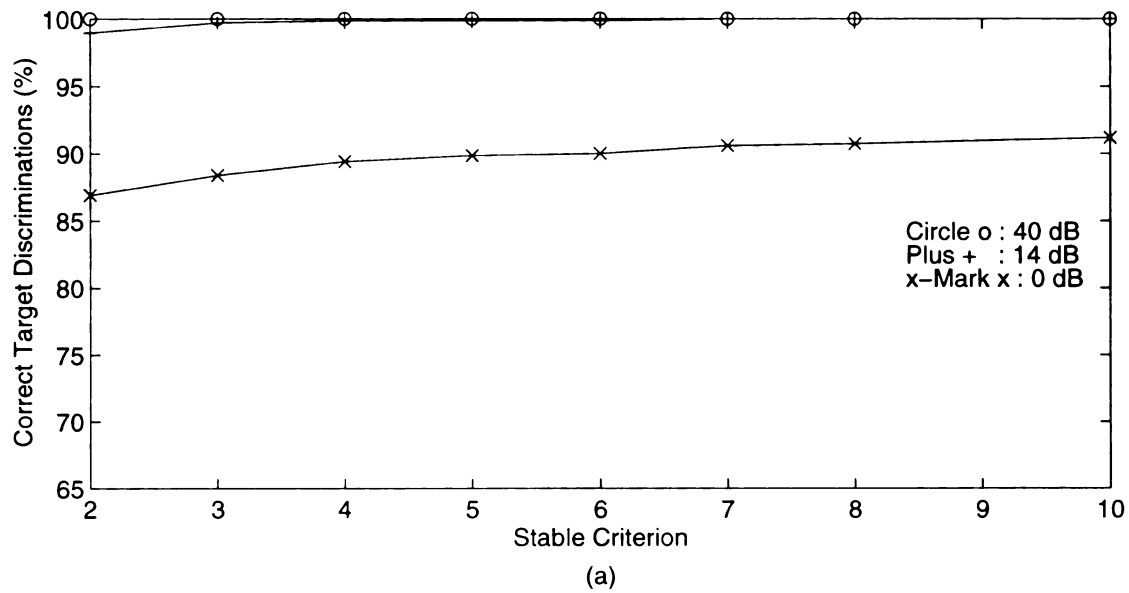


Figure 5.8 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 3 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs stability criterion.

(b) Wrong target discrimination performances vs. stability criterion.

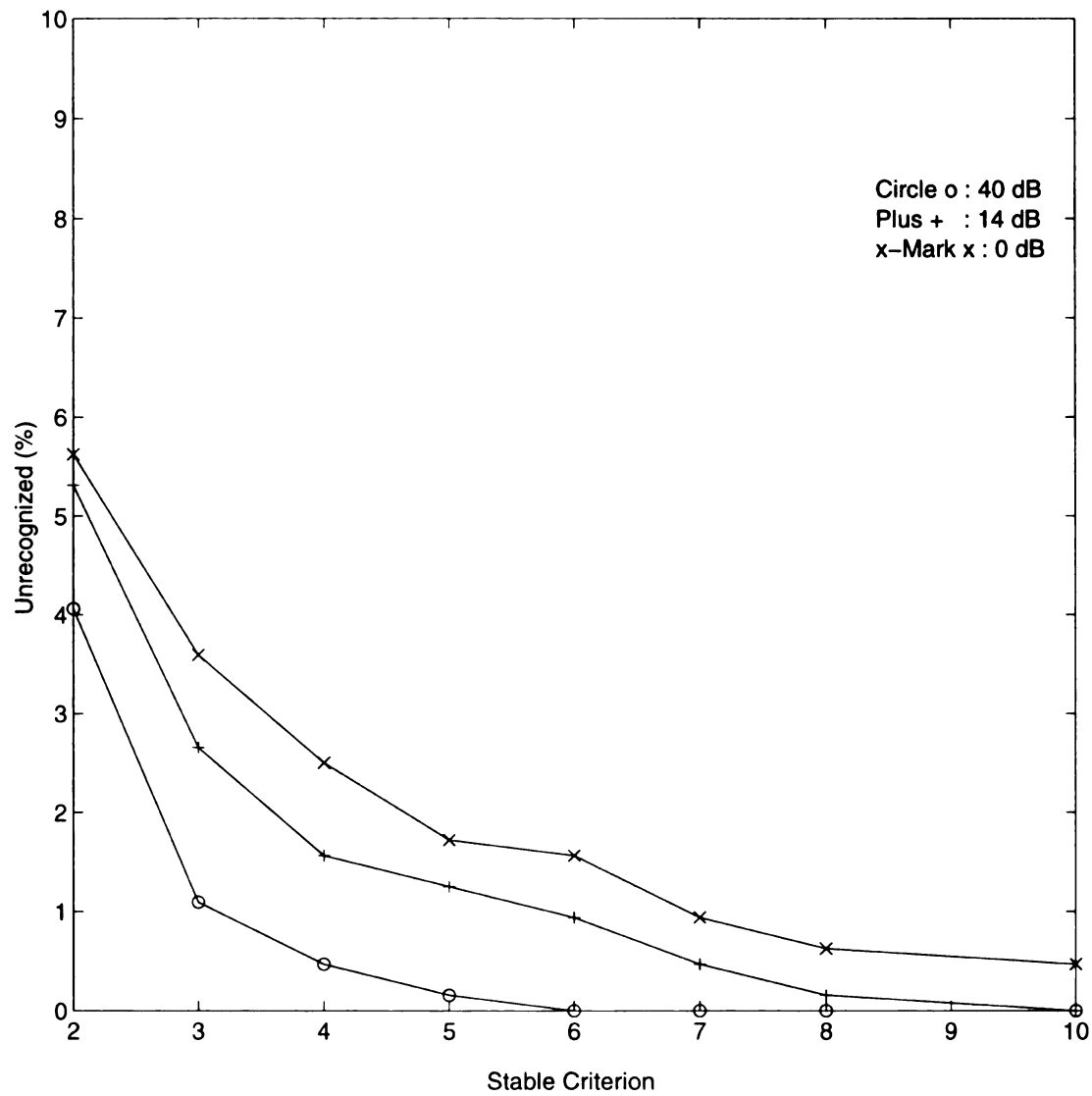


Figure 5.9 Unrecognized discrimination of the RCAAM/di with an initial order of 3 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns.

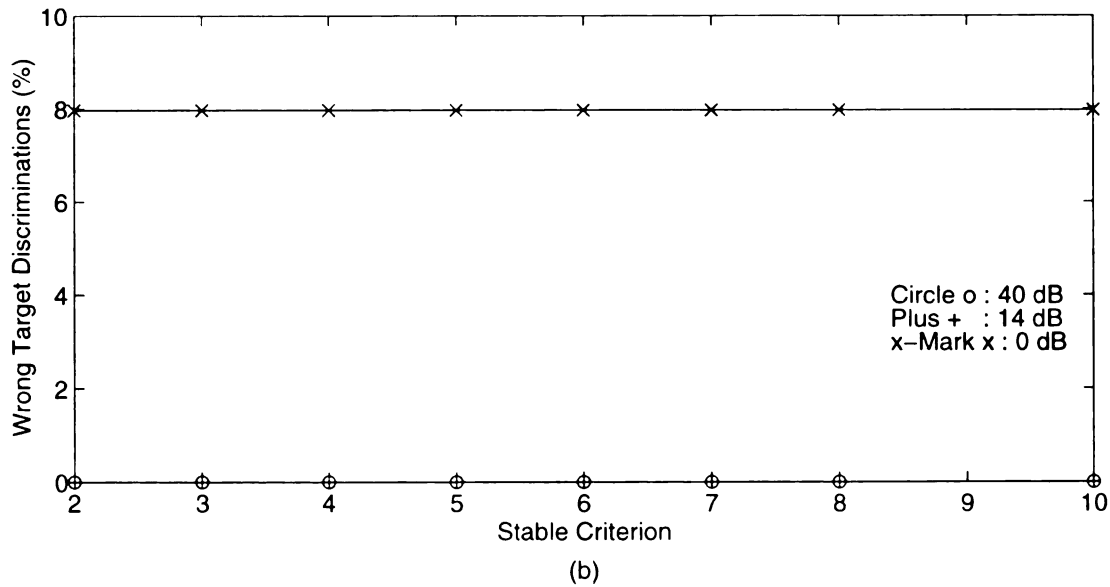
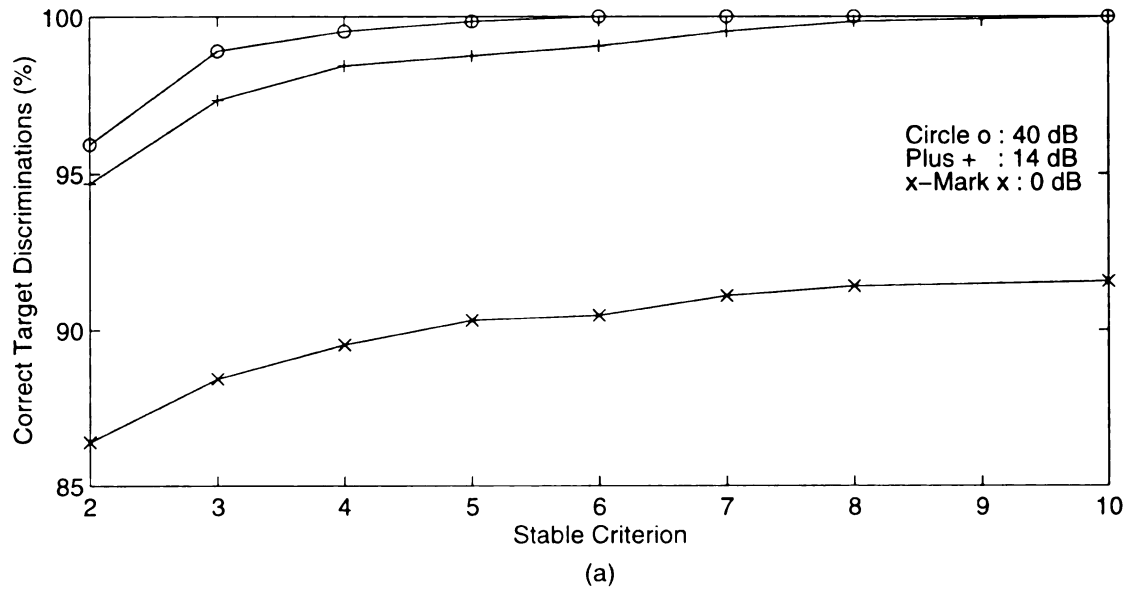


Figure 5.10 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 3 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs. stability criterion.

(b) Wrong target discrimination performances vs. stability criterion.

Then the high order correlation may deeply force the input into a spurious state and trap it there for many iterations. The network will thus need several iterations to compensate the initial overemphasis and then gradually turn it toward a stable state if possible. Here the high initial order thus slows down the convergence, if possible. This factor can explain why the di6 needs a higher α to converge the unstored pattern inputs, if possible, than the di3 and di2. Since the RCAAM/di has its output feedback to network input, the current output state depends on the previous output state. Thus RCAAM/di is a causal system and its convergence is successively improved iteration by iteration. RCAAM/di also accumulates previous correlation gains in its memory. As the recurrent operation goes on, the converging continues but the adaptive elasticity is gradually declining. Since a high initial order acts similarly but not exactly as a long term iteration accumulation, a high initial order RCAAM/di will have low adaptive elasticity. Therefore, a RCAAM/di with a small initial order has high adaptive elasticity to converge unstored pattern inputs to stable states.

We know the radar target responses become less consistent within some aspect ranges. Therefore, if an input is a little closer to one specific stored pattern than its own target patterns, then a high initial order will force the input to initially approach the wrong pattern. The strong momentum introduced by high initial order is then hardly released by future recurrent feedbacks, since the updated input state might have become much closer to the inconsistent pattern. We mentioned that the recurrent feedback of the RCAAM/di will enhance the target group idea. For inconsistent aspect patterns, networks with small initial orders may have better discriminations than ones with high initial orders, since multiple consistent attractions from group patterns, under low correlation order, will prevail and then

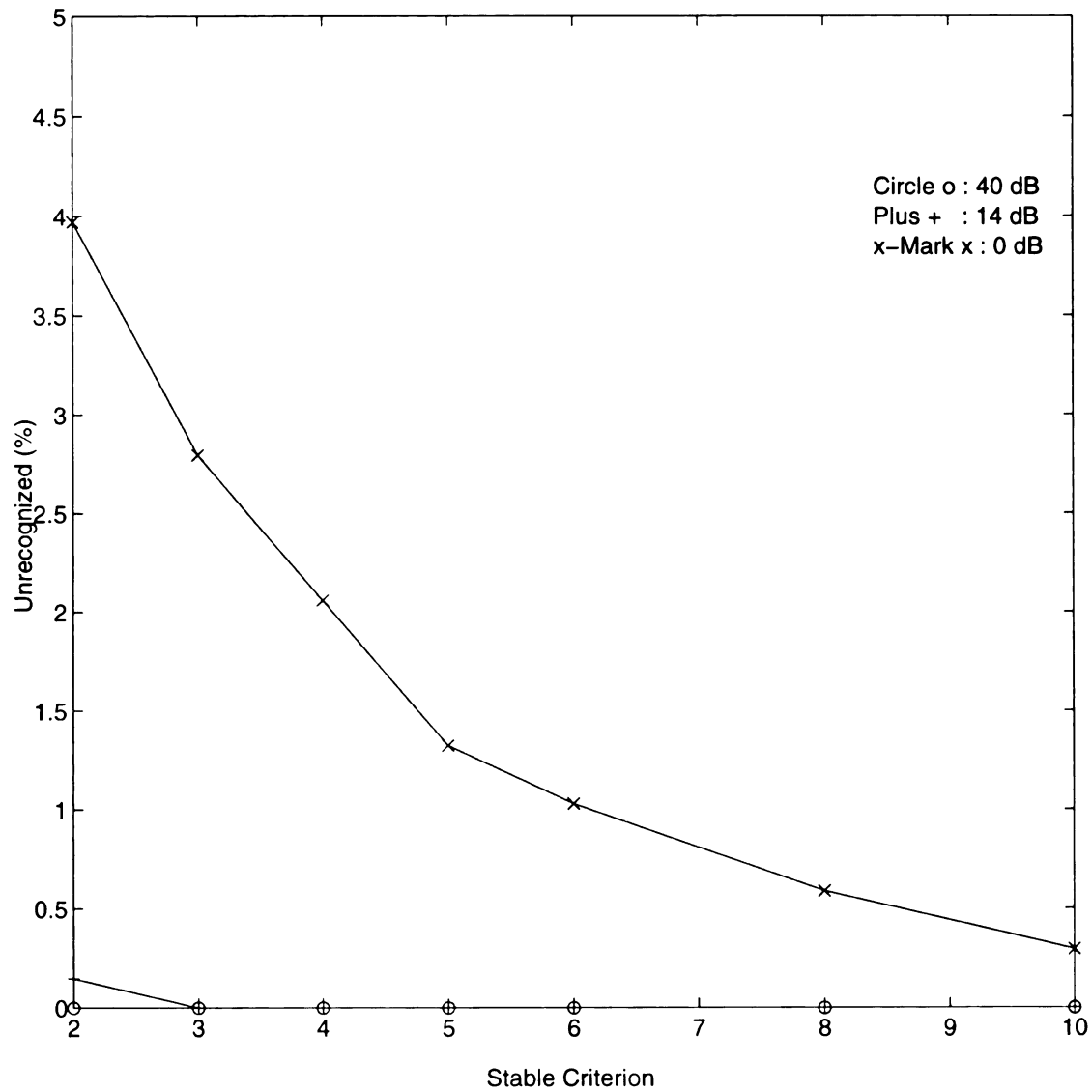


Figure 5.11 Unrecognized discrimination of the RCAAM/di with an initial order of 6 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns.

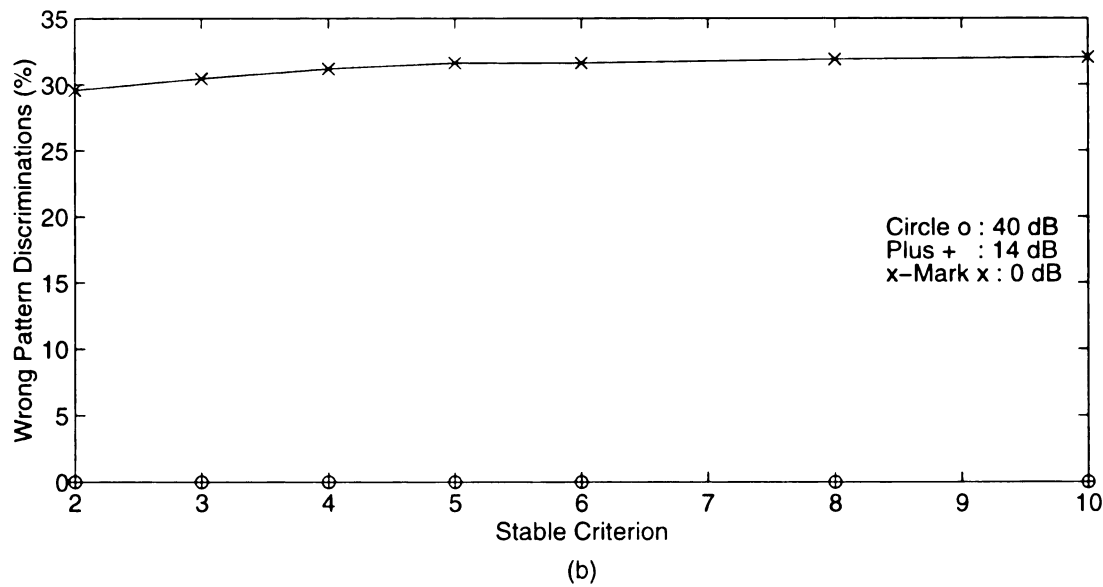
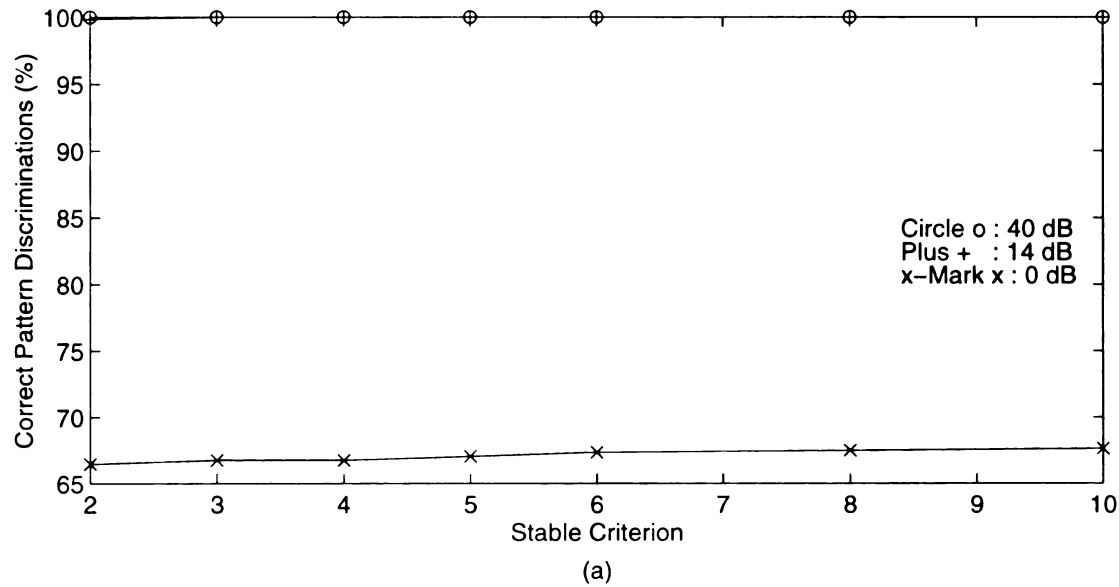


Figure 5.12 Correct and wrong pattern discrimination performances of the RCAAM/di with an initial order of 6 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.

(a) Correct pattern discrimination performances vs. stability criterion.

(b) Wrong pattern discrimination performances vs. stability criterion.

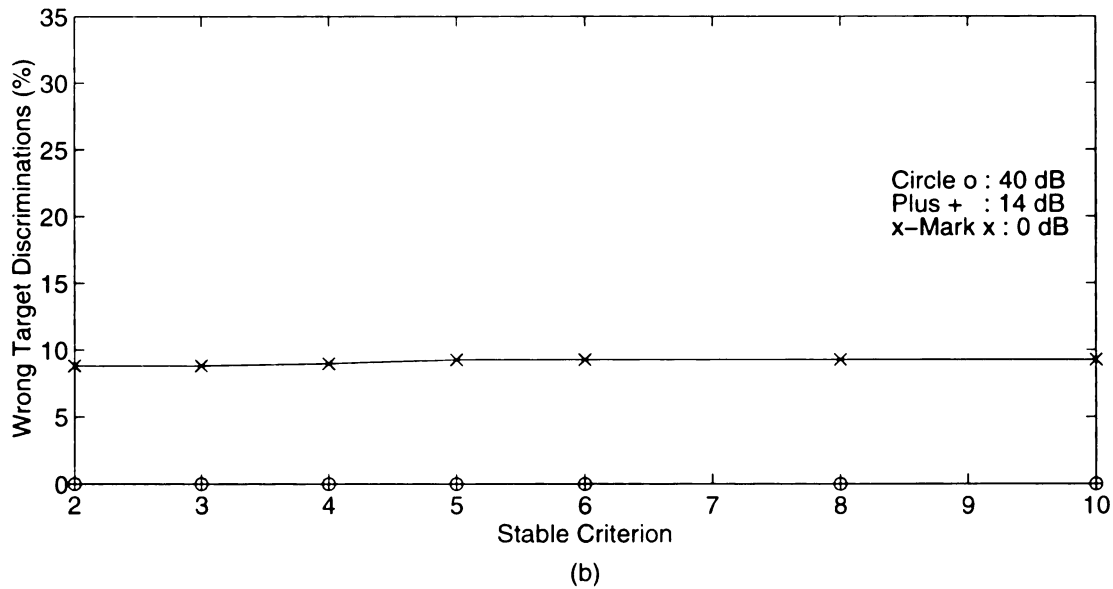
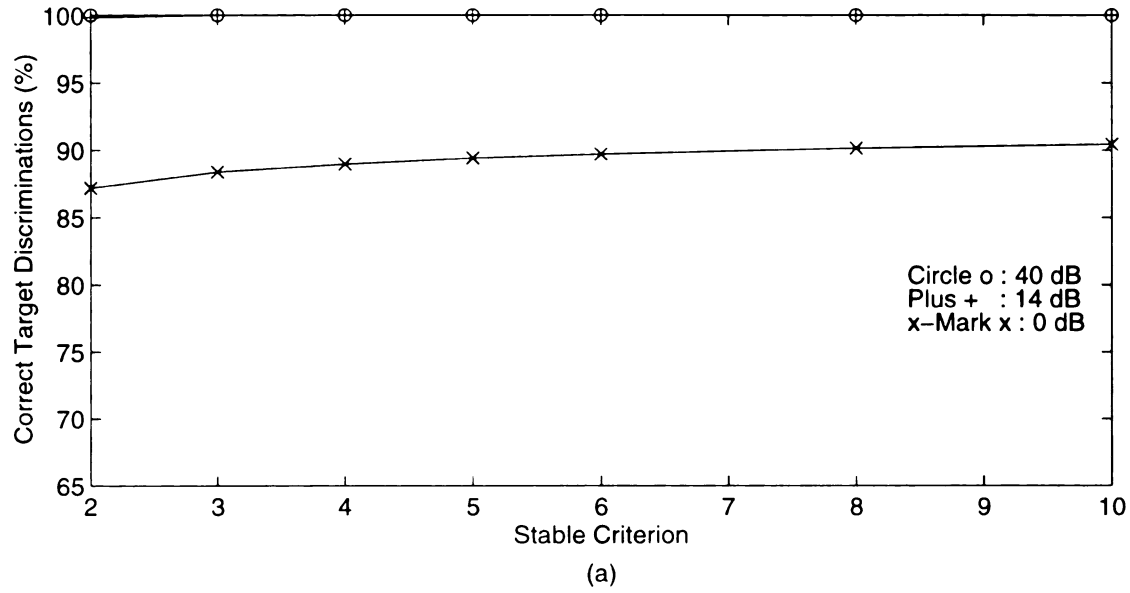


Figure 5.13 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 6 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs stability criterion.

(b) Wrong target discrimination performances vs. stability criterion.

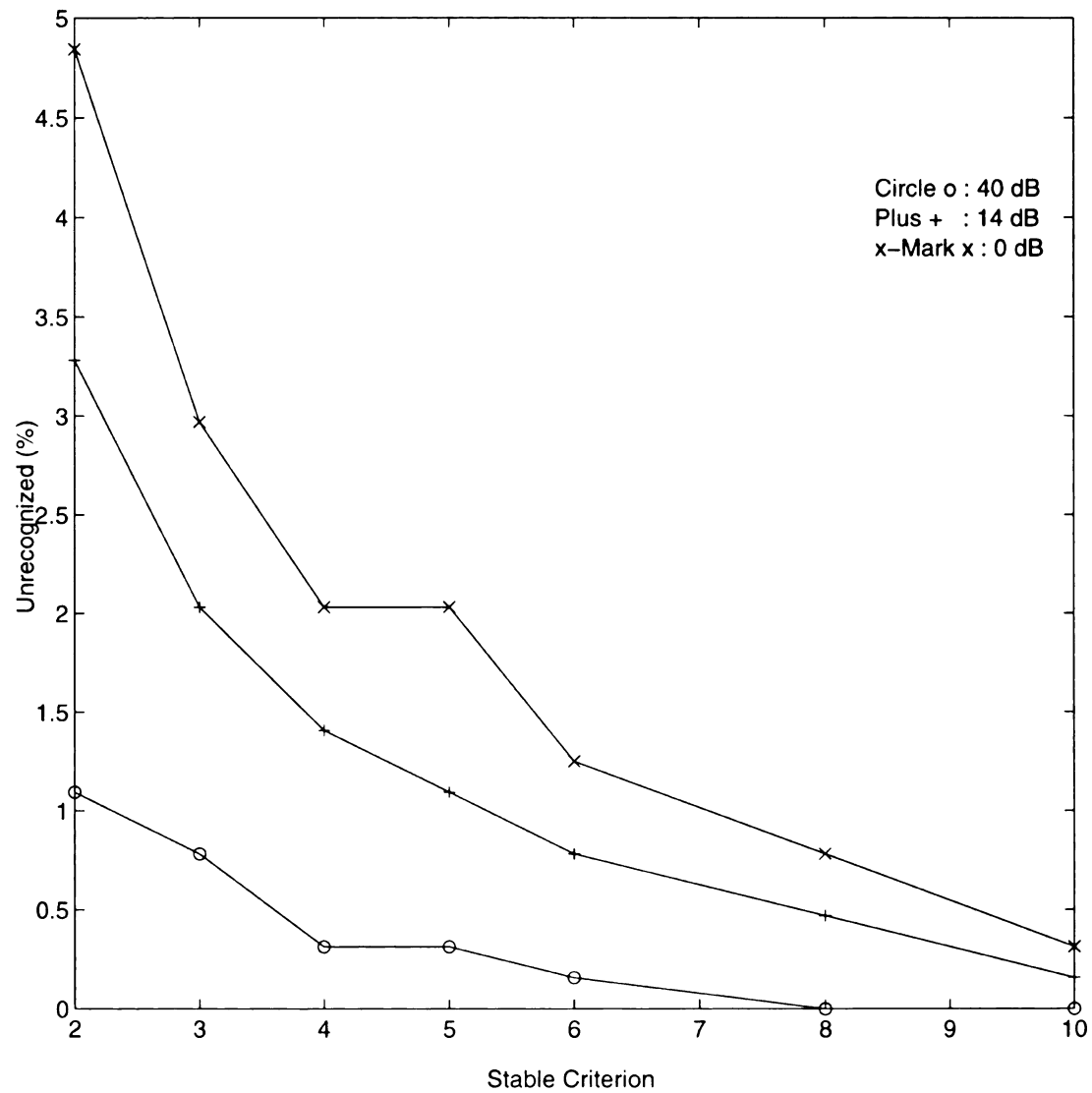


Figure 5.14 Unrecognized discrimination of the RCAAM/di with an initial order of 6 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns.

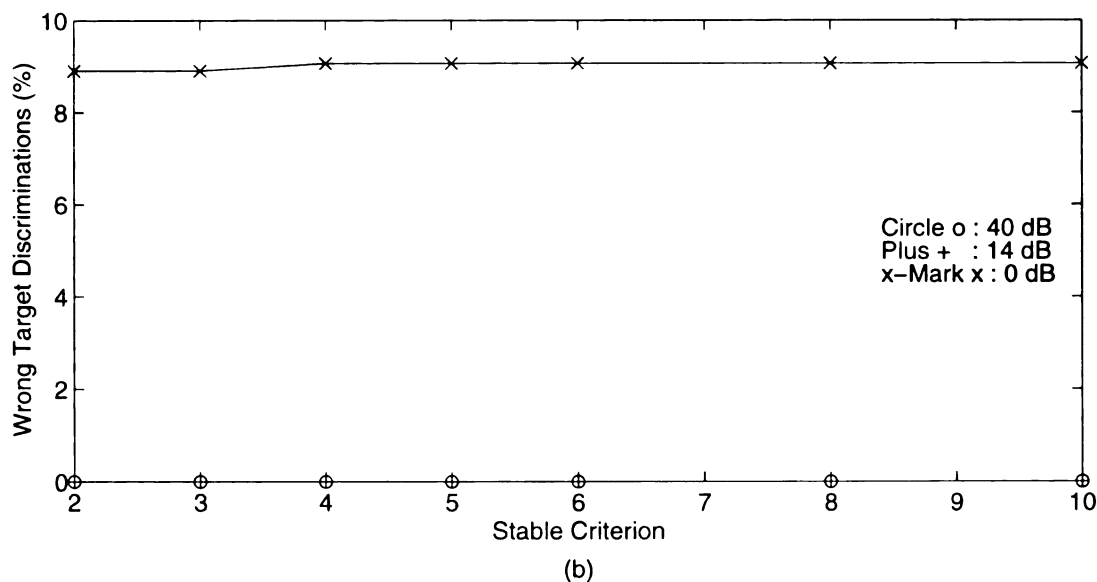
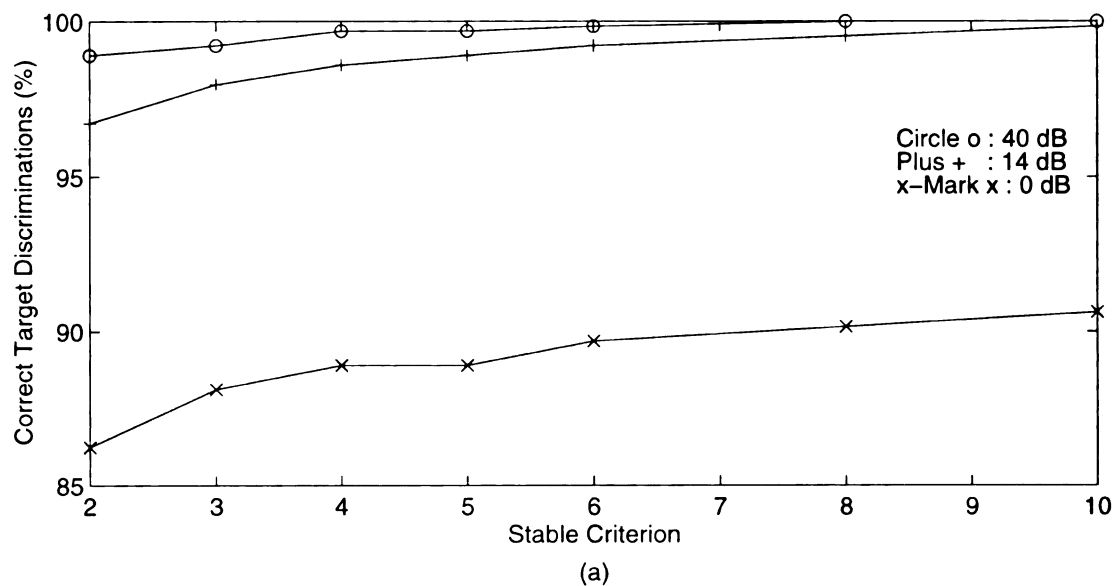


Figure 5.15 Correct and wrong target discrimination performances of the RCAAM/di with an initial order of 6 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs stability criterion.

(b) Wrong target discrimination performances vs. stability criterion.

iteratively adapt the input to its own group before possible high order correlation manipulation.

5.3.2 RCAAM/di performances with respect to initial order effect

In this subsection, we will analyze the performances of RCAAM/di with fixed stability criterions vs. initial order. First let's abbreviate the RCAAM/di with a stability criterion of q by using 'RCAAM/di_q' or 'di_q'. **Figure 5.16** shows unrecognized discrimination of RCAAM/di with a stability criterion of 2 (i.e. RCAAM/di₂ or di₂) vs. initial order for contaminated stored pattern inputs, while **Figure 5.17** presents the correct pattern discrimination in (a) and wrong pattern discrimination in (b). **Figure 5.18** presents the correct target discrimination of di₂ vs. initial order in (a) and wrong target discrimination in (b). The correct pattern and target discriminations rise along with initial order increasing for light contaminations, while they look like downward curves for the heavy noise case (0 dB). And the wrong pattern and target discriminations fall along with initial order increasing for light contaminations, while they look like upward curves for the heavy noise case.

For light contamination, the increasing curves can be explained by factor 1 in the above subsection. Under heavy noise, increasing initial order will greatly converge spurious states for RCAAM/di with small initial orders, and appropriately enhance the seriously soft elasticity to attract the distorted pattern by target group correlation gains. Too soft an elasticity may not only have low efficiency but also induce a coding scheme problem. For small initial order, an increment in the initial order will help to overcome the coding scheme problem described later. But the adaptive elasticity will become too small to turn some spurious states back as described by factor 2 in the above subsection, if the initial order is too

large. This explains why the downward curve has a maximum occurring between the smallest and the largest initial order, and the upward curve has a minimum occurring between the smallest and the largest initial order.

Figure 5.19 shows the unrecognized discrimination of RCAAM/di_2 for unstored pattern inputs, while **Figure 5.20** presents the correct target discrimination in (a) and wrong target discrimination in (b). In the previous subsection, we know the high initial order RCAAM/di has higher correct performances at a low *sc* and flatter performance curves than the low initial order one. Since the *sc* is only set to 2, we only see the increasing portion. To generalize a rule to regulate the *sc* effect on the performances of RCAAM/di along with initial order, we simulate another two sets with stability criterions of 5 and 10, RCAAM/di_5 and RCAAM/di_10. **Figure 5.21** to **Figure 5.25** show the simulation results of RCAAM/di_5, while **Figure 5.26** to **Figure 5.30** present the performances of RCAAM/di_10. Let's analyze the unrecognized discriminations for three stability criterions of 2, 5 and 10 with contaminated stored pattern inputs, i.e. **Figure 5.16**, **Figure 5.21** and **Figure 5.26**. All three unrecognized discriminations decrease as expected when the initial order changes from 1 to 6.

Figure 5.21 shows that the unrecognized discrimination of RCAAM/di15 is larger than the one of RCAAM/di05 under light contamination (40dB and 14 dB). It may be explained as a coding scheme effect. By definition, a lightly contaminated stored pattern should still be close to its true pattern. Consider three close patterns belonging to the same target. Suppose one of these three stored patterns is lightly contaminated, then the sum of weighted stored patterns is a vector most near the linear combination of those three patterns

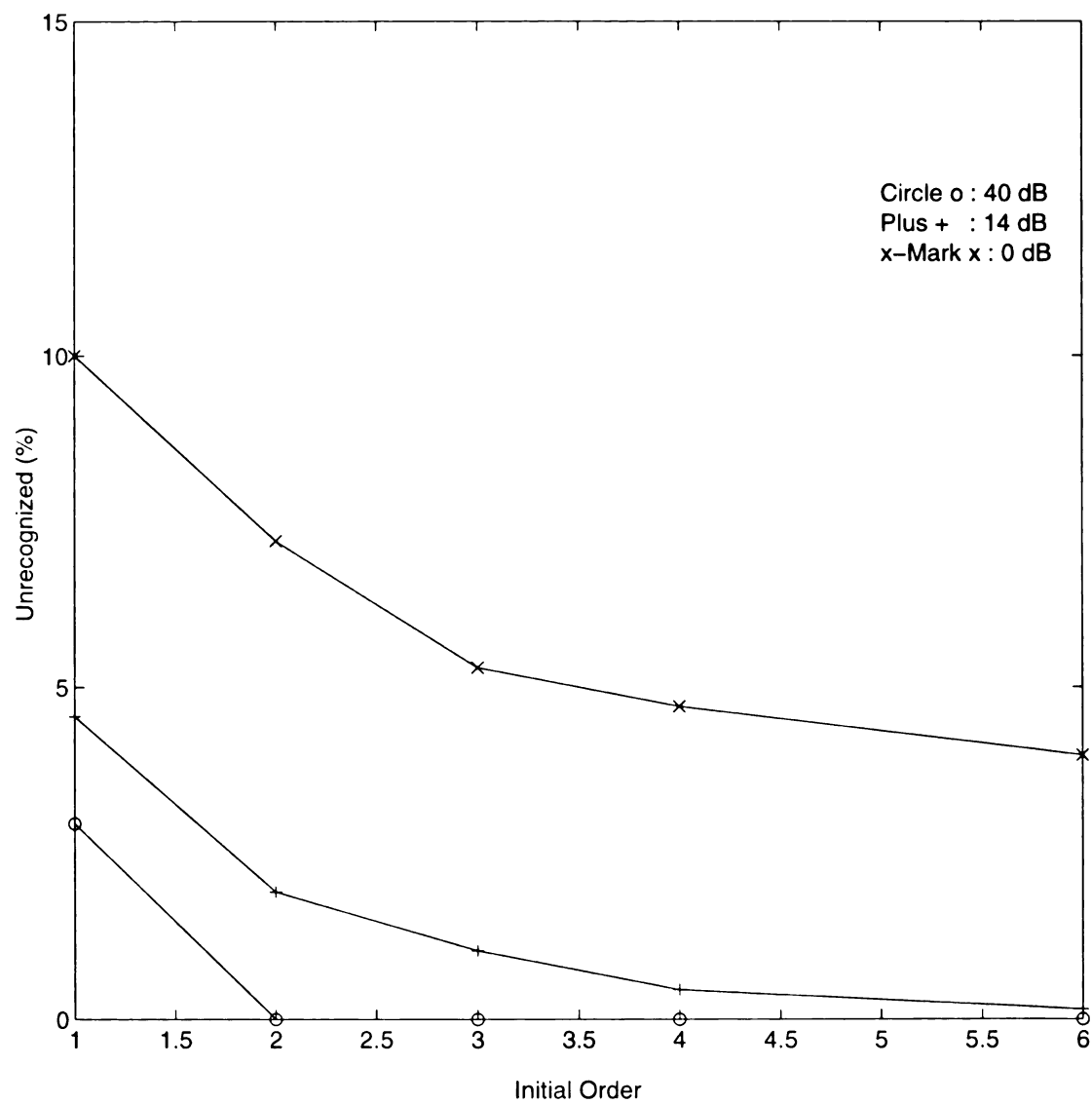


Figure 5.16 Unrecognized discrimination of the RCAAM/di with a stability criterion of 2 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 68 stored patterns.

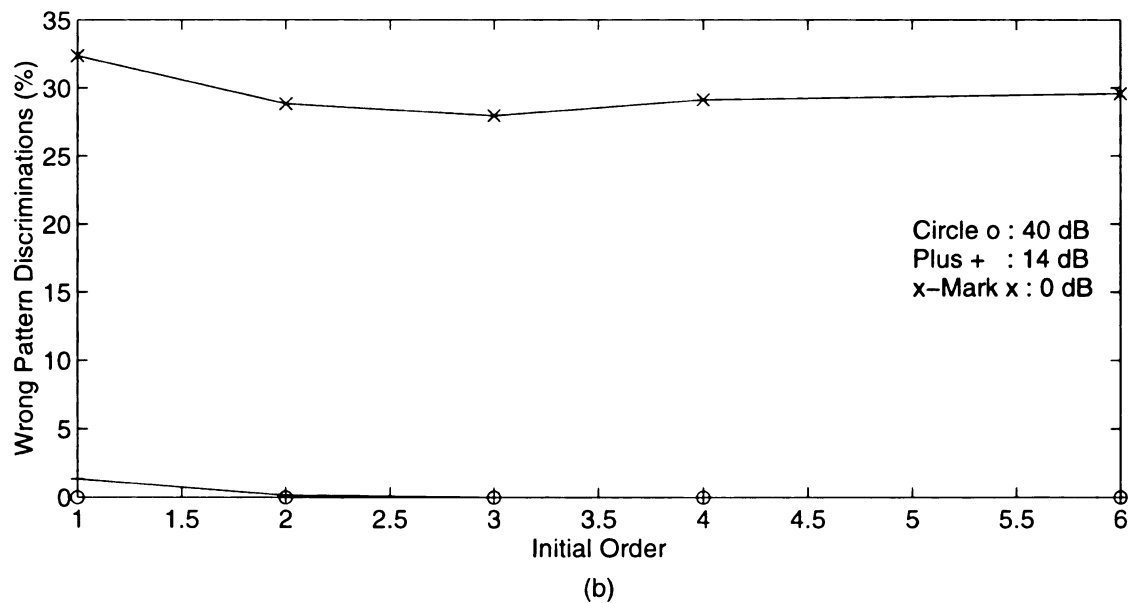
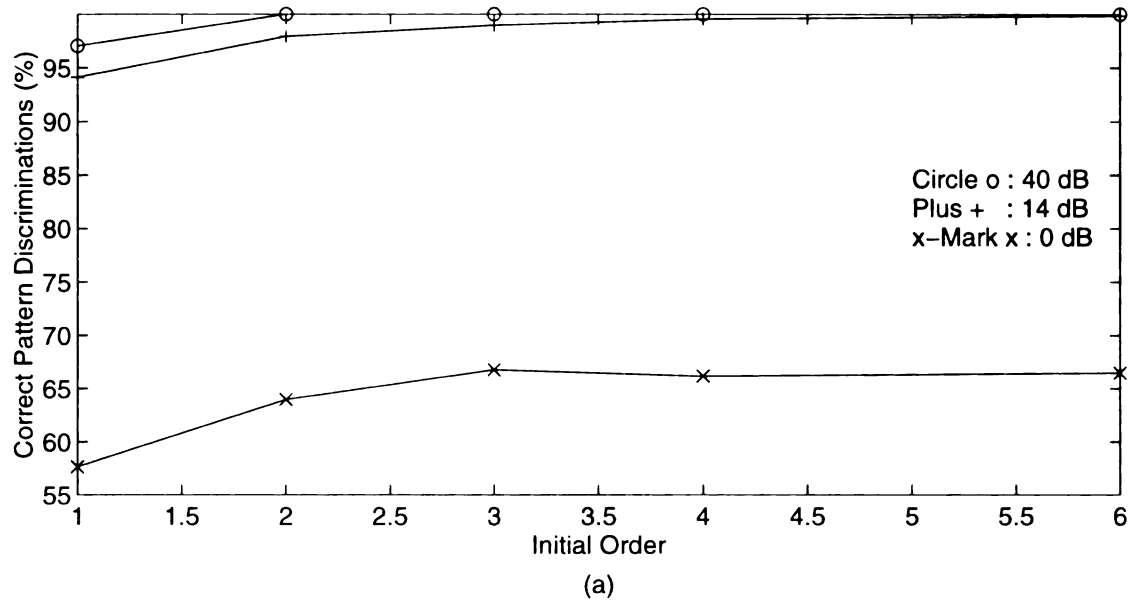


Figure 5.17 Correct and wrong pattern discrimination performances of the RCAAM/di with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for 68 stored patterns.

(a) Correct pattern discrimination performances vs. initial order.

(b) Wrong pattern discrimination performances vs. initial order.

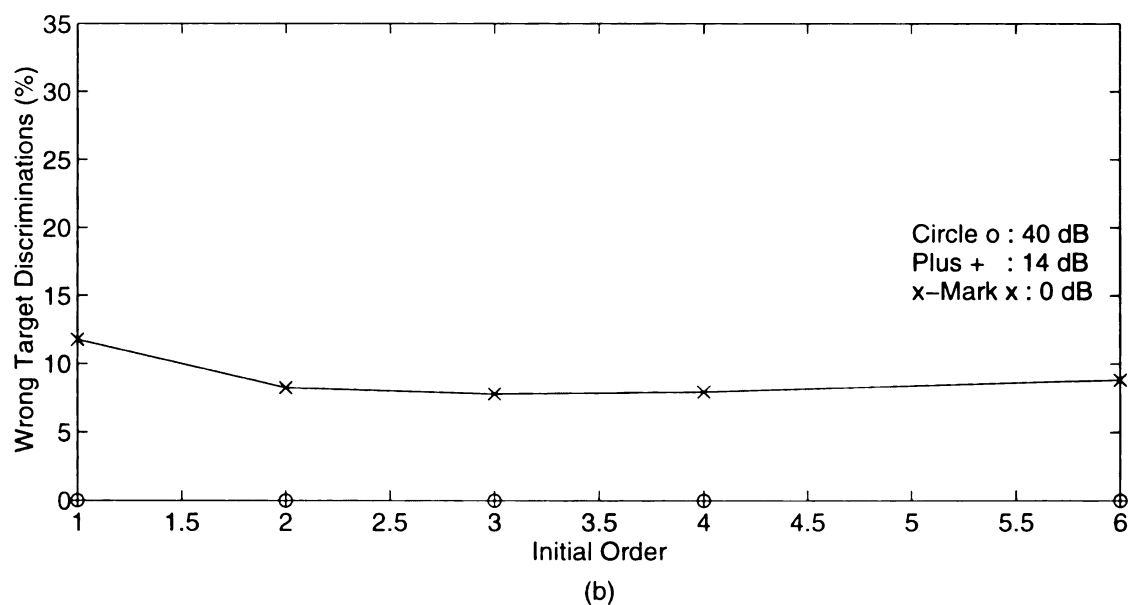
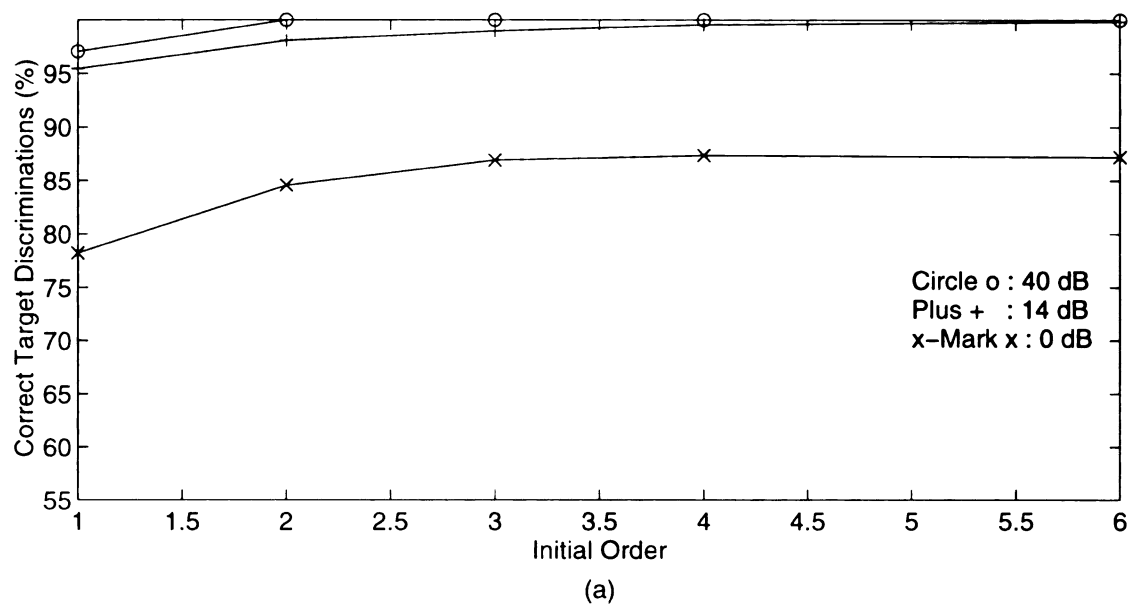


Figure 5.18 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order.

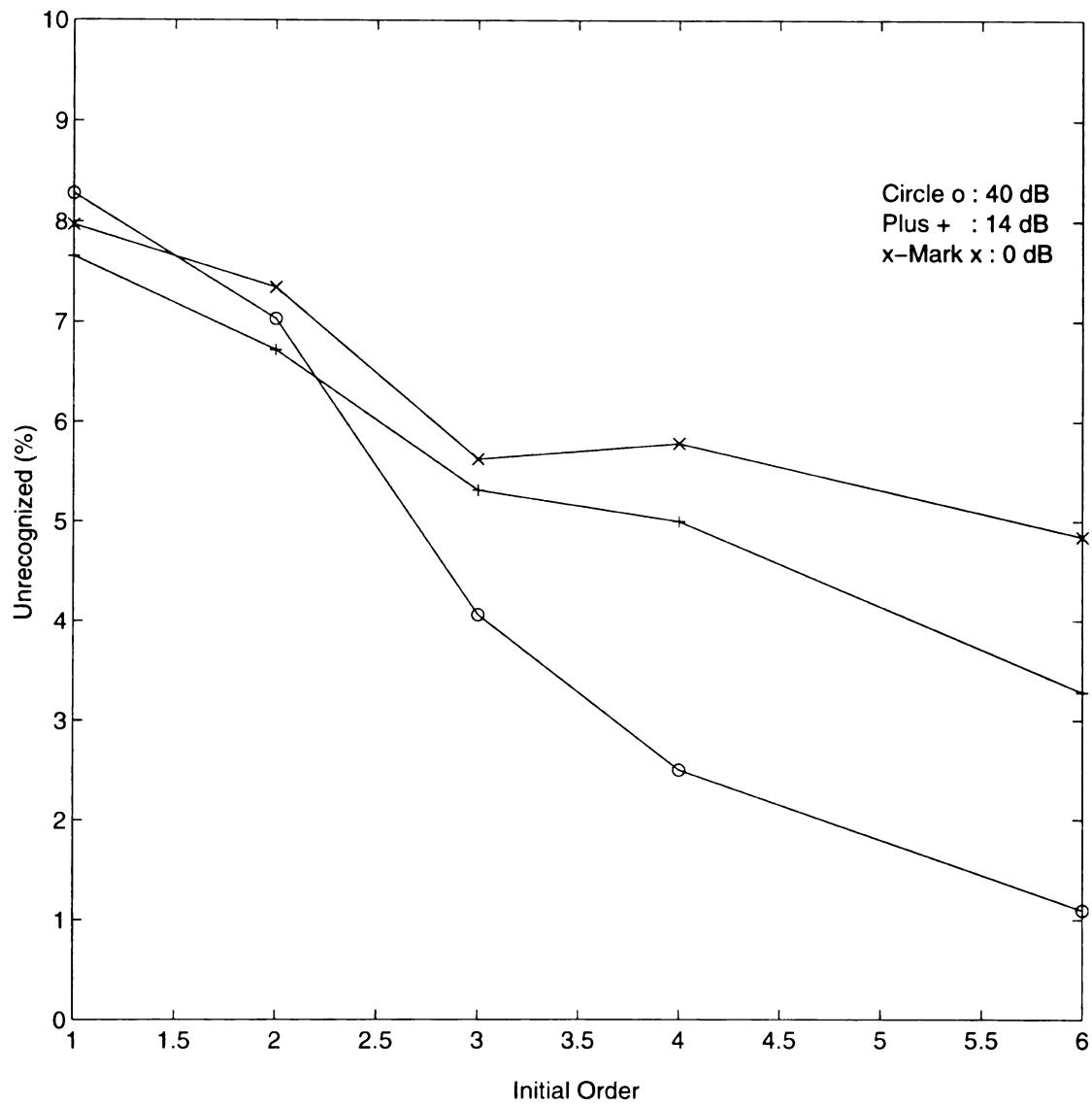


Figure 5.19 Unrecognized discrimination of the RCAAM/di with a stability criterion of 2 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 64 unstored patterns.

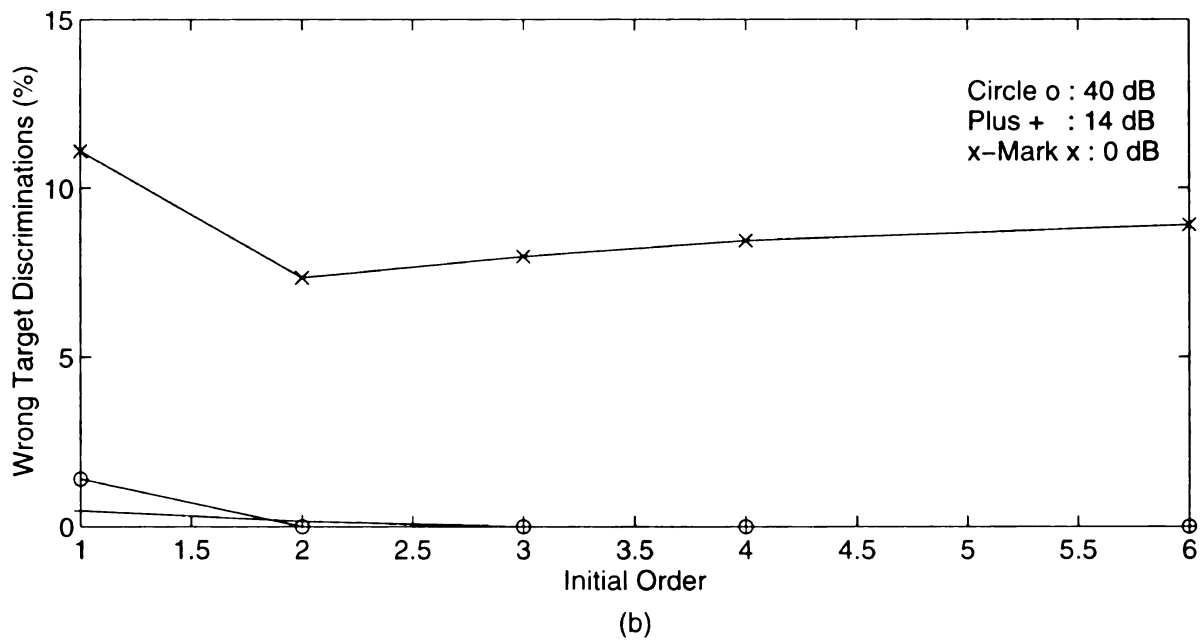
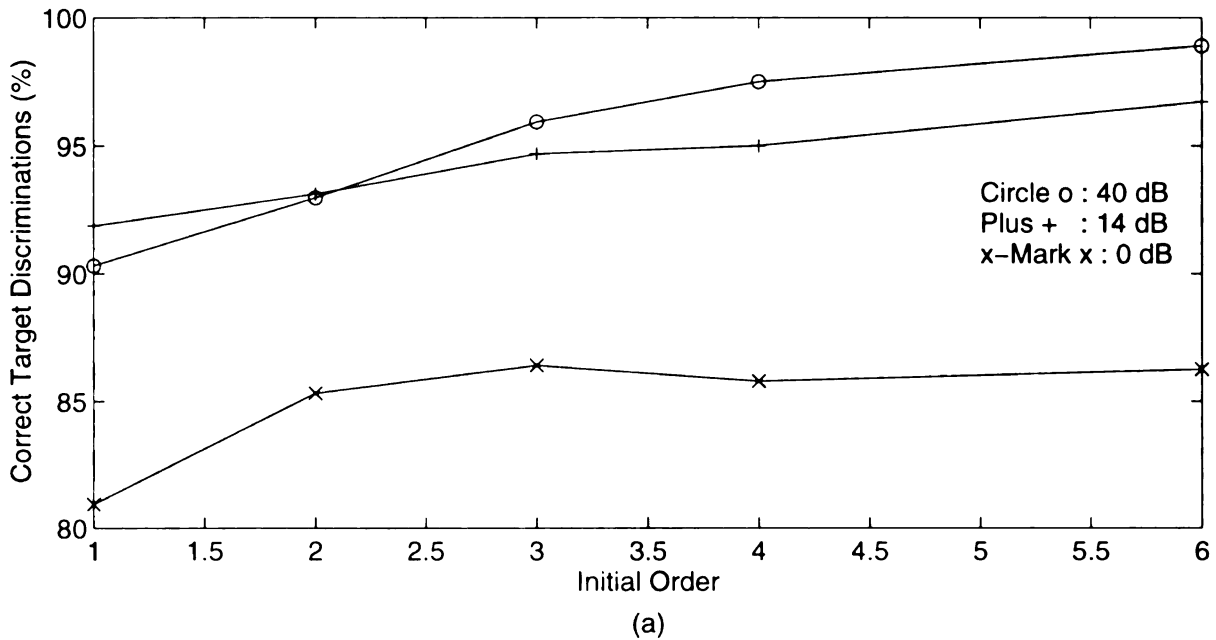


Figure 5.20 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order.

weighted. Then the coding scheme, 3-bit coding 7 levels, and the nonlinear threshold function (Sign function) may result in a state code which is not particularly close to one of those three stored patterns but close to another stored pattern. Therefore, with a stability criterion of 5, an initial order of 0 might leave the lightly distorted pattern in a wrong stored pattern. And an initial correlation with one order higher may pull it to an intermediate state, nor correct neither wrong state, then this intermediate state will contribute to the unrecognized discrimination while a wrong pattern, possible correct target, discrimination is eliminated. Therefore, the curves rise for 40 and 14 dB SNR's.

Furthermore another time increase on the initial order may converge the intermediate state to the correct stored state. This time the correct discrimination will be credited while the intermediate state is eliminated. The curves thus fall. This hypothesis can be verified by investigating **Figure 5.22** and **Figure 5.23** . In **Figure 5.22** , the correct pattern discrimination doesn't increase the same amount, while the wrong pattern discrimination decreases when the initial order changes from 0 to 1. This indicates that some portion of eliminated wrong discriminations goes to intermediate states instead of correct stored states. From **Figure 5.23** (a), you can see the correct target discrimination decreases when the initial order increases from 0 to 1 for the light contamination case (40 dB). This indicates a wrong pattern discrimination can be a correct target discrimination, and the wrong pattern discrimination can be converged to an intermediate state instead of a correct target when the initial order only increases by one. From the above subsection, we know a higher **sc** converges more states or has higher capability to turn a spurious state into a stable state. The coding scheme effect won't bother the RCAAM/di with a **sc** of 10, while it interferes with the

RCAAM/di with a **sc** of 5. Compared to the RCAAM/di_10, the RCAAM/di_5 still leaves some states convergeable. From the correct and wrong discrimination performances for three cases, all the correct and wrong discriminations are pulled up when the stability criterion is increased. These are consistent with the previous performances and discussions.

Let's investigate the unrecognized discriminations for unstored pattern inputs, i.e. **Figure 5.19** , **Figure 5.24** and **Figure 5.29** . From the previous subsection discussion, we have two rules describing the initial order effect on the performance curve shapes and converging speeds along with **sc**. The higher initial order RCAAM/di has flatter performance curves but slow converging vs. **sc**. And factor 2 told us a high initial order RCAAM/di won't converge unknowns for unstored pattern inputs until a high **sc** is given, since it has lost a lot of adaptive elasticity . Under contaminations of 40 dB and 14 dB SNR, consider the initial order ranges from 1 to 6 with unstored pattern inputs. In **Figure 5.19** , the RCAAM/di with an initial order of 6 has the minimum unknowns and the RCAAM/di with an initial order of 1 has maximum unknowns. When the **sc** is increased to 5, the above rules are invoked. In **Figure 5.24** , the RCAAM/di6's lose their places, while the RCAAM/di1's and RCAAM/di2's become new minima. When the **sc** continues increasing to 10, then factor 2 shows up. In **Figure 5.29** , the RCAAM/di6's not only lose their minima but also become the maxima, while the RCAAM/di3's and RCAAM/di4 continue converging their unknowns. The downward curves of correct target discriminations for stability criteria of 5 and 10, **Figure 5.25** and **Figure 5.30** , again show the consistency.

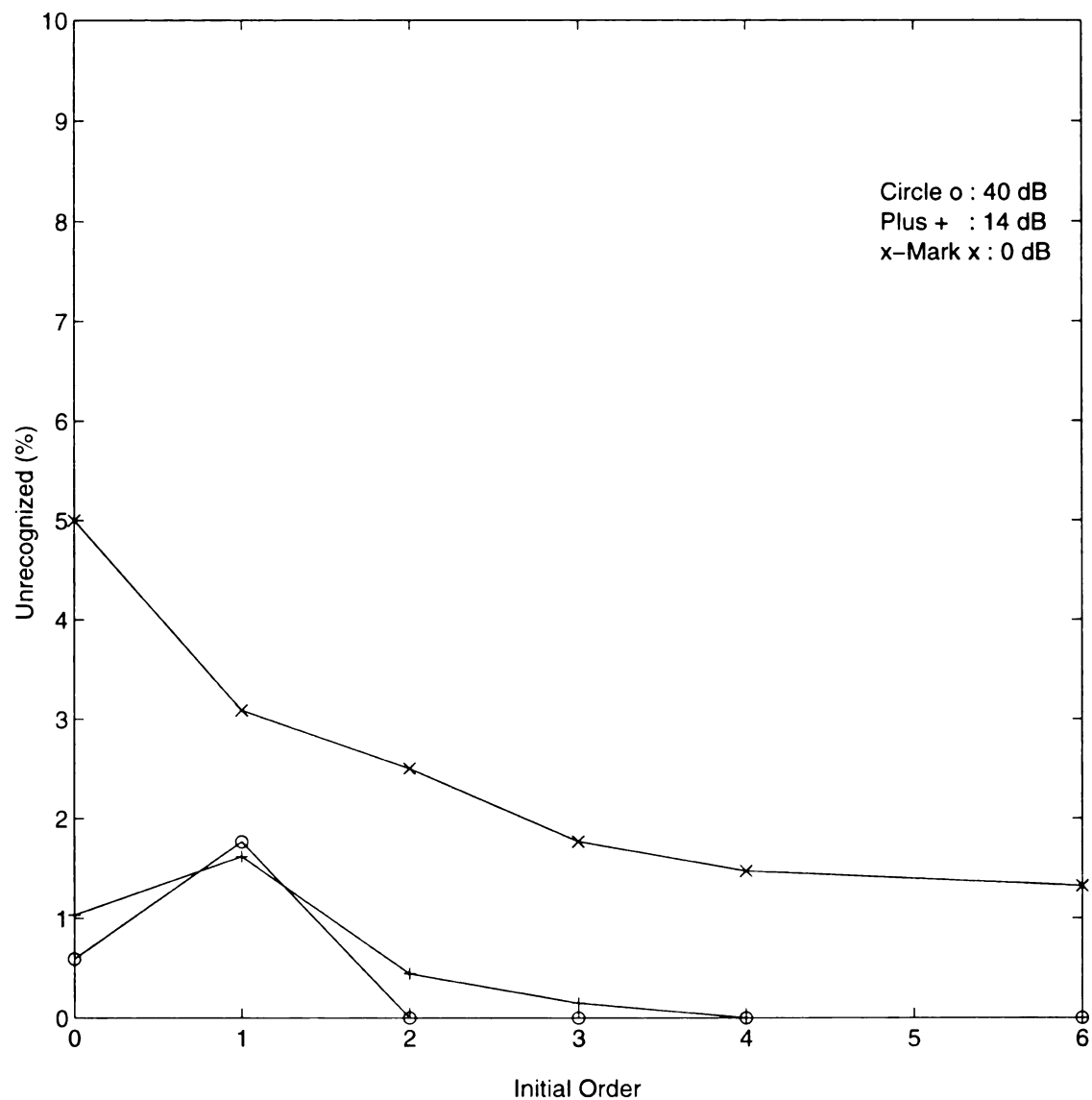


Figure 5.21 Unrecognized discrimination of the RCAAM/di with a stability criterion of 5 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 68 stored patterns.

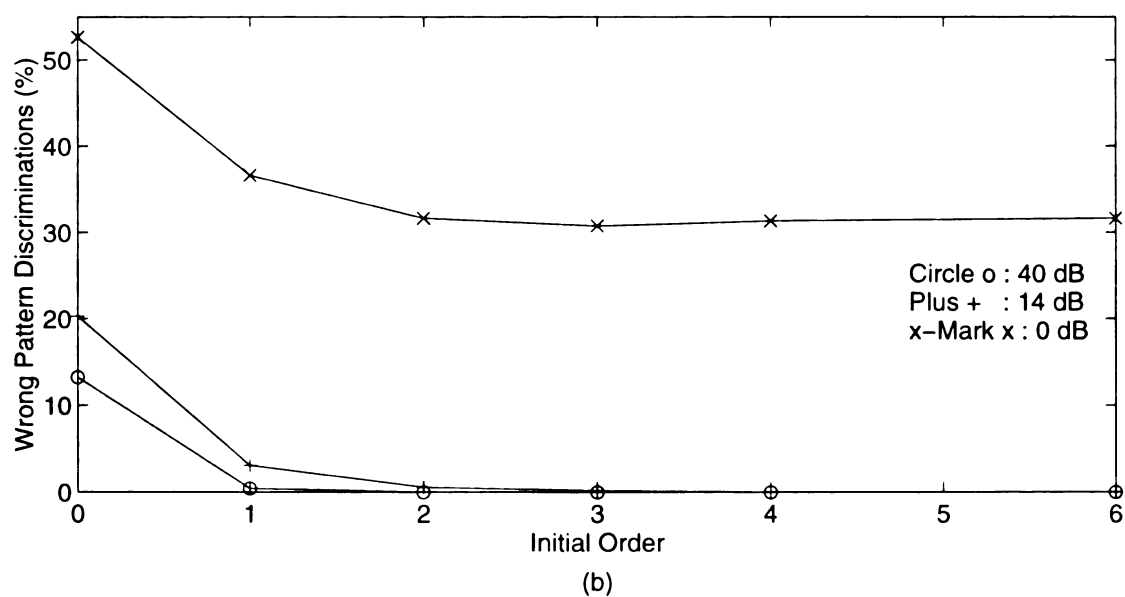
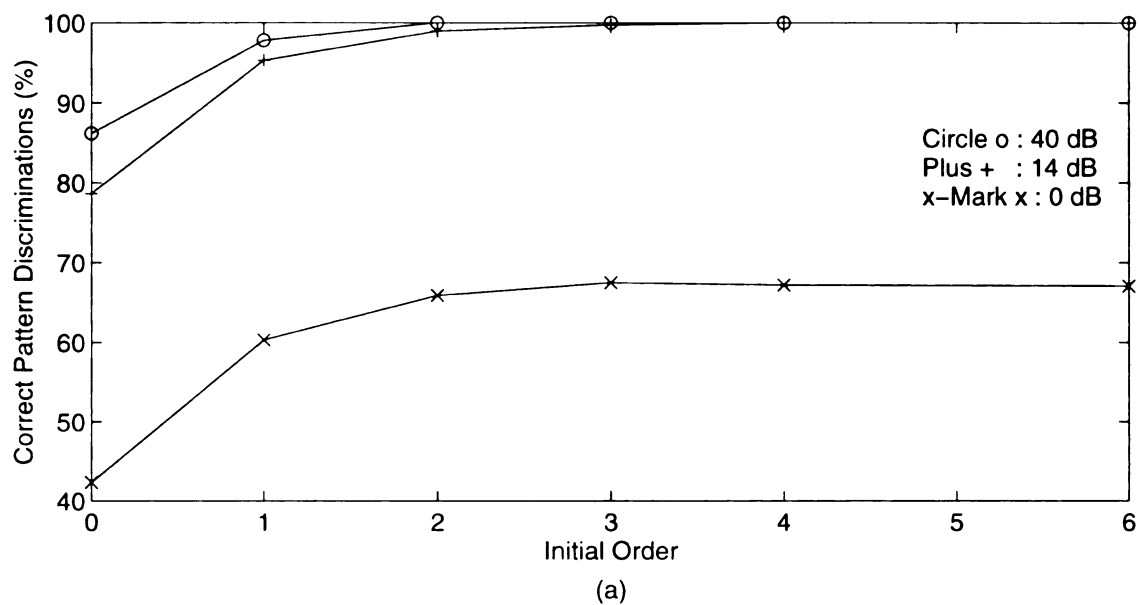


Figure 5.22 Correct and wrong pattern discrimination performances of the RCAAM/di with a stability criterion of 5 under 40, 14 and 0 dB SNR vs. initial order for 68 stored patterns.

(a) Correct pattern discrimination performances vs. initial order.

(b) Wrong pattern discrimination performances vs. initial order.

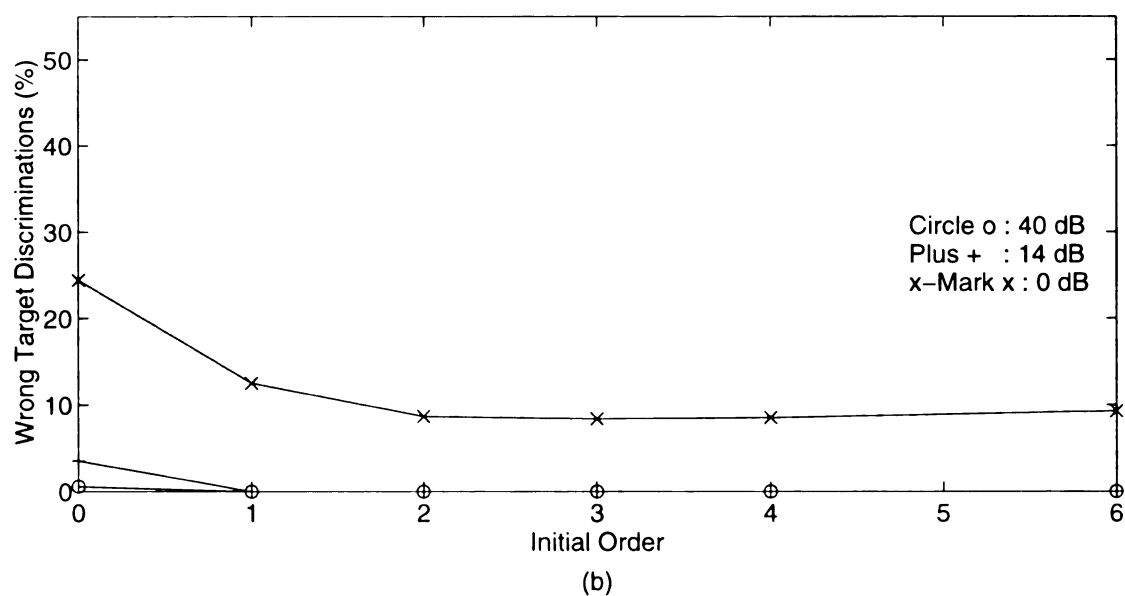
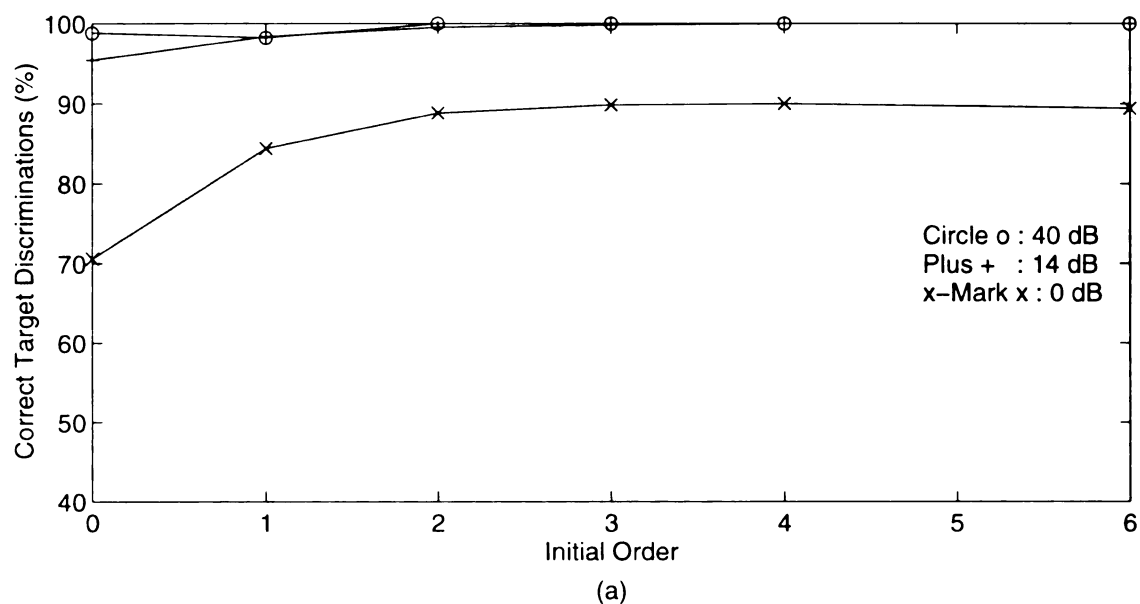


Figure 5.23 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 5 under 40, 14 and 0 dB SNR vs. initial order for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order.

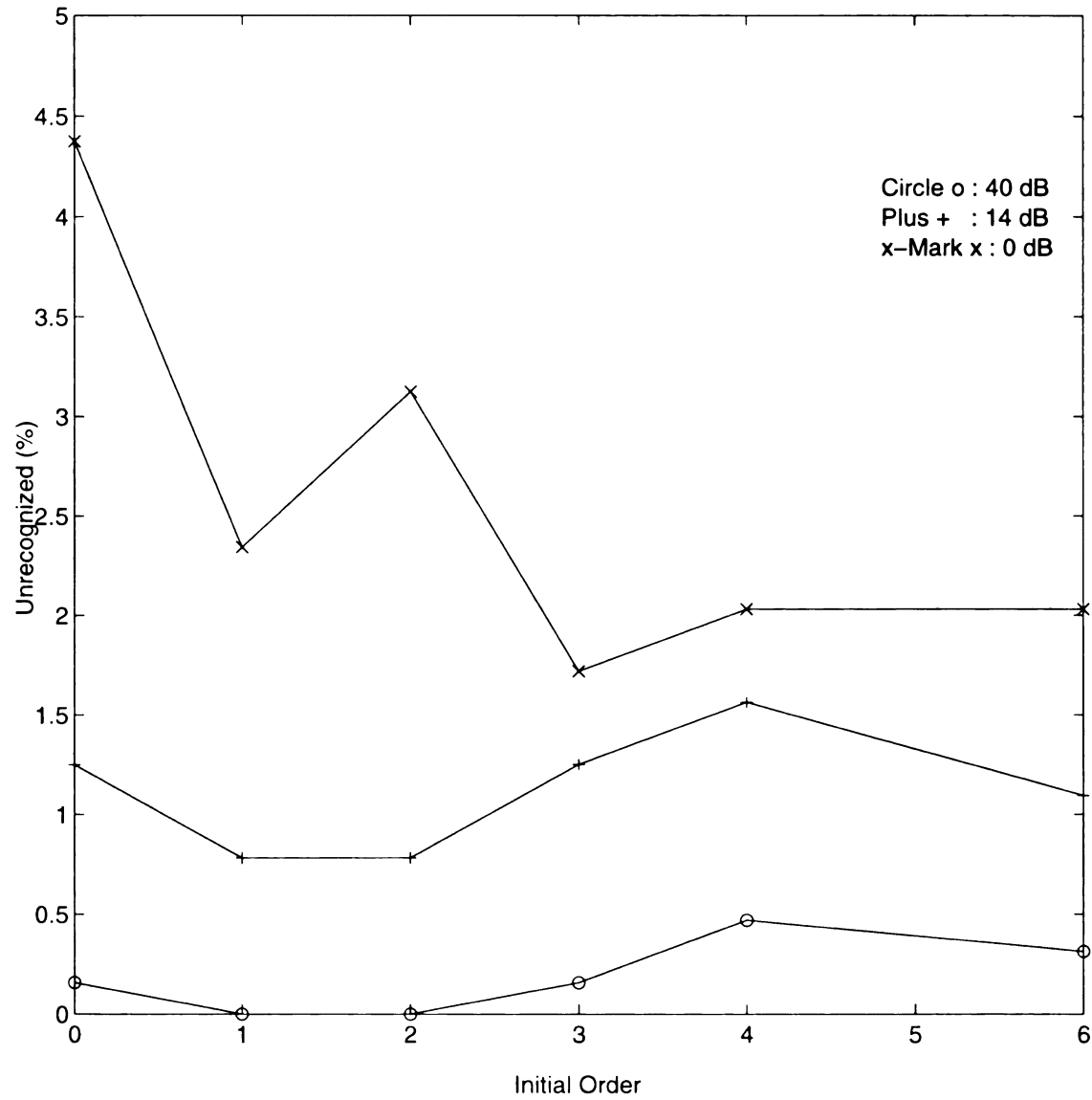


Figure 5.24 Unrecognized discrimination of the RCAAM/di with a stability criterion of 5 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 64 unstored patterns.

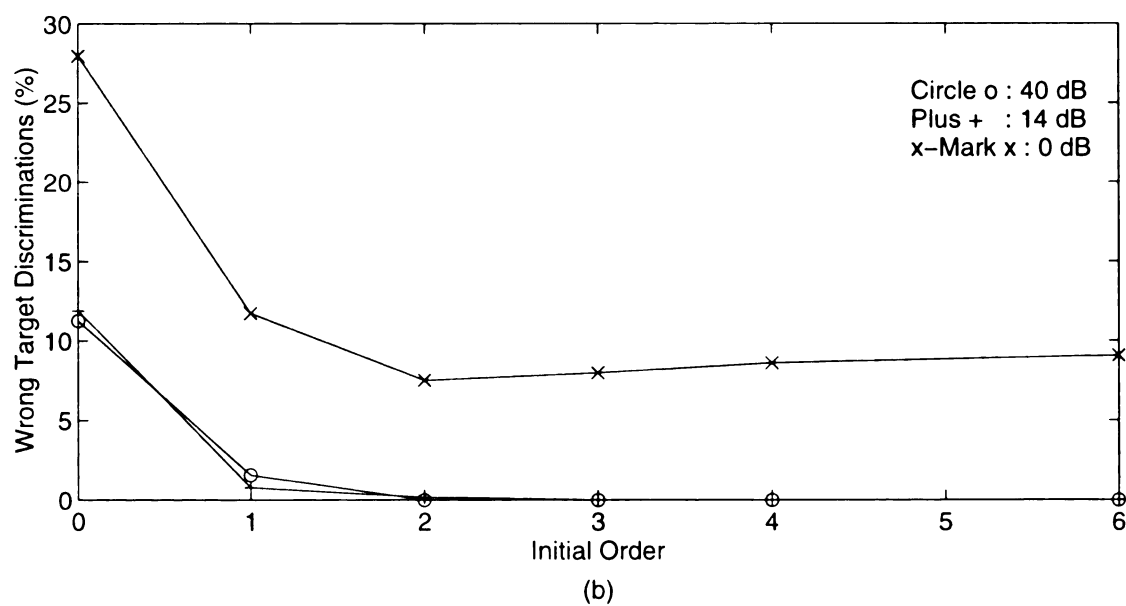
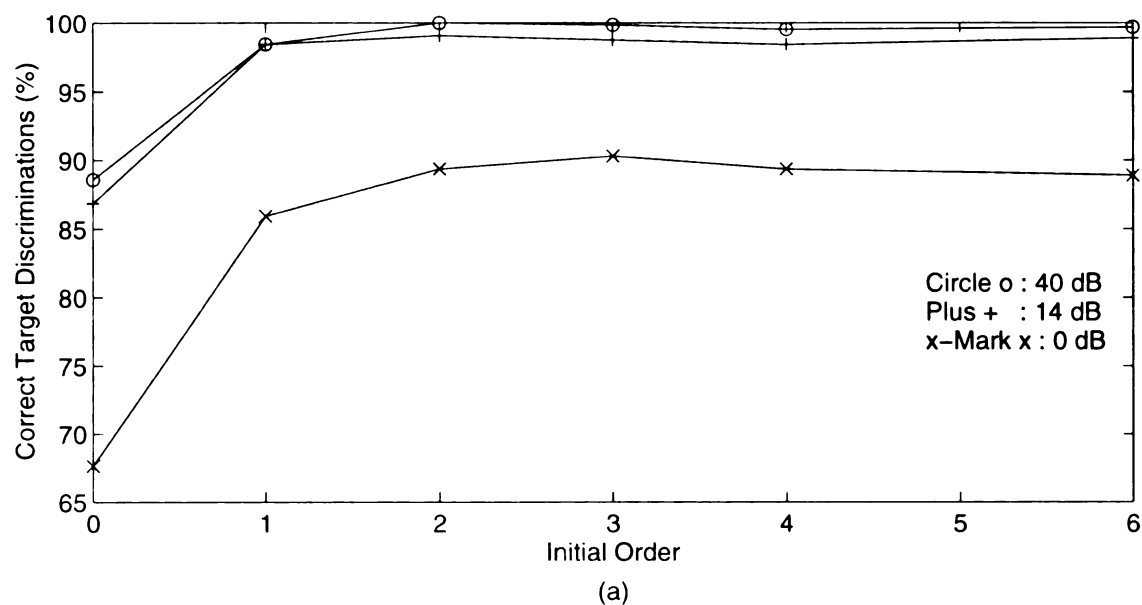


Figure 5.25 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 5 under 40, 14 and 0 dB SNR vs. initial order for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order.

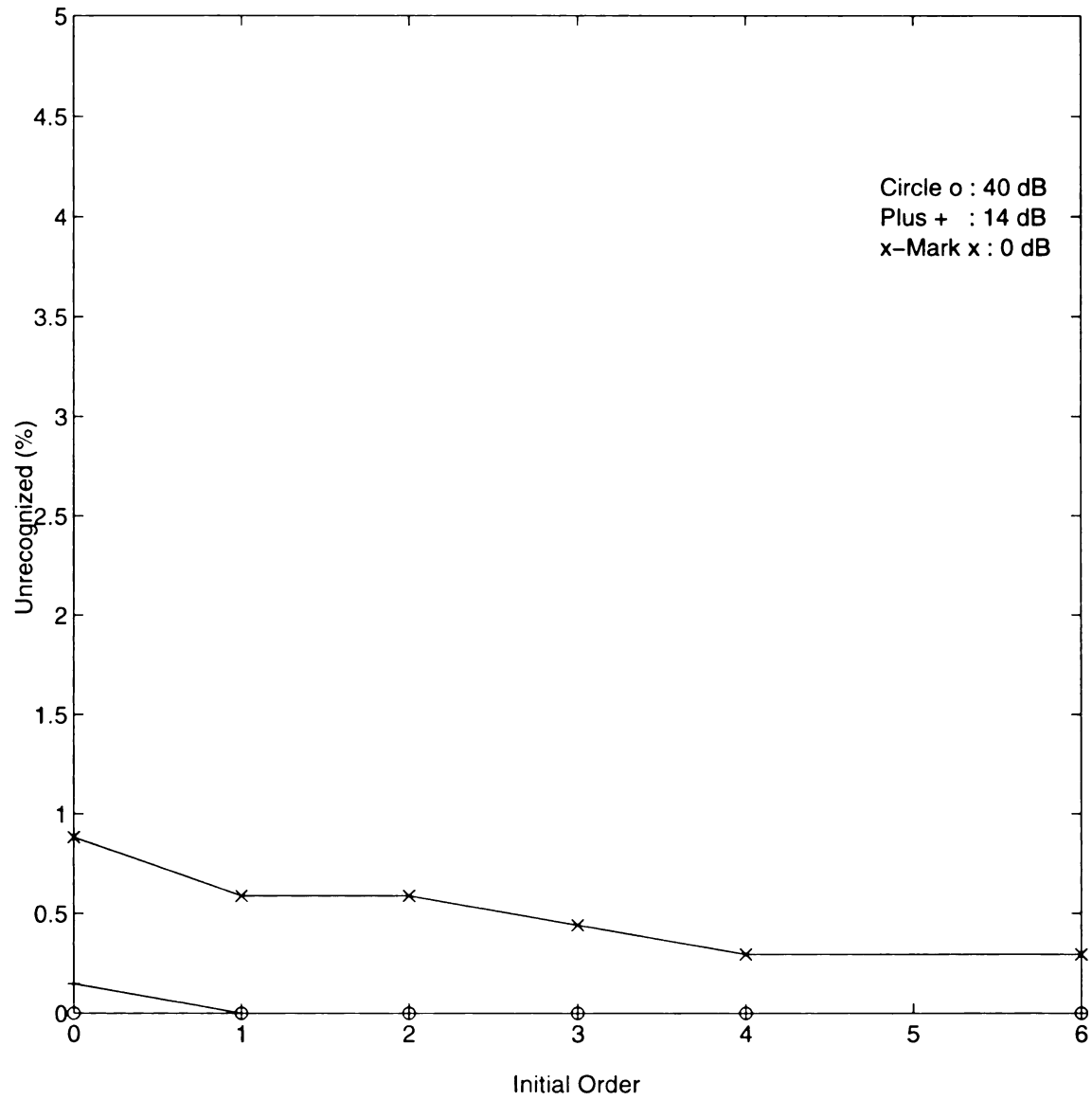


Figure 5.26 Unrecognized discrimination of the RCAAM/di with a stability criterion of 10 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 68 stored patterns.

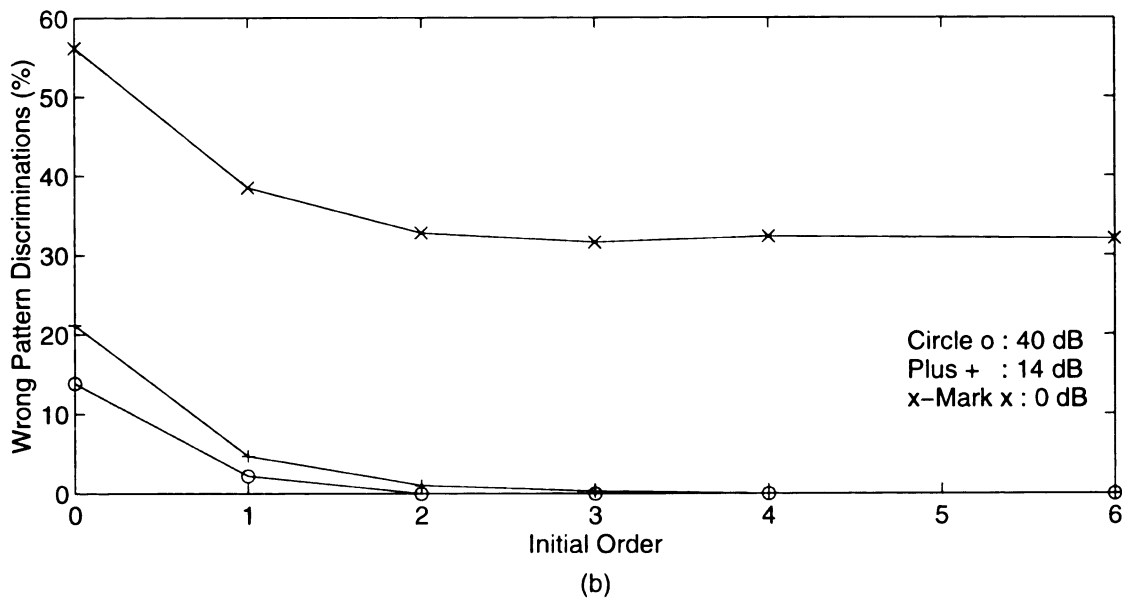
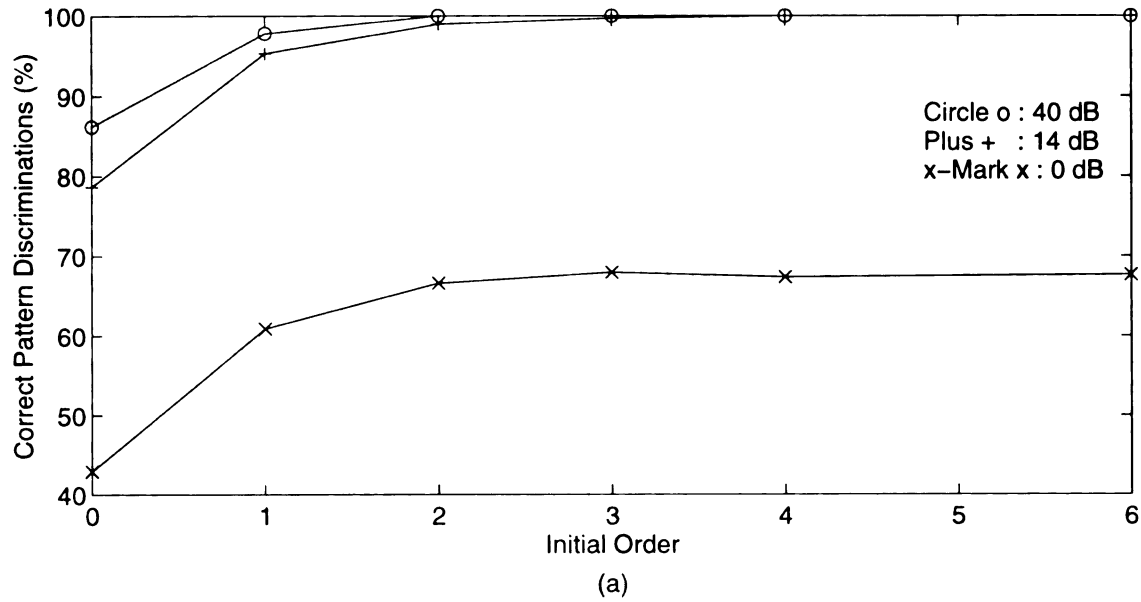


Figure 5.27 Correct and wrong pattern discrimination performances of the RCAAM/di with a stability criterion of 10 under 40, 14 and 0 dB SNR vs. initial order for 68 stored patterns.

(a) Correct pattern discrimination performances vs. initial order.

(b) Wrong pattern discrimination performances vs. initial order.

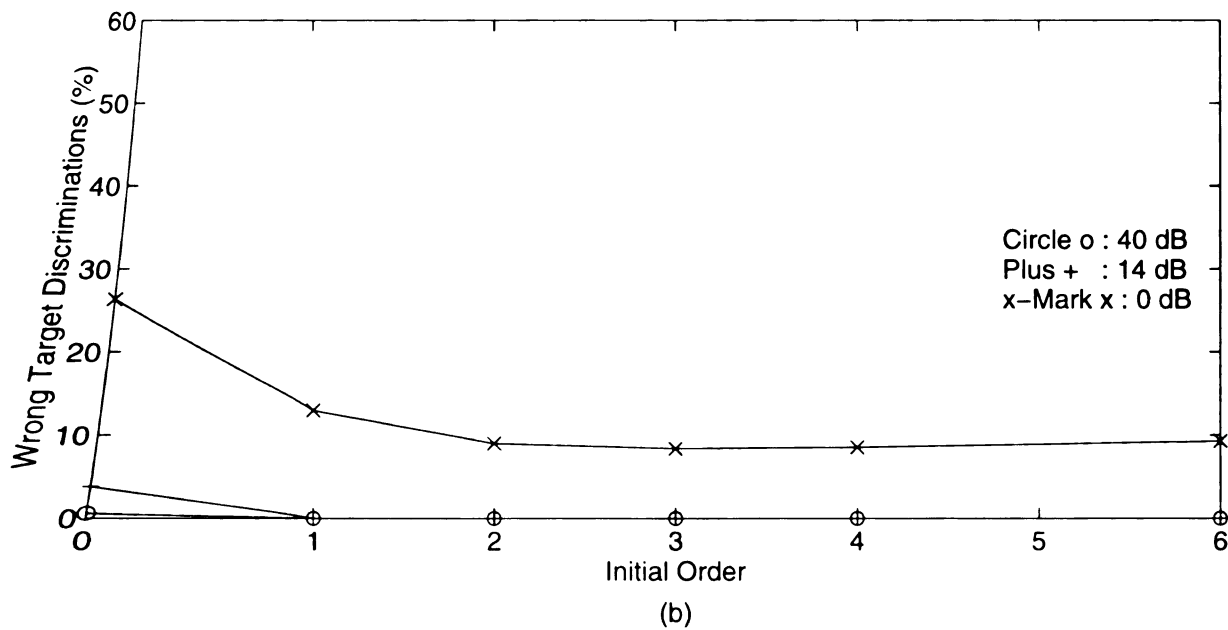
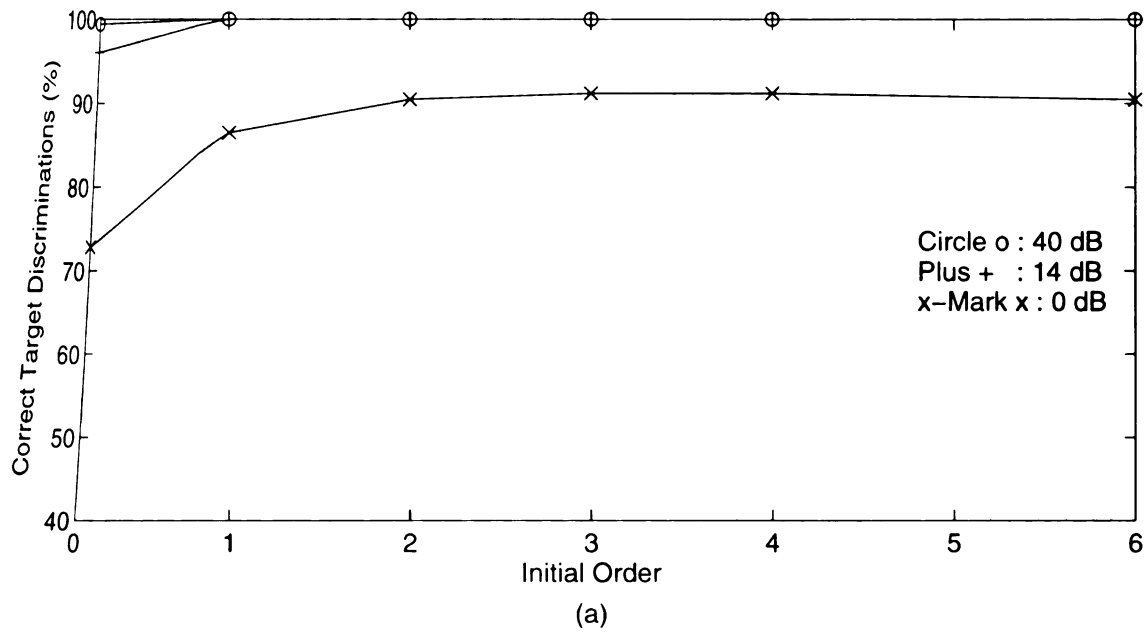


Figure 5.28 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 10 under 40, 14 and 0 dB SNR vs. initial order for the 68 colored patterns belonging to 4 targets..

) Correct target discrimination performances vs initial order.

) Wrong target discrimination performances vs. initial order.

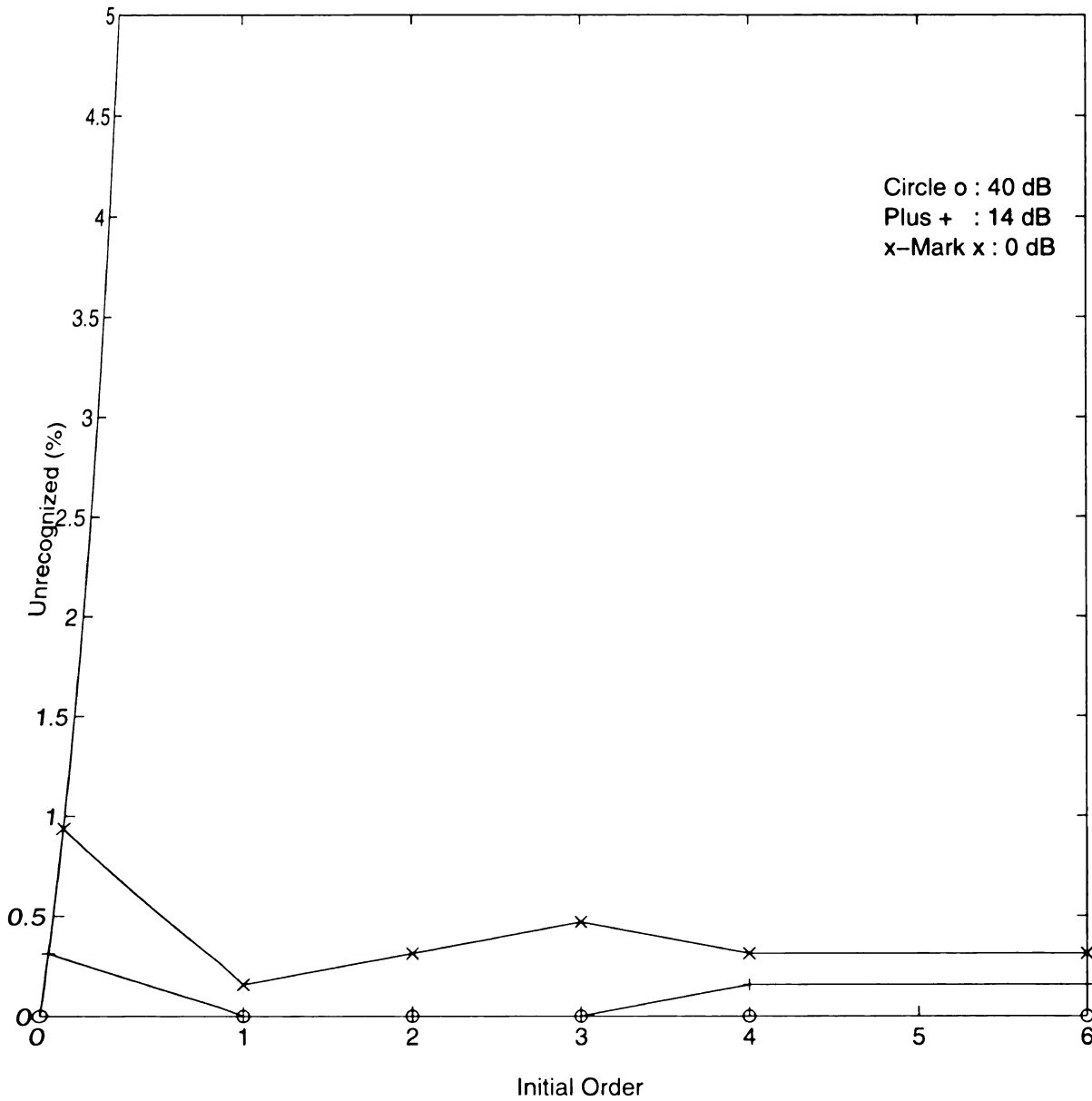


Figure 5.29 Unrecognized discrimination of the RCAAM/di with a stability criterion of 10 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 64 unstored patterns.

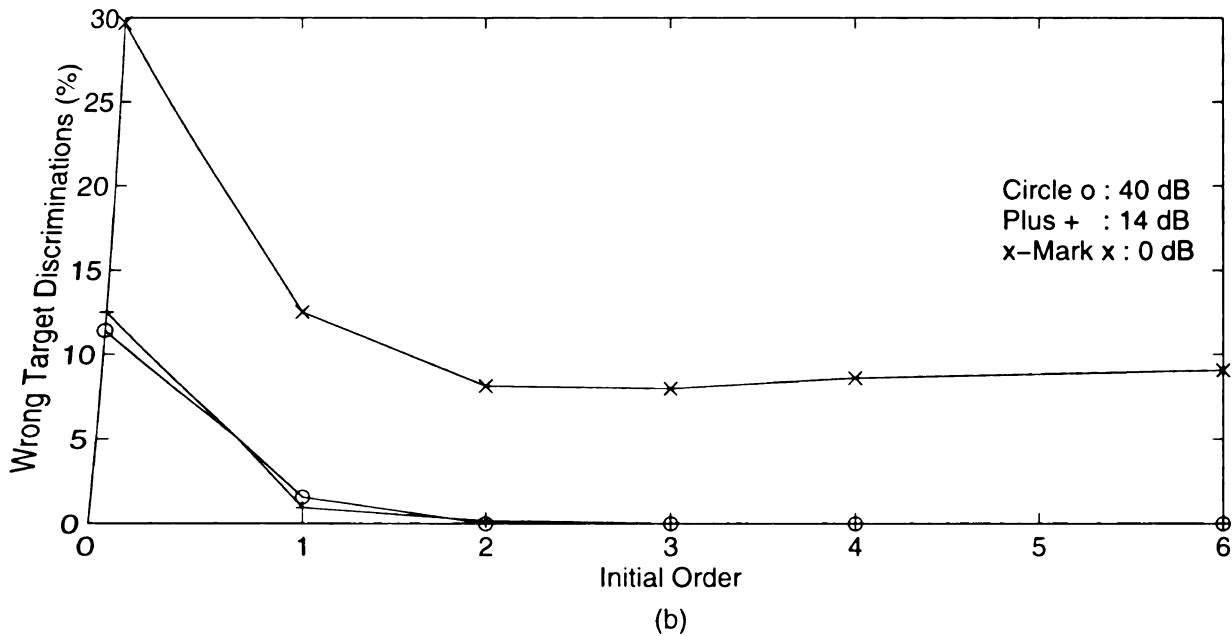
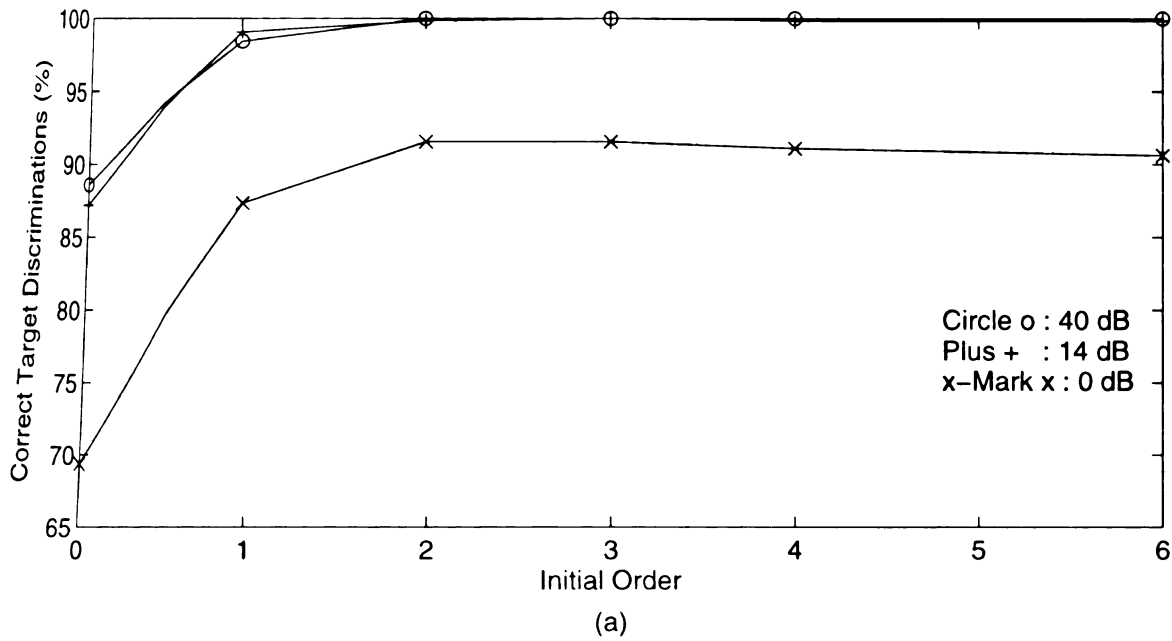


Figure 5.30 Correct and wrong target discrimination performances of the RCAAM/di with a stability criterion of 10 under 40, 14 and 0 dB SNR vs. initial order for the 64 stored patterns belonging to 4 targets..

a) Correct target discrimination performances vs initial order.

b) Wrong target discrimination performances vs. initial order.

5.3.3 RCAAM/di performances for the same averaged order group

In this subsection we evaluate the performances of RCAAM/di's with similar discrimination orders. The comparisons of RCAAM/di's belonging to a same discrimination order group can help us select and design an RCAAM/di which will satisfy our specifications under constrained available memory. Suppose a RCAAM/di with an initial order of p and a stability criterion of q (i.e. $dipq$) requires k iterations to satisfy its stability criterion for an input, then we say the order required by RCAAM/ $dipq$ for this input discrimination is $p+k$.

Figure 5.31 shows the averaged orders required for the discriminations (**aord**) of RCAAM/di's with the initial order change from 0 to 6 vs. stability criterion for stored pattern inputs, while **Figure 5.32** presents the averaged order required for the discriminations of unstored pattern inputs. For a fixed initial order, the **aord** changes linearly along with **sc**. The slope is larger for small **sc**, and is approaching 1 when **sc** become large. This is especially apparent for the RCAAM/di with small initial order, such as di_0 and di_1 . With **sc** fixed to 5, the di_0 has its **aord** larger than the di_1 's, di_2 's, di_3 's and di_4 's, while the di_1 has its **aord** still larger than the di_2 's. The phenomena indicate the soft elasticity associated with low initial orders will struggle longer to adapt to stored patterns under heavy contaminations. Since a low initial order, 0 or 1, is unable to explicitly instruct the possibly efficient routes to converge in the beginning, a slightly advantaged pattern after the feedback mechanism of RCAAM/di might become more ambiguous due to the linear combination of weighted patterns and the coding scheme problem. From **Figure 5.31** and **Figure 5.32**, we can see that an initial order larger than 1 will greatly reduce (or eliminate) the inefficiency in **aord** resulting from the too soft elasticity problem.

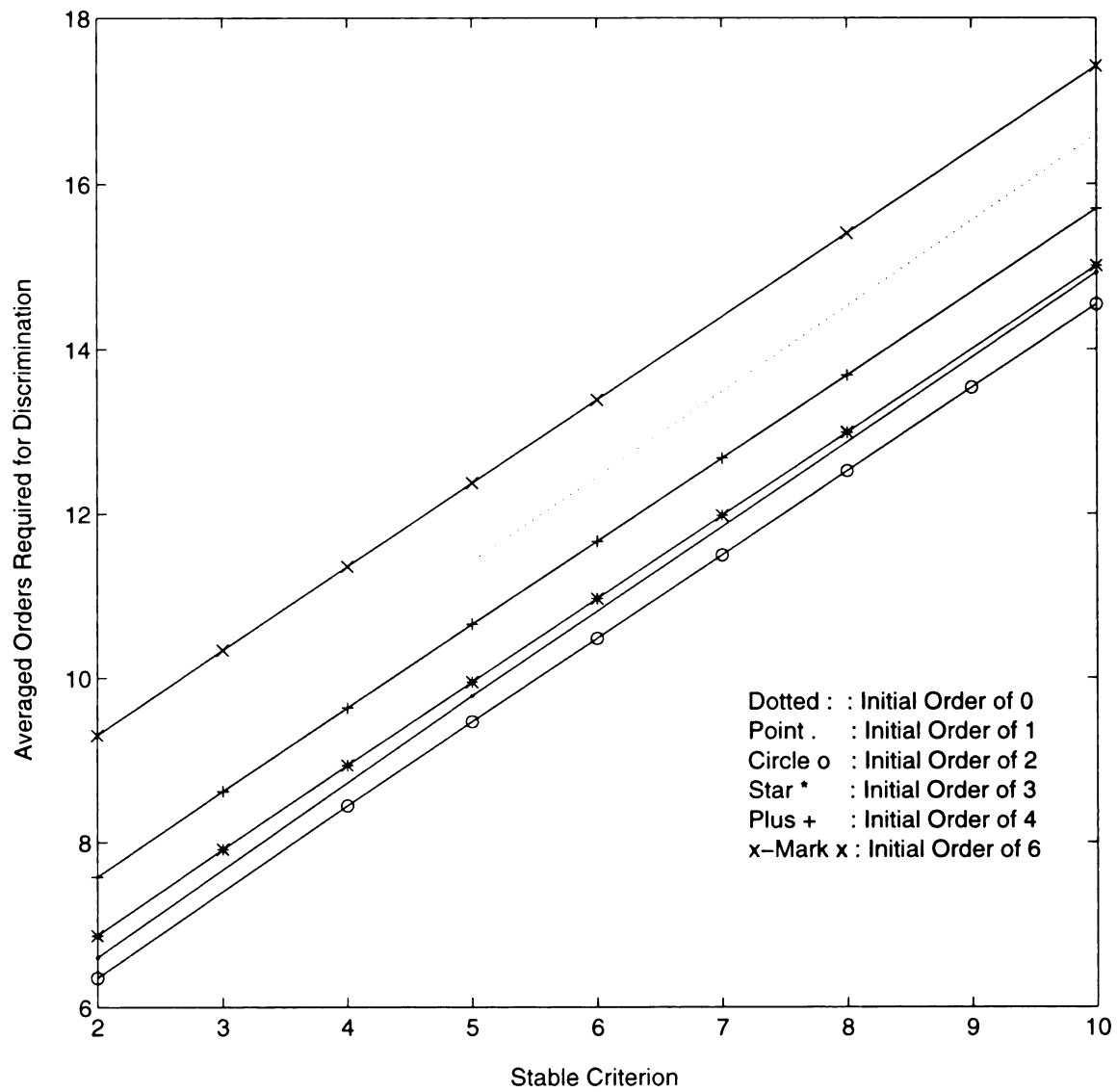


Figure 5.31 Averaged orders required for target discrimination of the RCAAM/di with contaminated stored pattern inputs vs. stability criterion.

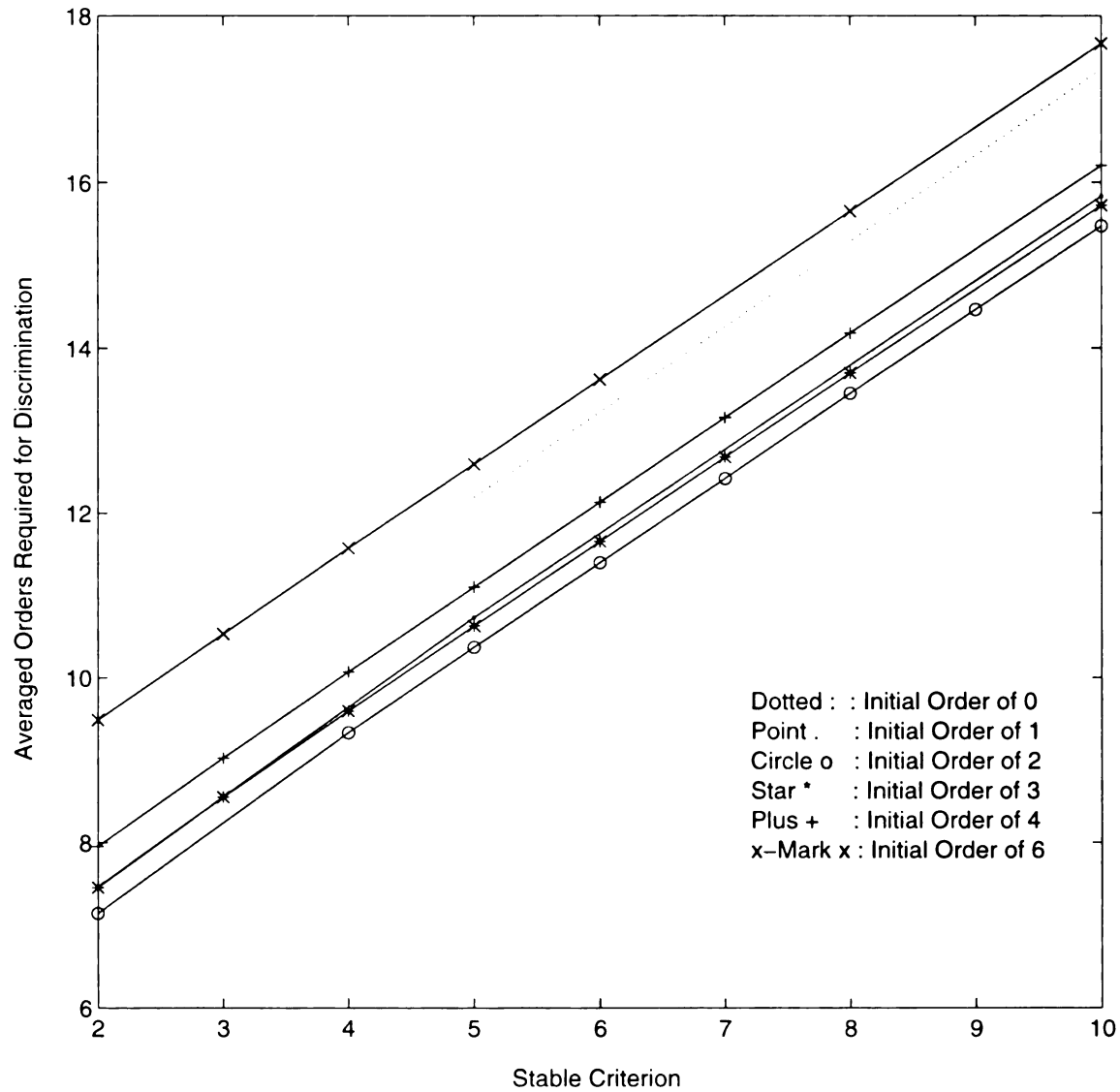


Figure 5.32 Averaged orders required for target discrimination of the RCAAM/di with contaminated unstored pattern inputs vs. stability criterion.

We categorize four RCAAM/di groups by using their **aord** of stored pattern inputs, group **aord9**, group **aord10**, group **aord11** and group **aord13**. Group **aord9** defines the RCAAM/di's with $9 < \mathbf{aord} < 10$, group **aord10** defines the RCAAM/di's with $10 < \mathbf{aord} < 11$, group **aord11** defines the RCAAM/di's with $11 < \mathbf{aord} < 12$, and group **aord13** defines the RCAAM/di's with $13 < \mathbf{aord} < 14$.

Figure 5.33 to **Figure 5.36** show the performance comparisons of 4 RCAAM/di's of group **aord9**, di25 with **aord**=9.4696, di35 with **aord**=9.9516, di44 with **aord**=9.6372 and di62 with **aord**=9.3125. In **Figure 5.33**, The di25 covers the largest order range from 8.2 to 11.2, while the di62 has the flattest curve ranging from 8.8 to 10.1. The di25 has the highest contamination observability, since we can easily judge the contamination degree from its converging iterations. Thus, with low contamination, the di25 only requires averaged iterations of $(8.2-2)=6.2$ to satisfy its **sc** of 5, while it needs averaged iterations of $(11.2-2)=9.2$ to discriminate a stored pattern input with a contamination of -6 dB. Compared to a 3 iteration range of di25, di62 only has a $(10. - 8.8) = 1.3$ iteration range to judge the contamination degree. We may say that the di25 is more economic than the others under light contaminations, and then works harder than the di44 and di62 under heavy noise. From **Figure 5.34**, the di44 has the best performances. The di25 works better in heavy noise, while the di62 leaves more unknowns than the others. The wrong discrimination performances are almost the same for all 4 RCAAM/di's. **Figure 5.35** shows the **aord** of unstored pattern inputs for group **aord9**. Compared to **Figure 5.33**, the aord of 40 dB in **Figure 5.35** is similar to the aord of 3 dB in **Figure 5.33**. Therefore, to a RCAAM/di an unstored pattern is analogous to a stored pattern contaminated with an SNR of about 4 dB. This observation

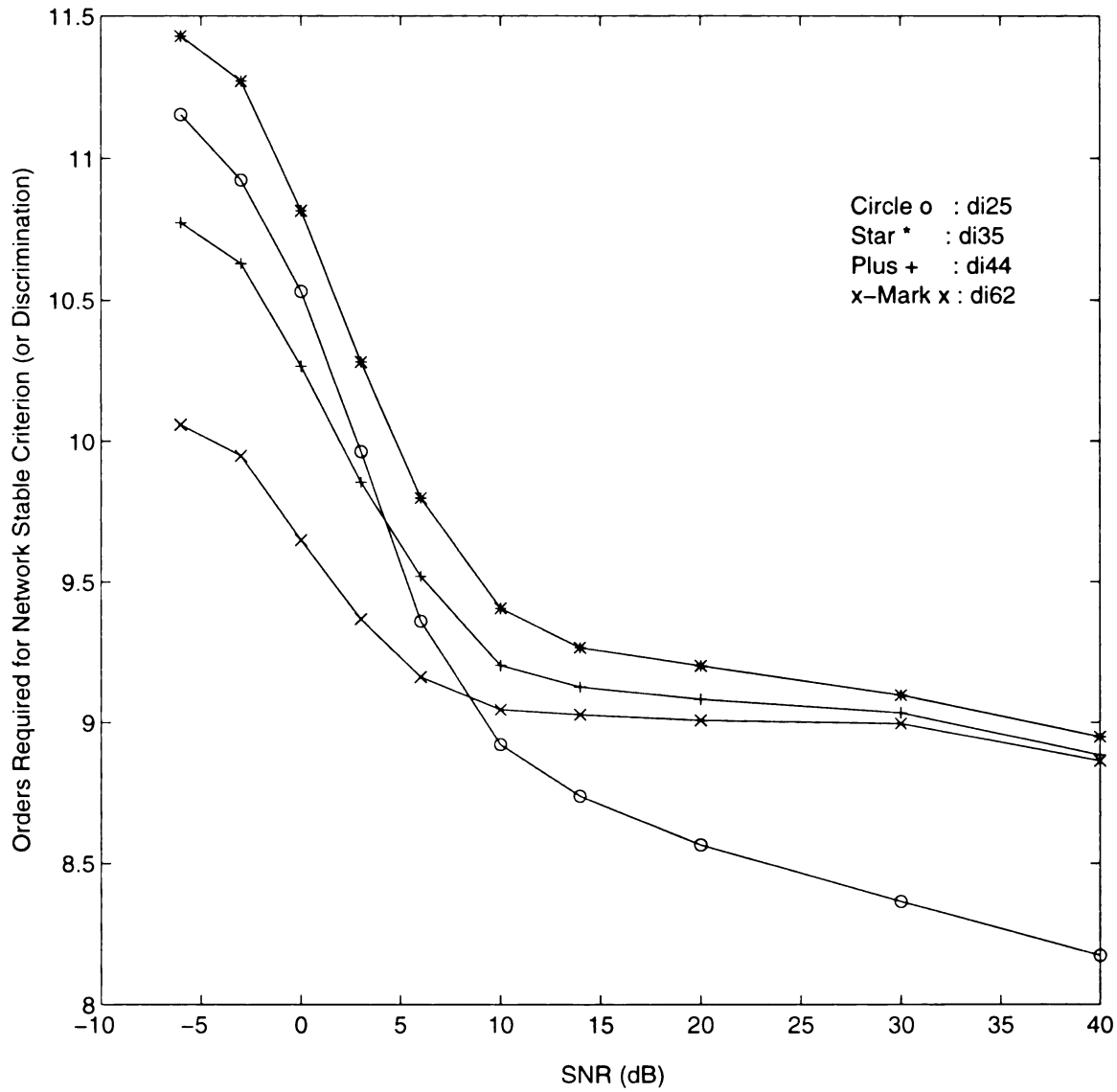


Figure 5.33 Orders required for stability criterion of the RCAAM/di with an average order between 9 and 10 vs. SNR.

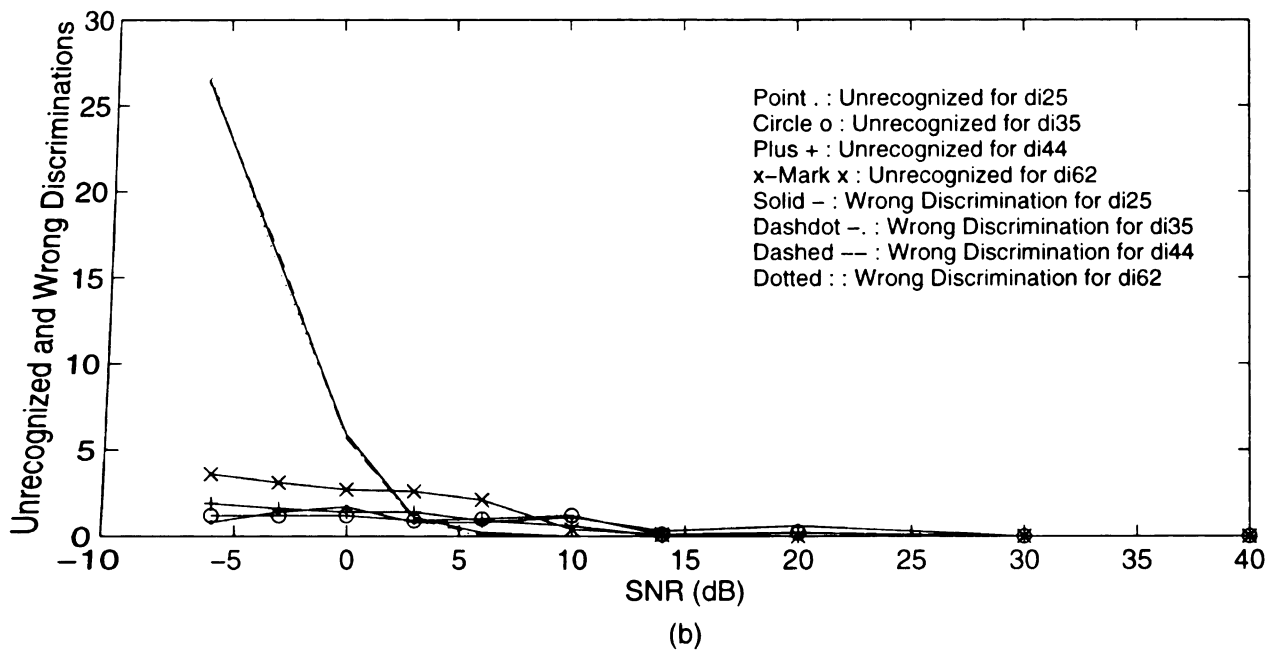
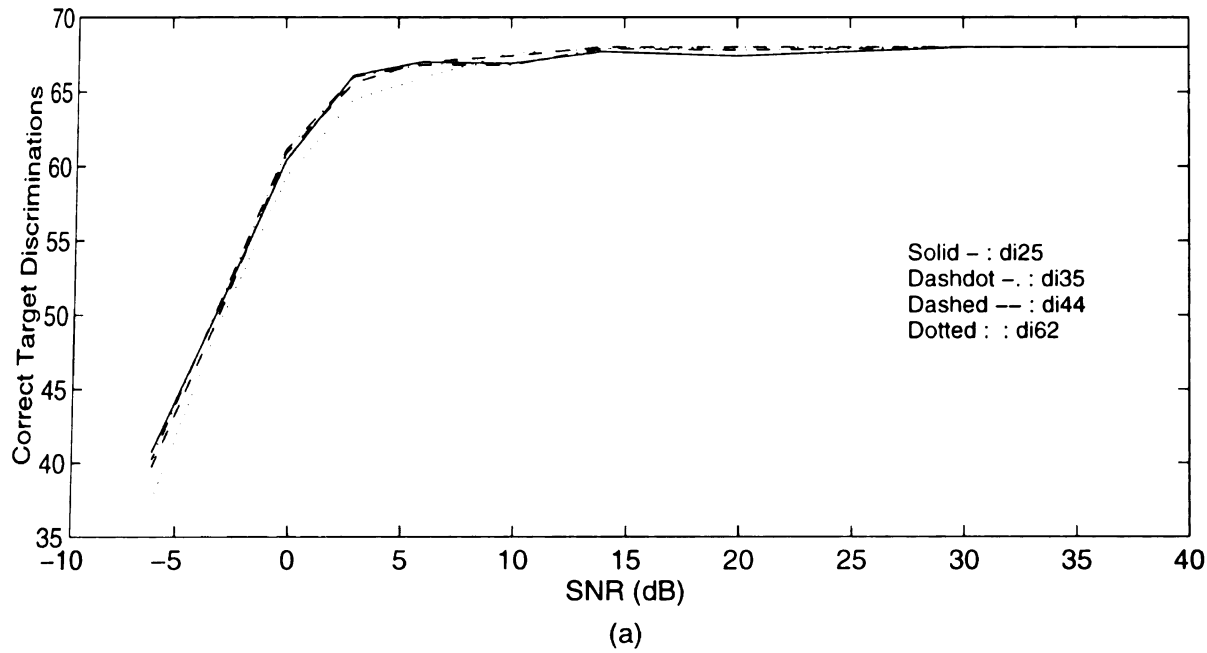


Figure 5.34 Performances of the RCAAM/di with an average order between 9 and 10 vs. SNR for 68 stored patterns.

(a) Correct target discriminations vs. SNR for 68 stored patterns.

(b) Unrecognized and wrong target discriminations vs. SNR for 68 stored patterns.

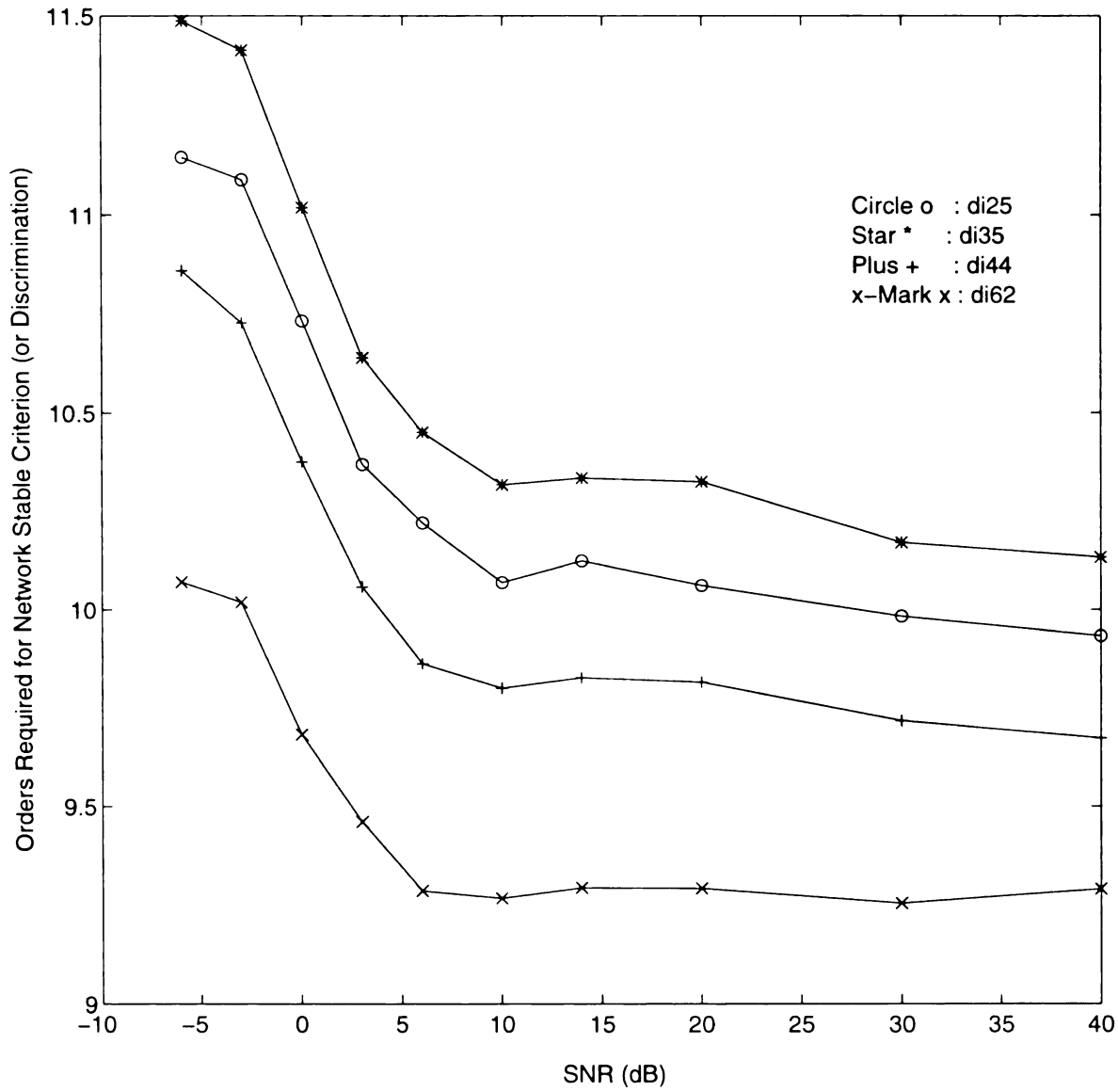


Figure 5.35 Orders required for unstored pattern discriminations of the RCAAM/di with an average order of stored patterns between 9 and 10 vs. SNR.

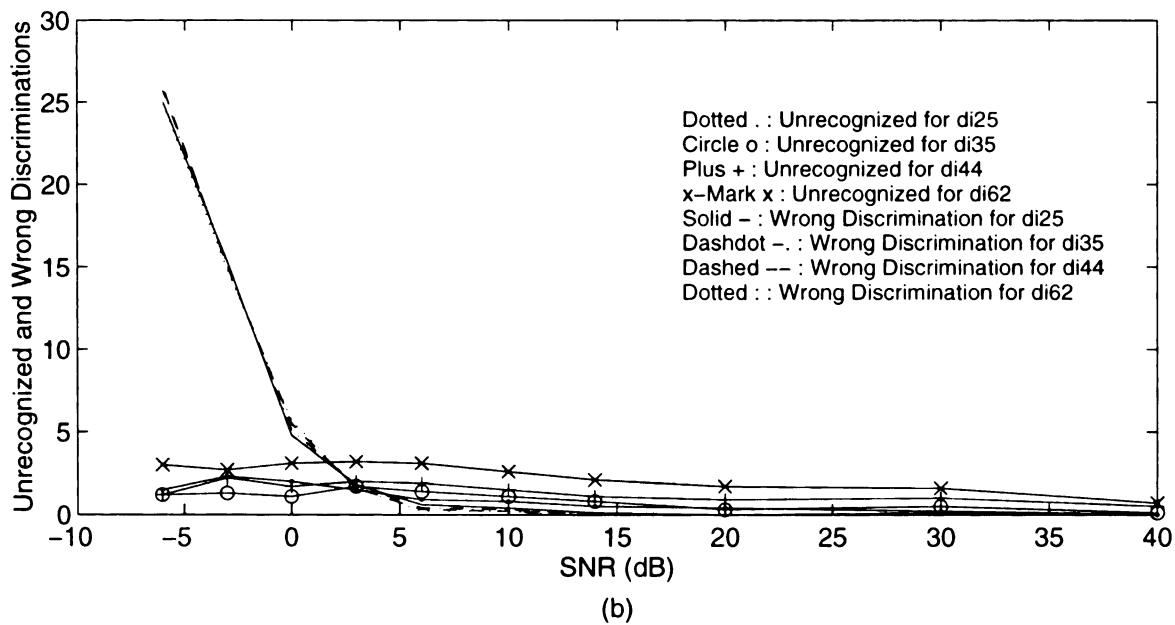
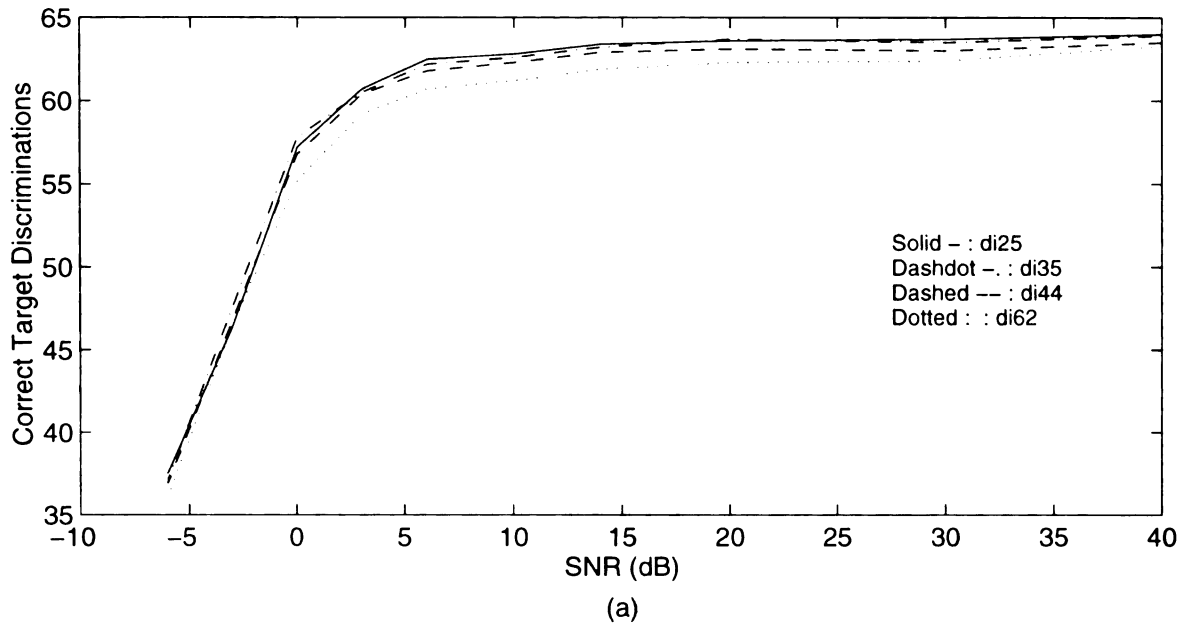


Figure 5.36 Performances of the RCAAM/di with an average order of stored patterns between 9 and 10 vs. SNR for 64 unstored patterns.

(a) Correct target discriminations vs. SNR for 64 unstored patterns.

(b) Unrecognized and wrong target discriminations vs. SNR for 64 unstored patterns.

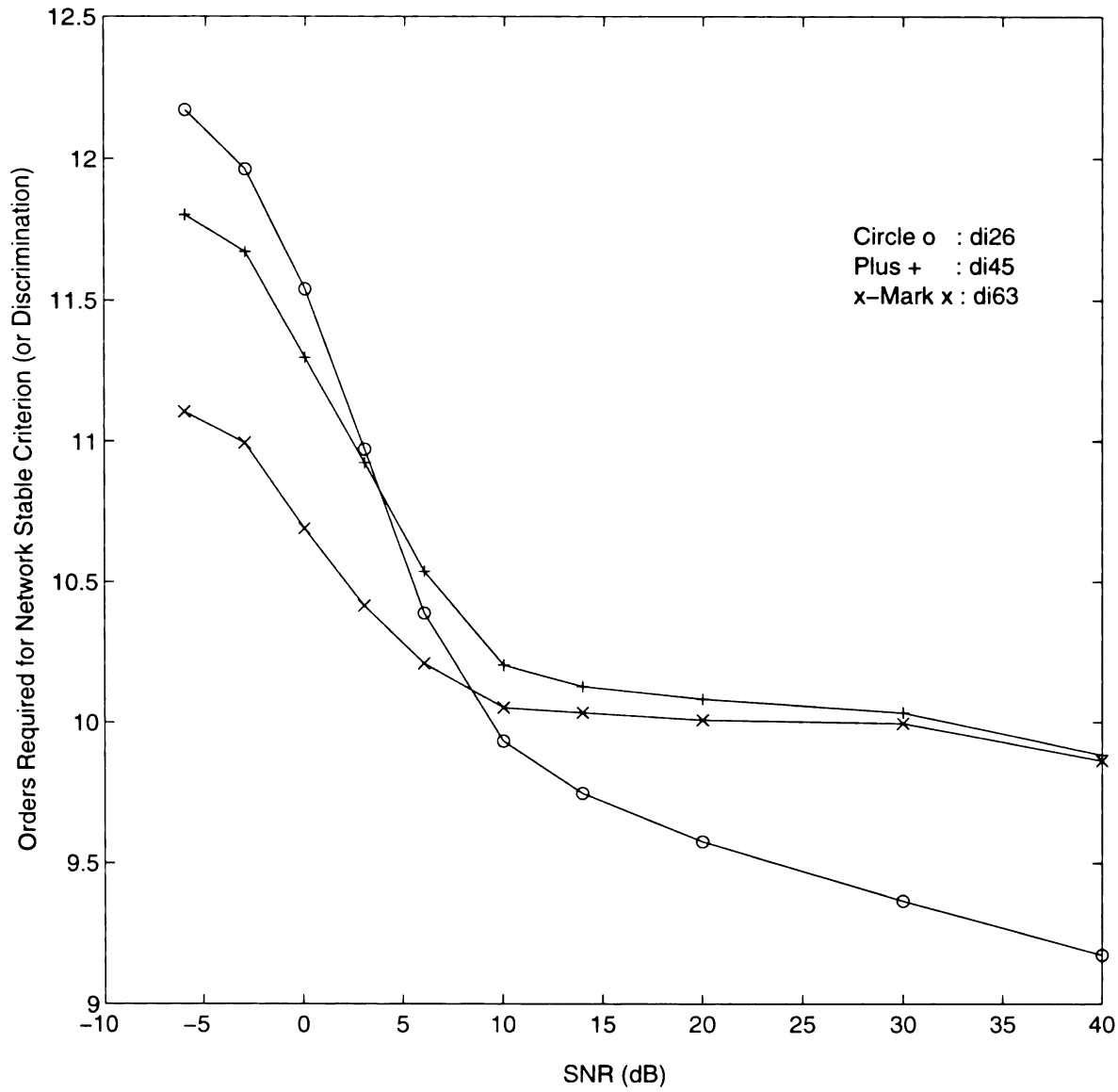


Figure 5.37 Orders required for stability criterion of the RCAAM/di with an average order between 10 and 11 vs. SNR.

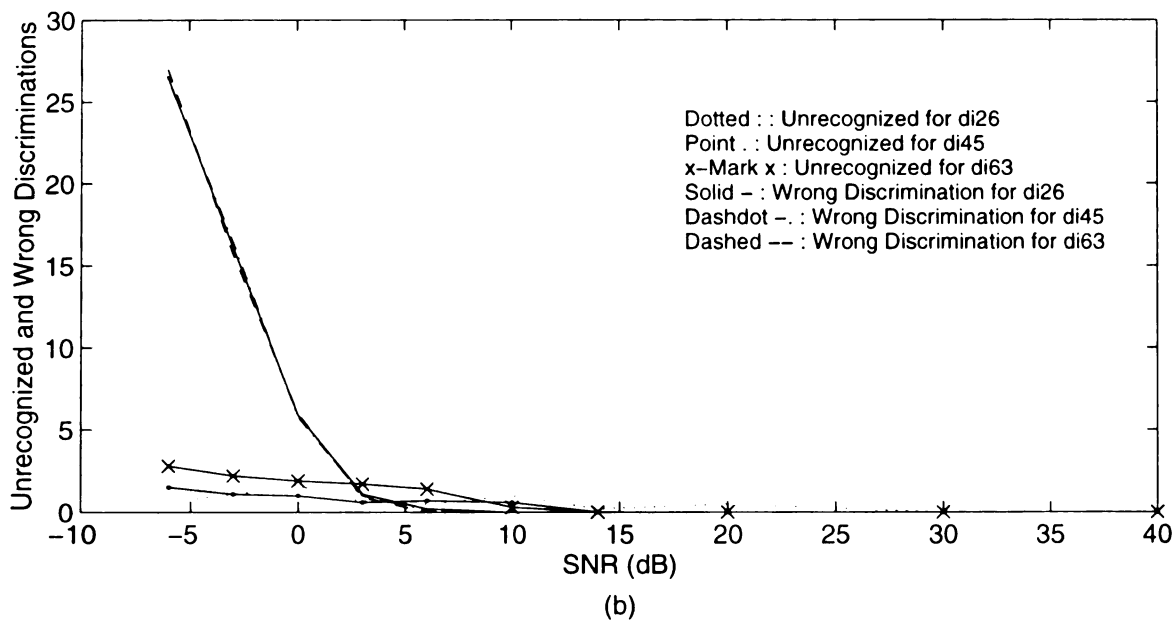
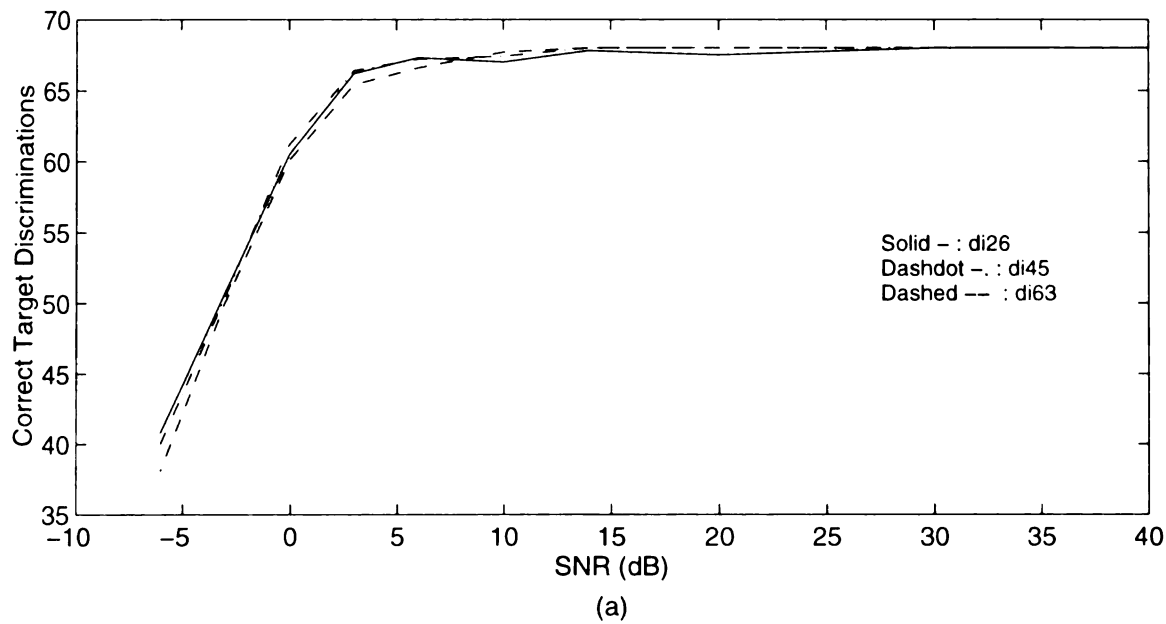


Figure 5.38 Performances of the RCAAM/di with an average order between 10 and 11 vs. SNR for 68 stored patterns.

(a) Correct target discriminations vs. SNR for 68 stored patterns.

(b) Unrecognized and wrong target discriminations vs. SNR for 68 stored patterns.

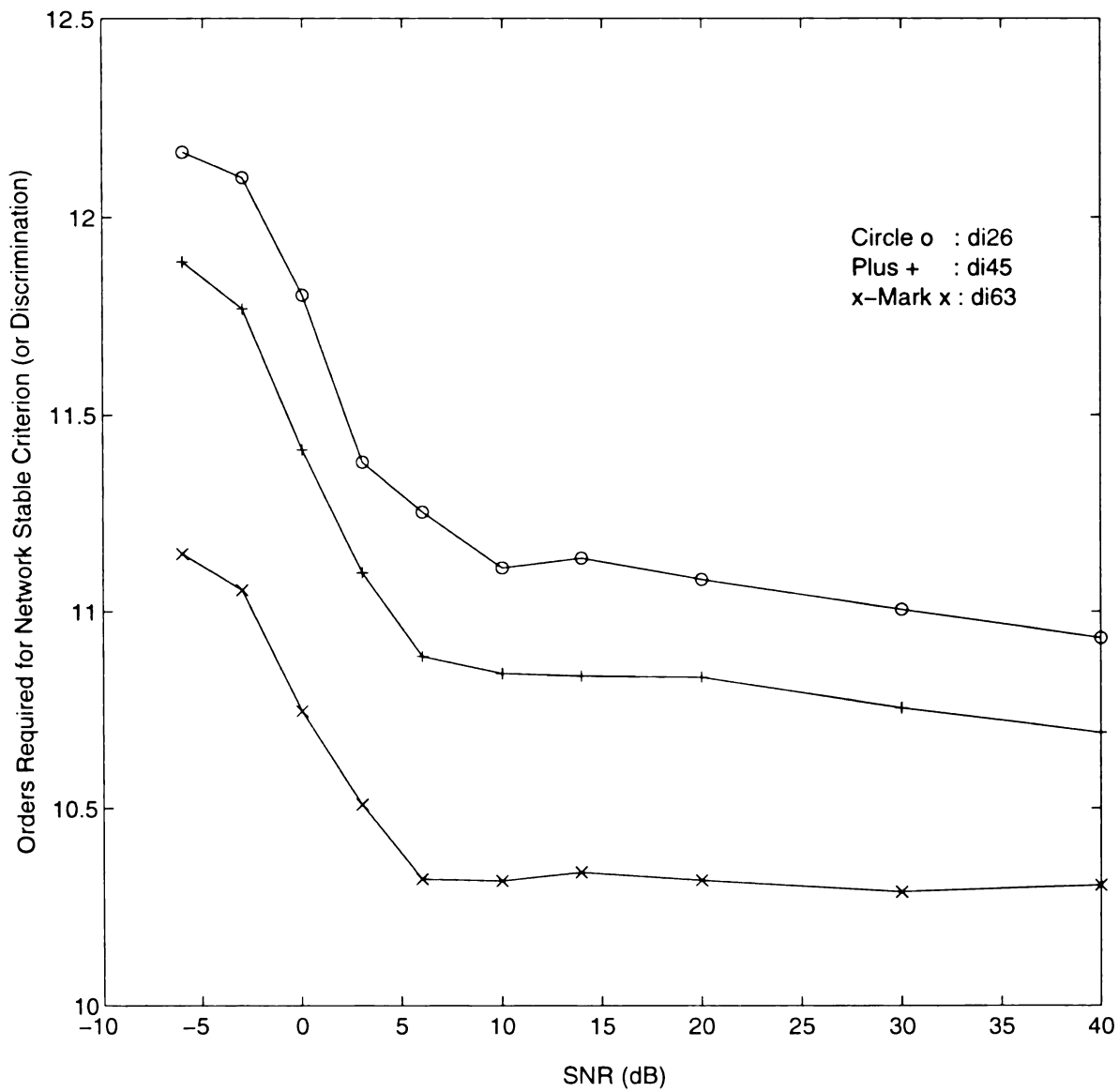


Figure 5.39 Orders required for unstored pattern discriminations of the RCAAM/di with an average order of stored patterns between 10 and 11 vs. SNR.

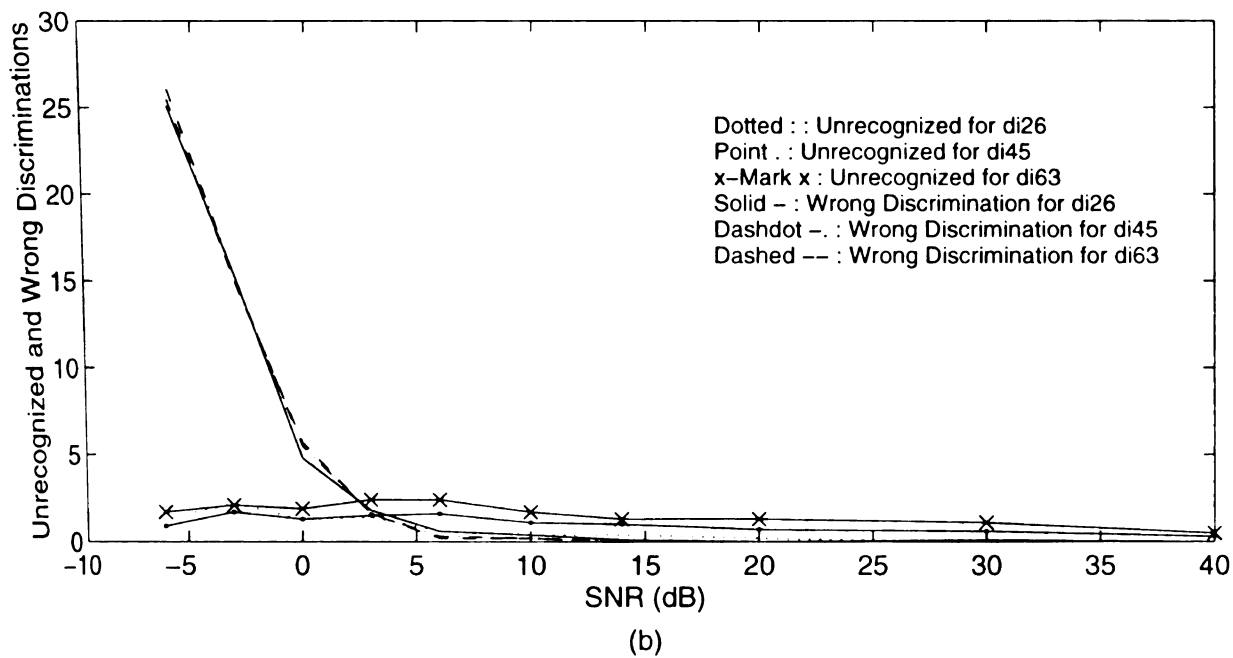
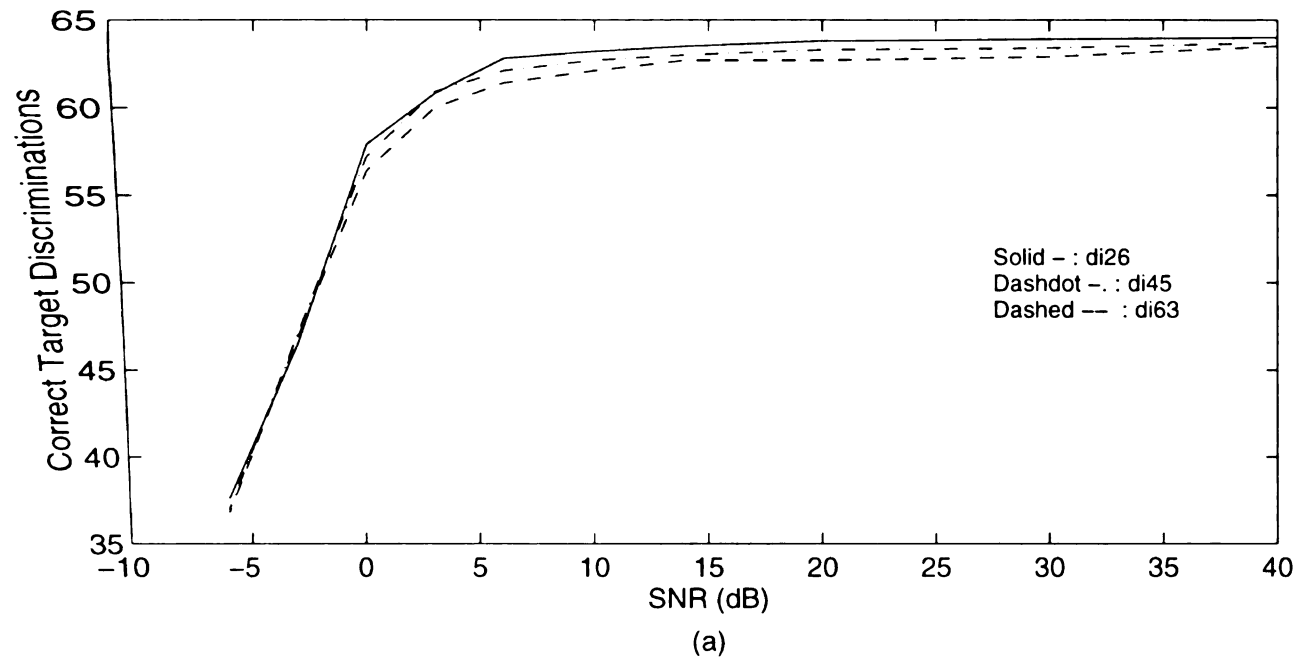


Figure 5.40 Performances of the RCAAM/di with an average order of stored patterns between 10 and 11 vs. SNR for 64 unstored patterns.

(a) Correct target discriminations vs. SNR for 64 unstored patterns.

(b) Unrecognized and wrong target discriminations vs. SNR for 64 unstored patterns.

is also consistent with the hypothesis of factor 2. **Figure 5.36** presents the performance comparisons of group **aord9** with noisy unstored pattern inputs. Again, the wrong discrimination performances are about the same. The di62 has the worst performances, while the di25 (and di35) has the best performances. Again, the deficient performances of di62 for noisy unstored pattern inputs can be explained by factor 2 described in the first subsection of the current section.

Figure 5.37 to **Figure 5.40** show the performance comparisons of 3 RCAAM/di's of group **aord10**, di26 with **aord**=10.4828, di45 with **aord**=10.6562 and di63 with **aord**=10.3365. The di45 and di63 have similar **aord** for lightly contaminated stored pattern inputs in **Figure 5.37** , and their performances are also close to each other in **Figure 5.38** . Again the di63's performances become worse than the di26's when the contaminations of stored patterns are getting severe. For unstored pattern inputs, the di26 has the best performances, the di45 has fair performances, and the di63 has the worst performances in **Figure 5.40** . **Figure 5.41** to **Figure 5.44** show the performance comparisons of 3 RCAAM/di's of group **aord11**, di27 with **aord**=11.4979, di46 with **aord**=11.6647 and di64 with **aord**=11.3581. Compared to group **aord10**, the **aord**'s of di45 and di63 are almost equal to the **aord**'s of di46 and di64 plus 1 for distorted stored pattern inputs. It means an increment of 1 in stability criterions of di45 and di63 didn't apparently improve their performances much for fairly contaminated stored pattern inputs. Or, equivalently, the di45 and di63, compared to the di46 and di64, already have the ability to correctly discriminate lightly contaminated stored patterns.

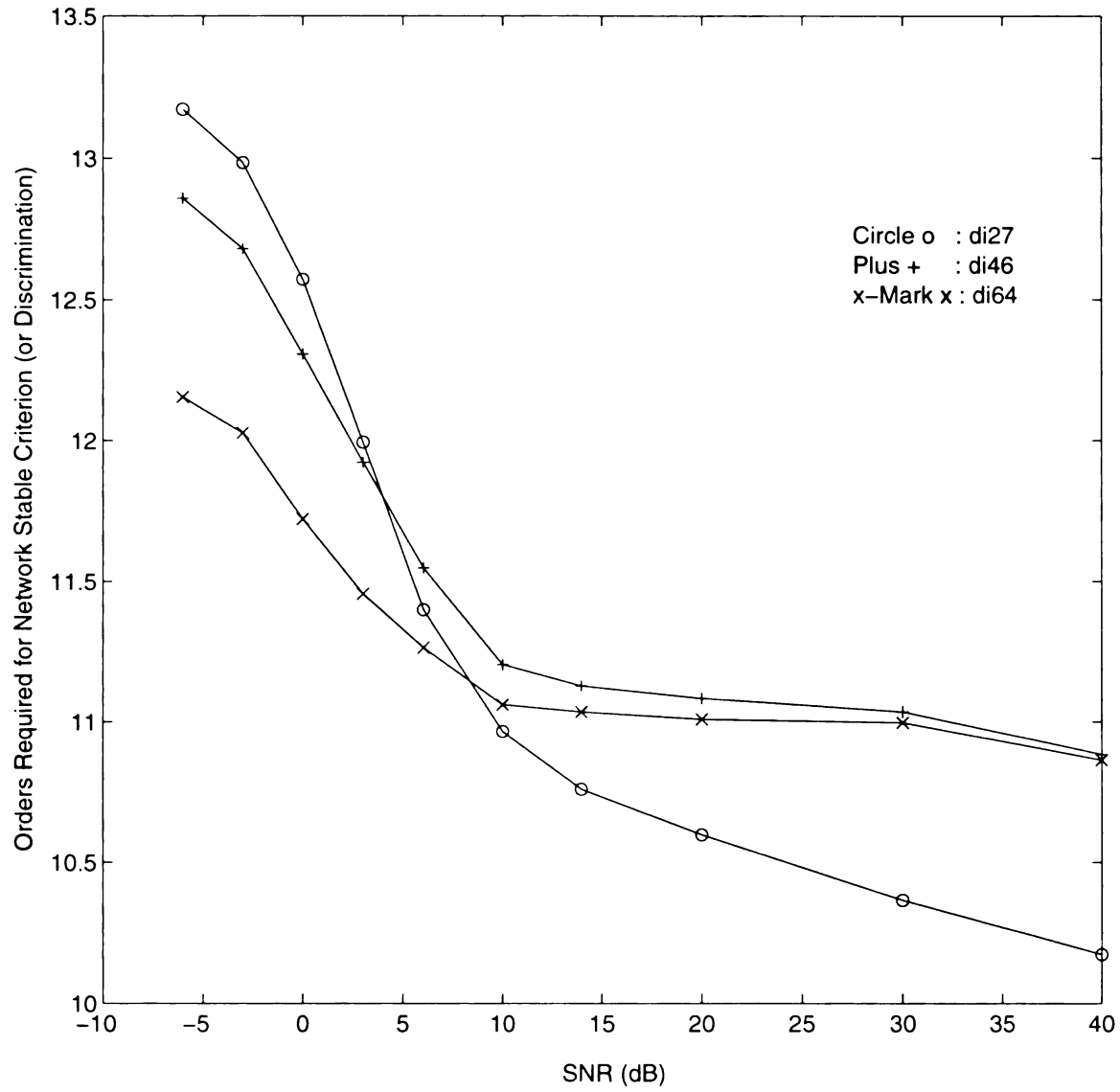


Figure 5.41 Orders required for stability criterion of the RCAAM/di with an average order between 11 and 12 vs. SNR.

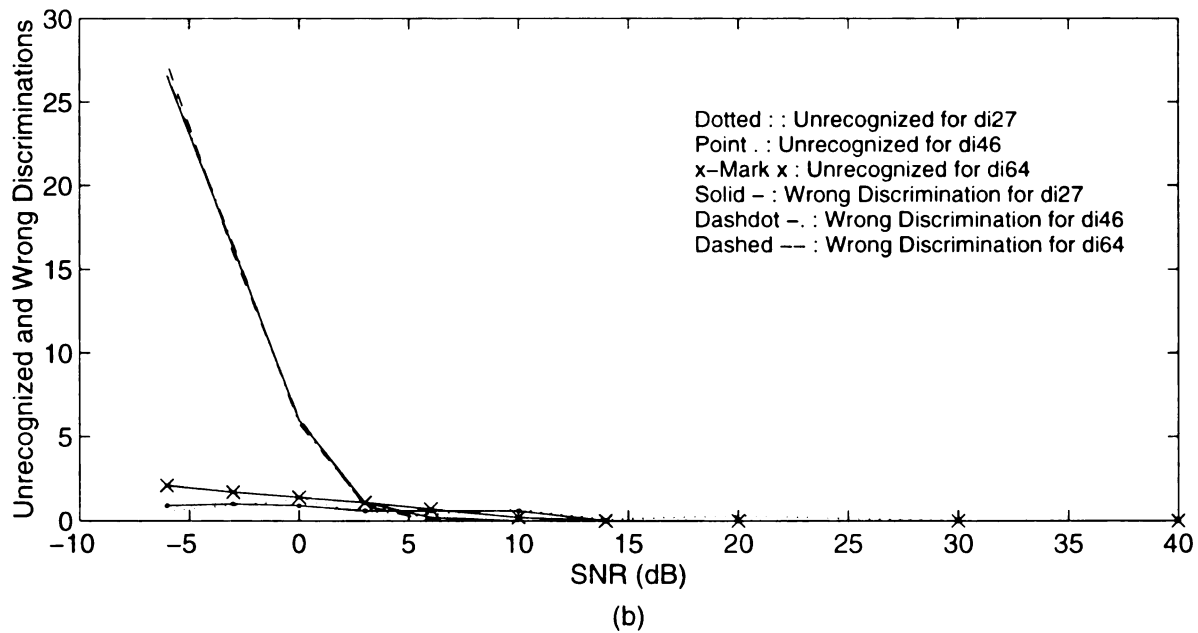
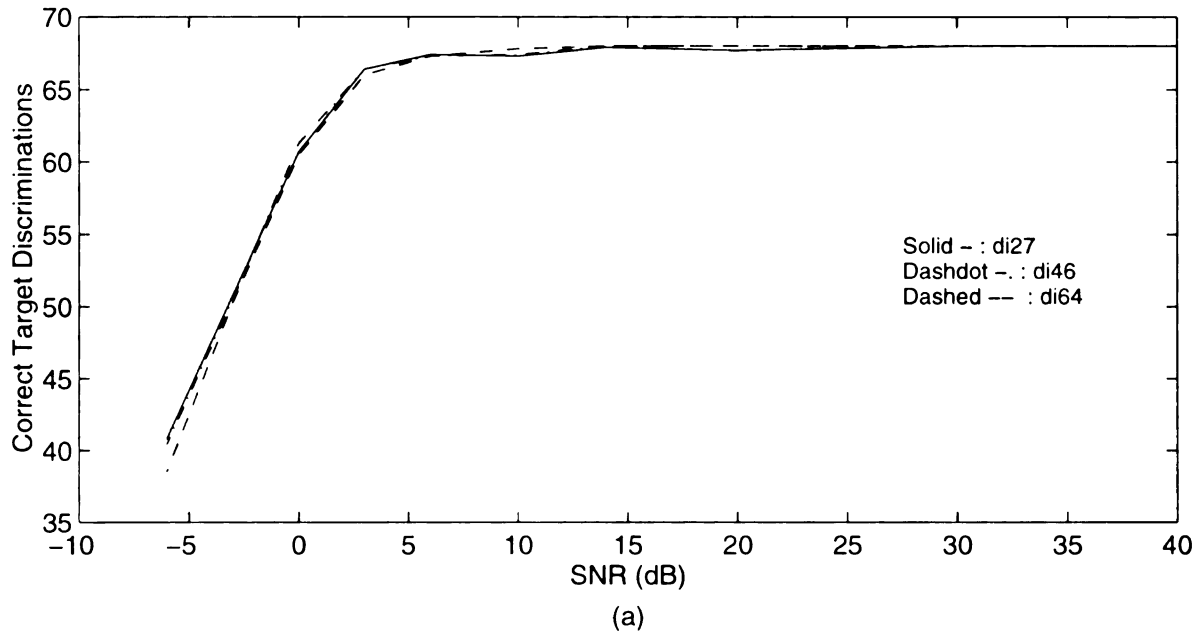


Figure 5.42 Performances of the RCAAM/di with an average order between 11 and 12 vs. SNR for 68 stored patterns.

(a) Correct target discriminations vs. SNR for 68 stored patterns.

(b) Unrecognized and wrong target discriminations vs. SNR for 68 stored patterns.

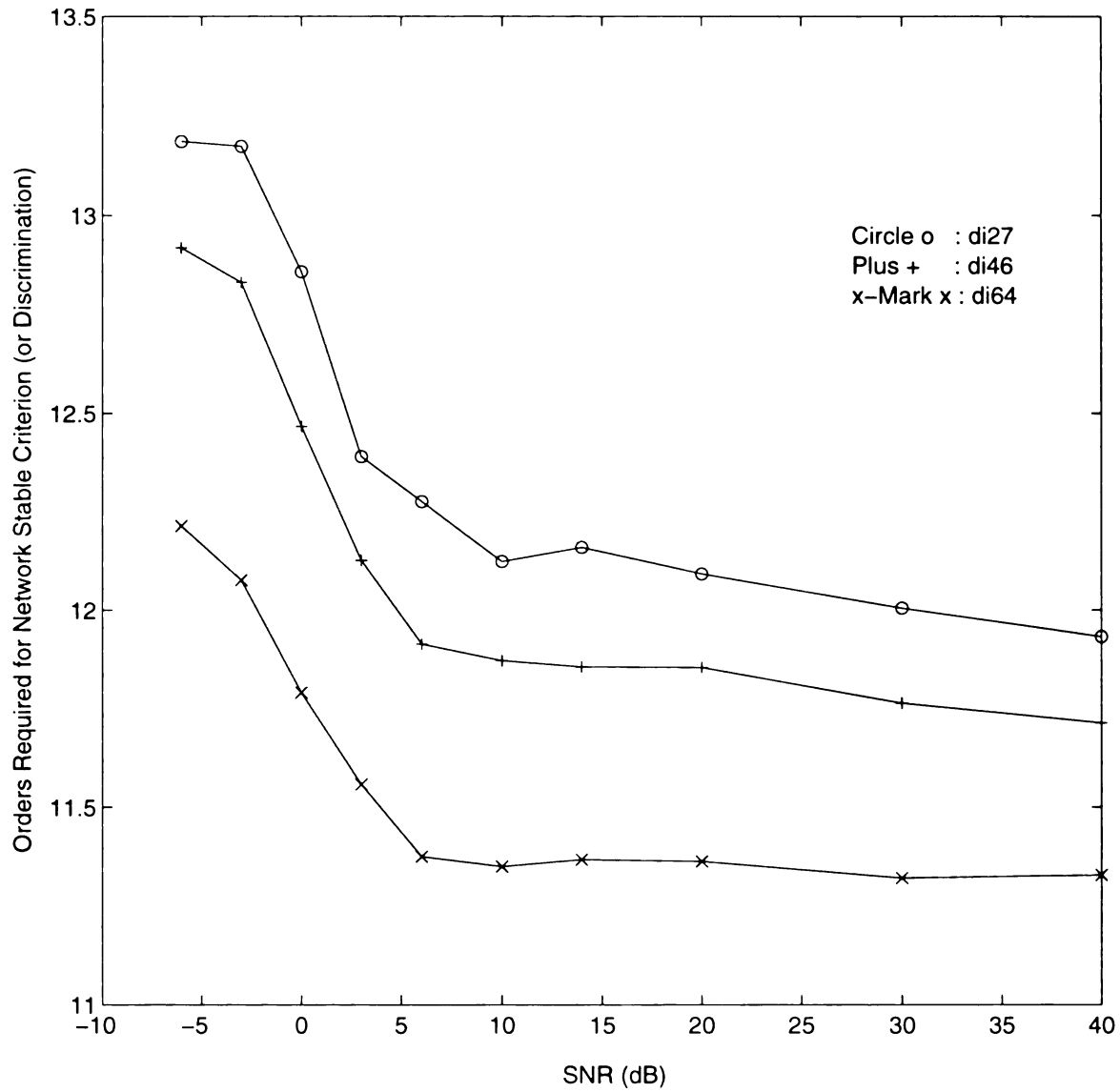


Figure 5.43 Orders required for unstored pattern discriminations of the RCAAM/di with an average order of stored patterns between 11 and 12 vs. SNR.

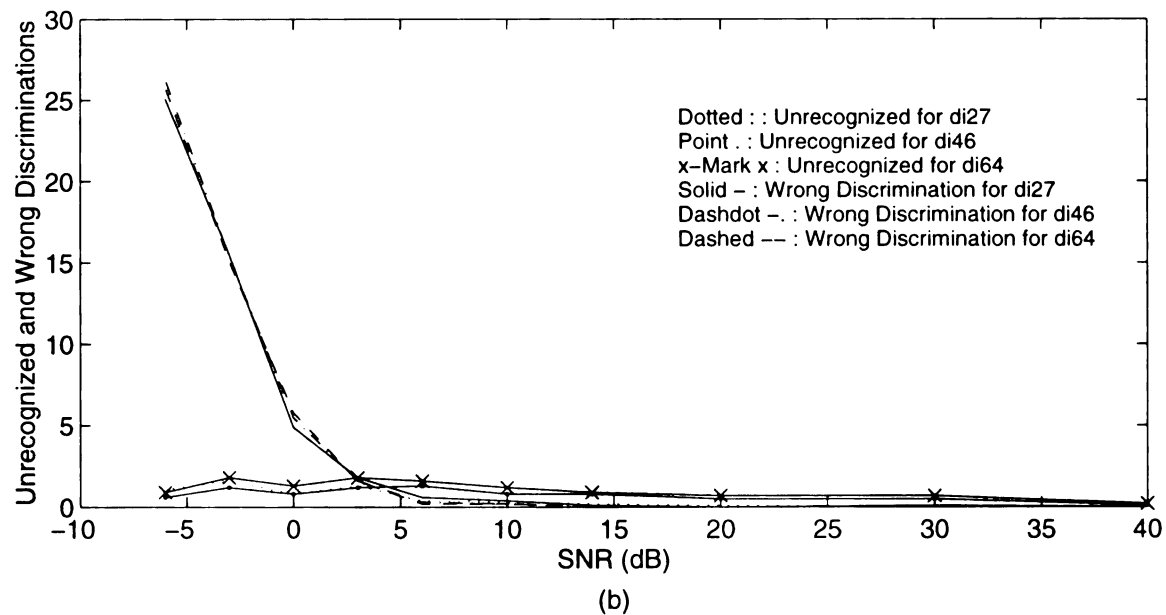
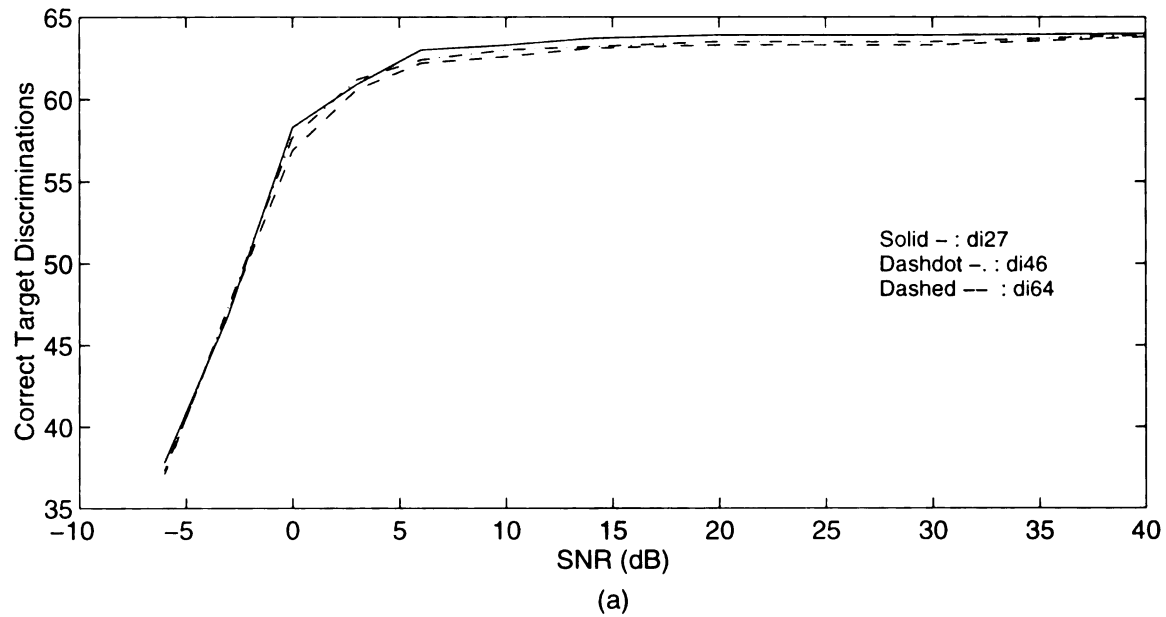


Figure 5.44 Performances of the RCAAM/di with an average order of stored patterns between 11 and 12 vs. SNR for 64 unstored patterns.

(a) Correct target discriminations vs. SNR for 64 unstored patterns.

(b) Unrecognized and wrong target discriminations vs. SNR for 64 unstored patterns.

Figure 5.45 to **Figure 5.48** show the performance comparisons of 3 RCAAM/di's of group **aord**13, di29 with **aord**=13.5368, di48 with **aord**=13.6843 and di66 with **aord**=13.3875. Comparing **Figure 5.45** with **Figure 5.41** , again the **aord**'s of di46 and di64 are almost equal to the **aord**'s of di48 and di66 plus 2 for distorted stored pattern inputs, but the aord of di29 has increments larger than 2 for 10 dB and 20 dB. Therefore, by comparing **Figure 5.46** with **Figure 5.42** , the di29 has great improvements for 20dB and 10 dB where the di27 still left few unknowns behind. This indicates that some contaminated stored patterns for 20 dB or 10 dB reside on spurious states which are stable for at least 7 recurrent iterations but become unstable on the 8th or 9th iteration. The spurious states would be considered as stable outputs when an **sc** of 7 is used by a RCAAM/di2. If the spurious states become unstable on the 8th iteration after the previous 7 stable iterations, then the di29 should at least take another 9 iterations to satisfy its **sc** of 9. Since the **aord** is calculated from the discrimination order divided (or averaged) by 68 stored patterns and 10 time simulations, the increments of **aord** for 20 dB and 10 dB for the di29 are small. This time the di29 has the best performances for both contaminated stored and unstored pattern inputs, while the di66 still has trouble to converge unknowns for unstored pattern inputs because of low elasticity (factor 2).

5.4 Performance of the Recurrent Correlation Accumulation Adaptive Memory with fixed input (RCAAM/fi)

In this section we simulate the RCAAM/fi's with fixed initial order but variable stability criteria, and the RCAAM/fi's with fixed **sc** but variable initial orders. Since the RCAAM/fi always has the original input as the network recurrent operation input and no

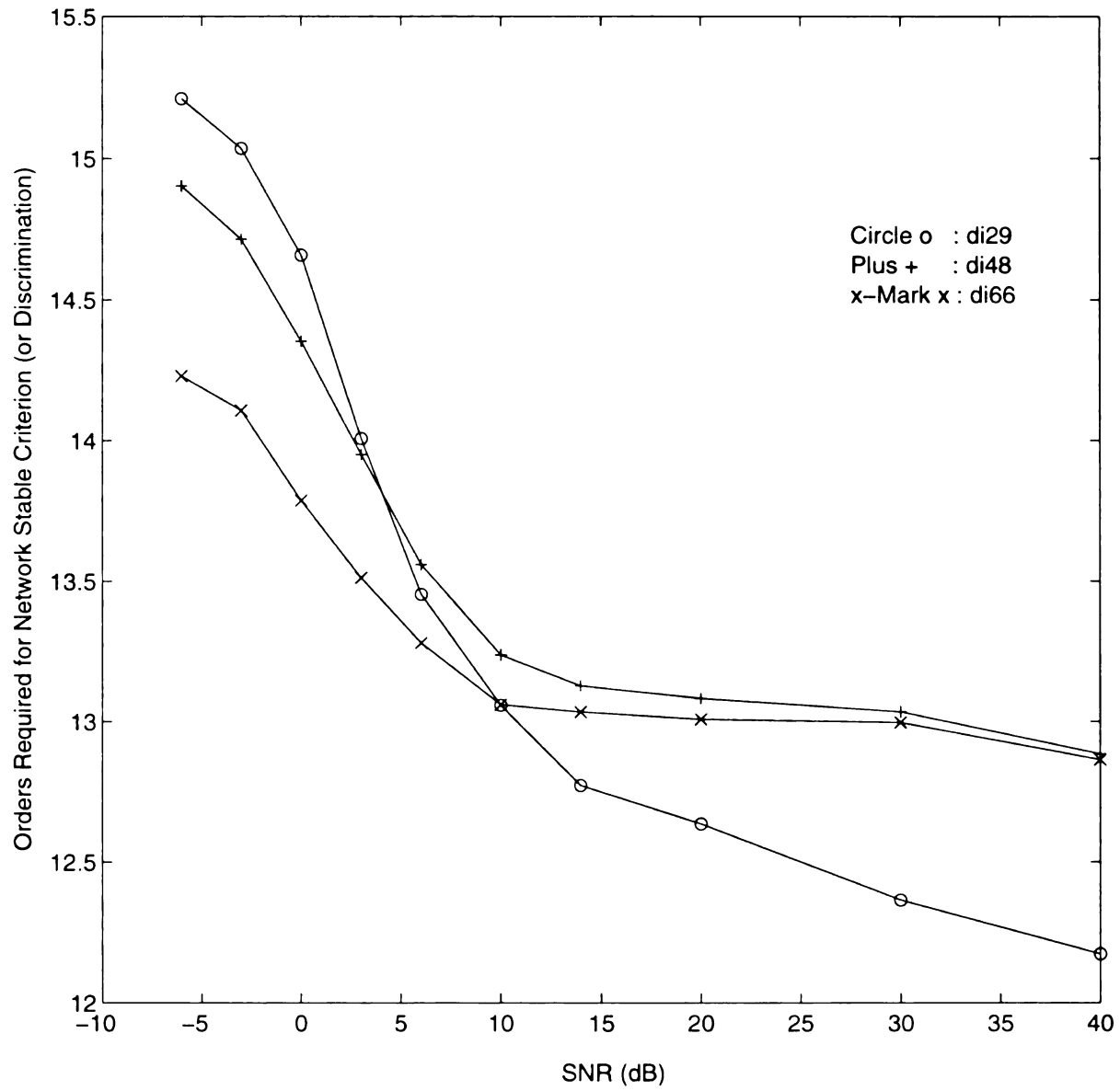


Figure 5.45 Orders required for stability criterion of the RCAAM/di with an average order between 13 and 14 vs. SNR.

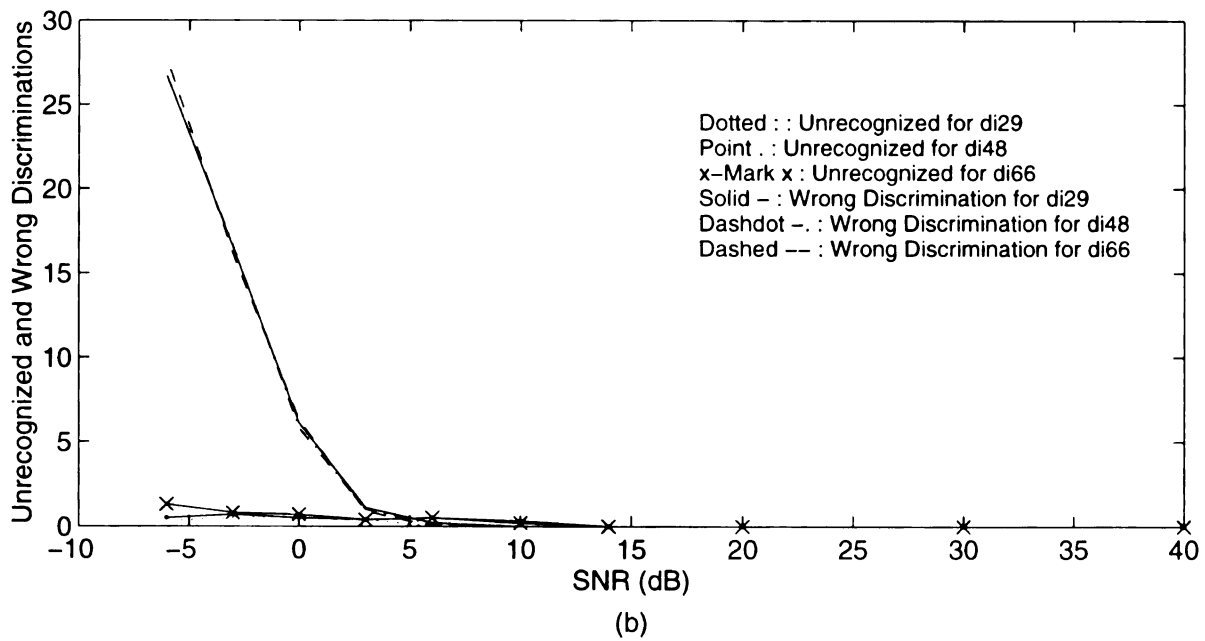
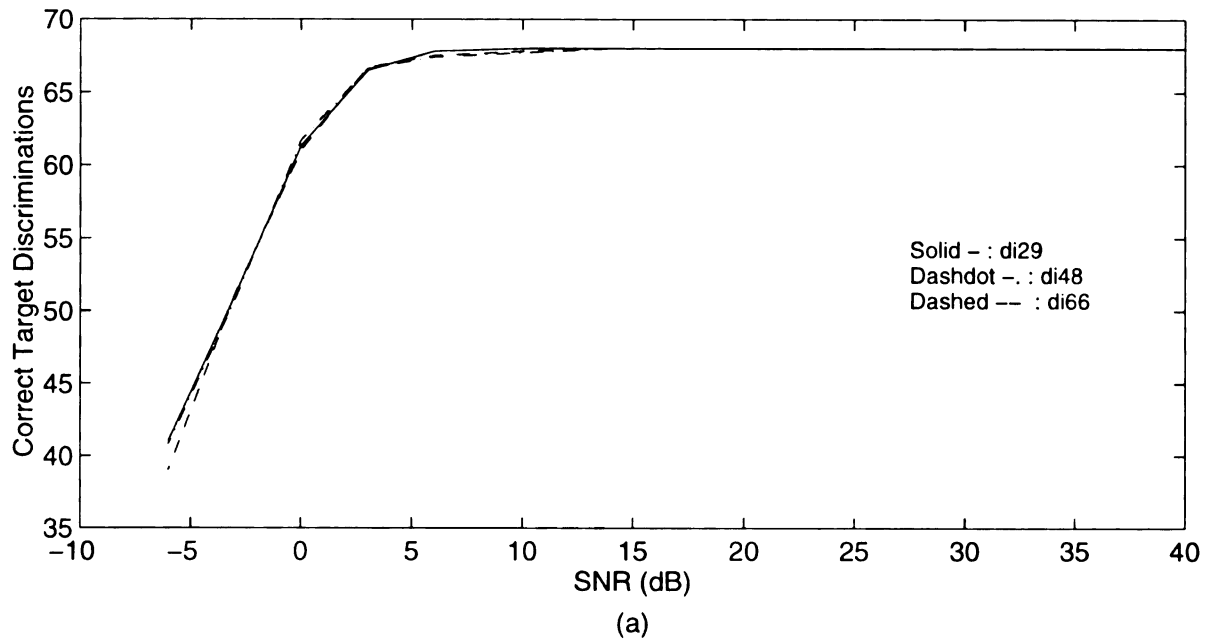


Figure 5.46 Performances of the RCAAM/di with an average order between 13 and 14 vs. SNR for 68 stored patterns.

(a) Correct target discriminations vs. SNR for 68 stored patterns.

(b) Unrecognized and wrong target discriminations vs. SNR for 68 stored patterns.

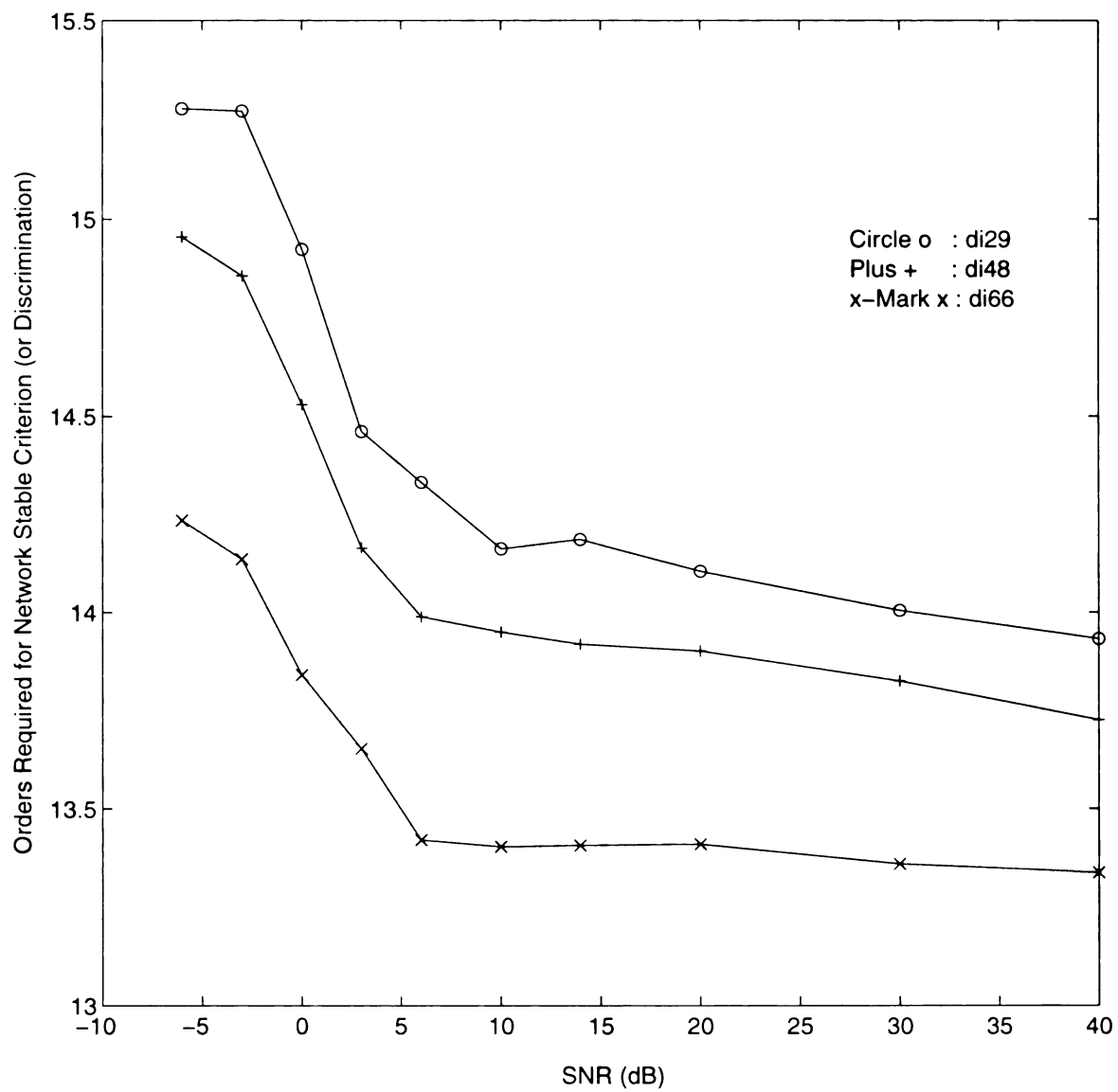


Figure 5.47 Orders required for unstored pattern discriminations of the RCAAM/di with an average order of stored patterns between 13 and 14 vs. SNR.

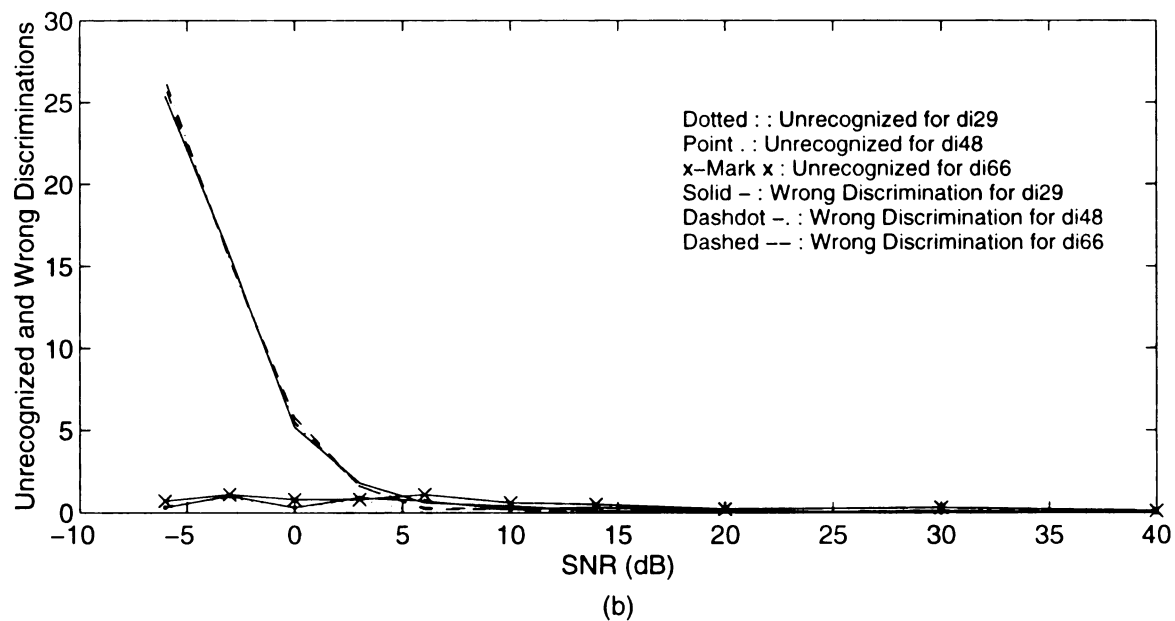
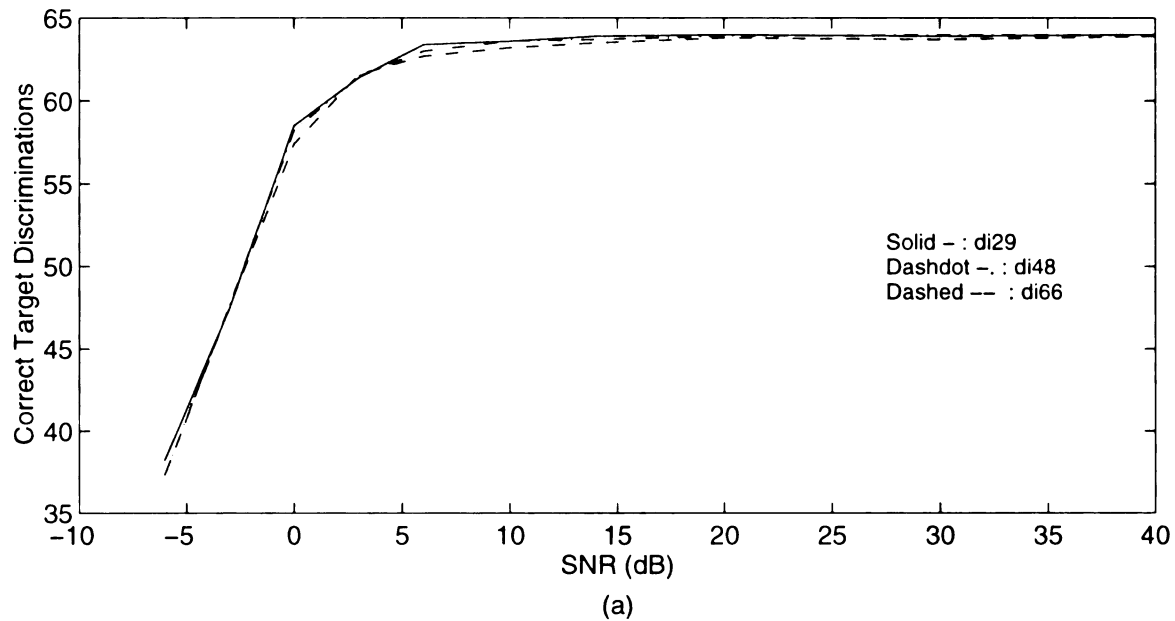


Figure 5.48 Performances of the RCAAM/di with an average order of stored patterns between 13 and 14 vs. SNR for 64 unstored patterns.

(a) Correct target discriminations vs. SNR for 64 unstored patterns.

(b) Unrecognized and wrong target discriminations vs. SNR for 64 unstored patterns.

feedback to the network input from output, its performances are supposed to be independent of initial orders. But an inappropriate selection of maximum convergent criterion and initial order may result in divergence, if the correlation with the largest scale among all stored patterns is negative. The aord of RCAAM/fi is also shown here.

Figure 5.49 shows the unrecognized discriminations of RCAAM/fi with an initial order of 1 (RCAAM/fi1 or fi1) vs. **sc** for stored patterns under 40, 14 and 0 dB SNR. With 40 dB an **sc** as small as 2 can converge all unknowns, with 14 dB an **sc** of 10 then converges all unknowns, but an **sc** as large as 40 will still leave 8.4% unknown for 0 dB. This indicates that the RCAAM/fi can converge unknowns for lightly contaminated stored patterns, but can't converge them under severe noise even when a high **sc** is adopted. The bidirectional amplifications of bipolar form and the recurrent mechanism of RCAAM/fi should account for this phenomenon. For example, the bipolar correlation between $[-1 \ 1]$ and $[-1 \ 1]$ will have a positive amplification of 2, while the correlation between $[-1 \ 1]$ and $[1 \ -1]$ gives a negative amplification of -2. Under severe distortion and with inconsistent responses resulting from uncertain beginning point allocation, the encoded input bipolar code may have negative correlation gain with some stored patterns. Suppose the correlation between the input and the stored pattern $\mathbf{x}$ has a negative gain, $-G_x$, which has the largest scale among all correlation gains. Then the RCAAM/fi1 will have a positive amplification of $(-G_x)^{1-2k+1}$ on stored pattern $\mathbf{x}$ at an odd iteration, $2k+1$, while it has a negative amplification of $(-G_x)^{1-2k}$ on stored pattern $\mathbf{x}$ at an even iteration, $2k$. Since a correlation gain with the largest scale will dominate the convergence, especially at a large iteration or with a high **sc**, the RCAAM/fi will finally oscillate between the negatively amplified pattern $\mathbf{x}$ and the positively amplified pattern $\mathbf{x}$.

Therefore, it can't converge if there is no maximum convergent cycle set.

Figure 5.50 shows the correct pattern discrimination of RCAAM/fi1 in (a) and wrong pattern discrimination in (b) vs. **sc** for stored patterns, while **Figure 5.51** presents the correct target discrimination in (a) and wrong target discrimination in (b). Under light contaminations, there are no wrong discriminations from RCAAM/di1, but the correct discriminations increase along with **sc**. Under 0 dB SNR, the correct target discrimination increases by $(83.5 - 77.3) = 6.2\%$ when the **sc** changes from 2 to 10, but the wrong discrimination only increases by 1%. Therefore, the existence of **sc** effectively improves the performances of RCAAM/fi.

Figure 5.52 shows the unrecognized discrimination of RCAAM/fi1 vs. **sc** for unstored pattern inputs under 40, 14 and 0 dB SNR, while **Figure 5.53** presents the correct target discrimination in (a) and wrong target discrimination in (b). Under light contaminations, there are no wrong discriminations from RCAAM/di1, but the correct discriminations increase by 11.3% for 40 dB and 9.8% for 14 dB when the **sc** changes from 2 to 10. For 0 dB, the correct target discrimination increases by 4.4% when the **sc** changes from 2 to 10, but the wrong discrimination only increases by 2%. Similarly as described in RCAAM/di case, the RCAAM/fi1's performances for unstored pattern inputs under severe distortion haven't behaved worse than its performances for stored pattern inputs.

Figure 5.54 shows the unrecognized discriminations of RCAAM/fi with an **sc** of 2 (RCAAM/fi_2 or fi_2) vs. initial order for stored patterns under 40, 14 and 0 dB SNR. With an SNR of 40 dB, there is no unknown left. With 0 dB, there is a strange oscillation along with initial order. Let's discuss this phenomenon after checking pattern and target discrimination performances. **Figure 5.55** shows the correct pattern discrimination of

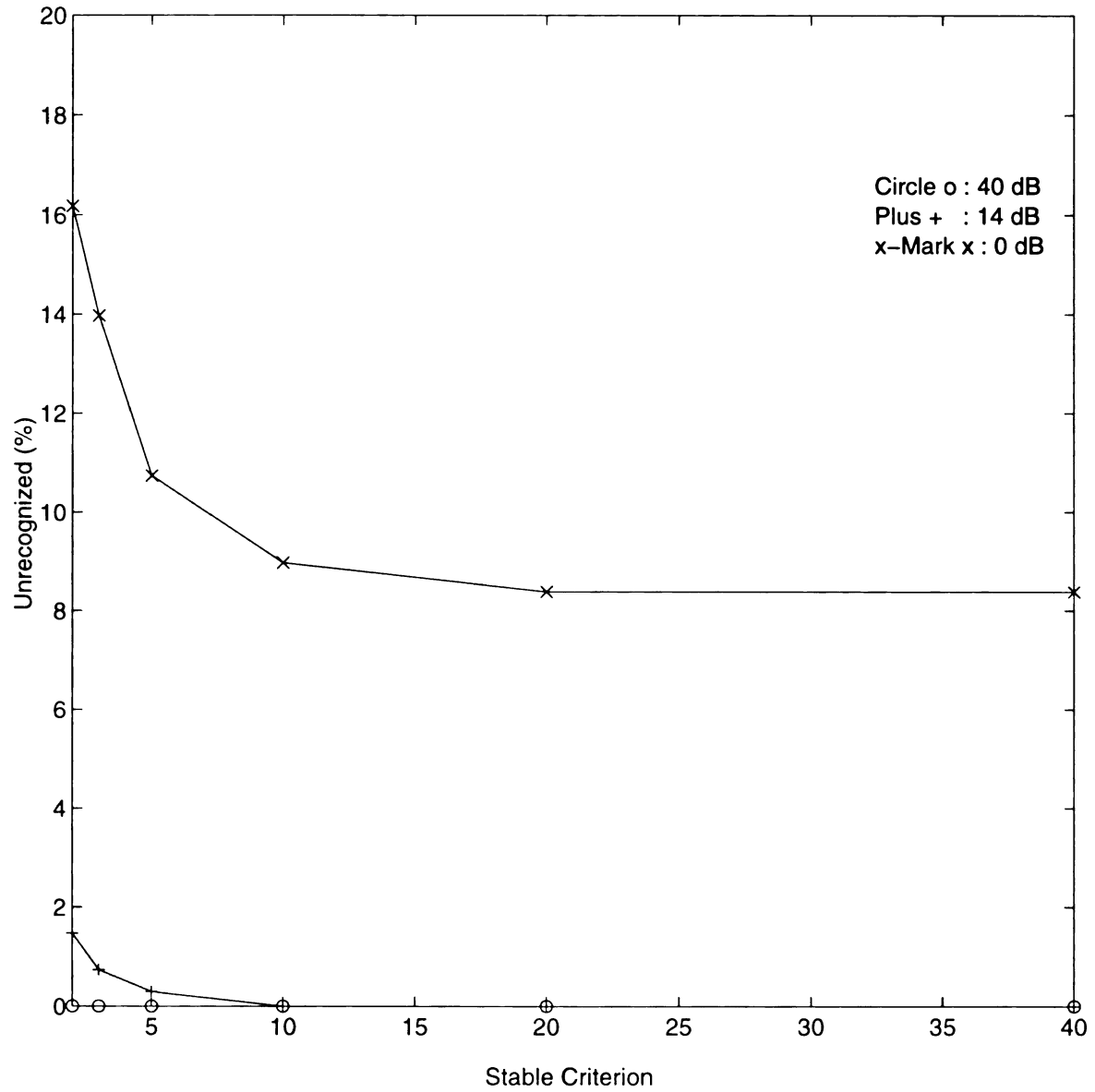


Figure 5.49 Unrecognized discrimination of the RCAAM/fi with an initial order of 1 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns.

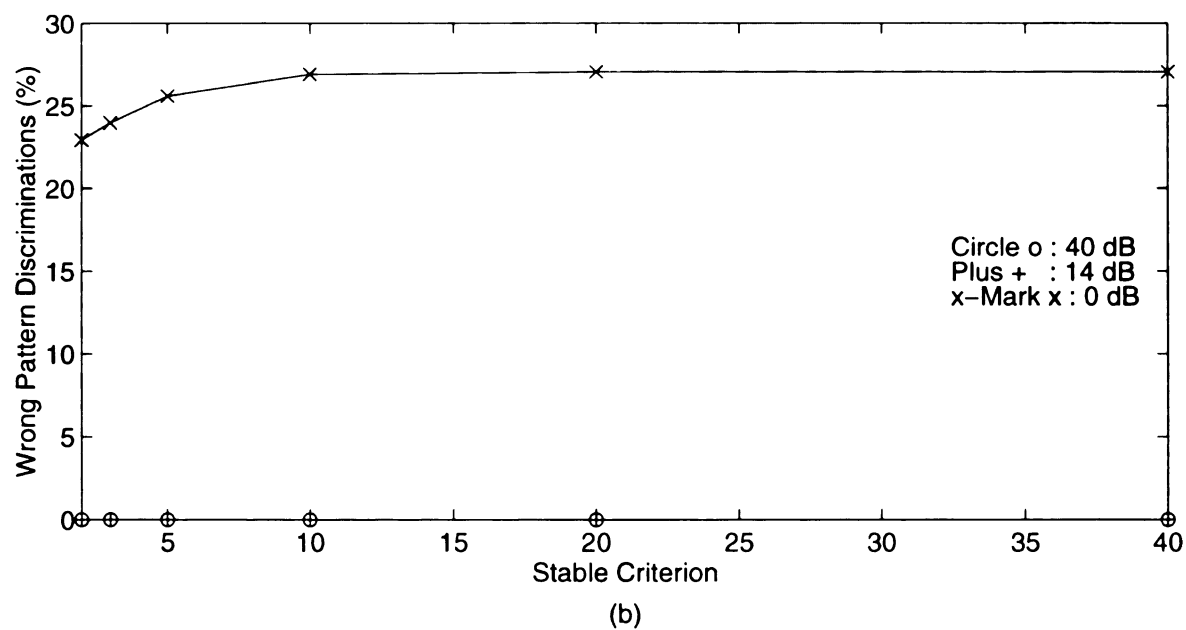
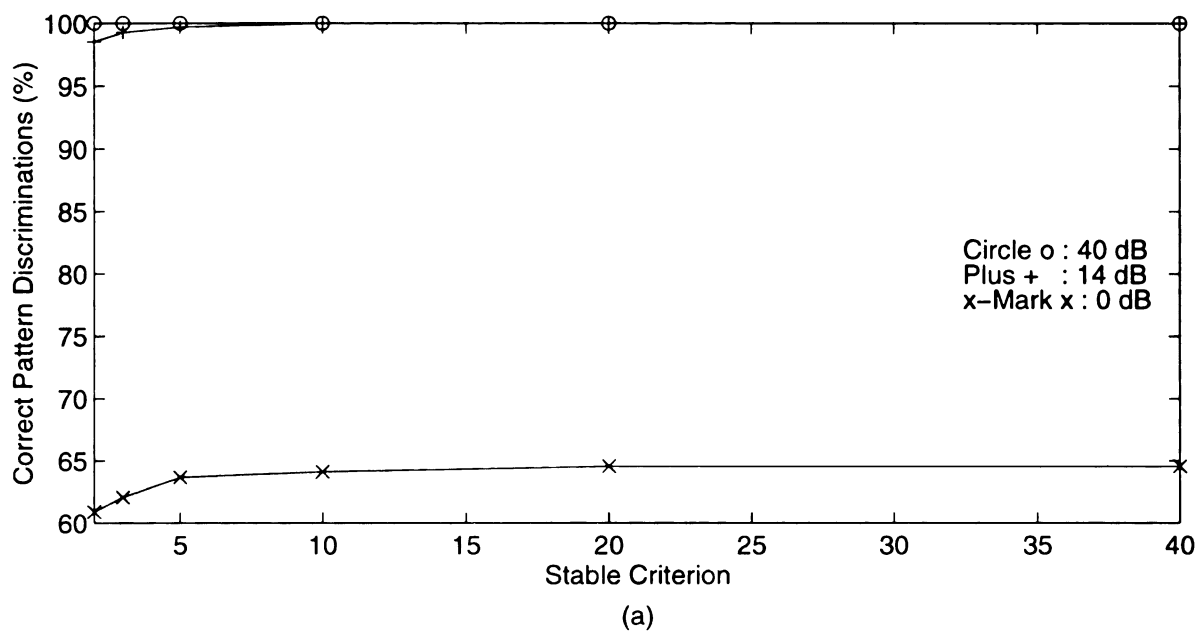


Figure 5.50 Correct and wrong pattern discrimination performances of the RCAAM/fi with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.

(a) Correct pattern discrimination performances vs. stability criterion.

(b) Wrong pattern discrimination performances vs. stability criterion.

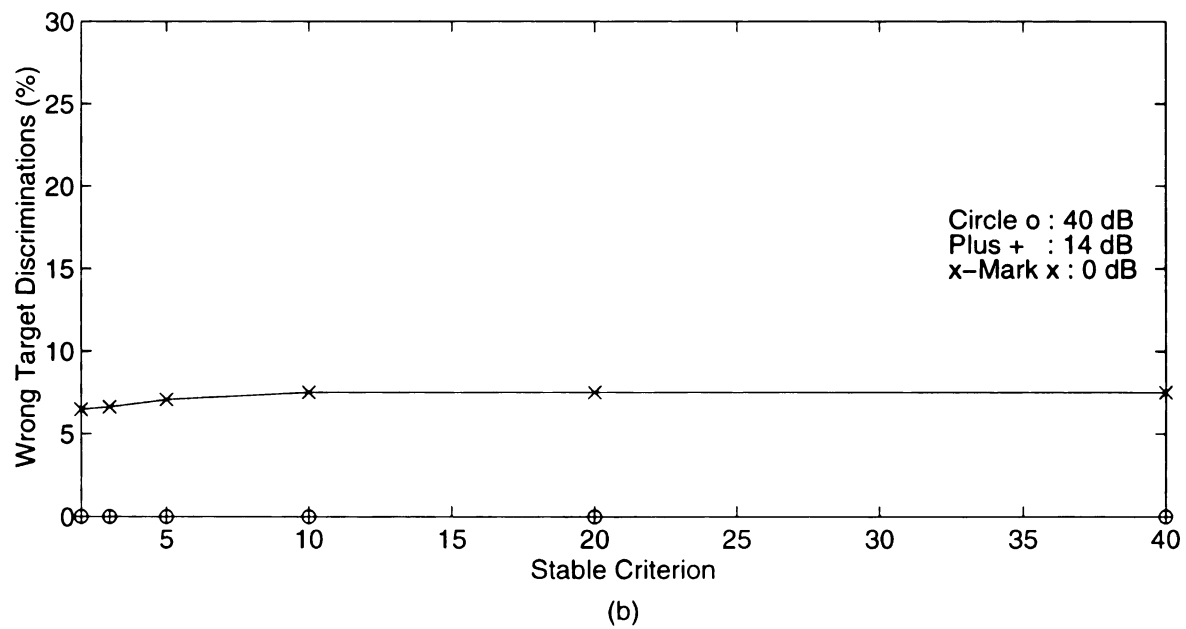
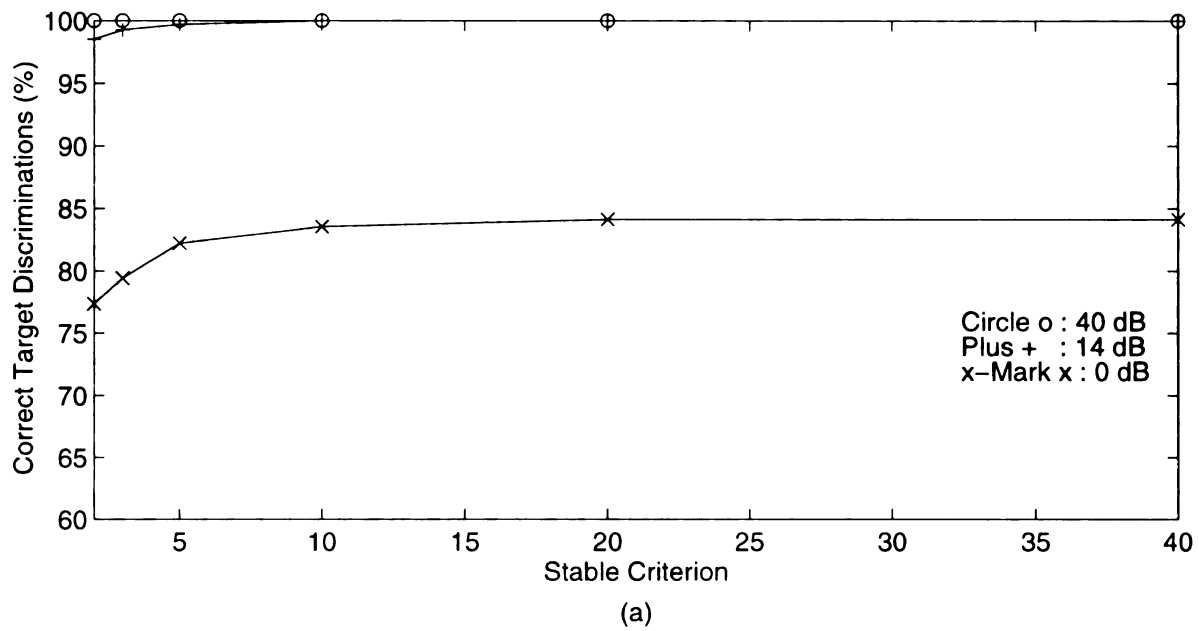


Figure 5.51 Correct and wrong target discrimination performances of the RCAAM/fi with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs stability criterion.

(b) Wrong target discrimination performances vs. stability criterion.

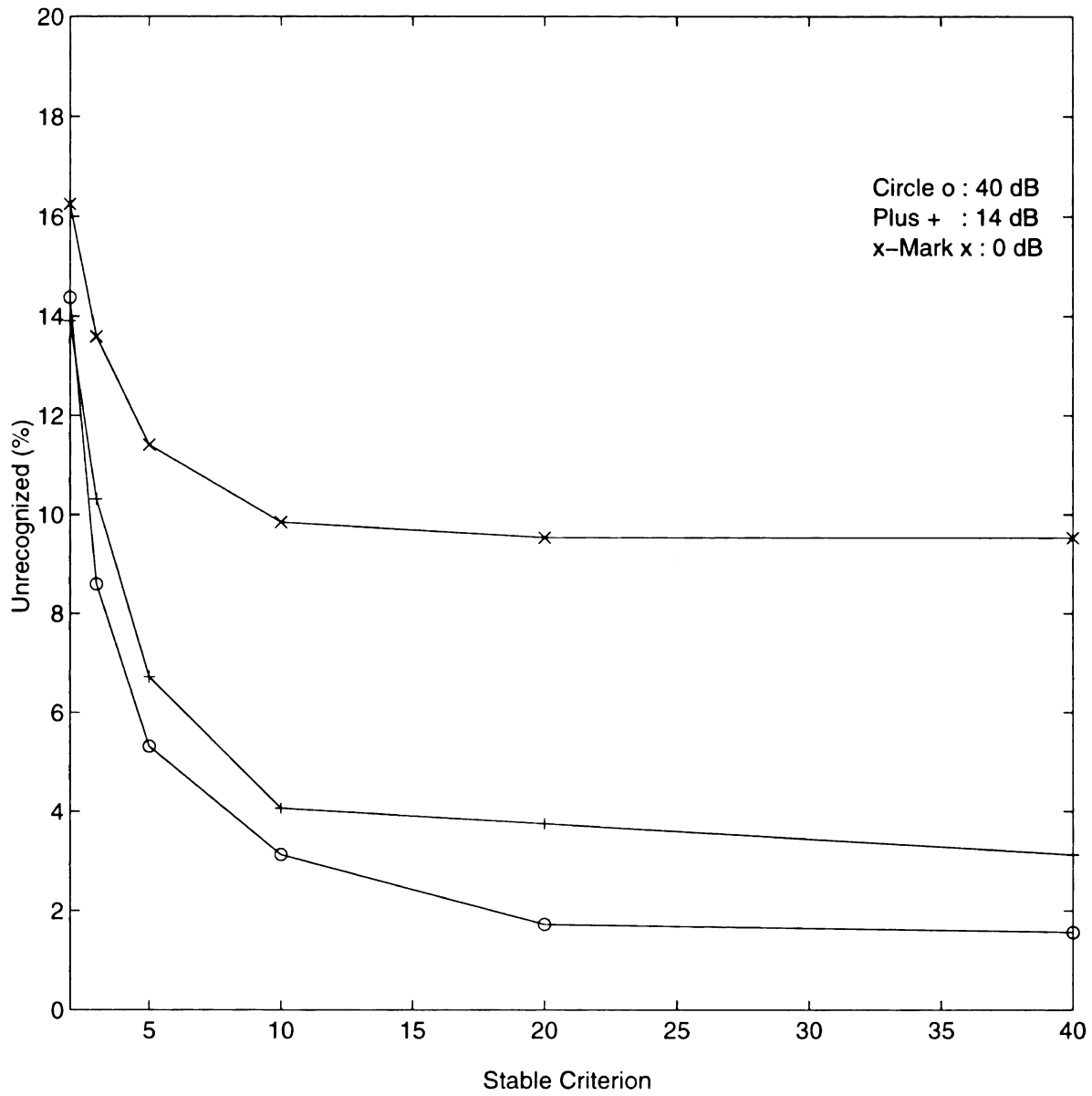


Figure 5.52 Unrecognized discrimination of the RCAAM/fi with an initial order of 1 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns.

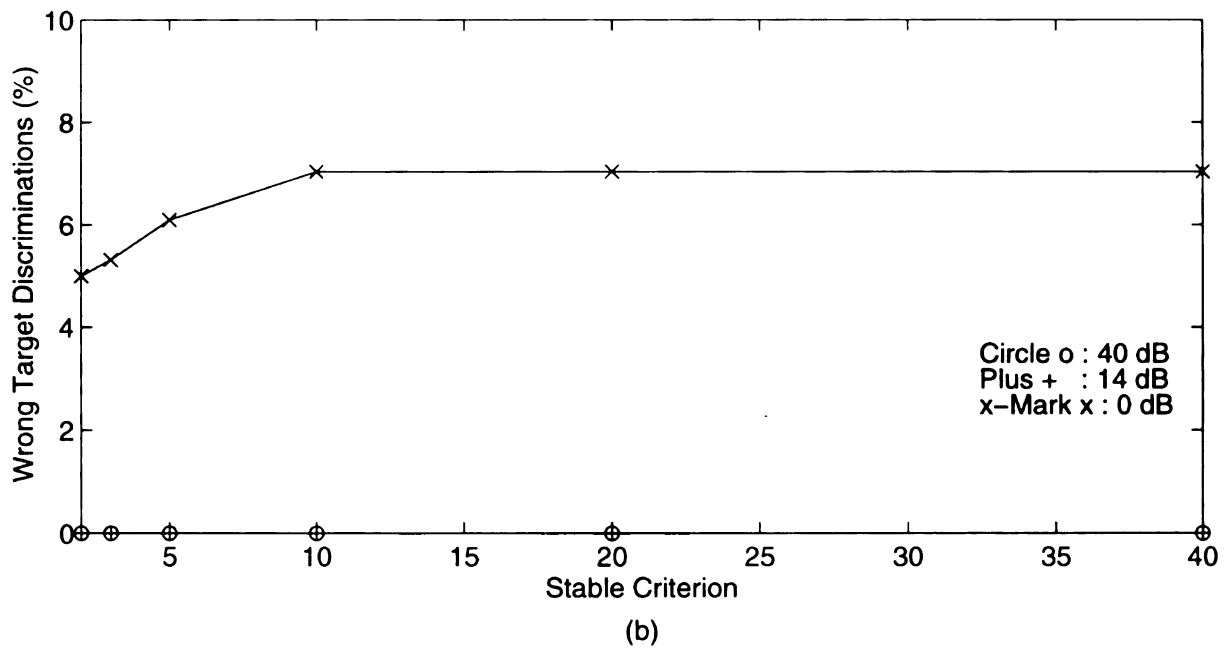
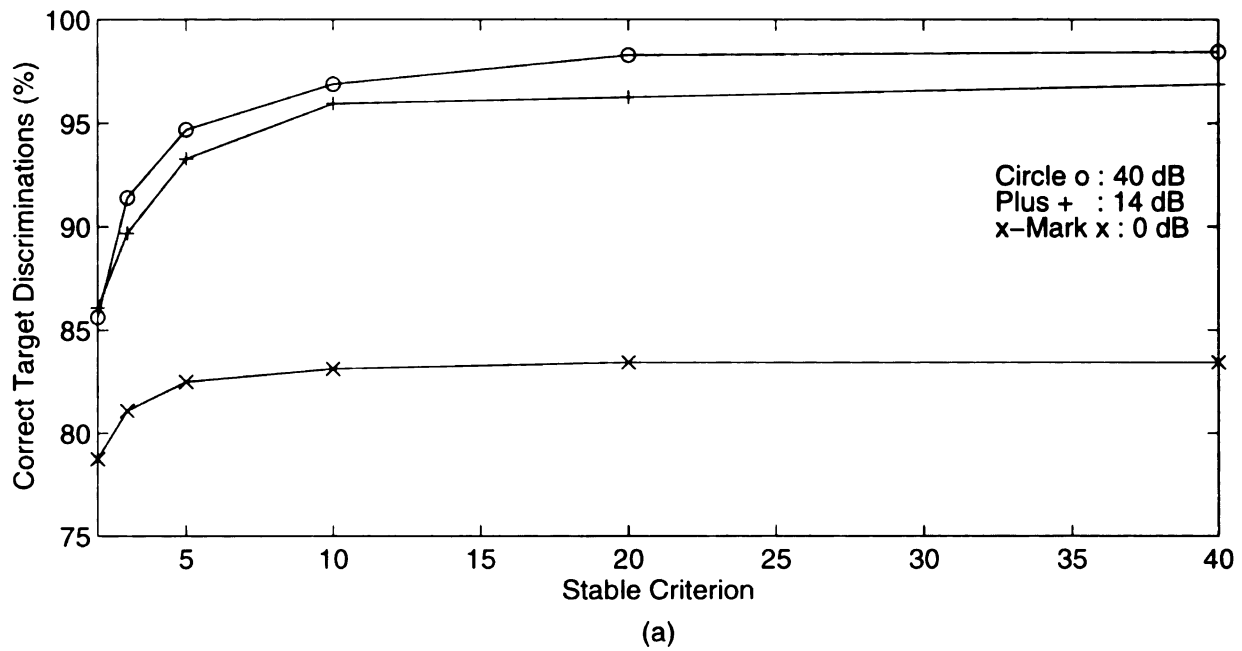


Figure 5.53 Correct and wrong target discrimination performances of the RCAAM/fi with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs. stability criterion.

(b) Wrong target discrimination performances vs. stability criterion.

RCAAM/fi_2 vs. initial order for stored pattern inputs in (a) and the wrong pattern discrimination in (b), while **Figure 5.56** presents the correct target discrimination in (a) and wrong target discrimination in (b). From the latter two figures, you can find out that the oscillation occurring in **Figure 5.54** results from the oscillating wrong pattern or target discrimination. In our simulations, we have set a maximum convergent cycle within which the recurrent updating procedure continues until either the sc is satisfied or the recurrent iteration reaches the maximum convergent cycle. If an iteration reaches the maximum convergent cycle, the recurrent updating procedure stops and the final state is given as the network output. We have set the maximum convergent cycle to 40, 60 or 120 iterations. Therefore, with an even initial order p , the final accumulation gain of a negative correlation will give a positive amplification, since an even p plus a even maximum convergent cycle gives an even integer. This final state will contribute to wrong target discrimination, if a pattern has a negative correlation gain, whose scale is largest among all correlation gains, and it belongs to a target different to the input's. This explains why the fi22, fi42 and fi62 have the peaks in wrong discriminations. By definition, stored patterns are slightly distorted by light contaminations. A large negative correlation gain rarely exists between a slightly contaminated stored pattern and any stored pattern. For an odd initial order p , the final accumulation gain of a negative correlation will give a negative amplification, since an odd p plus a even maximum convergent cycle gives an odd integer. Then RCAAM/fi can't recognize a negative stored pattern as one of the stored patterns since it only memorizes and recognizes the stored patterns. Therefore, the fi12, fi13 and fi15 leave those negatively amplified patterns unrecognized when the iteration reaches the maximum convergent cycle.

Figure 5.57 shows the unrecognized discrimination of RCAAM/fi_2 vs. initial order for distorted unstored pattern inputs. Again the unrecognized performance oscillates along with the initial order for 0 dB. **Figure 5.58** shows the correct target discriminations of RCAAM/fi_2 vs. initial order for distorted unstored pattern inputs in (a), and the wrong target discriminations in (b). In (b), the oscillation of wrong discrimination becomes larger. The wrong discriminations for odd initial orders have the same performances and those for even initial orders also have the same performances. The correct target discriminations for lightly contaminated unstored patterns are worse than those in **Figure 5.53** since the RCAAM/fi_2's shown in **Figure 5.58** only use an **sc** of 2.

Figure 5.59 shows that the **aord**'s of RCAAM/fi1 change along with **sc** for contaminated stored and unstored pattern inputs. Under the same **sc**, the **aord** of unstored patterns is larger than the **aord** of stored patterns. We should notice that the RCAAM/fi's **aord** is variable with respect to a maximum convergent cycle, since the discrimination order will be equal to initial order plus the maximum convergent cycle if an oscillation occurs. Compared to **Figure 5.31** and **Figure 5.32**, the RCAAM/fi has a higher **aord** than the RCAAM/di. Since the feedback mechanism allows RCAAM/di to sufficiently employ the adaptive elasticity in accordance with an appropriate **sc** to speed up convergence, the RCAAM/di will converge sooner than the RCAAM/fi under heavy contaminations. And the RCAAM/di doesn't suffer the oscillation problem which occurs in RCAAM/fi, so its **aord** is always smaller than a fair maximum convergent cycle. In other words, the RCAAM/fi keeps the original input intact at the expense of slowing down its converging when the recurrent accumulative adaptations continue.

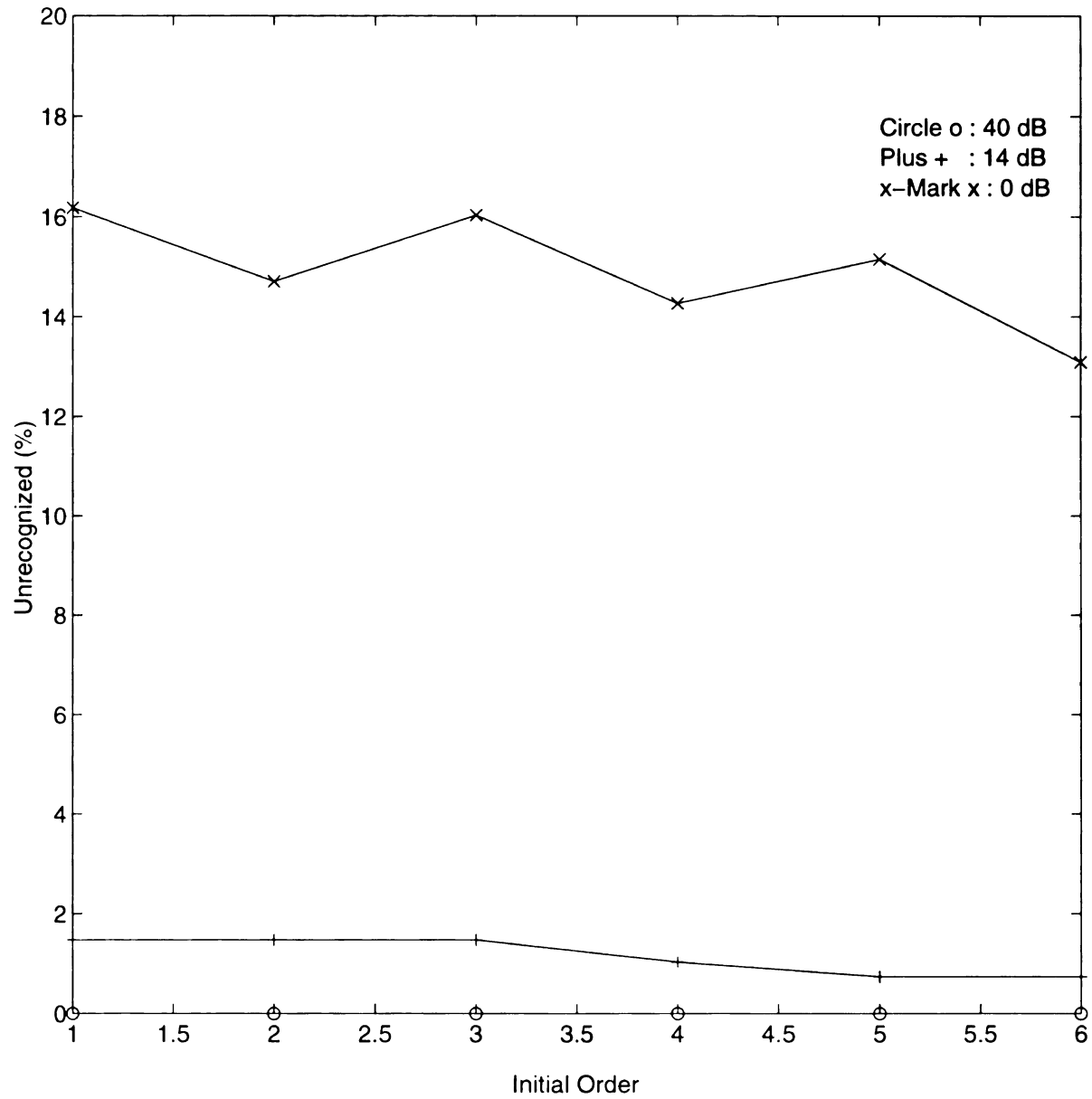


Figure 5.54 Unrecognized discrimination of the RCAAM/fi with a stability criterion of 2 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 68 stored patterns.

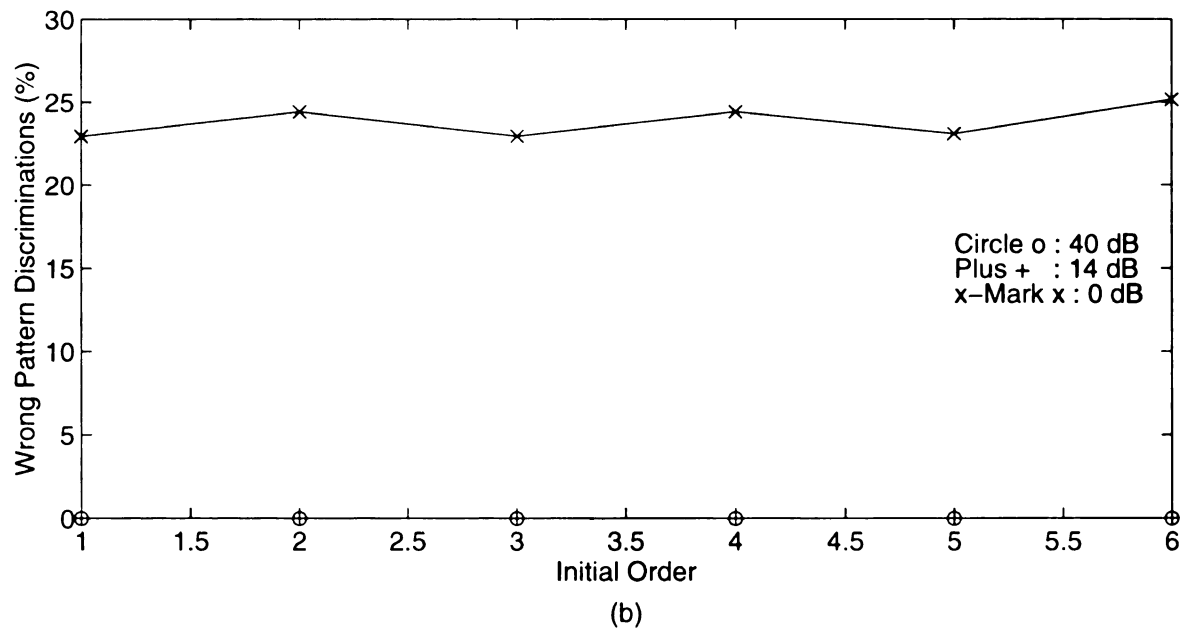
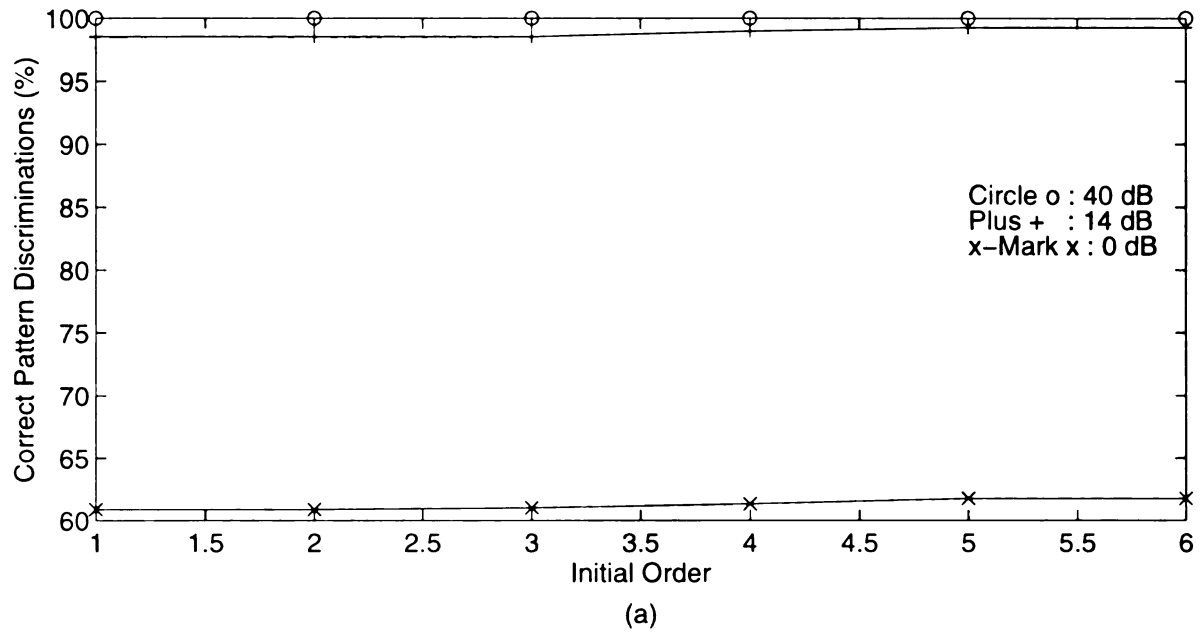


Figure 5.55 Correct and wrong pattern discrimination performances of the RCAAM/fi with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for 68 stored patterns.

(a) Correct pattern discrimination performances vs. initial order.

(b) Wrong pattern discrimination performances vs. initial order.

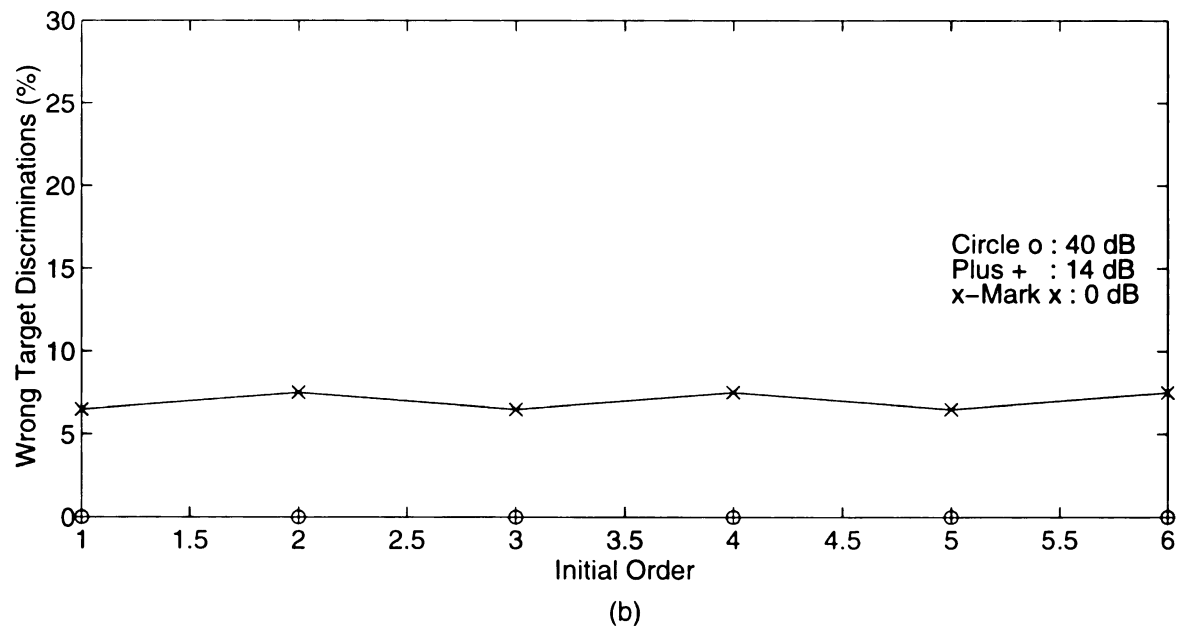
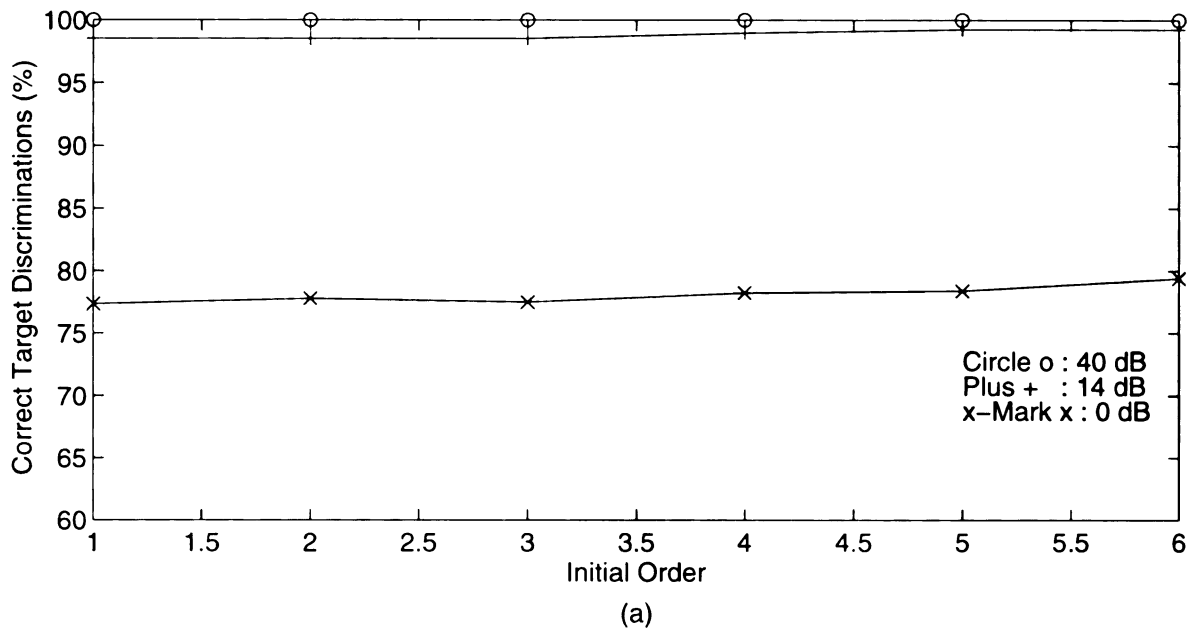


Figure 5.56 Correct and wrong target discrimination performances of the RCAAM/fi with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order.

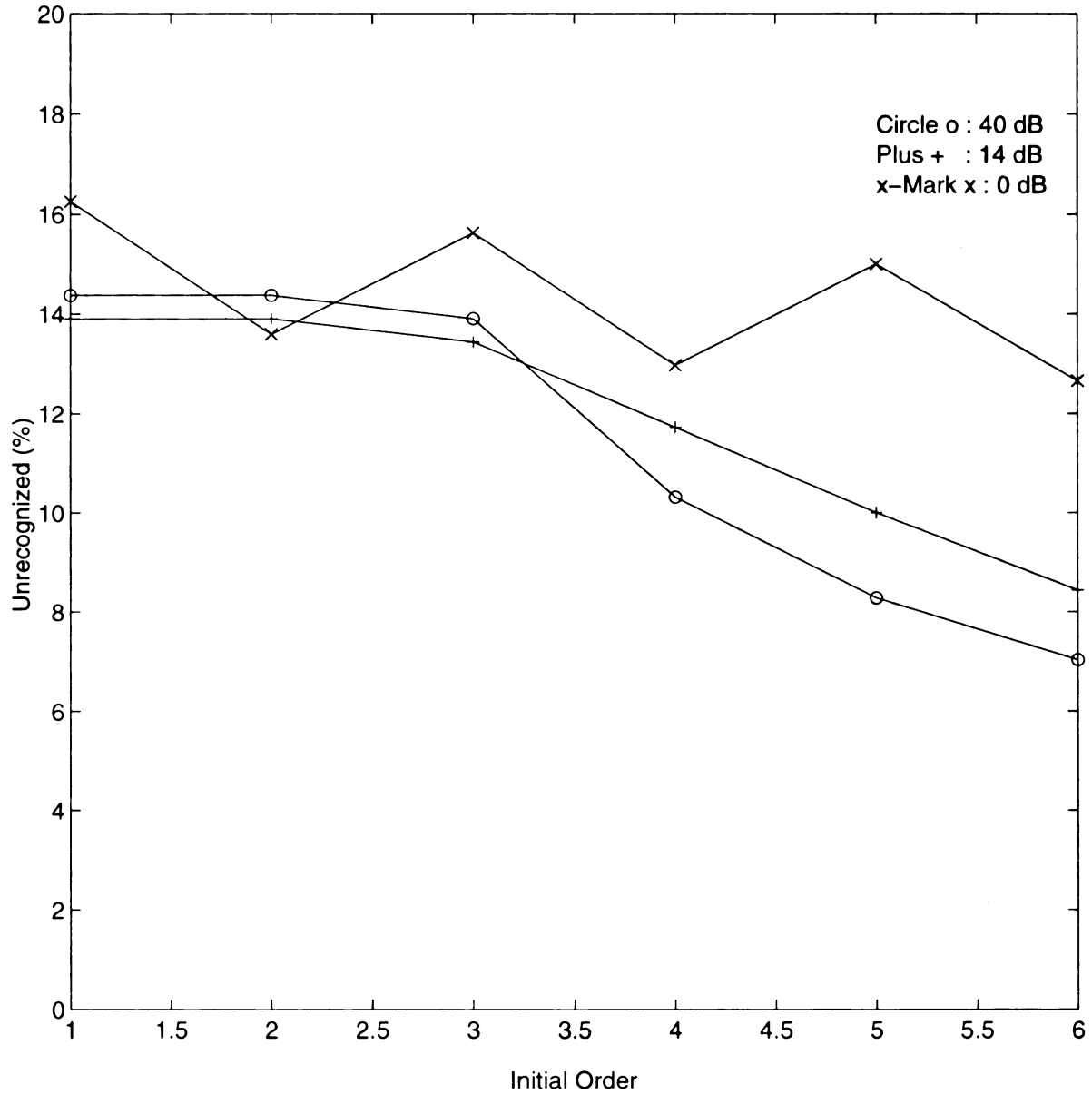


Figure 5.57 Unrecognized discrimination of the RCAAM/fi with a stability criterion of 2 under 40 dB, 14 dB and 0 dB SNR vs. initial order for 64 unstored patterns.

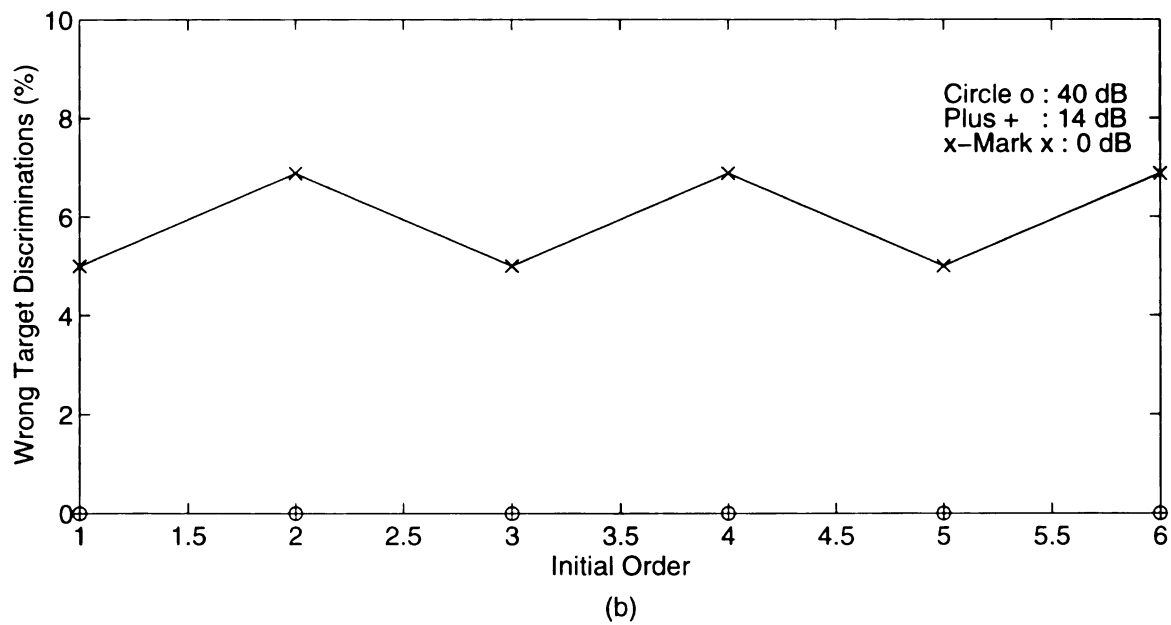
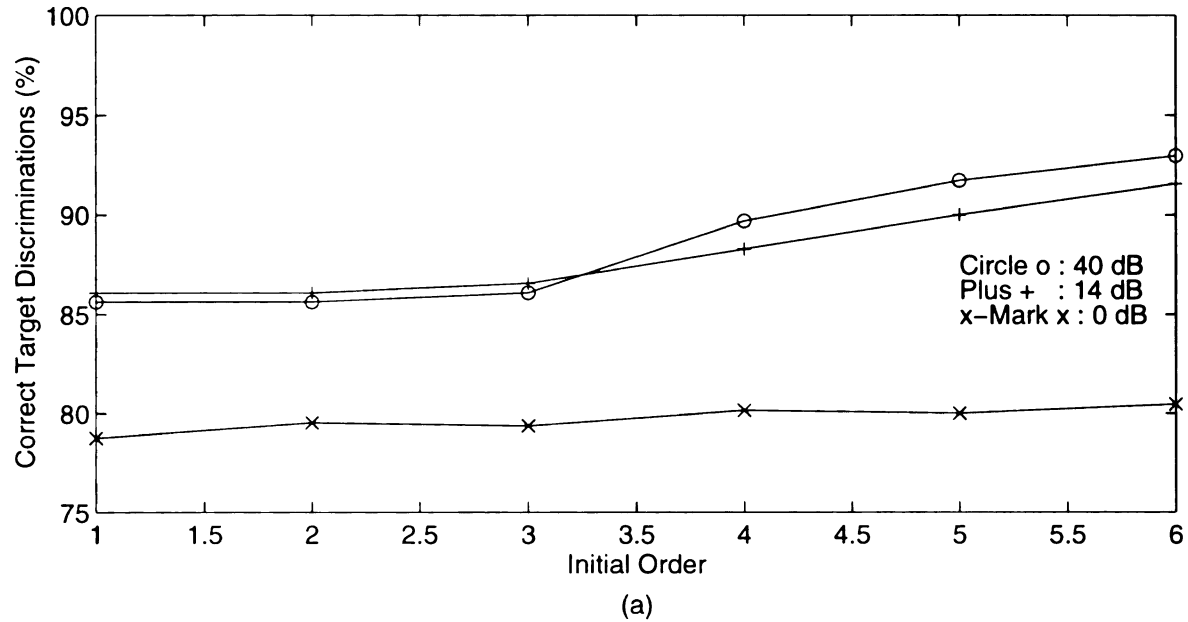


Figure 5.58 Correct and wrong target discrimination performances of the RCAAM/fi with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order.

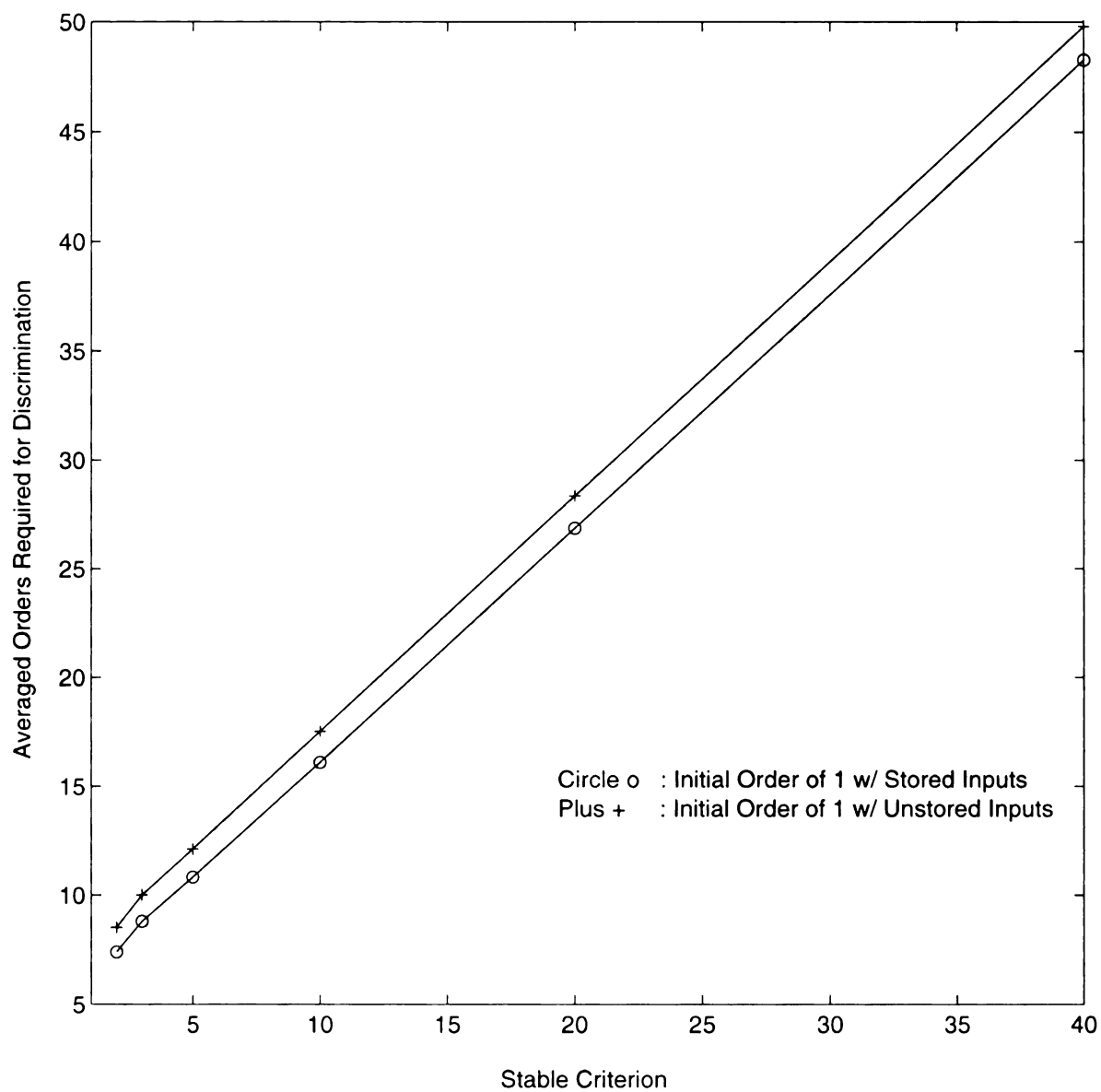


Figure 5.59 Averaged orders required for target discrimination of the RCAAM/fi with contaminated stored pattern inputs vs. stability criterion.

5.5 Performance of the Recurrent Correlation Accumulation Adaptive Memory with analog input and digital output (RCAAM/ad)

Since 3-bit coding with 7 quantization levels will have nonlinearity interference when contamination becomes serious in this section, we use analog inputs to eliminate the nonlinearity problem. We simulate the performances of RCAAM/ad with different initial orders and stability criterions. From section 5.2 we know the only difference between RCAAM/fi and RCAAM/ad is the network input form. RCAAM/ad is used with analog input form instead of digital (bipolar) form used by RCAAM/fi, with an initial order of 1 (RCAAM/ad1 or ad1) under different stability criterion circumstances and RCAAM/ad with its *sc* fixed to 2 (RCAAM/ad_2 or ad_2) under different initial order cases.

For four targets using RCAAM/ad, we have 68 100-point aspect responses to store. First of all, we calculate each energy of the 68 100-point responses then normalize them to 1 by their respective energy. These 68 normalized patterns are then stored in RCAAM/ad. When an input (of 100 responses) is input to the network, we calculate its energy and then add the corresponding noise with the desired simulation SNR. Then we calculate the energy of the noisy input, and finally normalize the noisy input to 1. The normalized input is then presented at the network input as the network process input pattern.

Figure 5.60 shows the unrecognized discrimination of RCAAM/ad with an initial order of 1 (RCAAM/ad1) vs. *sc* for contaminated stored pattern inputs for 40 dB, 14 dB and 0 dB SNR. **Figure 5.61** shows the correct pattern discrimination of RCAAM/ad1 vs. *sc* in (a) and wrong pattern discrimination in (b), while **Figure 5.62** presents the correct target discrimination in (a) and wrong target discrimination in (b). We can find RCAAM/ad1 has

excellent no error target discrimination even under severe distortion (0 dB SNR). Therefore, for distorted stored pattern inputs, RCAAM/ad has the best converged discrimination performances among three versions of \RCAAM. Comparing (b) of **Figure 5.61** with (b) of **Figure 5.62**, we realize that all wrong pattern discriminations result from the recognition of different aspect angles of a target, and thus will contribute to the correct target discriminations.

There is a strange phenomenon observed in that the correct discrimination of RCAAM/ad12 (i.e. $sc=2$) with an SNR of 0 dB is larger than the one with 40 dB. This phenomenon didn't occur in the RCAAM/fi. When the analog signal form is adopted, the pattern linearity isn't distorted at all. Most aspect responses are consistent within a fair angle range, and then there may be some adjacent analog aspect patterns that are very close to each other. If this happens, we may call these patterns with very close neighbors uneasy patterns. Then the correlation gains for its neighbors will be also large and close, if the input is an uneasy pattern. Therefore, the output may be a linear combination of the uneasy pattern and its adjacent stored patterns, since each analog stored patterns in RCAAM/ad has its own associated bipolar stored patterns. If the neighbor patterns are very close, then their linear combination state may become a spurious stable state which is stable only for few iterations. Therefore, an sc of 2 might still leave it in a spurious state, while a higher sc can force it to leave the spurious state. A large distortion may shift an uneasy pattern in a way to bias favor itself or one of its neighbors. Therefore, with a low sc of 2, the unrecognized discrimination of RCAAM/ad1 with 0 dB SNR is less than the one for 40 dB. And a higher sc will eliminate this strange phenomenon. Why wouldn't the same phenomenon have happened to the

RCAAM/fi with a **sc** of 2 ? The 3-bit coding with 7 quantization levels is a nonlinear procedure and the coding mapping from analog signal to digital form increases the correlation discrimination resolution. This means that the coding scheme will interpret uneasy patterns with higher discrimination resolutions of correlations, and thus the RCAAM/fi with a **sc** of 2 can discriminate easier based on correlation gains than the RCAAM/ad_2 under light contamination. The resolutions of correlation discrimination for analog and bipolar signals will be formally analyzed in the next section.

Figure 5.63 shows the unrecognized discrimination of RCAAM/ad1 vs. **sc** for contaminated unstored pattern inputs under 40 dB, 14 dB and 0 dB SNR. **Figure 5.64** presents the correct pattern discrimination of RCAAM/ad1 vs. **sc** for contaminated unstored patterns in (a) and wrong pattern discrimination in (b). Again the RCAAM/ad1 has excellent no error target discrimination for seriously distorted unstored patterns. And the unrecognized discrimination for 0 dB SNR is again less than the one for 40 dB SNR. But this time the cause is different for the previous case. Since each unstored test pattern is resident at the middle of two adjacent stored patterns in our simulations, a severely contaminated unstored pattern might have a better chance to shift toward either adjacent stored pattern than an uncontaminated unstored pattern. Therefore, a heavily distorted unstored pattern may be closer to a true stored pattern than a slightly distorted unstored pattern. This seemingly unreasonable phenomenon results from the deterministic allocation arrangements around stored and unstored test patterns. It is not expected in practical situation.

Figure 5.65 shows the correct target discrimination of RCAAM/ad with a **sc** of 2 (RCAAM/ad_2) vs. initial order for contaminated stored patterns in (a) and wrong target

discrimination in (b). **Figure 5.66** presents the correct target discrimination of RCAAM/ad_2 vs. initial order for contaminated unstored patterns in (a) and wrong target discrimination in (b). As explained in the previous section on RCAAM/fi, the performances of RCAAM/ad are independent of their initial orders. The phenomena occurred in the RCAAM/ad12 with contaminated stored patterns and the RCAAM/ad1 with distorted unstored patterns and also exists in RCAAM/ad_2 with initial orders around 1. We should notice that the phenomenon for stored patterns will be eliminated by larger initial orders, since a sufficiently larger initial order is capable of discriminating the uneasy stored patterns. **Figure 5.67** shows the **aord** of RCAAM/ad vs. **sc** for contaminated stored patterns and unstored patterns. From this figure, there are approximately linear relationships between **aord** and **sc** for both cases.

5.6 Comparisons of Correlation-based Discrimination Resolutions Between Analog and Bipolar Patterns

In the previous section, we found the RCAAM/ad with a fair **sc** has higher unrecognized discriminations for lightly contaminated stored patterns than the RCAAM/fi, though it has excellent correct and no error target discrimination performances for seriously distorted patterns. This indicates that the analog data used by RCAAM/ad may have lower correlation-based discrimination resolution than the digital data used by RCAAM/fi.

Figure 5.68 shows the orders required for the discriminations of RCAAM/fi1 vs. SNR for contaminated stored patterns, while **Figure 5.69** presents the orders required for the discriminations of RCAAM/ad1. With contamination equal to or larger than 6 dB, the RCAAM/fi12 and RCAAM/fi15 always have smaller discrimination orders than the RCAAM/ad12 and RCAAM/ad15, respectively. When the contaminations are less than 6 dB,

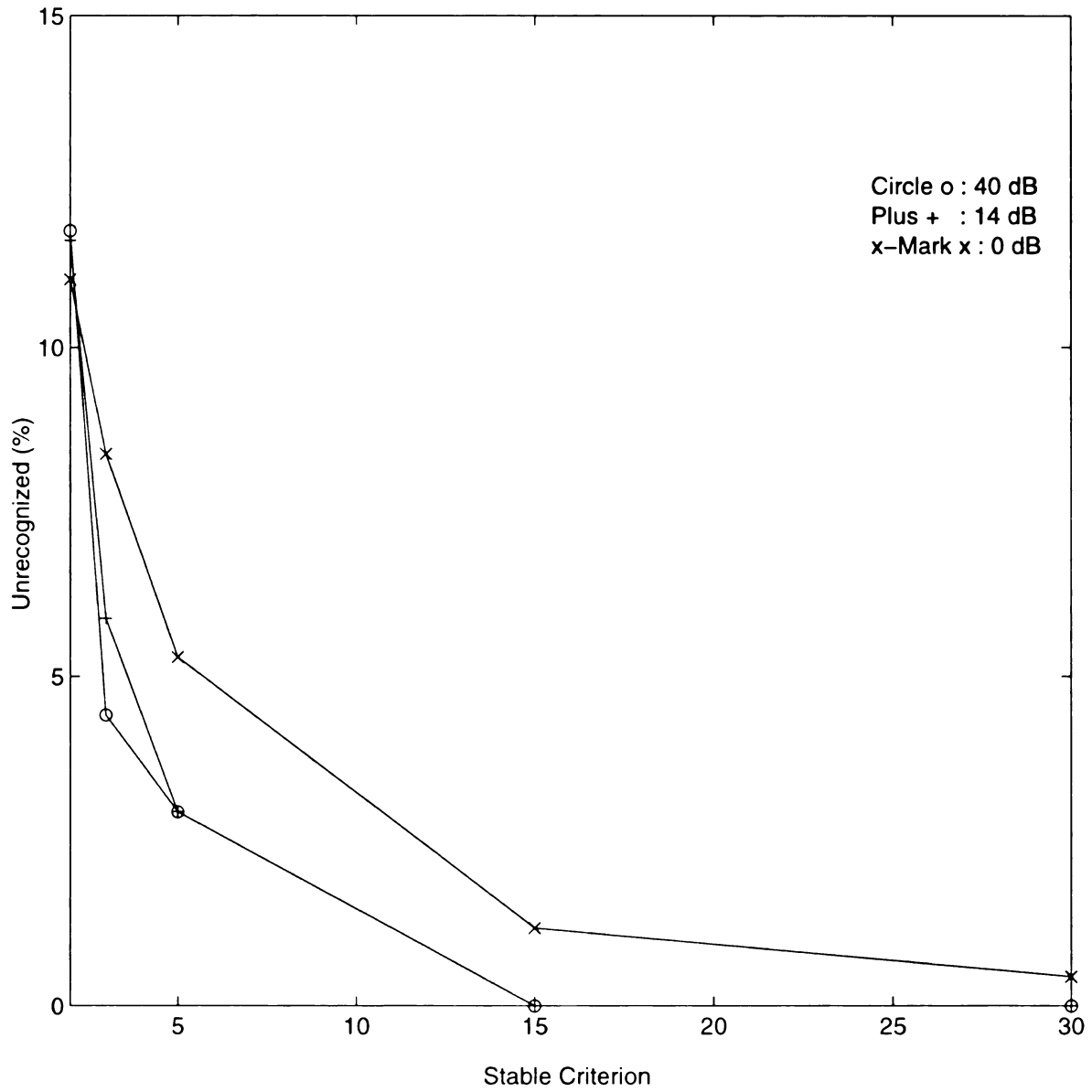


Figure 5.60 Unrecognized discrimination of the RCAAM/ad with an initial order of 1 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 68 stored patterns.

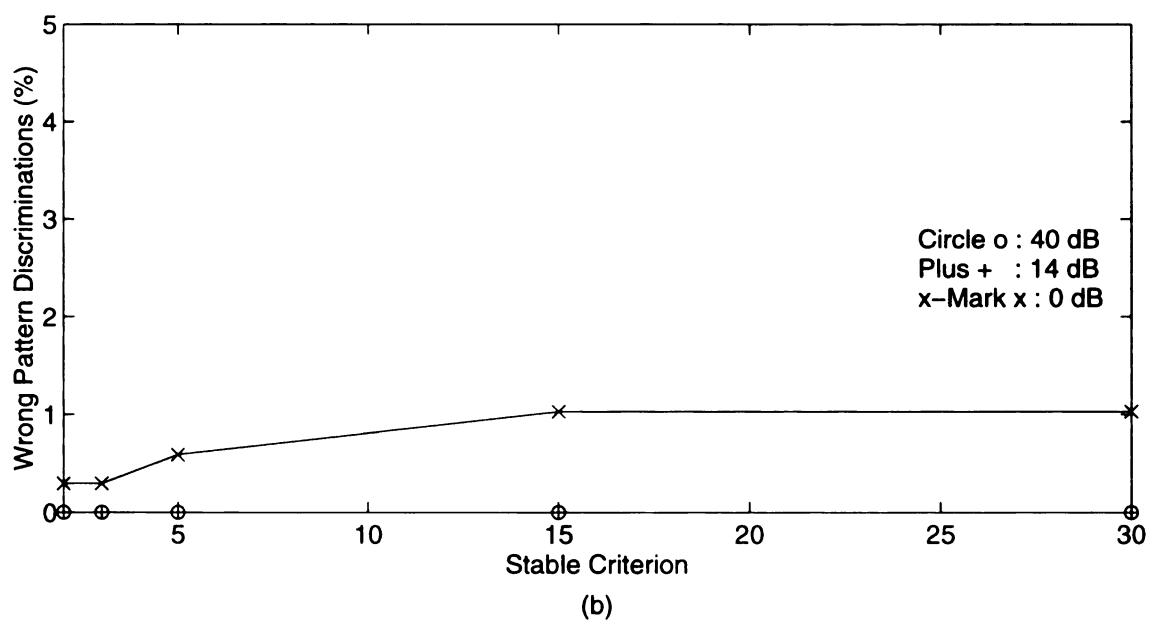
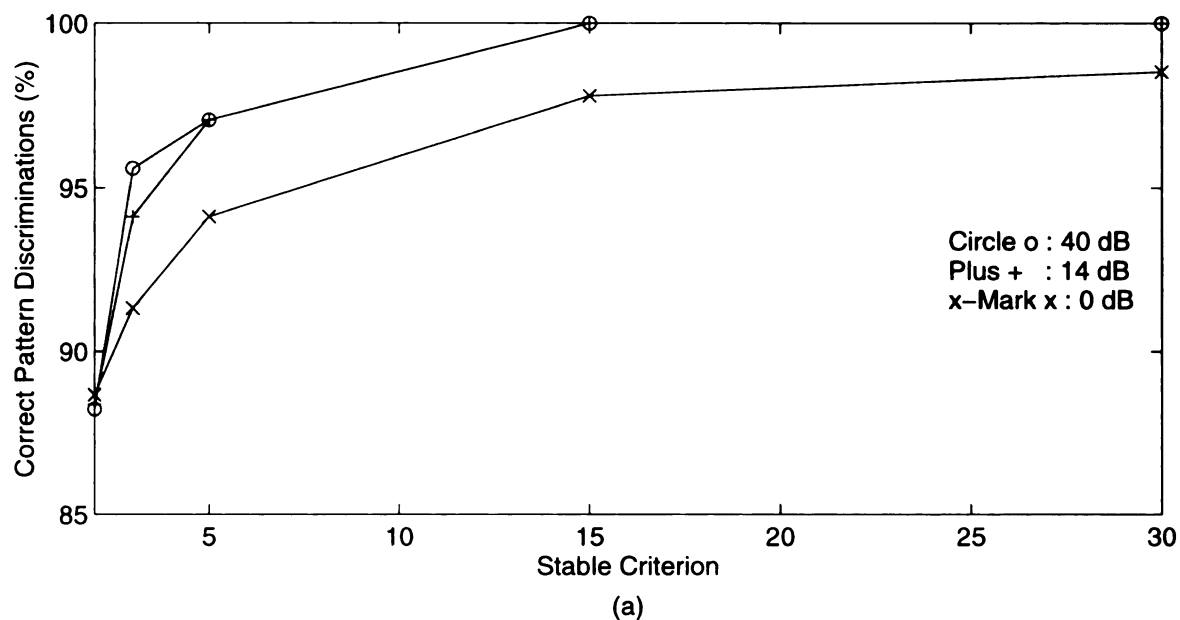


Figure 5.61 Correct and wrong pattern discrimination performances of the RCAAM/ad with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for 68 stored patterns.

(a) Correct pattern discrimination performances vs. stability criterion.

(b) Wrong pattern discrimination performances vs. stability criterion.

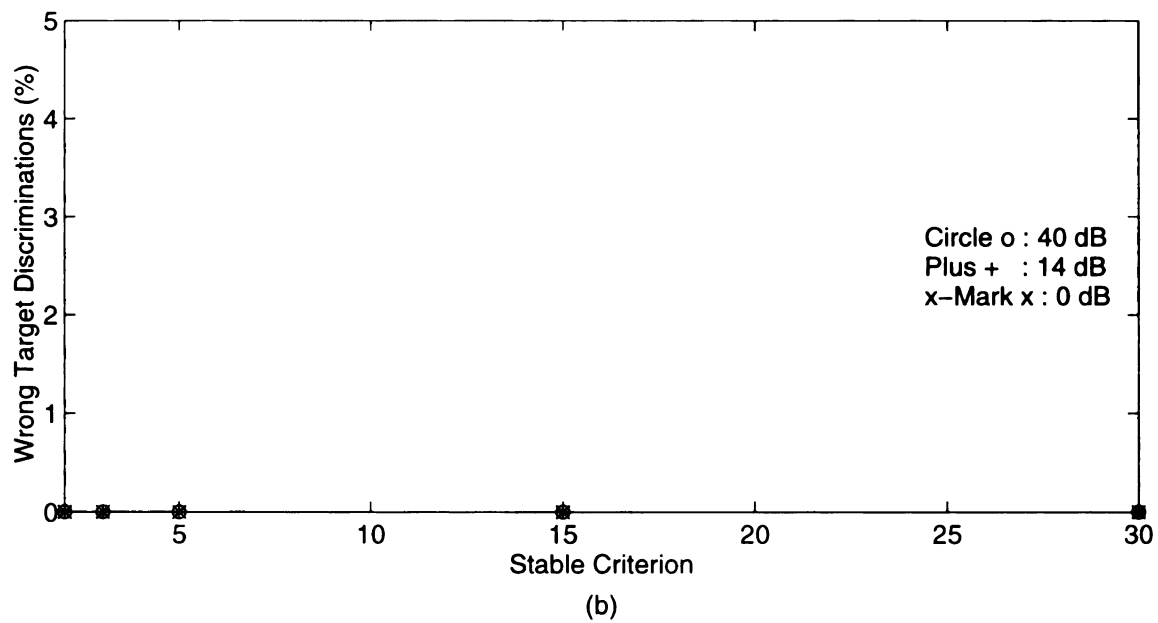
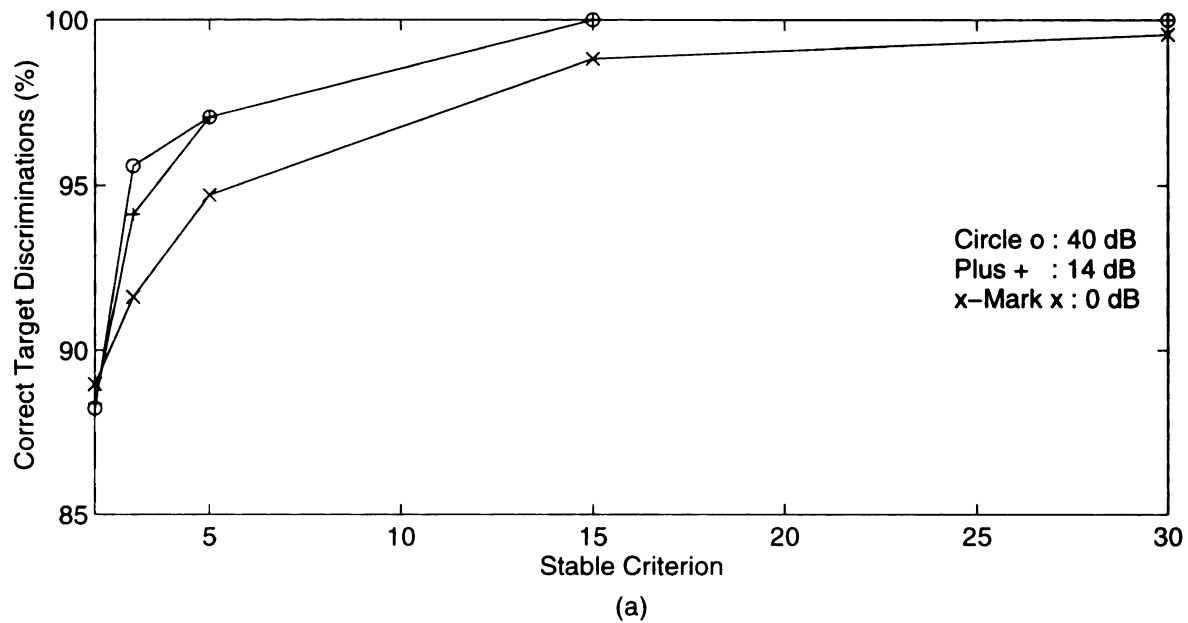


Figure 5.62 Correct and wrong target discrimination performances of the RCAAM/ad with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for the 68 stored patterns belonging to 4 targets..

- (a) Correct target discrimination performances vs stability criterion.
 (b) Wrong target discrimination performances vs. stability criterion.

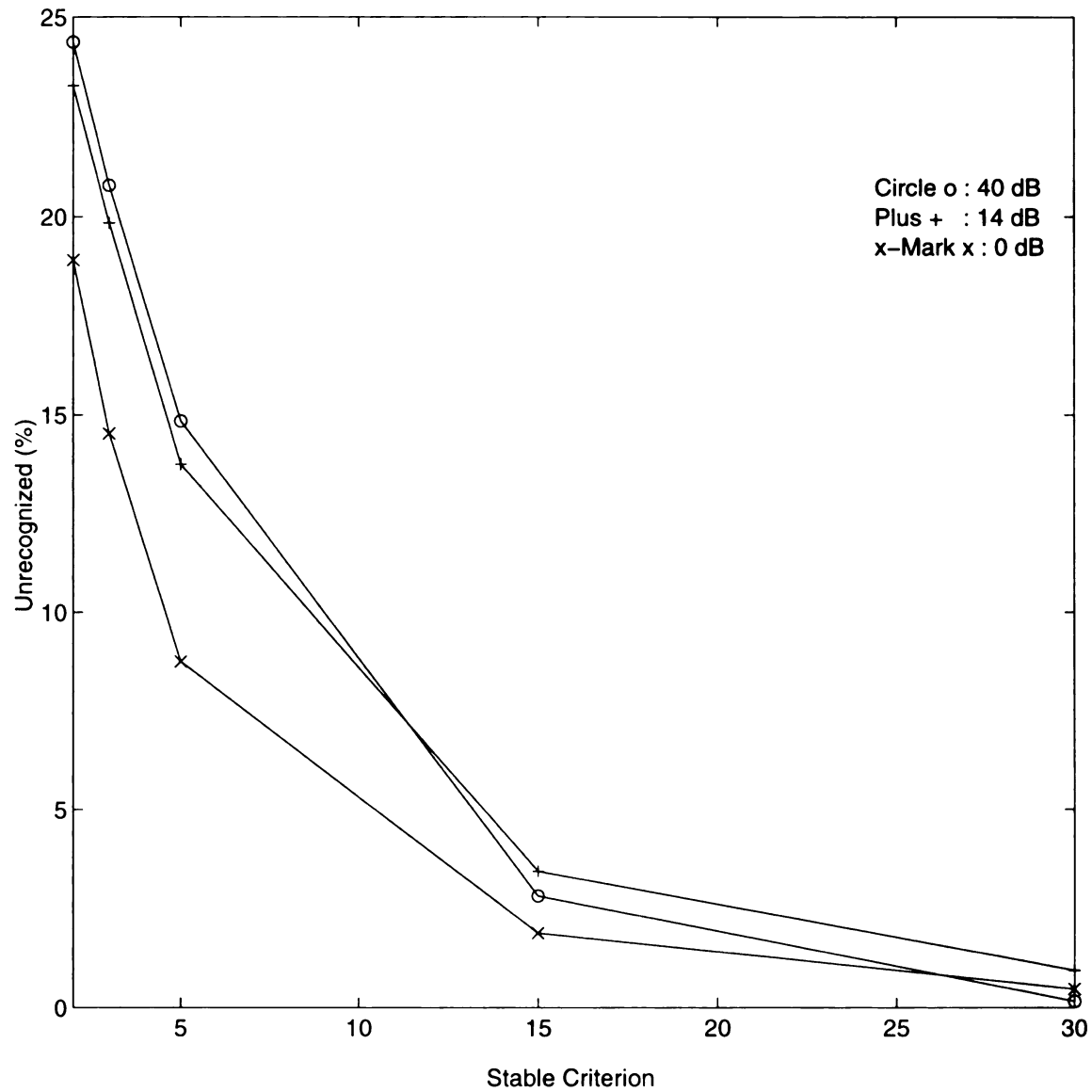


Figure 5.63 Unrecognized discrimination of the RCAAM/ad with an initial order of 1 under 40 dB, 14 dB and 0 dB SNR vs. stability criterion for 64 unstored patterns.

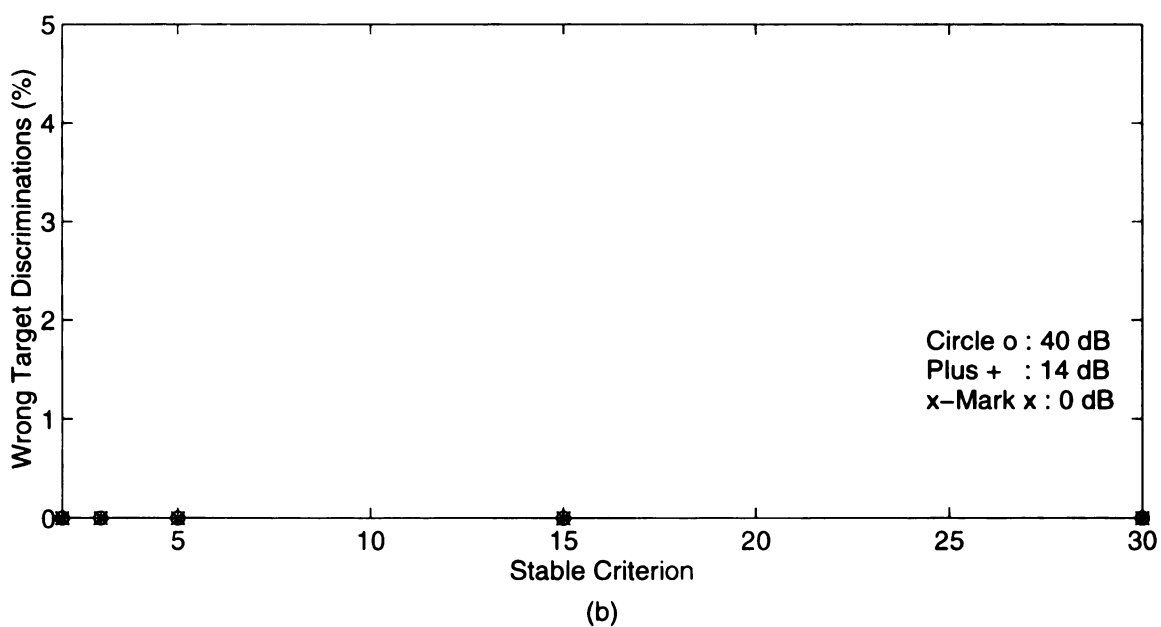
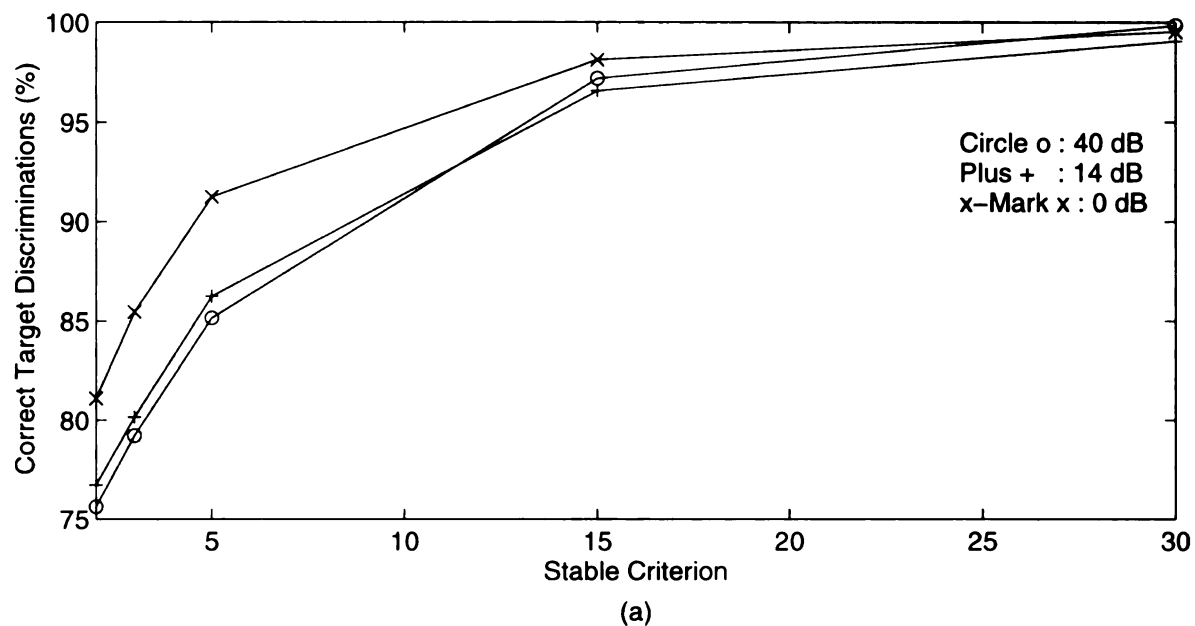


Figure 5.64 Correct and wrong target discrimination performances of the RCAAM/ad with an initial order of 1 under 40, 14 and 0 dB SNR vs. stability criterion for the 64 unstored patterns belonging to 4 targets..

- (a) Correct target discrimination performances vs stability criterion.
 (b) Wrong target discrimination performances vs. stability criterion.

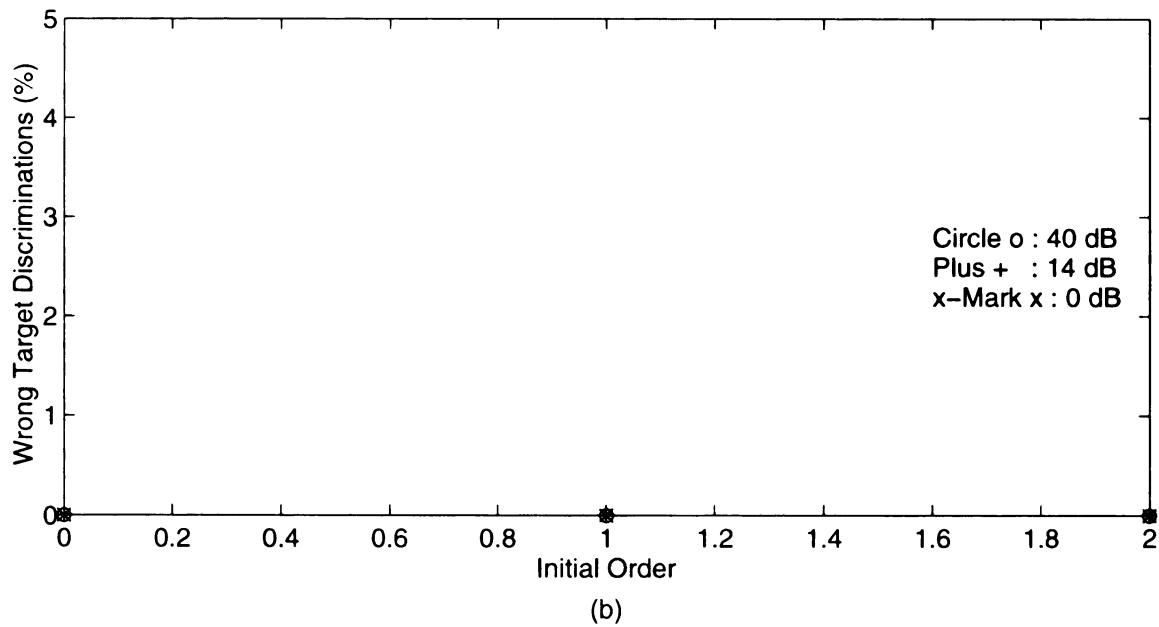
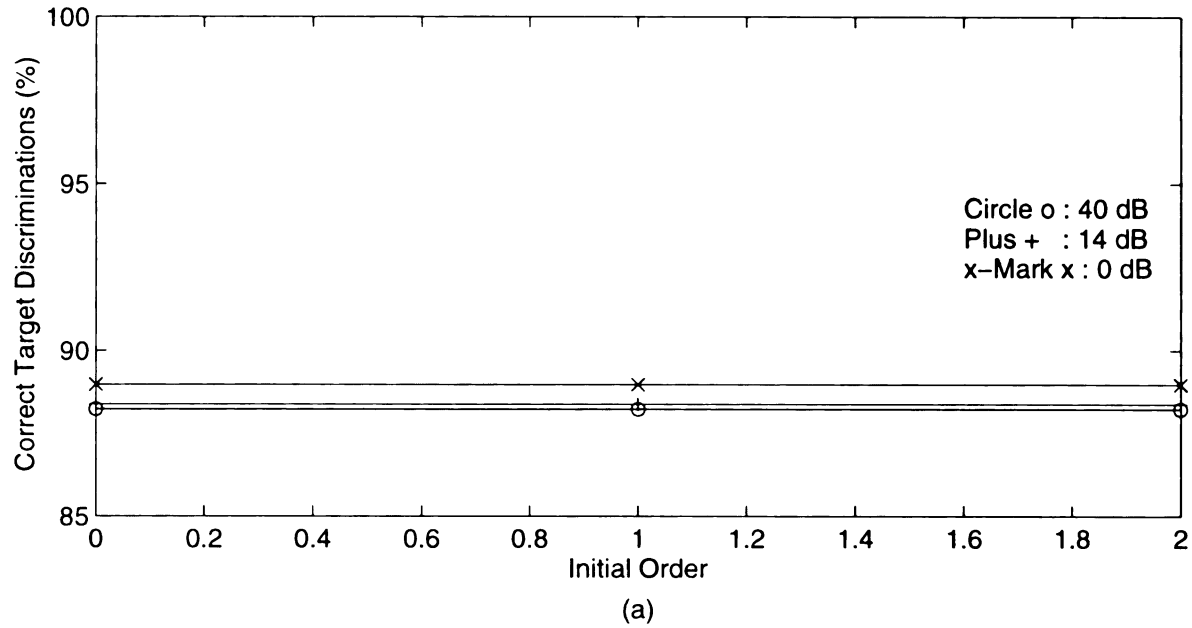


Figure 5.65 Correct and wrong target discrimination performances of the RCAAM/ad with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 68 stored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order.

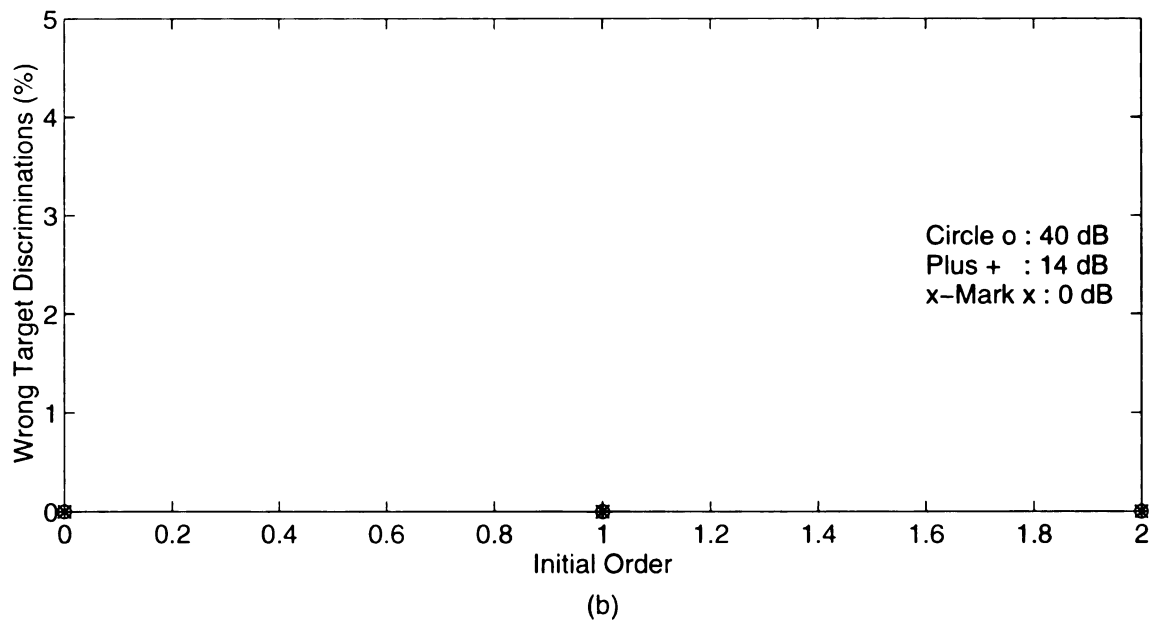
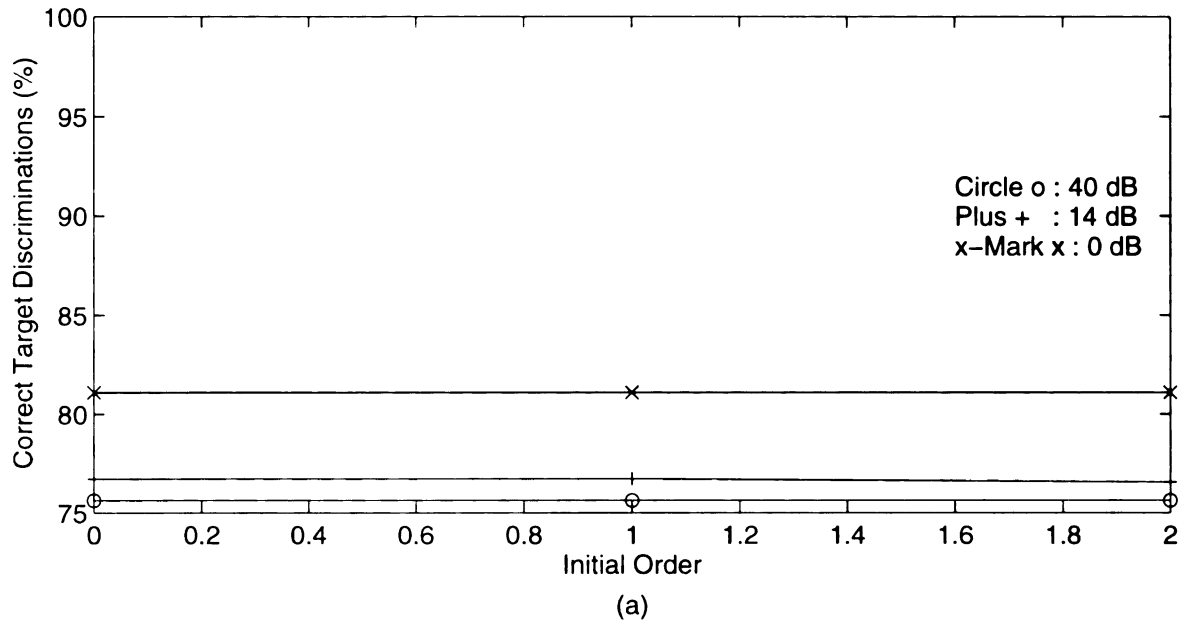


Figure 5.66 Correct and wrong target discrimination performances of the RCAAM/ad with a stability criterion of 2 under 40, 14 and 0 dB SNR vs. initial order for the 64 unstored patterns belonging to 4 targets..

(a) Correct target discrimination performances vs initial order.

(b) Wrong target discrimination performances vs. initial order.

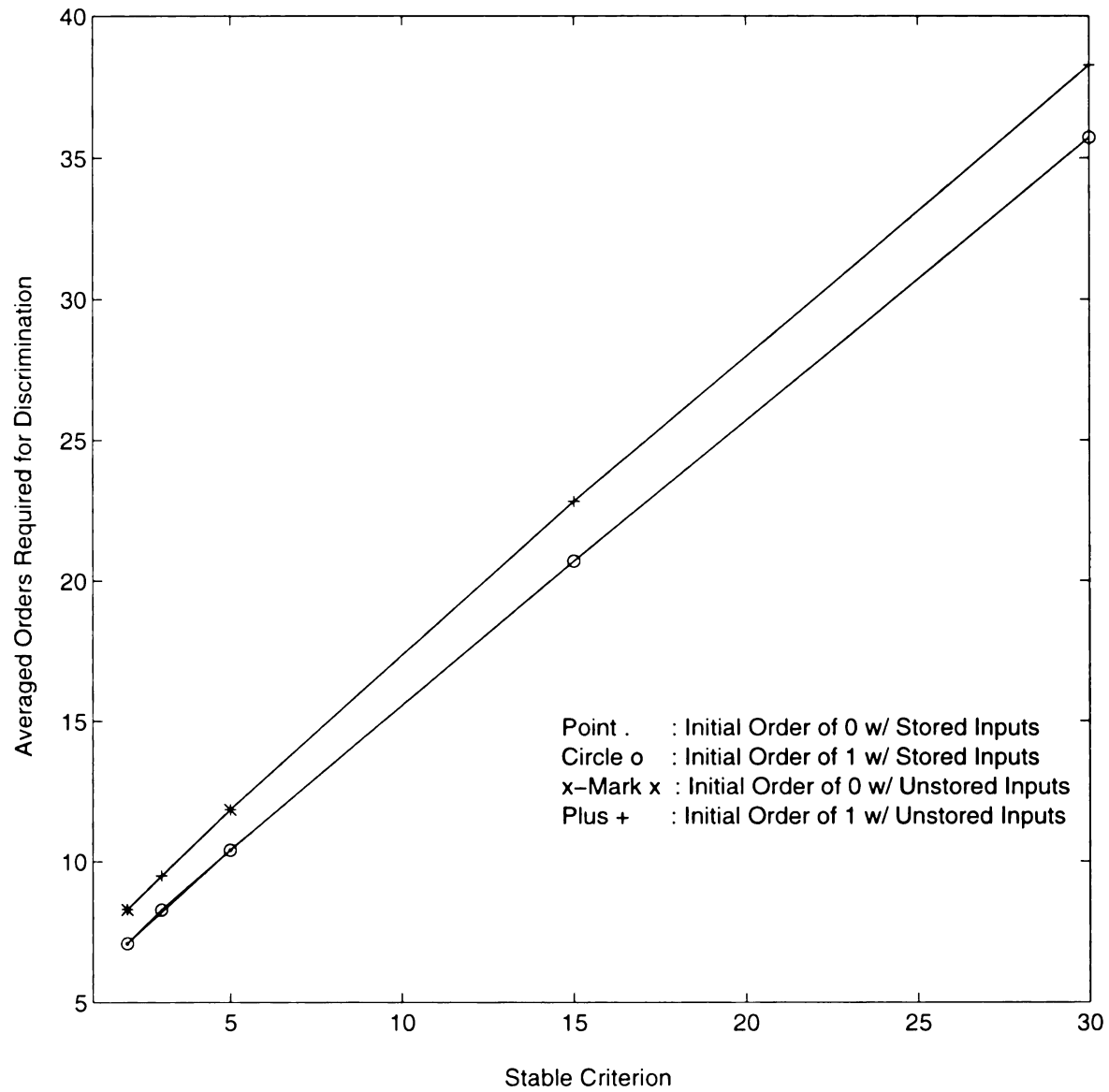


Figure 5.67 Averaged orders required for target discrimination of the RCAAM/ad with contaminated stored pattern inputs vs. stability criterion.

the RCAAM/fi1 has much larger discrimination orders than the RCAAM/ad. The latter phenomenon results from the code linearity breakdown corresponding to severe contaminations. As described in Chapter 3, the coding scheme we use will still have linearity if the contamination amplitude is statistically less than 1.5 quantization levels. When the contamination level breaks through the linearity range, the nonlinearity will result in many ambiguous states which are not apparently close to any stored pattern. Therefore, the discrimination order abruptly increases with a dramatic rate. And the former indicates the analog data carries lower discrimination resolution than the encoded bipolar data. Let's roughly prove this issue as follows.

In our simulation, there are 4 targets and each target has 17 aspect response patterns stored in the network. Suppose the i^{th} stored pattern, p_i , is a row vector and g_i is its autocorrelation, then $g_i = p_i * (p_i)^T$ where $*$ denotes matrix multiplication and $(p_i)^T$ stands for the transpose of the row vector p_i . Consider a RCAAM/fi or RCAAM/ad with initial order of 0 and the recurrent iteration k . Then, from section 2, the stored pattern p_i has autocorrelation gain or weight, g_i at iteration 1, and has autocorrelation gain or weight, g_i^k at iteration k , if the input is pattern p_i .

Assumption 1 : The output bipolar codes, associated to analog stored patterns in RCAAM/ad and associated to themselves in RCAAM/fi, are consistent within each target. Therefore, the output bipolar codes vary smoothly along with aspect angles for each target.

Assumption 2 : Only active patterns, whose correlation gains are greater than a threshold value, can participate in generating the next state in the output stage. This assumption is practical, since a small correlation gain will become annulled compared to a large one after

a few iterations.

Assume the target t has n active stored aspect patterns w.r.t. an input. Usually we set an active threshold by some percent, such as 30% or 20%, of the maximum correlation gain of the data form. Then define a normalized correlation gain for this target t w.r.t. the input by G_t where

$$\begin{aligned} G_t &= \sum_{i=1}^n g_i \quad \text{for iteration } 1 \\ G_t^k &= \sum_{i=1}^n g_i^k \quad \text{for iteration } k \end{aligned} \quad (11)$$

From the assumption 2, only active patterns can contribute to the normalized target correlation gain. Since assumption 1 has assumed the output codes are consistent within each target, summation can be used to enhance the same target code. For finite positive real values, we know the mean of sum is equal to or greater than the root of the product. Therefore,

$$\sum_{i=1}^n g_i^k \geq n \left(\prod_{i=1}^n g_i^k \right)^{\frac{1}{n}} = n \left(\prod_{i=1}^n g_i \right)^{\frac{k}{n}} \quad (12)$$

and then

$$G_t = \left(\sum_{i=1}^n g_i^k \right)^{\frac{1}{k}} \geq n^{\frac{1}{k}} \left(\prod_{i=1}^n g_i \right)^{\frac{1}{n}} \quad (13)$$

The equality appears only when all the correlation gains (g_i 's) have the same value. From the right most term, the normalized correlation gain G_t will decrease by $n^{1/k}$ when the recurrent

iterations continue. Finally this gives us a simplified way to evaluate and then compare the normalized target correlation gains between the analog data and bipolar data.

In the following simulations, we assume the active patterns in the output stage should have their correlation gains greater than a threshold value. Thus, under this assumption, only active patterns can contribute to the normalized target correlation gain. To evaluate the correlation-based discrimination resolution, we calculate the normalized target crosscorrelation gain between different targets. Then, for each target, we compare its normalized target autocorrelation gain to the other three normalized target crosscorrelation gains. If the ratios of target autocorrelation gain, 1 after normalization, to target crosscorrelation gains for a target are high then that target has high correlation-based recognition resolution. If the ratio of target autocorrelation gain to target crosscorrelation gains is around 1, the target has low correlation-based recognition resolution and then will not be easily recognized. Since each target has 17 aspect patterns stored, the appearance probability for each aspect is $1/17$ for a given target. In our simulations, the bipolar pattern has 300 bits and a maximum correlation gain of 300, while the normalized analog data has 100 values and a maximum correlation gain of 1. For example, if we set an active threshold by 30% of the maximum correlation gain, then, to a given input, a stored pattern in RCAAM/fi will be referred as active only if its correlation gain is greater than 90 ($300 \times 30\%$) with the input, and a stored pattern in RCAAM/ad will be considered as active only if its correlation gain is greater than 0.3 ($1 \times 30\%$) with the input. For each stored pattern, we calculate the correlations between itself and all 68 stored patterns and then evaluate the normalized target correlation gains for all 4 targets by only using those active correlations.

For example, a stored aspect pattern of target B52 should result in more active patterns for the B52 and less active patterns for the other three targets. After all 68 stored aspect patterns are calculated, the final normalized target correlation (autocorrelation and crosscorrelation) gains are given by the averaged value among 17 aspect patterns for each target. The following tables show the simulation results for iteration $k=1$; and the results for the other k 's are similar to $k=1$.

Table 5.1 shows the comparisons of normalized target correlation gains subject to a threshold of 30% between analog data and the encoded bipolar data. For bipolar input form, an active threshold of 30% means a stored pattern can be an active pattern only if it has a correlation greater than 90 ($300 \times 30\%$) with the bipolar input. It is equivalent to say an active stored pattern at least has 195 ($150 + 90/2$) bits out of 300 bits the same as the bipolar input. If an active threshold of 30% is assumed, the analog input form brings 21.76% stored patterns into active operation, while the encoded bipolar input form only involves 8.69% stored patterns. Since the target correlation gains are finally normalized with respect to the input target, the normalized correlation gains in each row can be regarded as an estimate of the input target recognition resolution in this 4 target memory. The input target can be easily recognized within a few iterations, if the normalized crosscorrelation gains of the other three targets w.r.t. the input target are much less than 1. For example, with a threshold of 30%, an analog aspect pattern of target F14 has a normalized crosscorrelation gain of 0.0409 with target B58 and a normalized crosscorrelation gain of 0.6048 with target TR1. Therefore, the RCAAM/ad recurrent operations are almost expended in distinguishing the input target, F14, from the target TR1, while the target B58 nearly doesn't bother the input pattern at all.

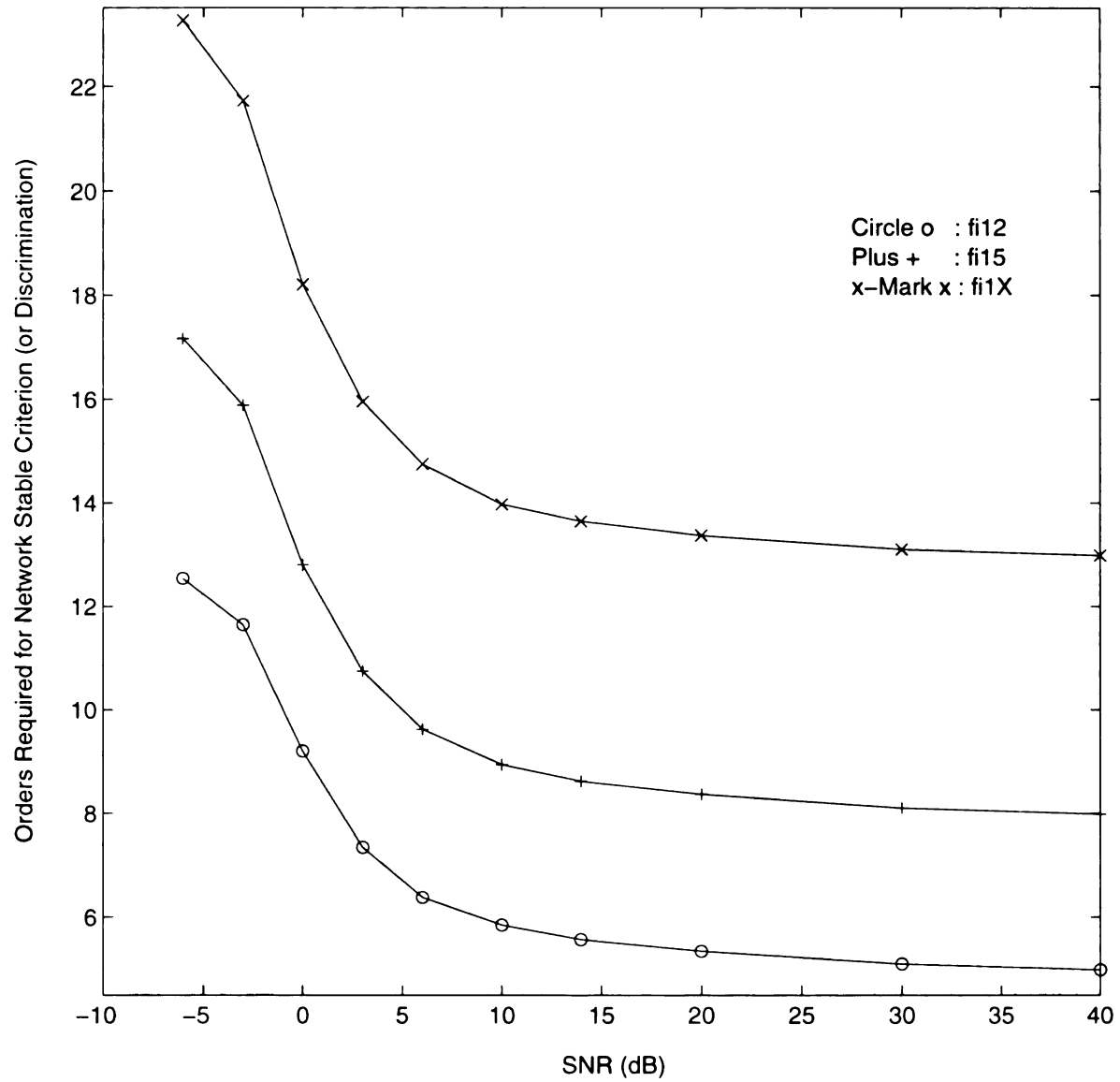


Figure 5.68 Orders required for the discriminations of RCAAM/fi1 vs. SNR for contaminated stored patterns.

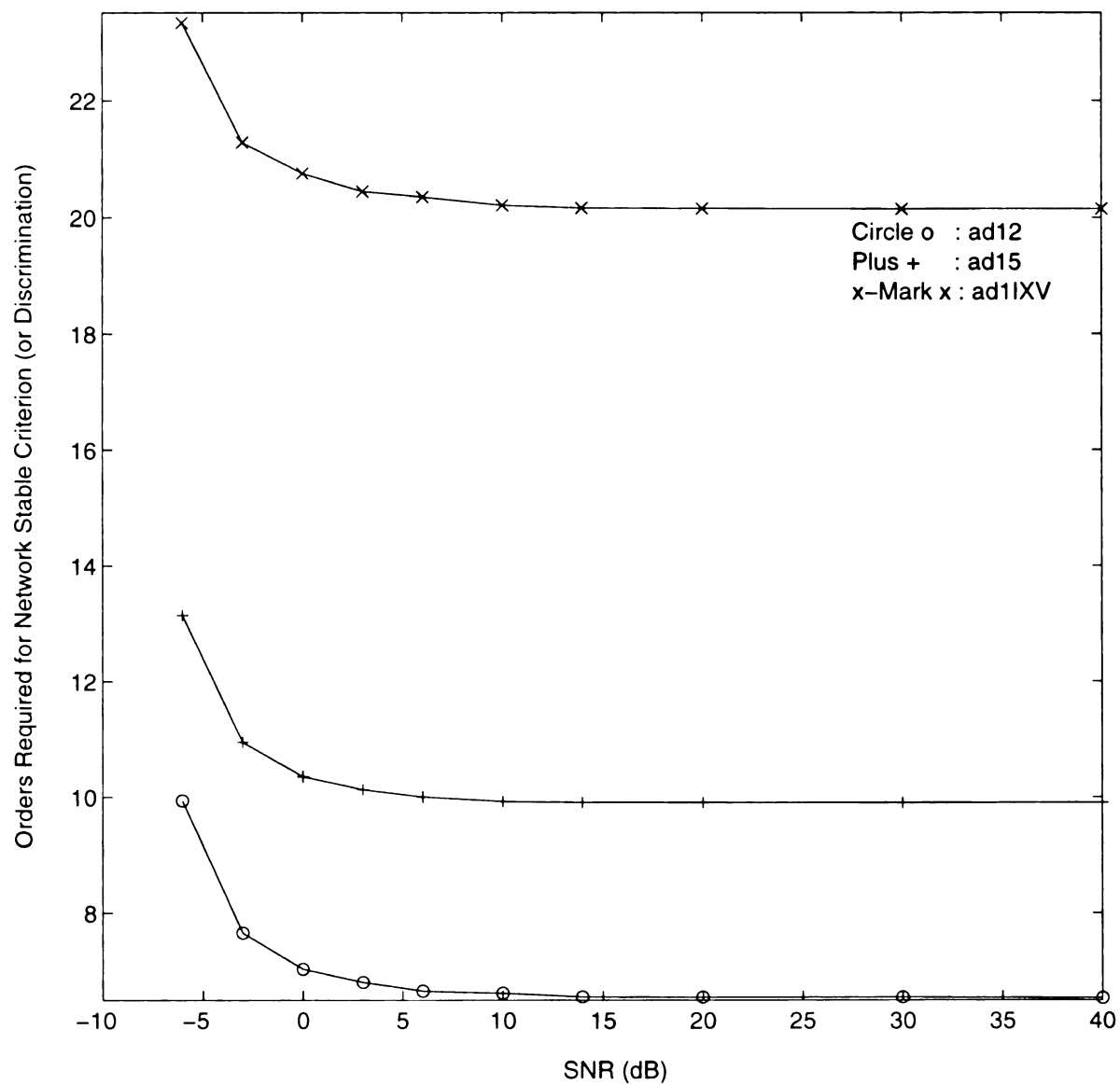


Figure 5.69 Orders required for the discriminations of RCAAM/ad1 vs. SNR for contaminated stored patterns.

Comparing the analog portion with the bipolar portion in Table 5.1, we can find the analog input data has lower correlation-based discrimination resolution than the encoded bipolar data. This indicates that for the same input, the encoded bipolar form data is more easily discriminated than the analog form, and the bipolar input form requires smaller discrimination order from the RCAAM/fi than the analog input form does from the RCAAM/ad. This comparison of correlation-based discrimination resolutions is consistent with the results of **Figure 5.68** and **Figure 5.69**. For bipolar input targets, the input target B52 has the most 0's in the normalized crosscorrelation gains w.r.t. the other targets. Thus the target B52 is the most easily recognized target under the RCAAM/fi network manipulation subject to an assumed threshold of 30%.

Table 5.2 shows the comparisons of normalized target correlation gains subject to a threshold of 20% between analog data and the encoded bipolar data. For bipolar input form, an active threshold of 20% means a stored pattern can be an active pattern only if it has a correlation greater than 60 ($300 \times 20\%$) with the bipolar input. It is equivalent to say an active stored pattern at least has 180 ($150 + 60/2$) bits out of 300 bits the same as the bipolar input. Under this assumed active threshold, the analog input form brings 39.1% stored patterns into active operation, while the encoded bipolar input form makes 20.02% stored patterns active. In Table 5.2, in comparison to Table 5.1, there are two normalized target crosscorrelation gains, the terms of TR1 w.r.t. input target B58 and the terms of B58 w.r.t. input target TR1, of bipolar form getting greater than those of analog form. Strictly speaking of those two cross targets, the analog form has higher discrimination resolution than the bipolar form. But, in general, the bipolar form still has a lot higher discrimination resolution than the analog

form.

Table 5.3 presents the comparisons of normalized target correlation gains subject to a threshold of 10% between analog data and the encoded bipolar data. This assumed active threshold makes the analog input form induce 64.06% stored patterns active and the encoded bipolar input form induce 55.97% stored patterns active. For bipolar input form, to be an active pattern only requires a correlation greater than 30 ($300 \times 10\%$) with the bipolar input. This means an active stored pattern may have 135 ($150 - 30/2$) bits out of 300 bits different than the bipolar input. Besides the two cross terms appearing in Table 5.2, there are another two normalized target crosscorrelation gains, the term of F14 w.r.t. input target B58 and the term of B58 w.r.t. input target F14, of bipolar form changing to greater than those of analog form.

Summation is used in (11) to enhance the same target code, since assumption 1 has assumed the output codes are consistent within each target. The assumption implicitly indicates a larger active threshold can confine active patterns within a more consistent region. Then the active correlation gains are closer, and the summation becomes more meaningful. The equality in (12) is approached when the correlation gains become closer. Therefore, a larger active threshold not only represents a better practical situation but also makes the estimates of the normalized target correlation gain formula more consistent and acceptable. Therefore, an active population of more than 50% not only becomes impractical, but also violates the assumptions.

Input Data Form	No. (%) of Active Patterns w.r.t. a Threshold of 30%	Input Target	Normalized Correlation Gain of B52 w.r.t. Input Target	Normalized Correlation Gain of B58 w.r.t. Input Target	Normalized Correlation Gain of F14 w.r.t. Input Target	Normalized Correlation Gain of TR1 w.r.t. Input Target
Analog	14.79 (21.76%)	Target B52	1.0000	0.2777	0.3623	0.2801
		Target B58	0.2176	1.0000	0.0507	0.1214
		Target F14	0.2289	0.0409	1.0000	0.6048
		Target TR1	0.1741	0.0964	0.5951	1.0000
Bipolar	5.91 (8.69%)	Target B52	1.0000	0	0	0
		Target B58	0	1.0000	0	0.0280
		Target F14	0	0	1.0000	0.1402
		Target TR1	0	0.0236	0.1518	1.0000

Table 5.1 Comparisons of normalized target correlation gains subject to an assumed active threshold of 30% between analog data and encoded bipolar data.

Input Data Form	No. (%) of Active Patterns w.r.t. a Threshold of 20%	Input Target	Normalized Correlation Gain of B52 w.r.t. Input Target	Normalized Correlation Gain of B58 w.r.t. Input Target	Normalized Correlation Gain of F14 w.r.t. Input Target	Normalized Correlation Gain of TR1 w.r.t. Input Target
Analog	26.59 (39.10%)	Target B52	1.0000	0.4431	0.5247	0.3753
		Target B58	0.3392	1.0000	0.1927	0.2505
		Target F14	0.3609	0.1740	1.0000	0.7168
		Target TR1	0.2524	0.2196	0.6999	1.0000
Bipolar	14.29 (21.02%)	Target B52	1.0000	0.1020	0.0144	0.0909
		Target B58	0.0608	1.0000	0.1115	0.2914
		Target F14	0.0071	0.0917	1.0000	0.3398
		Target TR1	0.0478	0.2567	0.3644	1.0000

Table 5.2 Comparisons of normalized target correlation gains subject to an assumed active threshold of 20% between analog data and encoded bipolar data.

Input Data Form	No. (%) of Active Patterns w.r.t. a Threshold of 10%	Input Target	Normalized Correlation Gain of B52 w.r.t. Input Target	Normalized Correlation Gain of B58 w.r.t. Input Target	Normalized Correlation Gain of F14 w.r.t. Input Target	Normalized Correlation Gain of TR1 w.r.t. Input Target
Analog	43.56 (64.06%)	Target B52	1.0000	0.5759	0.6286	0.4719
		Target B58	0.4585	1.0000	0.3614	0.3617
		Target F14	0.4655	0.3390	1.0000	0.7909
		Target TR1	0.3360	0.3253	0.7720	1.0000
Bipolar	38.06 (55.97%)	Target B52	1.0000	0.4132	0.1662	0.2858
		Target B58	0.2666	1.0000	0.3867	0.5235
		Target F14	0.1007	0.3658	1.0000	0.5391
		Target TR1	0.1794	0.5209	0.5624	1.0000

Table 5.3 Comparisons of normalized target correlation gains subject to an assumed active threshold of 10% between analog data and encoded bipolar data.

5.7 Performances of RCAAM-GI Cascade Networks

From previous RCAAM/di, RCAAM/fi, and RCAAM/ad simulation performances, we see that the unrecognized discriminations reduce when the stability criterion increases. At the same time, the correct target discriminations of the RCAAM/di and RCAAM/fi are apparently improved while the wrong target discriminations rise only slightly. For the RCAAM/ad, the correct target discriminations increase without raising the wrong target discriminations. Although a higher **sc** will improve discrimination performances, especially for the networks with low initial orders, a higher **sc** will take more processing (calculation) time as well as larger computing memory (i.e. larger **aord**) to satisfy the higher stability criterion. And a high initial order not only has weakened adaptive elasticity and contamination observability but also raises the **aord**, i.e. computing space. A low initial order gives a high adaptive elasticity, and a low **sc** neither renders all convergeable correct target discriminations nor converges all possible wrong discriminations. Therefore, a low initial order and **sc** will quite often leave an ambiguous input in semi-stable states, which are stable only for a few iterations. The Generalized Inverse (GI) network is further trained after an initial correlation-based memory coding. It has the hybrid characteristic of being capable of converging the spurious states within attractive basins to their associative target codes. We should remember that the GI network has a noticeable advantage that it prefers to leave an ambiguous state unrecognized rather than converge it to a wrong target. Thus a combination network leaving the RCAAM with low initial orders and **sc** and cascaded with a GI network, will become more efficient without sacrificing possible convergence. This thought seems unreasonable, since taking advantage without losing profits anywhere is somehow contrary to nature.

5.7.1 Converging Efficiency Comparisons Between the RCAAM and the RCAAM-GI Cascade Network

Figure 5.70 shows the unrecognized discrimination comparisons between the RCAAM/di2 and the RCAAM/di22-GI cascade network along with stability criterion for both contaminated stored and unstored patterns with 40 dB and 0 dB SNR. The RCAAM/di22-GI cascade network converges all unknowns for all cases, while the RCAAM/di2 with a high sc of 10 still struggles for convergence for both contaminated stored and unstored patterns for 0 dB SNR. Therefore, a low sc of 2 in the RCAAM/di22-GI cascade network at least has saved the discrimination orders integrated by an increment of 8 in sc used by RCAAM/di2X where X denotes 10. Thus the RCAAM/di22-GI cascade network is much more efficient in processing time and space than the RCAAM/di2.

Figure 5.71 presents the unrecognized discrimination comparisons between the RCAAM/fi1 and the RCAAM/fi12-GI cascade network along with stability criterion for both contaminated stored and unstored patterns for 40 dB and 0 dB SNR. For the RCAAM/fi12-GI cascade network, the unknowns are not completely converged only for the heavy contamination case, i.e. 0 dB SNR. As discussed before, serious contamination will result in coding linearity failure and then the mechanism of RCAAM/fi will make the output state fall into oscillation if the dominant correlation gain is negative. Therefore, the existence of unknowns should be attributed to the nonlinear coding scheme and RCAAM/fi mechanism. It is noticeable that the RCAAM/fi12-GI cascade network leaves no unknowns for contaminated unstored patterns for 40 dB SNR, while the RCAAM/fi1 couldn't converge all unknowns even when an sc as large as 40 is used. This time the efficiency benefit becomes

huge when using an sc as small as 2 in RCAAM/fi12-GI instead of a sc as large as 40 in RCAAM/fi1IVX where IVX stands for 40. Later we will show that the performances of RCAAM/fi12-GI network for contaminated unstored patterns under the 40 dB won't drop, but get better than the RCAAM/fi1IVX.

Figure 5.72 shows the unrecognized discrimination comparisons between the RCAAM/ad1 and the RCAAM/ad12-GI cascade network along with stability criterion for both contaminated stored and unstored patterns for 40 and 0 dB SNR. For the RCAAM/ad12-GI cascade network, the unknowns are all converged for all cases, since the analog data has no nonlinearity interference problem even under severe distortions. The RCAAM/ad1 still needs an sc of 15, larger than the sc of 2 used by RCAAM/fi1, to converge unknowns for slightly contaminated stored patterns. The reason for this, as given in the previous section, is that the analog data has lower correlation-based discrimination resolution than the encoded bipolar data form. The RCAAM/ad1 converges the slightly contaminated unstored patterns only when a high sc of 30 is adopted. Compared to the RCAAM/di, the analog data form is acting on a converging delay, so that the effect has further emphasized the efficiency of the RCAAM/ad12-GI cascade network.

5.7.2 Noise Tolerance Comparisons Between the RCAAM and RCAAM-GI Cascade Network

From the last subsection, we know that the converging efficiency of the RCAAM-GI cascade network is much better than the RCAAM. But we don't know if the RCAAM-GI cascade network can still have good discrimination effect when its previous stage network, the RCAAM, only sets a low stability criterion. Here we simulate the RCAAM with three sc 's

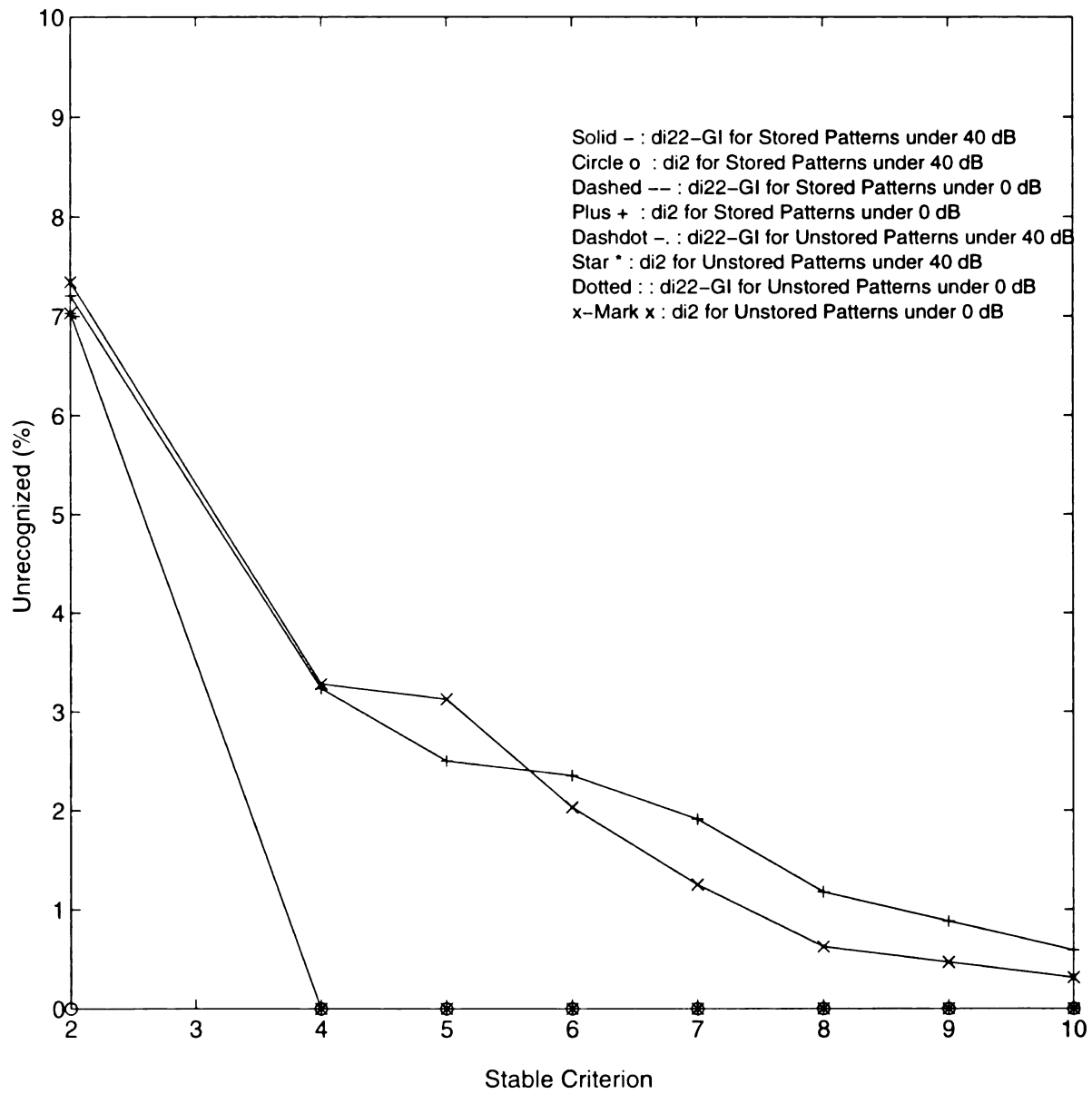


Figure 5.70 Unrecognized discriminations of the RCAAM/di2 and RCAAM/di22-GI vs. initial order for 40 dB and 0 dB SNR.

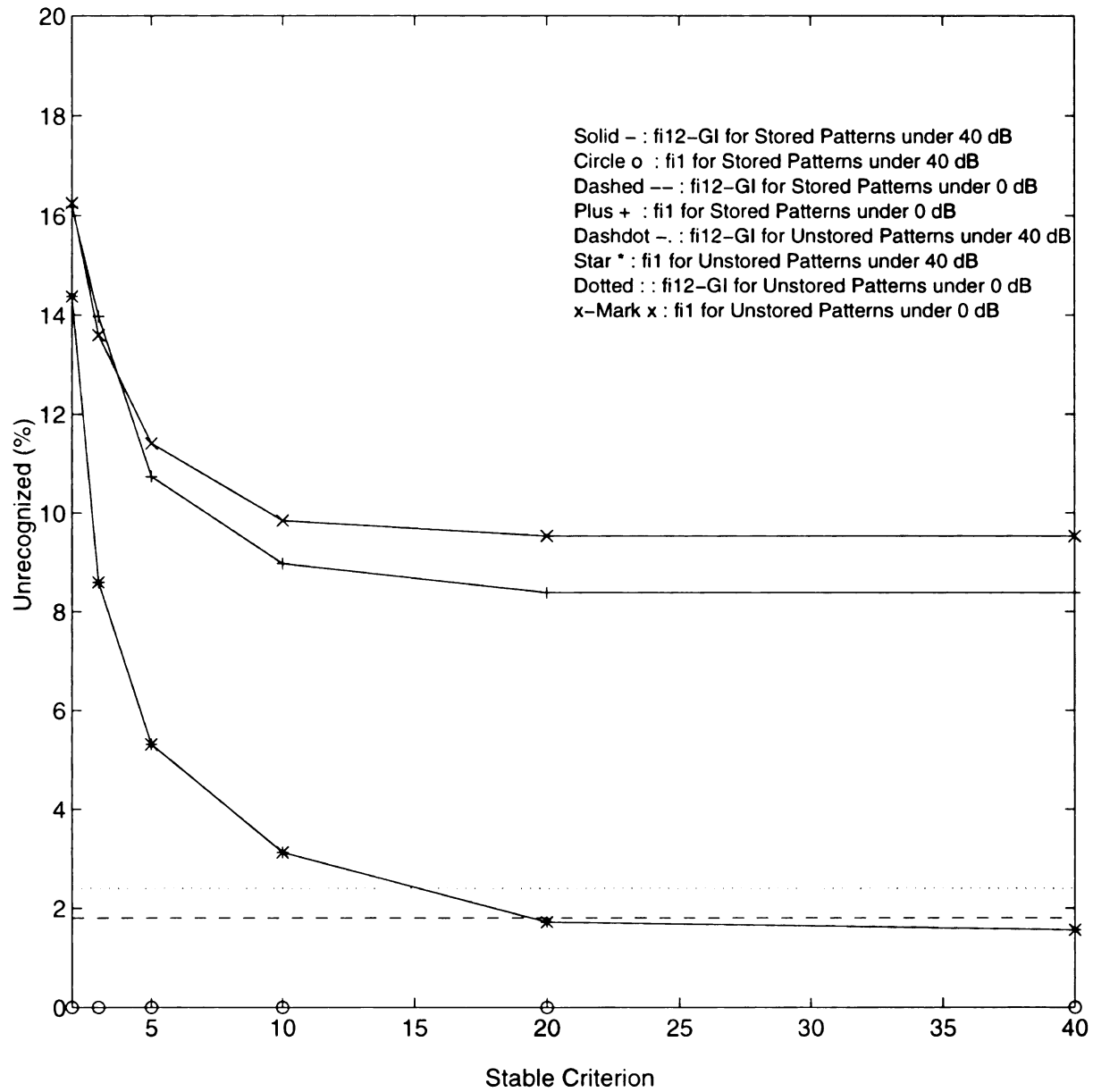


Figure 5.71 Unrecognized discriminations of the RCAAM/fi1 and RCAAM/fi12-GI vs. initial order for 40 dB and 0 dB SNR.

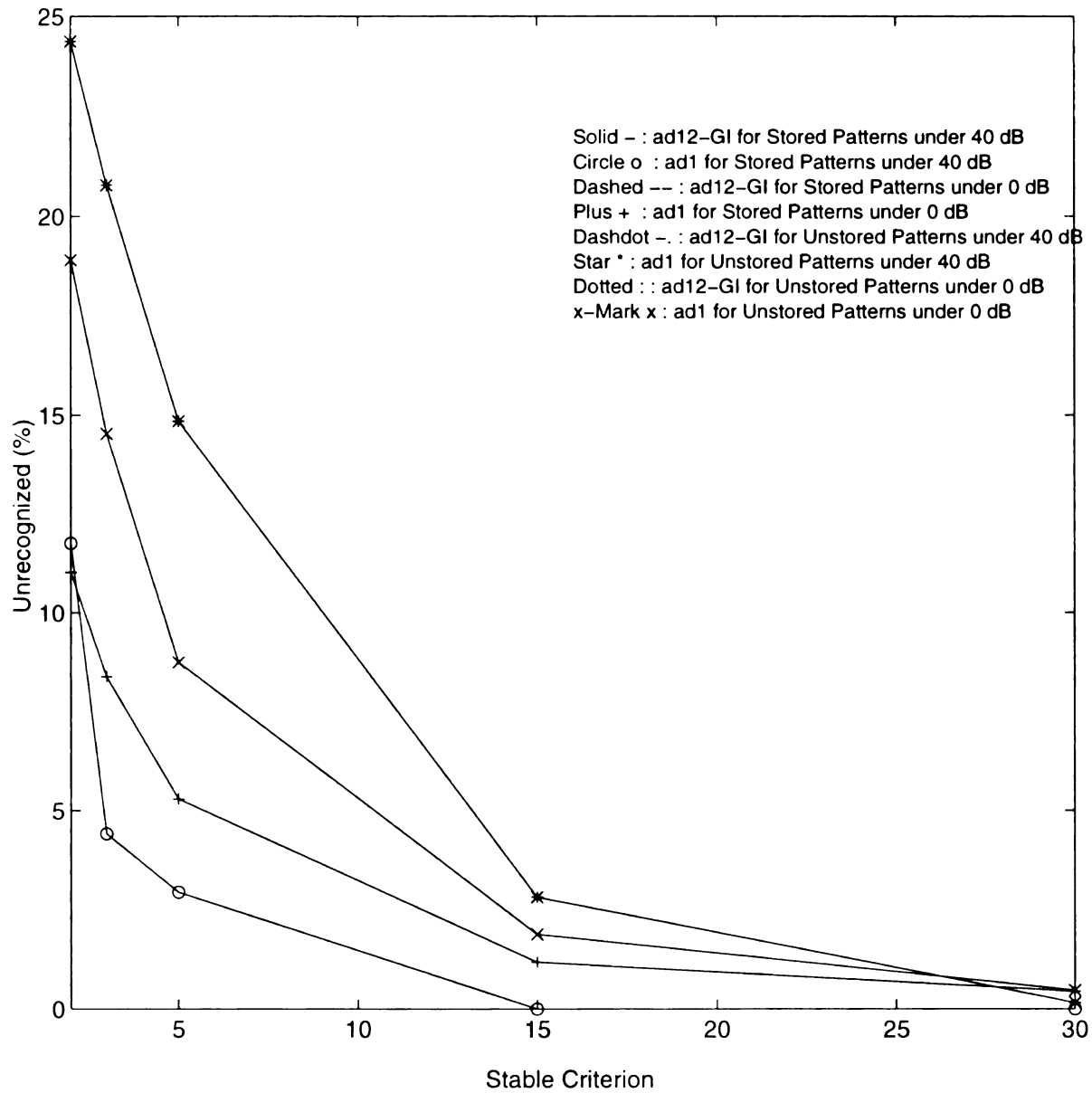


Figure 5.72 Unrecognized discriminations of the RCAAM/ad1 and RCAAM/ad12-GI vs. initial order for 40 dB and 0 dB SNR.

under different contamination levels, and then compare their discrimination performances with those of the RCAAM-GI cascade network using the lowest sc .

Figure 5.73 shows the correct target discriminations of the RCAAM/di22-GI cascade network and the RCAAM/di2 for contaminated stored patterns vs. SNR in (a), and the wrong target discriminations in (b). **Figure 5.74** presents the unrecognized discriminations of the RCAAM/di22-GI cascade network and the RCAAM/di2 for contaminated stored patterns vs. SNR in (a), and discrimination orders in (b). The correct target discriminations always increase when the RCAAM/di2 changes its sc from 2 to 5 and then from 5 to 10. Corresponding to the increment in sc , the wrong target discriminations show no change for fair contaminations, and only rise slightly for serious noises. In both correct and wrong target discrimination performances, the RCAAM/di2X's performances approximate the RCAAM/di22-GI's. It seems that the RCAAM/di2's performances will converge to the RCAAM/di22's while the sc continues increasing. The unrecognized discrimination performances for the RCAAM/di2 and RCAAM/di22-GI network also reveal this converging characteristic. Although the RCAAM/di22-GI cascade network only sets an sc of 2, it leaves the fewest unknowns behind for all contamination situations. The large differences between the discrimination orders needed for the RCAAM/di2 and the ones for the RCAAM/di22-GI network again demonstrates the RCAAM/di22-GI's high efficiency. Although the RCAAM/di2X expends a space about 8 orders higher than RCAAM/di22-GI network, it doesn't give better discrimination performances.

Figure 5.75 shows the correct target discriminations of the RCAAM/di22-GI cascade network and the RCAAM/di2 for contaminated unstored patterns vs. SNR in (a), and the

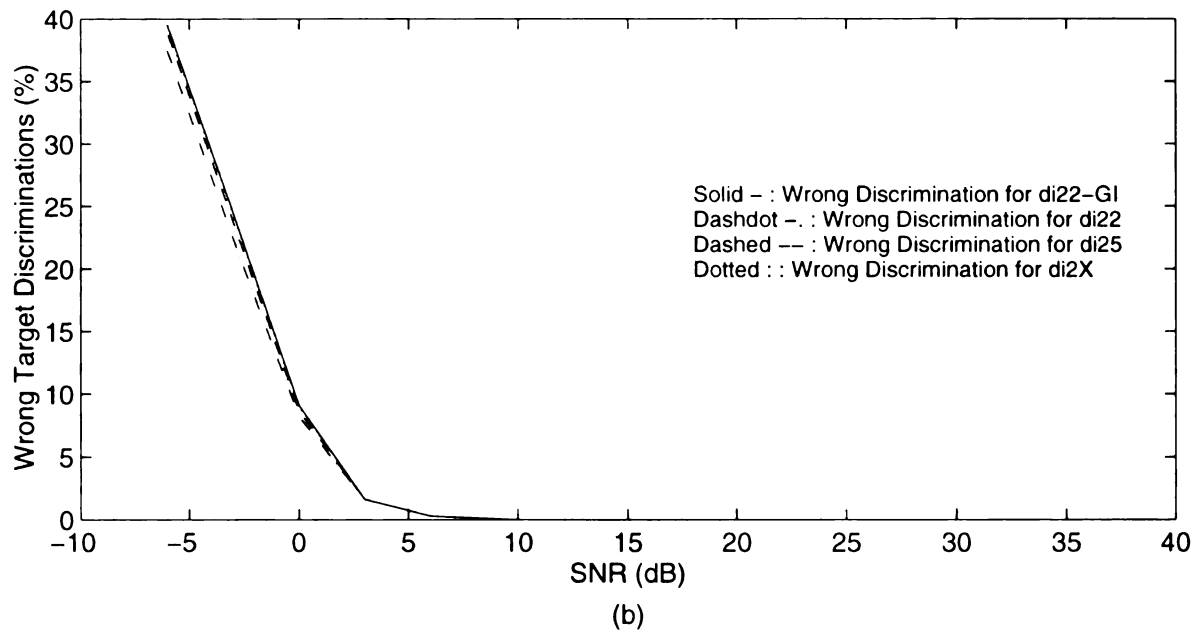
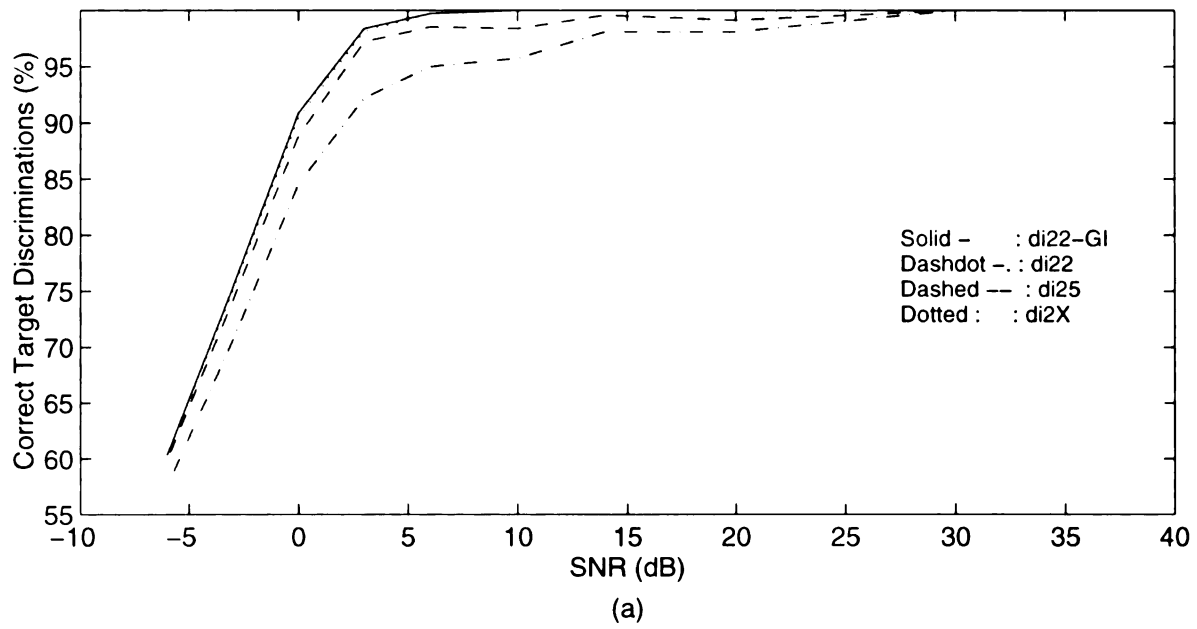


Figure 5.73 Performance comparisons of the RCAAM/di's with an initial order of 2 and RCAAM/di22-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.

(a) Correct target discrimination vs. SNR for the 68 stored patterns.

(b) Wrong target discrimination vs. SNR for the 68 stored patterns.

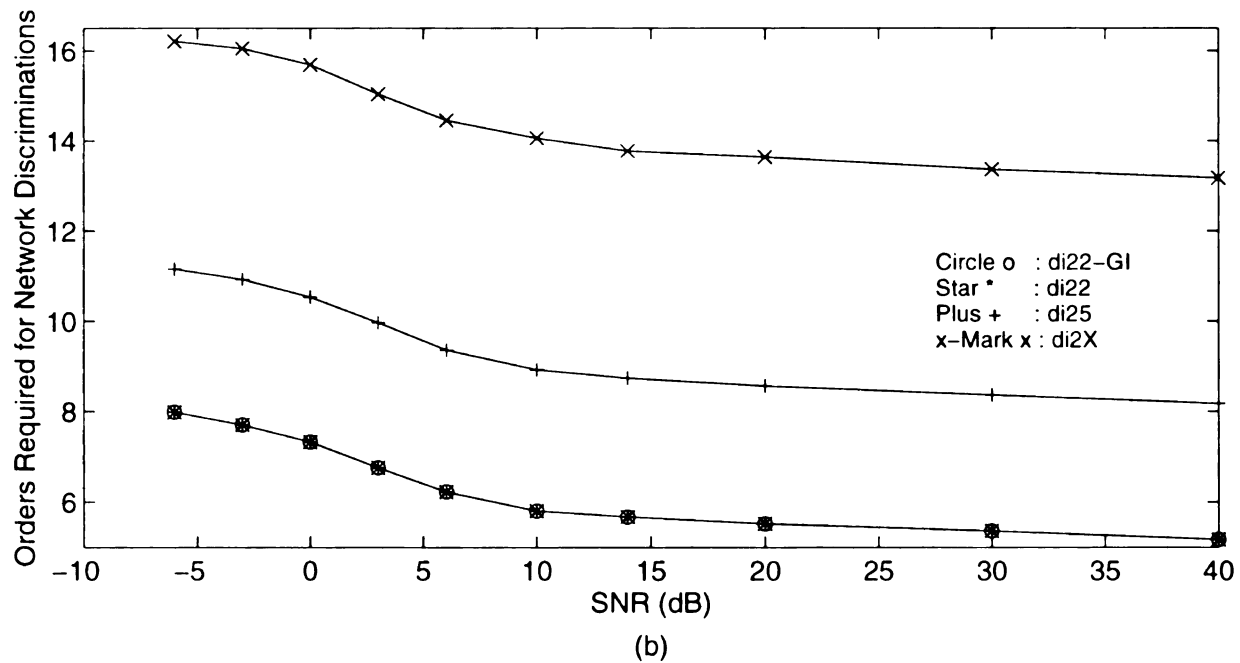
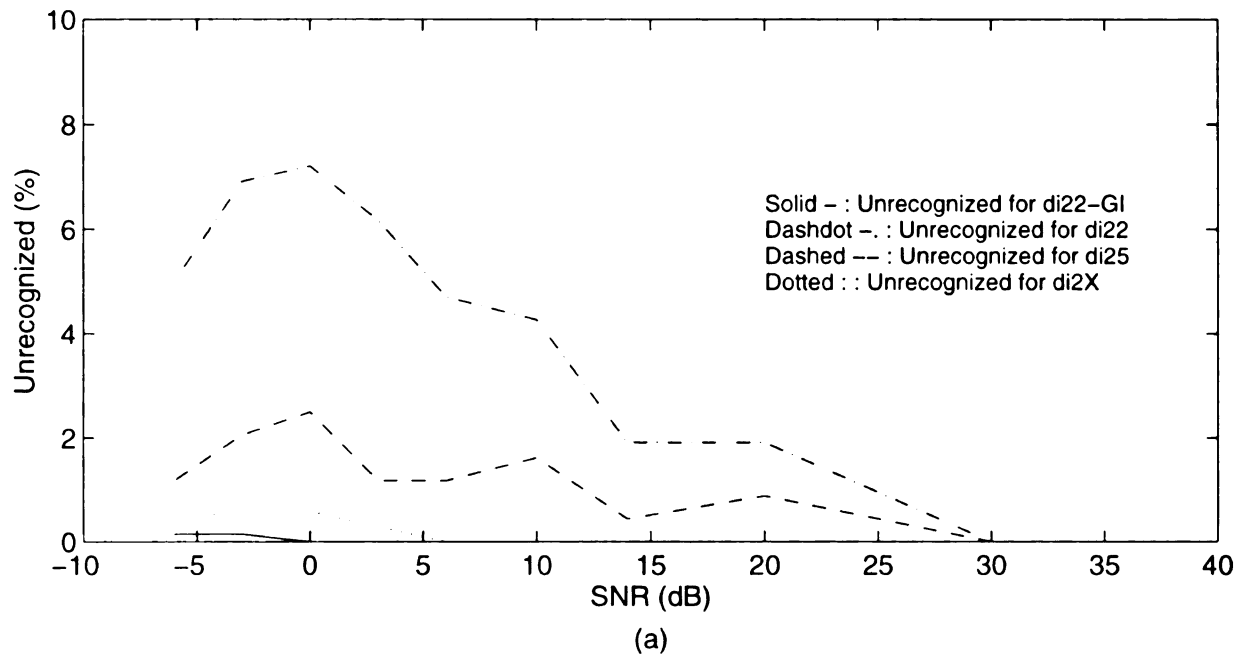


Figure 5.74 Unrecognized discriminations and discrimination orders of the RCAAM/di's with an initial order of 2 and RCAAM/di22-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.

(a) Unrecognized discriminations vs. SNR for the 68 stored patterns.

(b) Orders required for discriminations vs. SNR for the 68 stored patterns.

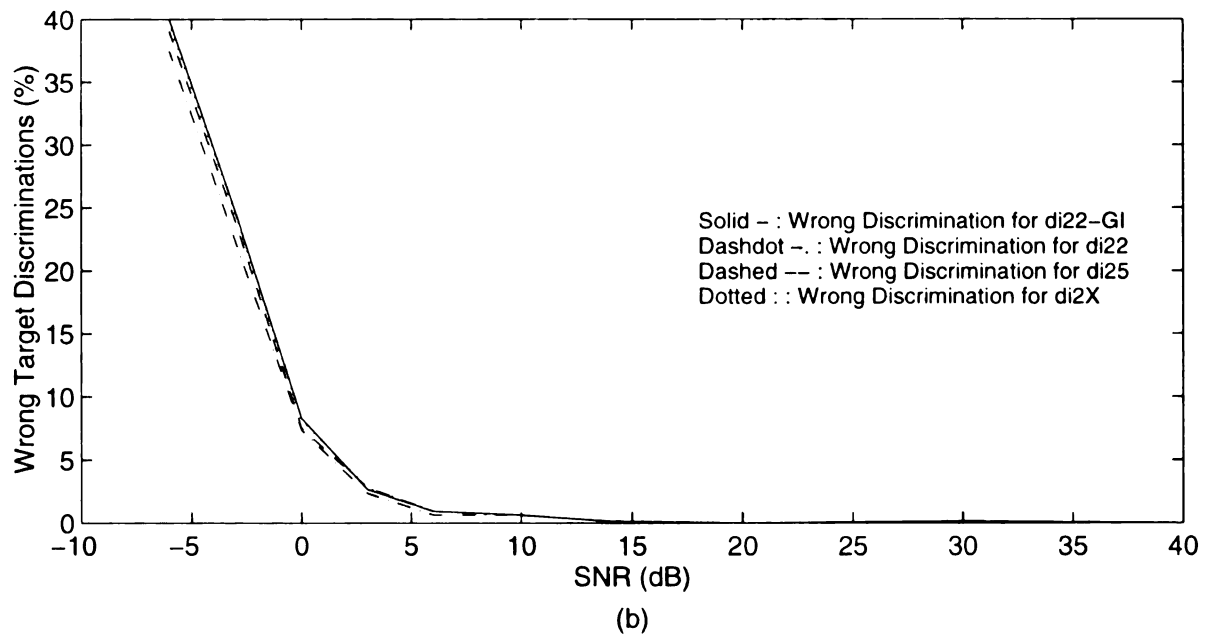
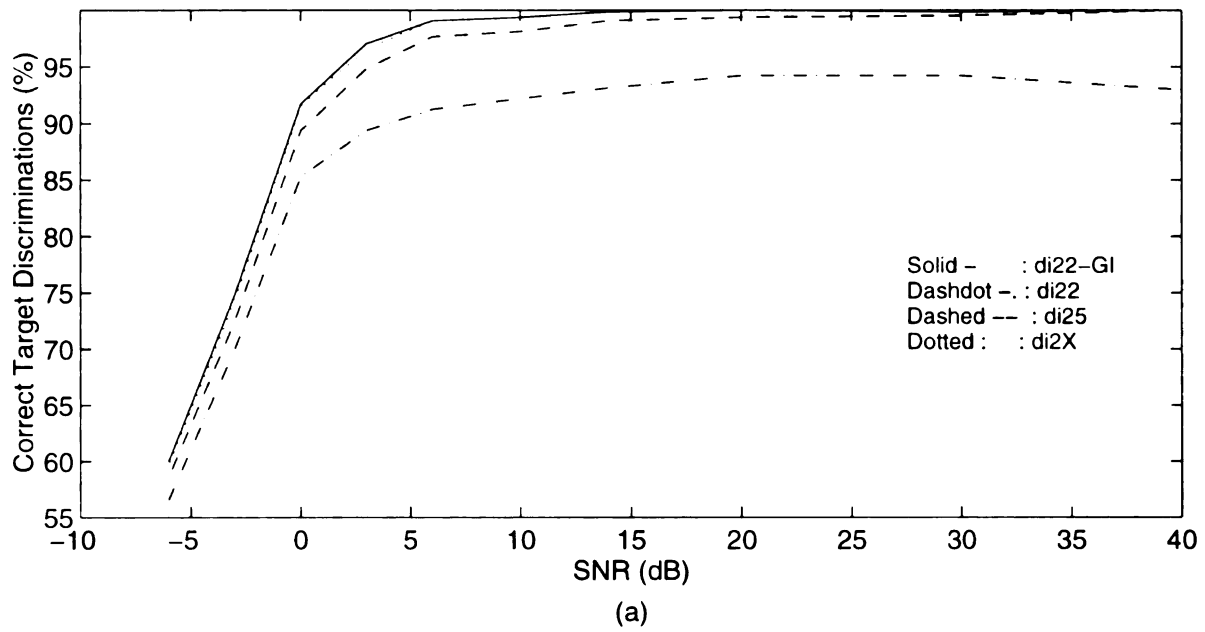


Figure 5.75 Performance comparisons of the RCAAM/di's with an initial order of 2 and RCAAM/di22-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

(a) Correct target discriminations vs. SNR for the 64 unstored patterns.

(b) Wrong target discriminations vs. SNR for the 64 unstored patterns.

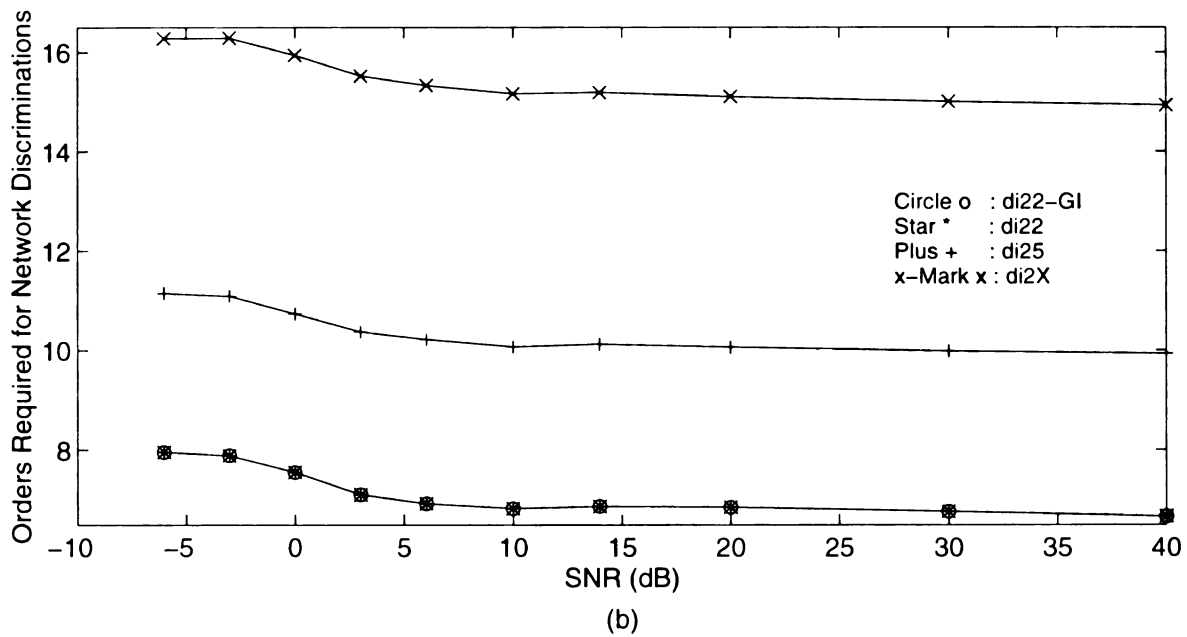
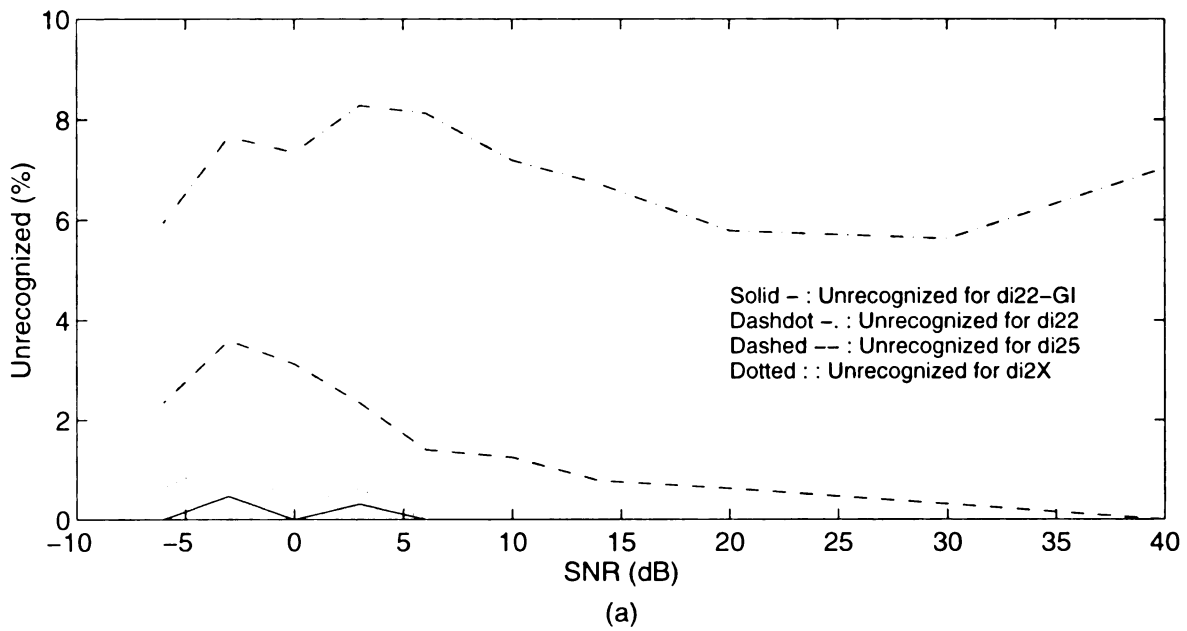


Figure 5.76 Unrecognized discriminations and discrimination orders of the RCAAM/di's with an initial order of 2 and RCAAM/di22-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

(a) Unrecognized discriminations vs. SNR for the 64 unstored patterns.

(b) Orders required for discriminations vs. SNR for the 64 unstored patterns.

wrong target discriminations in (b). **Figure 5.76** presents the unrecognized discriminations of the RCAAM/di22-GI cascade network and the RCAAM/di2 for contaminated unstored patterns vs. SNR in (a), and discrimination orders in (b). For unstored patterns, the RCAAM/di22-GI network's advantage of the RCAAM/di2 is again apparent. Specifically, the RCAAM/di25's correct target discrimination begins falling at 30 dB. The RCAAM/di22-GI network has the fewest unknowns and lowest discrimination orders for all contamination situations.

Figure 5.77 shows the correct target discriminations of the RCAAM/fi12-GI cascade network and the RCAAM/fi1 for contaminated stored patterns vs. SNR in (a), and the wrong target discriminations in (b). **Figure 5.78** presents the unrecognized discriminations of RCAAM/fi12-GI cascade network and the RCAAM/fi1 for contaminated stored patterns vs. SNR in (a), and discrimination orders in (b). It can be observed that the RCAAM/fi1's discrimination performances are inclined to converge to the RCAAM/fi12's, while the *sc* continues increasing. From the previous subsection, we have noticed that the RCAAM/fi12-GI cascade network is much more efficient in processing time and space than the RCAAM/fi1IVX, which set an *sc* as large as 40. By checking the ratios of correct to wrong target discrimination increments, especially for severe contaminations, we find that the RCAAM/fi12-GI network also has better discrimination performances than the RCAAM/fi1IVX. In **Figure 5.78**, the unrecognized rate doesn't apparently lowered, while the discrimination orders used by the RCAAM/fi1IVX get huge.

Figure 5.79 shows the correct target discriminations of the RCAAM/fi12-GI cascade network and the RCAAM/fi1 for contaminated unstored patterns vs. SNR in (a), and the

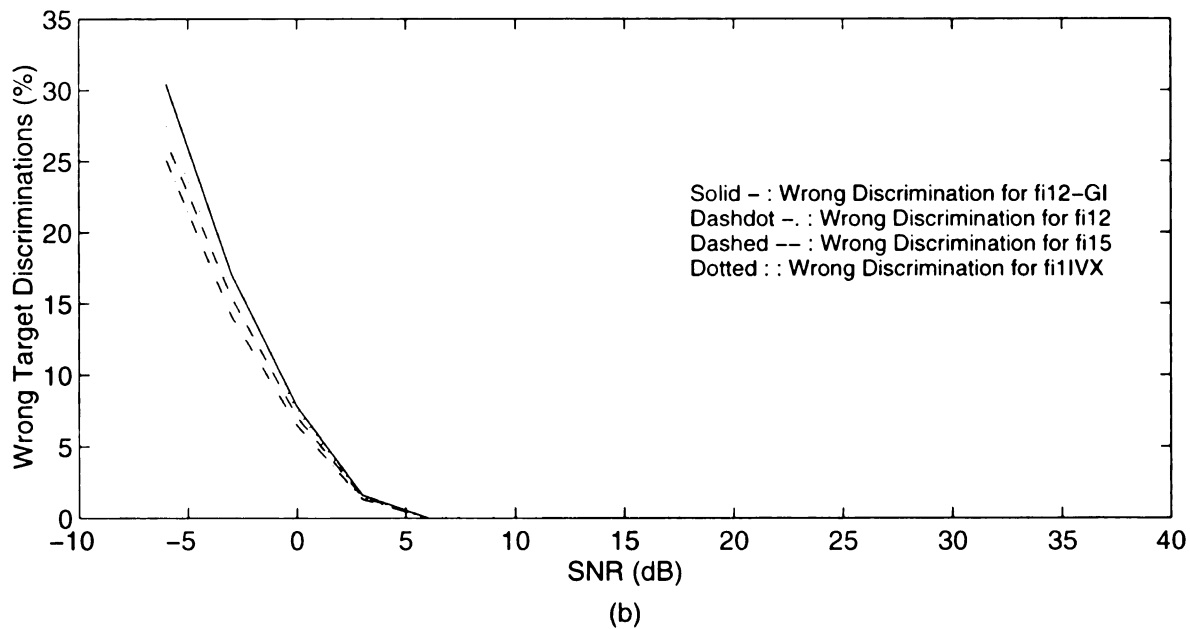
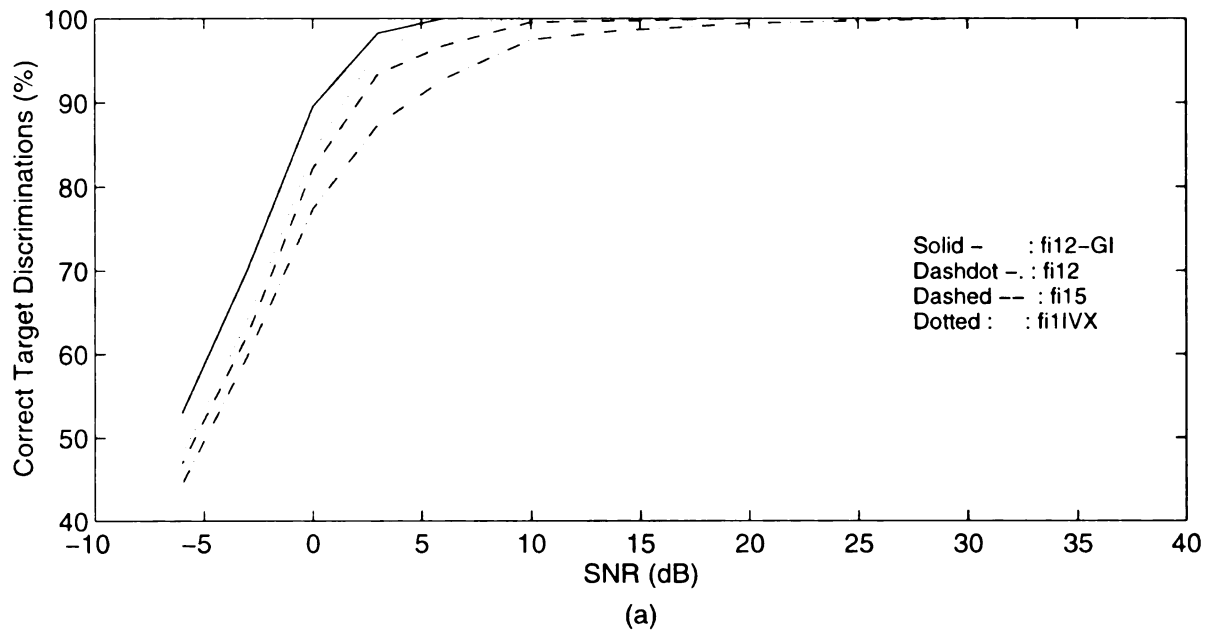


Figure 5.77 Performance comparisons of the RCAAM/fi's with an initial order of 1 and RCAAM/fi12-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.

(a) Correct target discriminations vs. SNR for the 68 stored patterns.

(b) Wrong target discriminations vs. SNR for the 68 stored patterns.

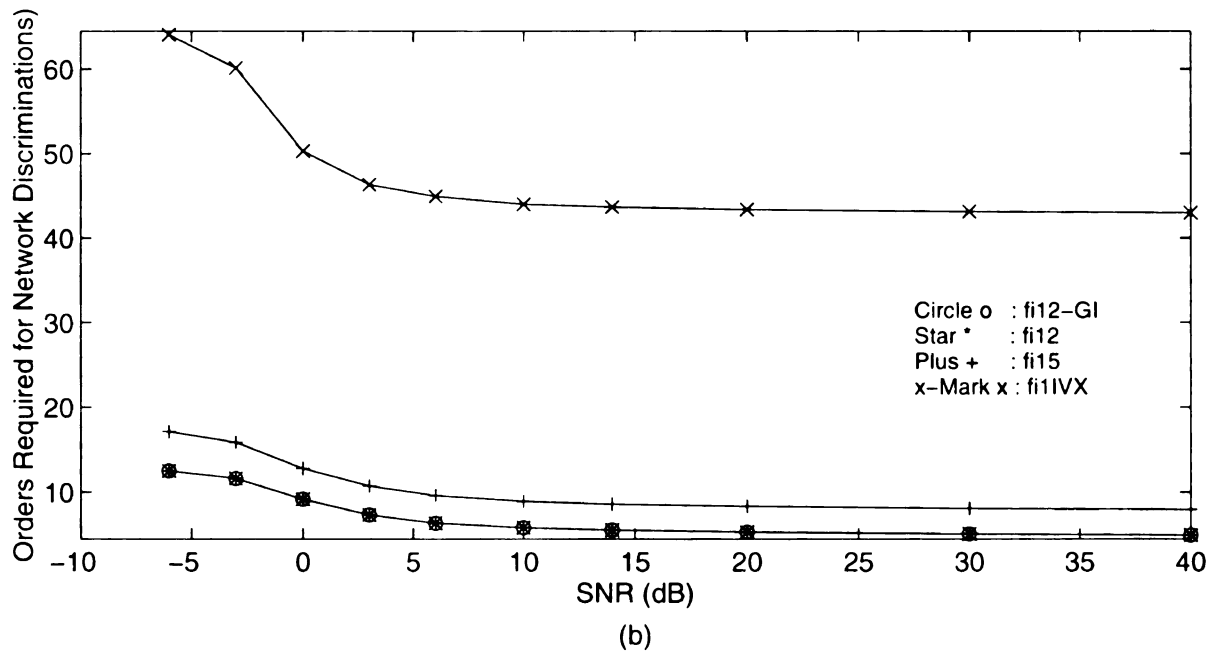
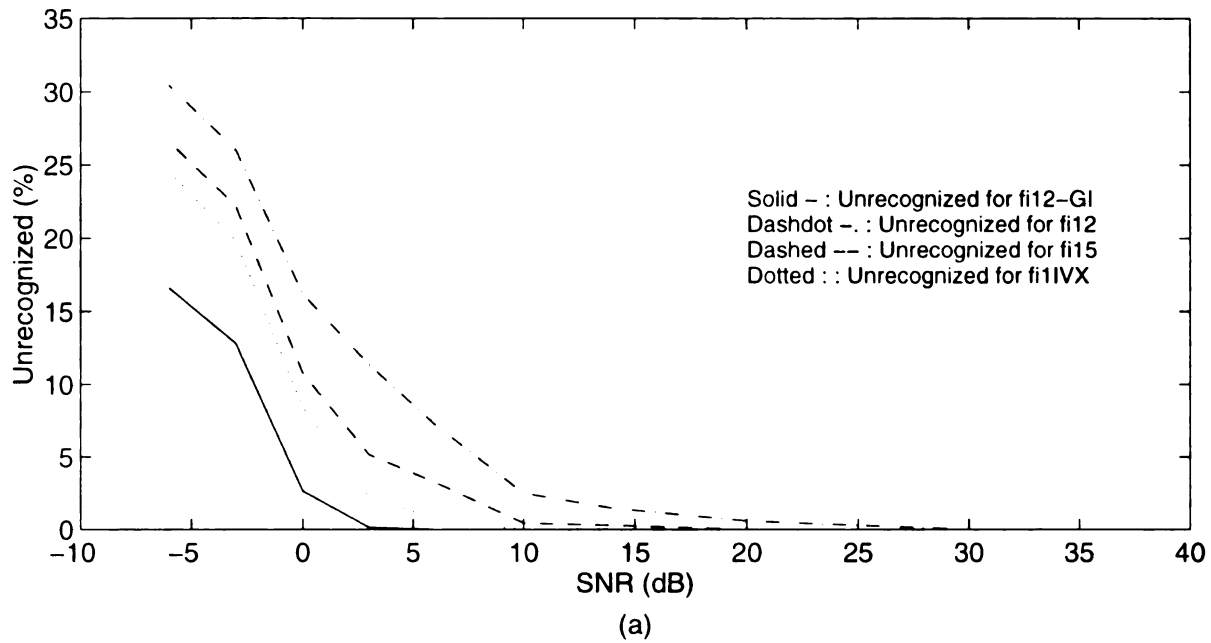


Figure 5.78 Unrecognized discriminations and discrimination orders of the RCAAM/fi's with an initial order of 1 and RCAAM/di12-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.

(a) Unrecognized discriminations vs. SNR for the 68 stored patterns.

(b) Orders required for discriminations vs. SNR for the 68 stored patterns.

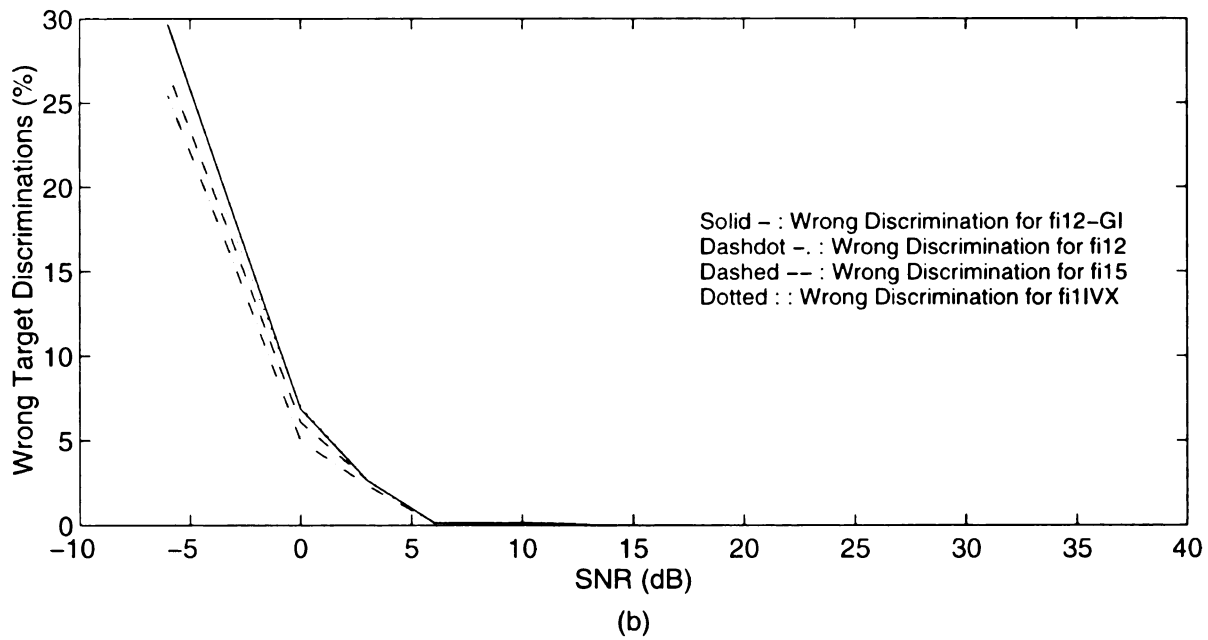
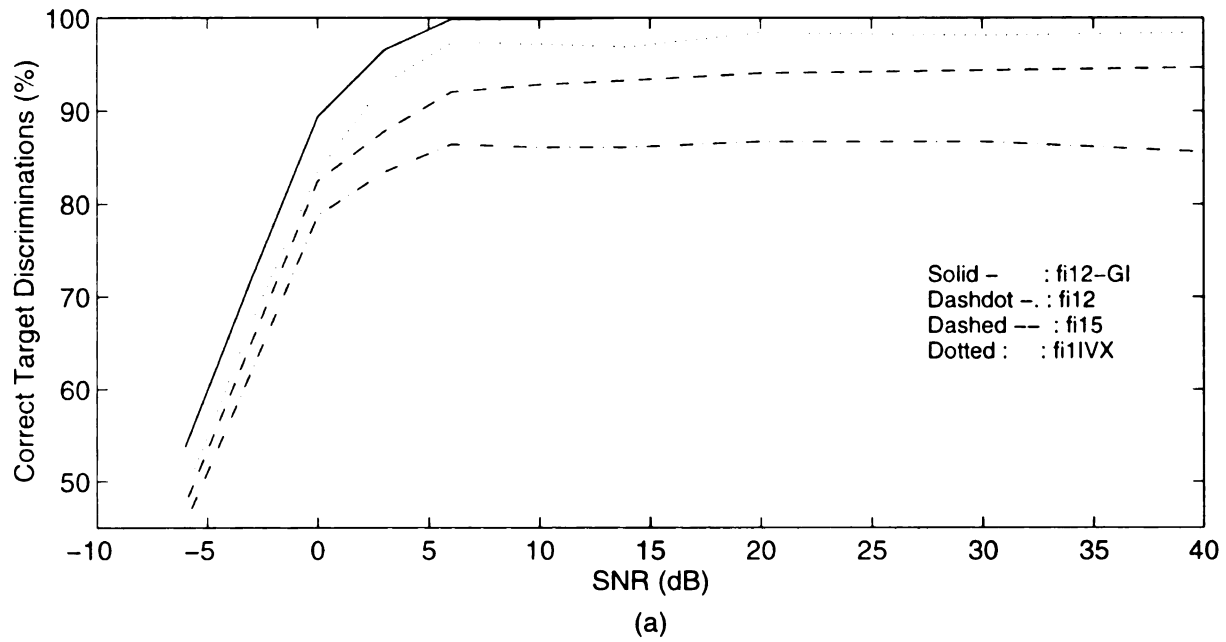


Figure 5.79 Performance comparisons of the RCAAM/fi's with an initial order of 1 and RCAAM/fi12-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

(a) Correct target discriminations vs. SNR for the 64 unstored patterns.

(b) Wrong target discriminations vs. SNR for the 64 unstored patterns.

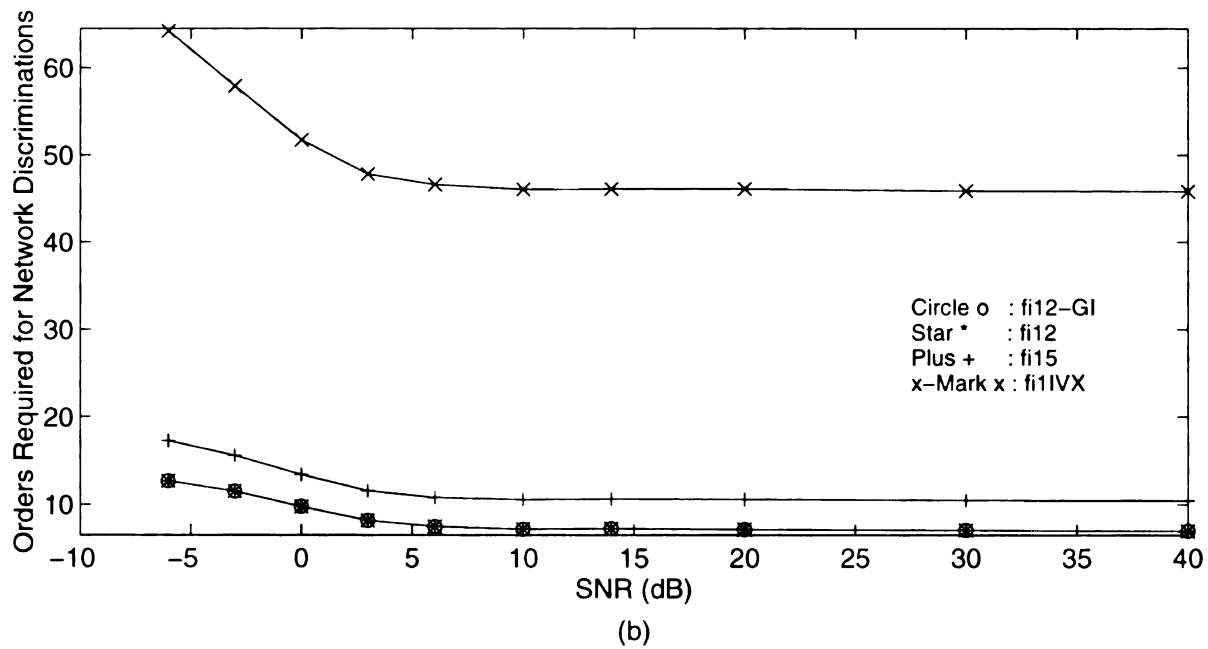
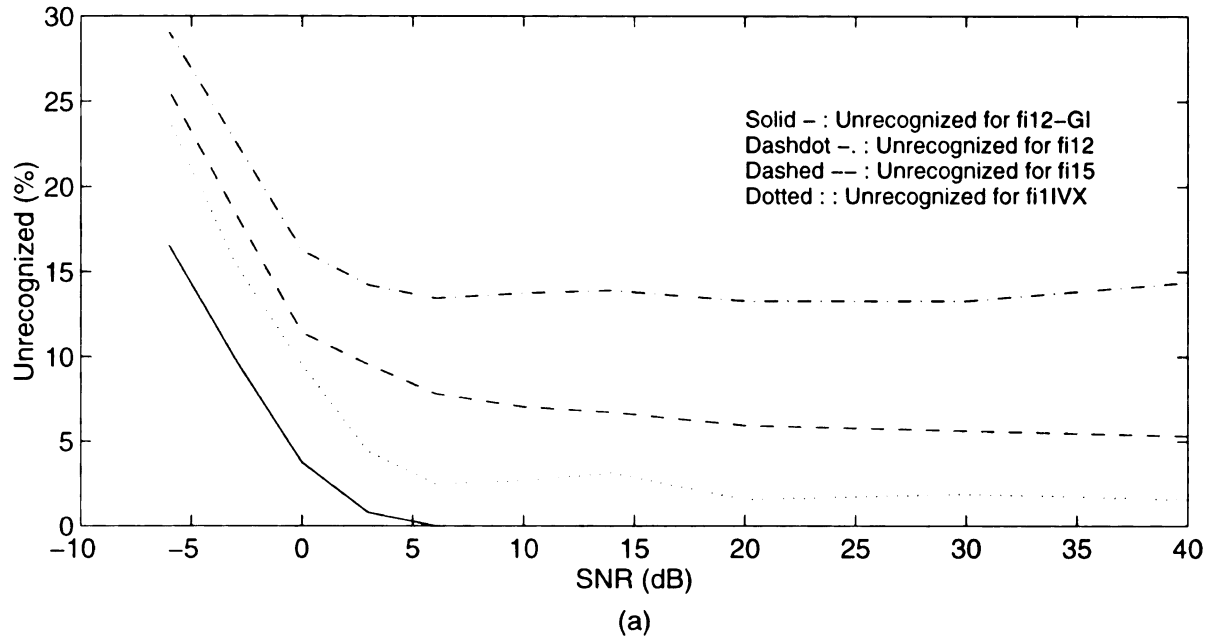


Figure 5.80 Unrecognized discriminations and discrimination orders of the RCAAM/fi's with an initial order of 1 and RCAAM/fi12-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

(a) Unrecognized discriminations vs. SNR for the 64 unstored patterns.

(b) Orders required for discriminations vs. SNR for the 64 unstored patterns.

wrong target discriminations in (b). **Figure 5.80** presents the unrecognized discriminations of the RCAAM/fi12-GI cascade network and the RCAAM/fi1 for contaminated unstored patterns vs. SNR in (a), and discrimination orders in (b). For unstored patterns, the RCAAM/fi12-GI cascade network's advantage of the RCAAM/fi1 is evident. All the RCAAM/fi1, including the one with a *sc* of 40, couldn't reach 100% correct target discrimination even for a slight contamination, while the cascade network with a low *sc* of 2 maintains the 100% correct target discrimination without difficulty until 3 dB. Although those unstable states, to which the RCAAM/fi1IVX is unable to converge, are ambiguous and oscillating, the RCAAM/fi12-GI cascade network still can converge them by their slightly revealed inclinations to correct stored patterns. The RCAAM/fi12-GI cascade network's excellent efficiency and great discrimination capability become much clearer in this case.

Figure 5.81 shows the correct target discriminations of the RCAAM/ad12-GI cascade network and the RCAAM/ad1 for contaminated stored patterns vs. SNR in (a), and the wrong target discriminations in (b). **Figure 5.82** presents the unrecognized discriminations of the RCAAM/ad12-GI cascade network and the RCAAM/ad1 for contaminated stored patterns vs. SNR in (a), and discrimination orders in (b). From the correct target discrimination performances, the analog data's deficiency, whose low correlation-based discrimination resolution makes it need a high discrimination order to converge, doesn't bother the RCAAM/ad12-GI cascade network. The RCAAM/ad12-GI cascade network always has 100% correct target discrimination until -3 dB by only using a low *sc* of 2, while the RCAAM/ad1 still struggles to converge a lot of unknowns until it sets a high *sc* of 15. The cascade network still retains the RCAAM/ad's remarkable rare wrong

target discrimination. Since the analog data has no nonlinearity inconsistency problem, the converging failures should be attributed to the low correlation-based discrimination resolution. The analog data form also brings us the low correlation-based discrimination resolution, while it is introduced to eliminate the bipolar coding inconsistency occurring with severe distortion. Then the low correlation-based discrimination resolution needs more orders for discrimination, while the elimination of the inconsistent disorder with severe contamination greatly reduces the discrimination order. Thus the analog data form raises the discrimination orders required for light contamination, but at the same time reduces the discrimination orders required for severe contamination. This is the reason why the RCAAM/ad's discrimination order curves w.r.t. SNR are flatter than the RCAAM/fi's and RCAAM/di's. This is a fair and balanced trade : getting advantage at one side and paying (or losing) profit on the other side in return. As discussed before, a flat discrimination order curve has low contamination observability.

Figure 5.83 shows the correct target discriminations of the RCAAM/ad12-GI cascade network and the RCAAM/ad1 for contaminated unstored patterns vs. SNR in (a), and the wrong target discriminations in (b). **Figure 5.84** presents the unrecognized discriminations of the RCAAM/ad12-GI cascade network and the RCAAM/ad1 for contaminated unstored patterns vs. SNR in (a), and discrimination orders in (b). Compared to contaminated stored patterns, the RCAAM/ad12-GI's discrimination performances almost haven't changed, while the RCAAM/ad1's decay a lot. Again, the cascade network discriminates without error until -6 dB. It's amazing that about 3% contaminated unstored patterns still stay in confused states for at least 15 iterations, when the GI network itself has

correctly converged about 24% unrecognized states left by the RCAAM/ad12. For both contaminated stored and unstored patterns, the RCAAM/ad1's discrimination performances again become converging to the RCAAM/ad12-GI's, while the stability criterion continues increasing.

From the above simulation result comparisons, the excellent efficiency, high converging capability and powerful discrimination ability of the RCAAM-GI cascade network are very impressive. The performance comparisons between the RCAAM and RCAAM-GI cascade network have offered an alternative way to appreciate what a great converging capability the GI network has. As soon as an ambiguous state moves into attractive basins of the stored patterns, the GI network will converge the state, even if still spurious, to one of the stored patterns. With this issue in mind, a network doesn't need to converge a contaminated input all the way to a real stable state. All it needs to do is to associate an input with all stored patterns a few times. Then the input, if not severely distorted, will reveal its inclination to some stored states, and thus move from an ambiguous state, if existing, into the realms of attractive basins. Now the network can leave the task of converging the attracted state to one of stored patterns to the GI network.

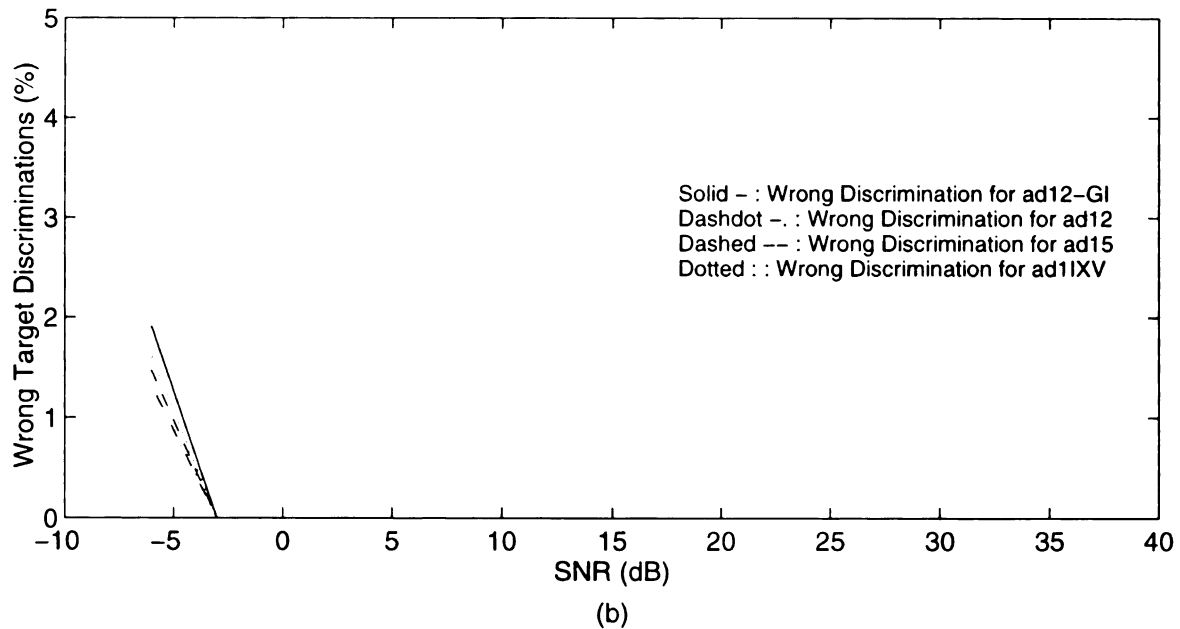
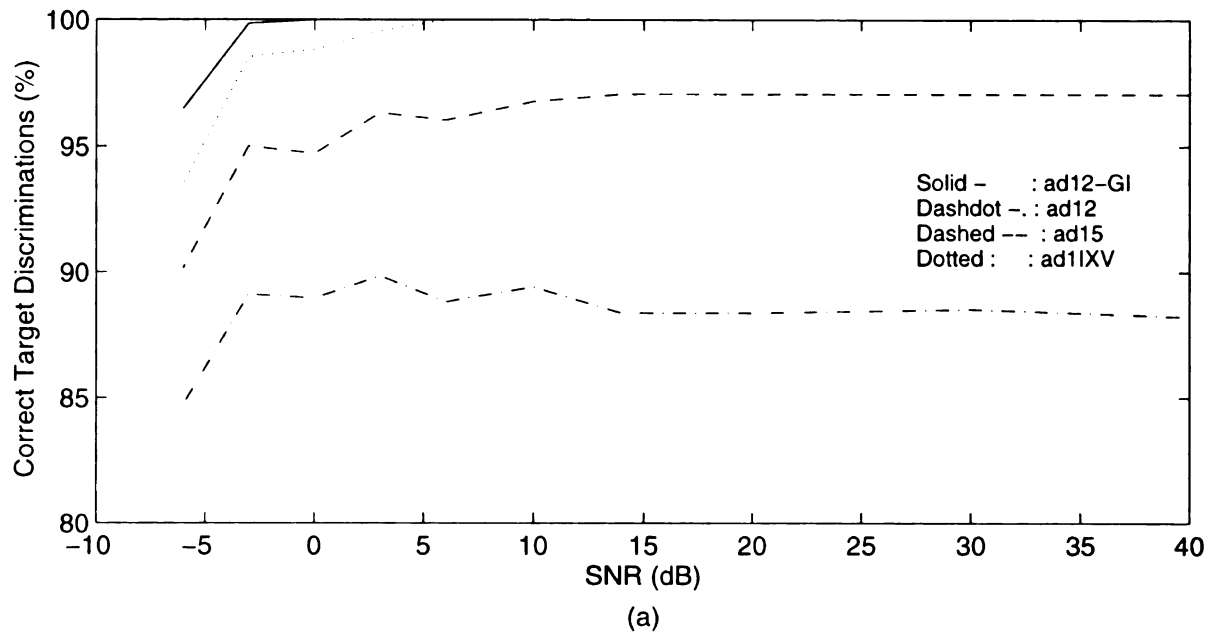


Figure 5.81 Performance comparisons of the RCAAM/ad's with an initial order of 1 and RCAAM/ad12-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.

(a) Correct target discriminations vs. SNR for the 68 stored patterns.

(b) Wrong target discriminations vs. SNR for the 68 stored patterns.

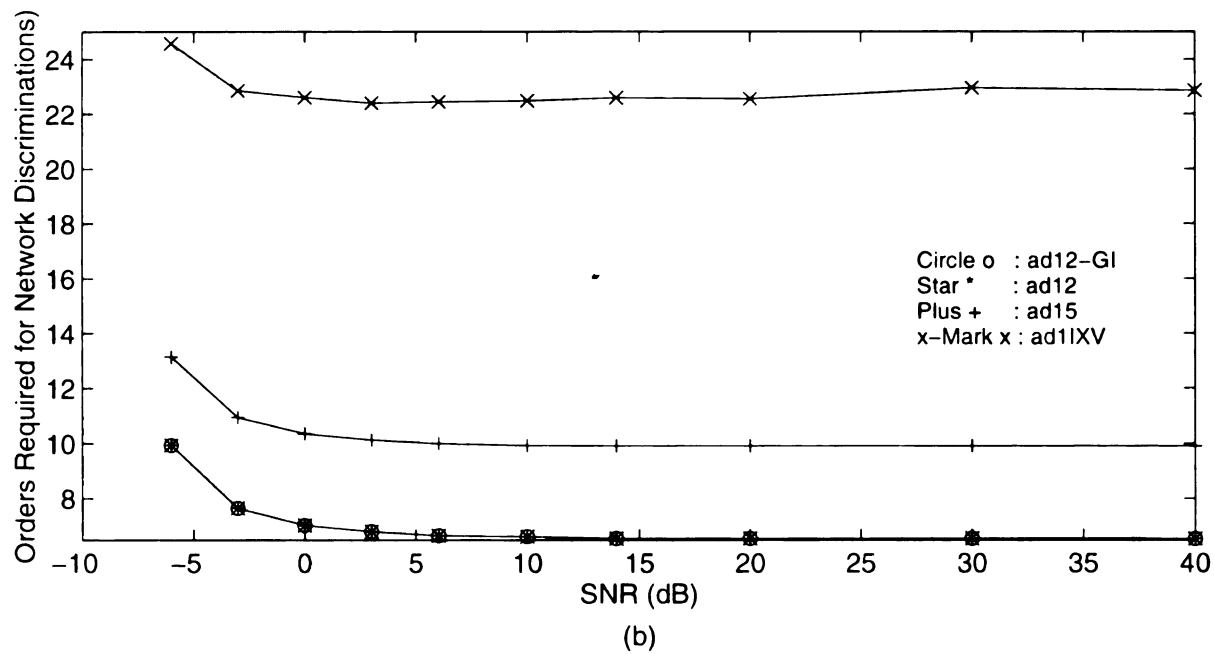
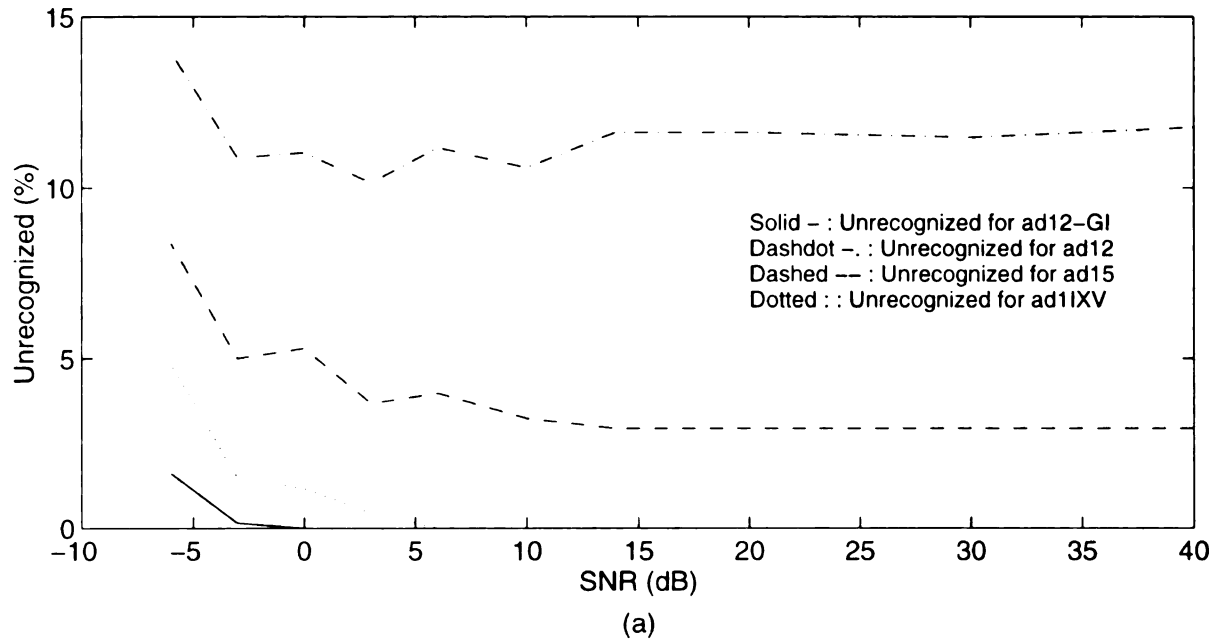


Figure 5.82 Unrecognized discriminations and discrimination orders of the RCAAM/ad's with an initial order of 1 and RCAAM/di12-GI cascade network vs. SNR for the 68 stored patterns belonging to 4 targets.

(a) Unrecognized discriminations vs. SNR for the 68 stored patterns.

(b) Orders required for discriminations vs. SNR for the 68 stored patterns.

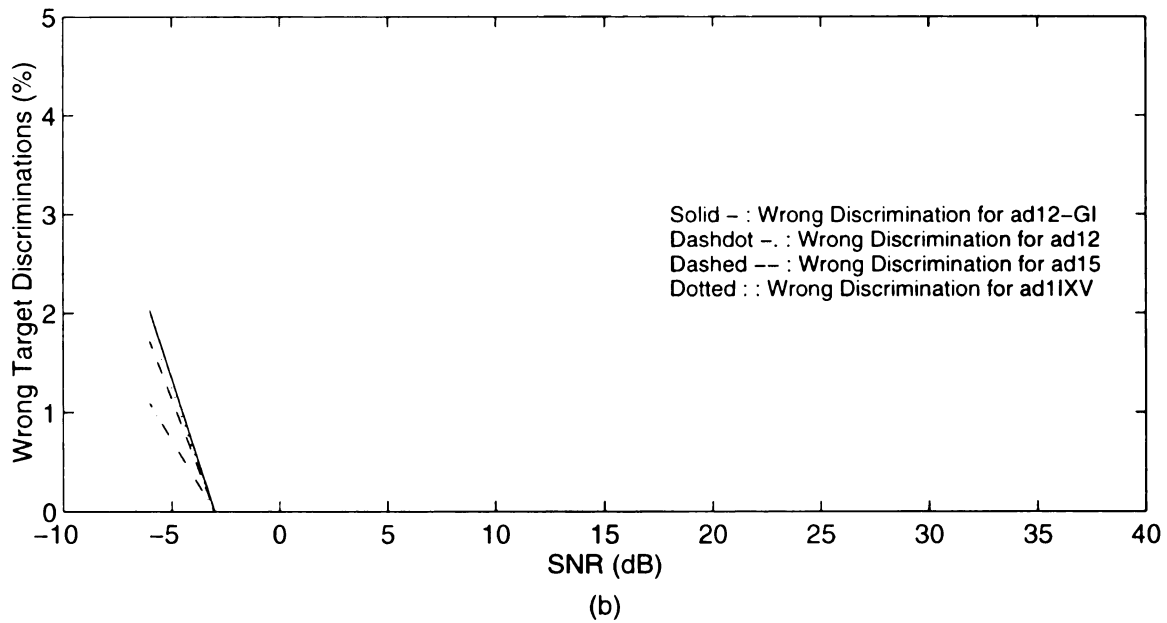
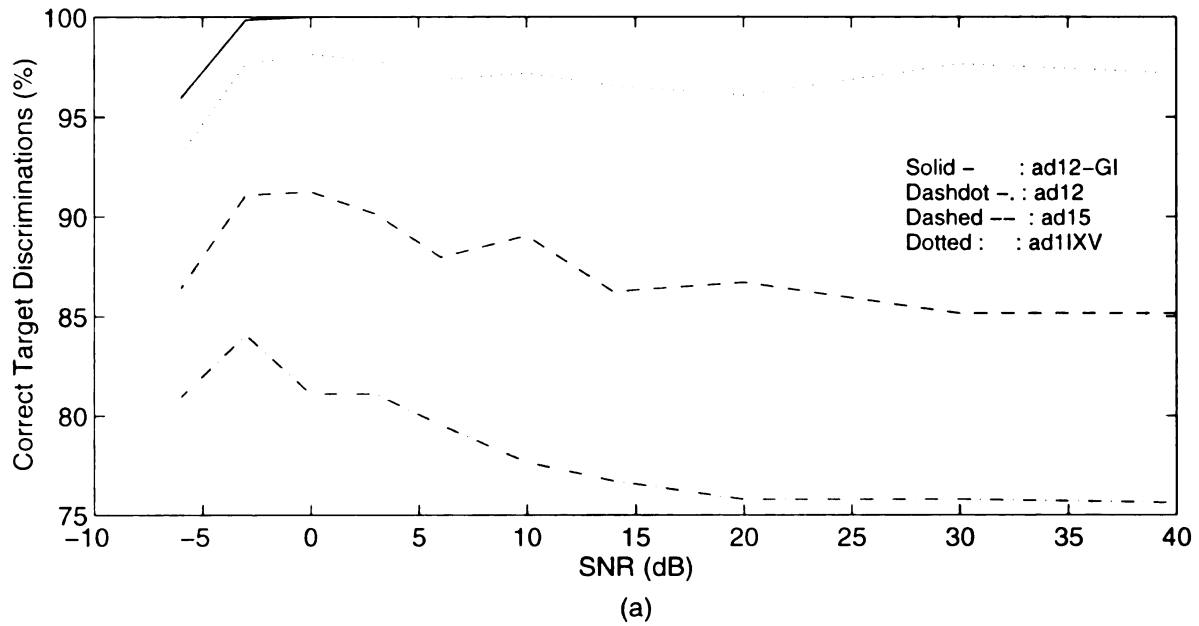


Figure 5.83 Performance comparisons of the RCAAM/ad's with an initial order of 1 and RCAAM/ad12-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

(a) Correct target discriminations vs. SNR for the 64 unstored patterns.

(b) Wrong target discriminations vs. SNR for the 64 unstored patterns.

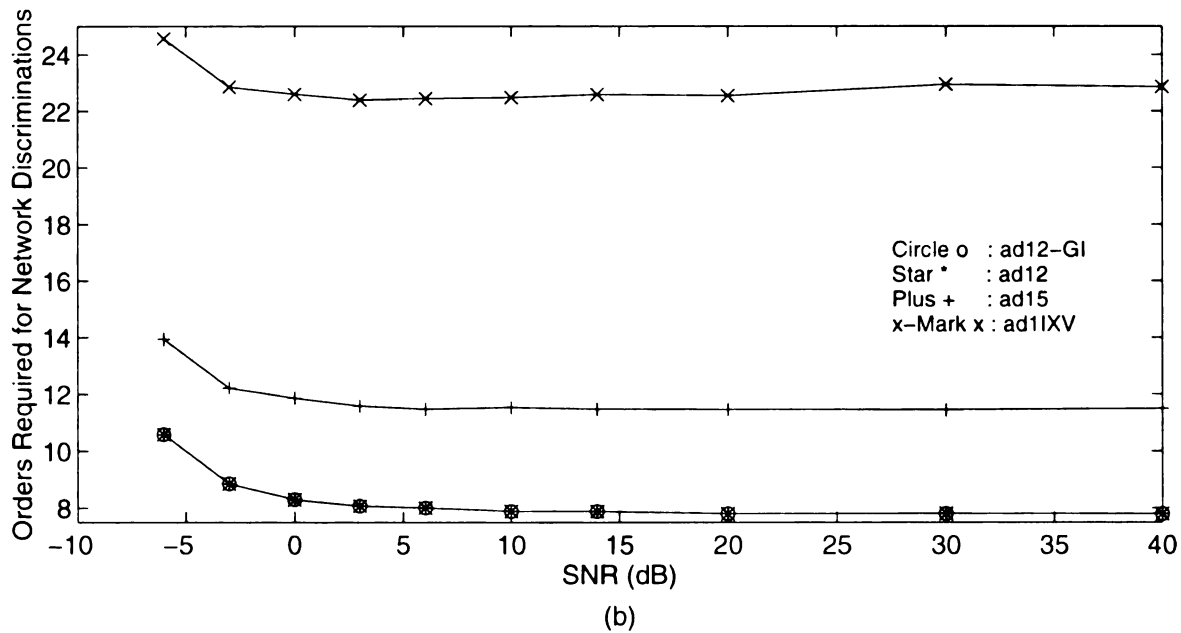
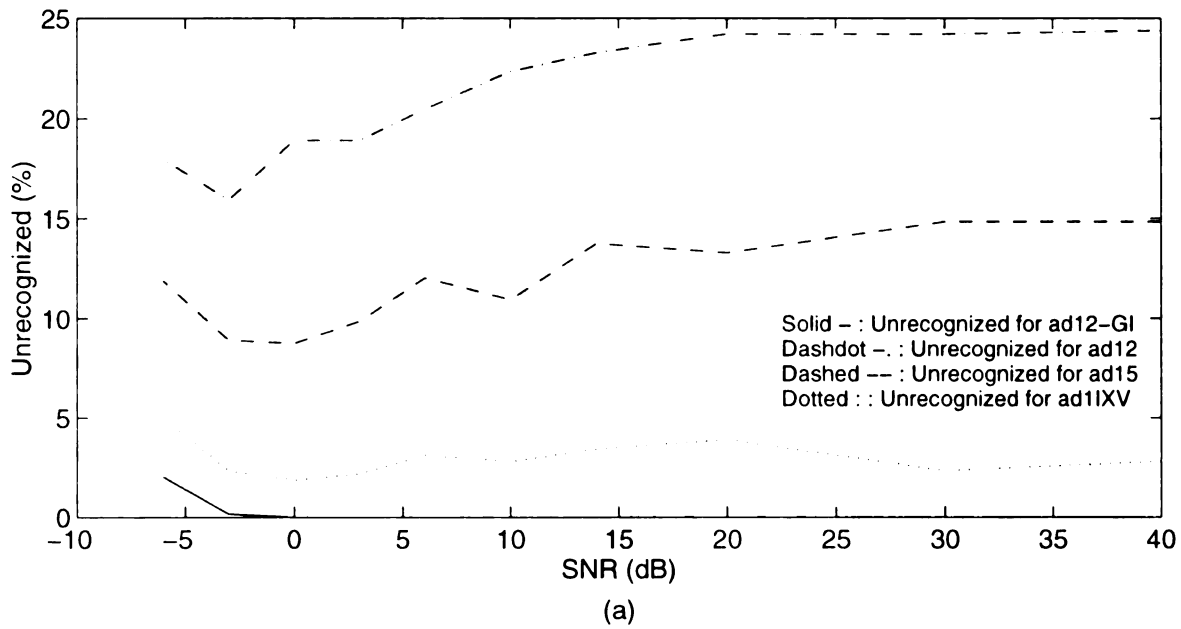


Figure 5.84 Unrecognized discriminations and discrimination orders of the RCAAM/ad's with an initial order of 1 and RCAAM/ad12-GI cascade network vs. SNR for the 64 unstored patterns belonging to 4 targets.

(a) Unrecognized discriminations vs. SNR for the 64 unstored patterns.

(b) Orders required for discriminations vs. SNR for the 64 unstored patterns.

CHAPTER 6

Target Discrimination using Neural Network with Spectrum Magnitude Response

6.1 Introduction

In the previous three chapters, we used the sampled backscatter time response as the network process information. We determined the beginning response time for each stored and unstored aspect responses by a simple detection algorithm, and then picked the first 100 time samples, starting from the detected beginning time, as time domain analog stored/unstored patterns. Then noises were added to the deterministic analog stored/unstored patterns to simulate the network tolerance performances. So we had assumed that the detection of the beginning response for target aspect responses is available and consistent even under various noises. In practical noise-limited situations, finding the same beginning response time used in training is very difficult. Therefore, the network must also store or train several time-shifted neighborhoods of the time segment pattern for each aspect angle, to increase tolerance for time-shifted patterns. This is impractical, since it dramatically reduces the network capacity.

Let's show a time-shifted case resulted from an inappropriate determination for the beginning response occurred in our previous time domain simulations. In our simulations, there are four targets and each target has 17 aspect patterns stored in networks. **Figure 6.1** shows the time response stored segments of the fourth and sixth aspect stored patterns of target F14. The aspect angle spacing between any two adjacent stored (or unstored) patterns of a target is 1.8° , therefore the aspect angle spacing between the fourth and sixth aspect

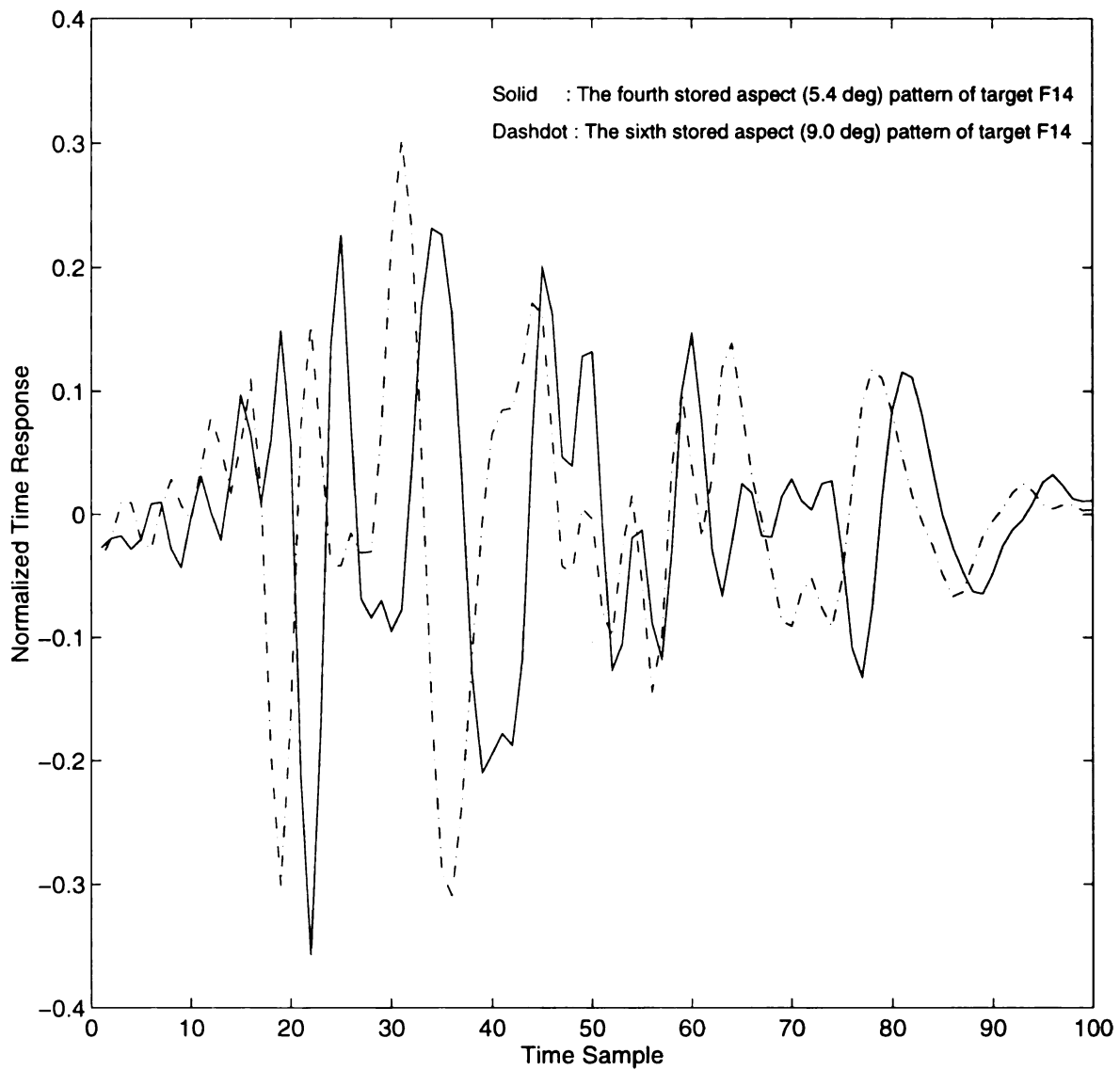


Figure 6.1 Two aspect (5.4° and 9°) stored patterns with an apparent time-shift resulted from an inappropriate beginning response determination occurred in time domain simulations.

stored pattern of target F14 is 3.6° . There is an apparent time-shift between these two aspect stored segments, since our detection to the beginning response failed. The analog correlation gain between these two stored aspect patterns is -0.2914, and the bipolar correlation gain is 14 after encoded by 3 bits coding 7 levels. Since the stored aspect responses are normalized, the analog autocorrelation gain for any stored pattern should be 1. The bipolar autocorrelation gain should have the value of 300, thus the correlation gain of 14 indicates the two stored patterns of target F14 with aspect angles spacing with 3.6° have 143 bits different among 300 bits. Therefore, no matter in analog or digital data form, the correlation-based inconsistency resulted from the confused allocation of the beginning responses is worse for these two aspect stored patterns. This correlation-based inconsistency will degrade the target group idea and then make network's discrimination only depend on individually distinct aspect patterns but consistent target clustering in our simulations.

In this chapter, we use the spectral magnitude, which is time-shift invariant, as the network process information to overcome the difficulty of consistently allocating the beginning time responses. Unfortunately, we use less information here than the time domain process since the phase is ignored. Also, the sharp specular peaks characteristic of a typical backscatter time response don't occur in the corresponding frequency spectrum.

6.2 Spectrum Magnitude Process and Normalization

To simulate target impulse response in the time domain, we measure the frequency responses of 4 targets in the frequency band 1-7 GHz with a frequency increment of 0.01 GHz. Thus, we have 601 measured spectral responses including both magnitudes and phases. Since an impulse in time domain has uniform spectral distribution in the frequency domain,

the transmitted wave generated by a HP-8720B network analyzer sweeps from 1 GHz to 7 GHz to emulate the uniform spectral distribution. To have a narrow sampling time, (8192-601) zeros are attached to the measured spectral responses to have 8192 total spectral responses. Then a 8192-point inverse FFT is used to create the corresponding scattering time responses. We then assume these time responses as measured by a time-domain radar system. To eliminate any spurious reflections within the measurement chamber and save space, we time-gate the responses and only adopt 820 responses for each aspect angle as measured effective responses by a time-domain radar system.

In time domain simulations, we picked the first 100 samples from 820 points as the aspect pattern, starting from the detected target beginning response time. But here we use the 820 time responses as time domain aspect pattern, and then use the FFT to transform the time responses back into the frequency domain to have frequency domain aspect pattern. Therefore, the simulation inputs are always 820 time responses in our spectral magnitude simulations. Since we measured the spectral responses in the frequency band 1-7 GHz, the FFT transform should have effective spectral responses in frequency band 1-7GHz. To be consistent with the time domain simulations, we'd like to use 100 spectral samples as frequency domain aspect pattern. We also want to fully utilize the spectral responses within that bandwidth, so the transferred spectral responses should have 100 effective frequency samples in frequency band 1-7GHz and zeros outside the band. After calculation, a 1358 point FFT is used to have the band covered by 100 frequency samples, 18 to 117. Then we pick the 100 frequency samples residing from 18 to 117 and only use the spectral magnitude portion as network processing patterns. To systematically process data, each aspect spectrum

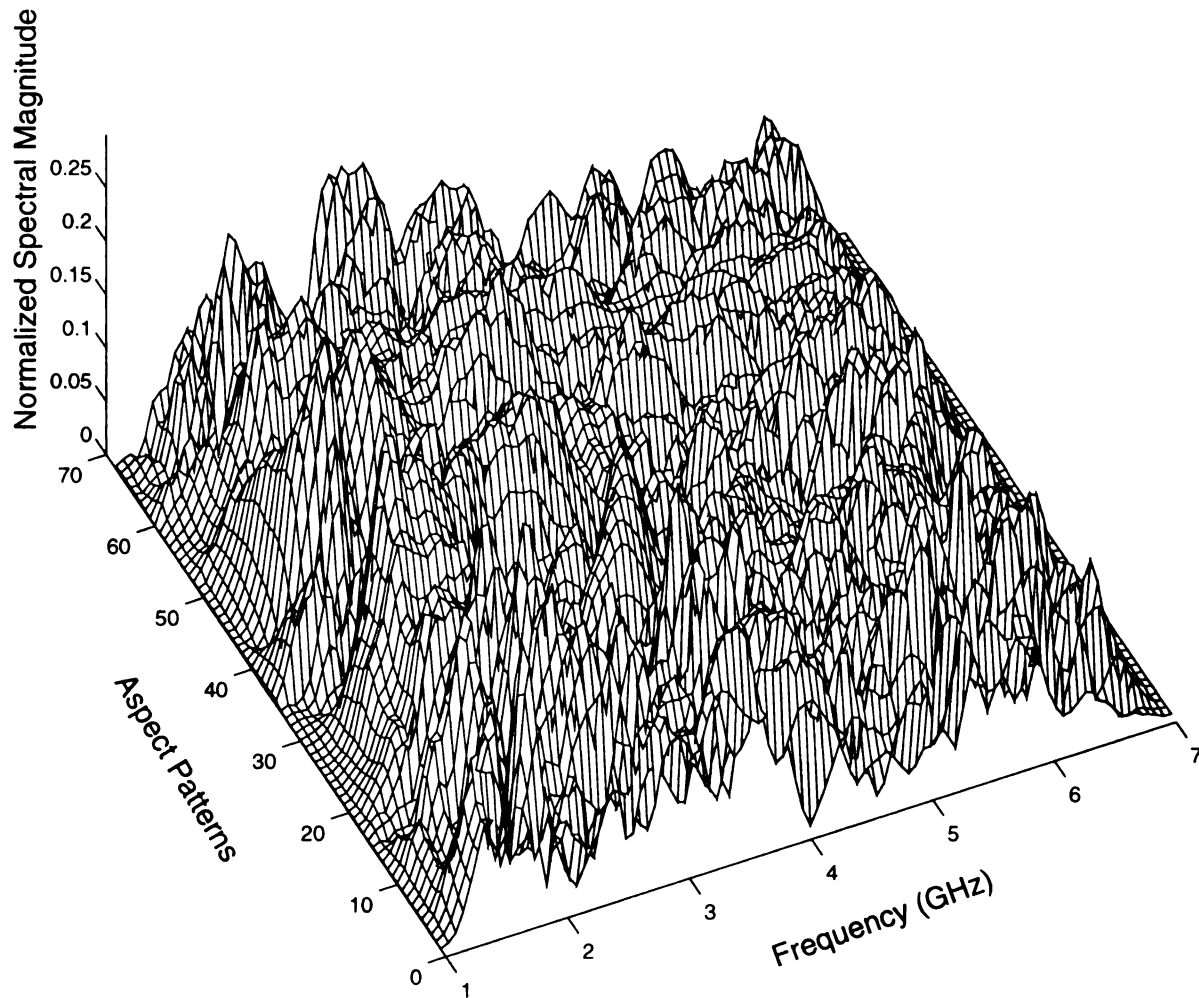


Figure 6.2 68 aspect FFT spectrum magnitude patterns, with normalized energy of 1, used for spectrum process network trainings/storage. Spectrum process networks simulate 4 targets, each target has 17 trained/stored aspect spectral patterns. The effective frequency band is 1-7 GHz.

magnitude pattern is normalized to energy 1. **Figure 6.2** shows the 68 training/stored spectral magnitude patterns obtained from the 68 aspect time responses where sample point 1 corresponds to 1 GHz and point 100 to 7 GHz. Since only one half of the signal information is utilized, the spectrum magnitude patterns are not as distinct from each other as the time domain patterns.

In digital data process simulations, not only 7 quantization levels are again used but also 5 levels are tried. since the dynamic oscillation range for spectrum magnitude patterns is much smaller than time responses, 5 quantization intervals may be enough to represent the small dynamic oscillation range. Then we use 3 bits to binomially encode the 5 and 7 numerical intervals, so each binomial pattern will have 300 bits for both 5 and 7 quantization levels. The code assignment for 7 quantization levels encoded by 3 bipolar bits is same as the time domain one presented in chapter 3. The code assignment for 5 quantization levels encoded by 3 bipolar bits is shown below :

Bit1Bit2 Bit3	-1 -1	-1 1	1 1	1 -1
-1	2	3	4	
1	1		5	

Table 6.1 Code assignment of 5 quantization levels encoded by three bipolar bits.

The above code assignment has a great linear range, and the statistical linear range is ± 3.5 levels. Therefore, the linear range almost covers all 5 levels. The linearity only fails on two specific situations that the signal in level 1 will recognize itself closer to the level 5 than the level 4, and the signal in level 5 will recognize itself closer to the level 1 than the level 2. Thus the 5-level code has higher linearity than the 7-level code, and this is the major reason that we also use the 5-level code in spectrum magnitude simulations. The great linearity can prevent correlation-based networks from wrong similarity recognition under contaminations. At the same time the code with low quantization levels will result in low correlation-based discrimination resolution, and it will be analyzed later.

There are some problems with analog spectrum process networks, since the spectral magnitude carries less information than the time response signal. In the time responses, the location of specular peaks is a good measurement for discrimination. In contrast, the spectral magnitude doesn't oscillate around its mean value nor have a lot of sharp peaks. Apparently, the frequency magnitude changes much slower than the time domain signal, and its variance is much less than its corresponding time response. Therefore, the spectral magnitude distribution is more uniform and the discrimination among these spectrums becomes harder. Since the spectrum magnitude is positive, the correlation between any two stored spectrum patterns is always a positive value. Therefore, before processing analog spectrum magnitude, we normalize spectrum magnitude patterns to satisfy the following two conditions :

- (1). Every stored pattern must be normalized to 0 mean value.
- (2). The energy of each pattern must be normalized to some uniform value.

The above condition (1) makes recurrent correlation process with a nonlinear threshold

function more efficient, while condition (2) enhances the linearity of the correlation operation.

Suppose $\{A(i) | i=1, \dots, n\}$ is a stored spectrum magnitude pattern with energy 1. Then

$$\sum_{i=1}^n A(i)^2 = 1 \quad (1)$$

Letting $\underline{a}$ denote the mean of A , i.e.,

$$\underline{a} = \frac{1}{n} \sum_{i=1}^n A(i) \quad (2)$$

and $E\{A(i) - \underline{a}\} = 0$, then we have two processes by which to produce the normalized

pattern $\underline{A}$

(1). Prior Process :

For condition (2), we have

$$\sum_{i=1}^n [C \cdot (A(i) - \underline{a})]^2 = 1 \quad (3)$$

so that

$$C = \sqrt{\frac{1}{1 - n\underline{a}^2}} \quad (4)$$

and $\underline{A}(i) = C \cdot (A(i) - \underline{a})$.

If $A_M(i) = A(i) - \underline{a}$ has been calculated, then we use the following process.

(2). Posterior Process :

Condition (2) gives

$$\underline{A}(i) = \frac{A_M(i)}{\sqrt{\sum_{i=1}^n A_M(i)^2}} = D \cdot A_M(i) \quad (5)$$

where

$$D = \frac{1}{\sqrt{\sum_{i=1}^n A_M(i)^2}} \quad (6)$$

Due to

$$E\left\{\frac{A_M(i)}{\sqrt{\sum_{i=1}^n A_M(i)^2}}\right\} = D \cdot E\{A_M(i)\} = 0 \quad (7)$$

, the condition (2) is also satisfied.

Since we have assumed $\mathbf{A}$ has energy 1, then

$$\sum_{i=1}^n A_M(i)^2 = 1 - n\alpha^2 \quad (8)$$

and thus $\mathbf{C}=\mathbf{D}$.

Thus, we now have the normalized spectrum magnitude pattern $\underline{\mathbf{A}}$ satisfying both conditions (1) and (2) :

$$E\{\underline{\mathbf{A}}(i)\} = 0 \quad (9)$$

$$\sum_{i=1}^n \underline{\mathbf{A}}(i)^2 = 1 \quad (10)$$

Although we have normalized the spectral magnitude to overcome the deficiency caused by disregarding phase, the analog correlation processing algorithm still requires further improvement. As analyzed previously, the frequency spectra are more similar to each other than the time responses, and thus the crosscorrelations between any two stored spectral patterns almost always have positive values, even after the above normalization. Therefore,

we need an offset to further compensate this deficiency.

We statistically evaluate the crosscorrelations for all the stored spectral patterns to find out the average and the minimum, and then design the **offset**. Since the spectral patterns have been normalized, the maximum correlation gain, (i.e. autocorrelation,) among all stored patterns is 1, provided the input is equal to some stored pattern. Let $C_{\min}$ denote the minimum of crosscorrelations among all stored patterns. If a recurrent correlation associative network is to converge to the expected stored pattern, it needs to satisfy at least

$$1 - \text{Offset} > -(C_{\min} - \text{Offset}) \quad (11)$$

and thus

$$\text{Offset} < \frac{1 + C_{\min}}{2} \quad (12)$$

Typically, $C_{\min}$ has a negative value. If $-(C_{\min} - \text{Offset}) > 1 - \text{Offset}$ exists, there exists an input and one stored pattern that have negative normalized correlation gain and the gain scale is larger than the normalized input autocorrelation. Therefore, the stored pattern with minimum correlation gain will overcome the others including the input's true stored pattern when the recurrent iterations are even. If we use the value of **Offset** just satisfying the above inequality margin, then the negative gain dominator may still occur when inputs are contaminated. And the critical satisfaction also may make the network not converge or converge very slowly. We evaluate the mean of all crosscorrelations, then statistically and experimentally find that 2/3 of the mean is a good choice for the **Offset**. Therefore, the network processing algorithm for RCAAM/ad requires some modification to process the analog frequency spectrum magnitude.

Suppose $\{(\xi^i, \zeta^i) | i = 1, \dots, P\}$ are the associative pattern pairs stored in RCAAM/ad,

where ξ^i is a bipolar column vector of length m and ζ^i is a normalized analog spectral magnitude vector of length n , given by $\xi = [\xi^1 \ \xi^2 \ \dots \ \xi^p]$ and $\zeta = [\zeta^1 \ \zeta^2 \ \dots \ \zeta^p]$, and let the offset vector $\mathbf{F}_{\text{offset}}$ be a column vector of length $\mathbf{P}$ with each component value equal to the **Offset**. If $\mathbf{U}$ is an n -dimensional normalized analog spectral magnitude input, then we can construct the initial Analog-Digital RCAAM as follows :

$$M_o = \xi \cdot \{ \text{Diag} [(\zeta^T \cdot U - F_{\text{offset}}) \cdot \wedge(\text{init})] \cdot \zeta^T \} \quad (13)$$

$$OS_o = \xi \cdot \{ \text{Diag} [(\zeta^T U - F_{\text{offset}}) \cdot \wedge(\text{init})] \cdot F_{\text{offset}} \} \quad (14)$$

where $A \cdot \wedge q \equiv [A_1^q A_2^q \dots A_p^q]^T$ and

$$\text{Diag} (A) \equiv \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & 0 & \dots \\ \dots & & & \\ 0 & \dots & 0 & A_p \end{bmatrix}$$

if $A = [A_1 \ A_2 \ \dots \ A_p]^T$, and OS_o is an m by 1 column vector.

Then the current dynamic memory output is

$$V_o = \text{Sign} (M_o \cdot U - OS_o) \quad (15)$$

The initial order **init** is usually set to 0. Since the input has analog form (the same as ζ^i), and the output has binomial form (the same as ξ^i), we have fixed input ($U_k = U$) for this analog-digital hybrid memory. Then the dynamic memories $\mathbf{M}_k$, the dynamic **Offset** vector $\mathbf{OS}_k$ and the evolution outputs $\mathbf{V}_k$ at the recurrent iteration time k have

$$M_k = \xi \cdot \{ \text{Diag} [(\zeta^T U - \text{Offset}) \cdot \wedge(\text{init} + k)] \cdot \zeta^T \} \quad (16)$$

$$OS_k = \xi \cdot \{ \text{Diag} [(\zeta^T U - \text{Offset}) \cdot F_{\text{offset}}] \cdot F_{\text{offset}} \} \quad (17)$$

$$V_k = \text{Sign} (M_k \cdot U - OS_k) \quad (18)$$

We previously showed the inconsistent correlation gain resulted from inconsistent beginning response detections. Now let's check if the spectrum magnitude process can alleviate the problem. **Figure 6.3** shows the FFT spectrum magnitude patterns corresponding to the previous time-shifted responses, while **Figure 6.4** presents their normalized spectrum magnitude patterns. We can see that the whole waveform shifting effect has been eliminated for both types of spectrum magnitude patterns. The correlation gain for the two spectrum magnitude patterns is 0.897 and is 0.639 for the two normalized spectrum magnitude patterns. Compared to the negative correlation gain occurred in time domain, the spectrum magnitude waveform for these two aspect patterns are more consistent. After offsetting, the spectrum magnitude correlation gain is 0.349, and then still positive. In time domain, the 7-level bipolar coding gave the two aspect patterns a correlation gain of 14, thus two bipolar patterns had 157 bits in common and 143 bits different. In spectrum magnitude process, the 5-level coding has a correlation gain of 132, therefore two bipolar patterns have 216 bits in common and 84 bits different. The 7-level coding has a correlation gain of 62, therefore two bipolar patterns have 182 bits in common and 118 bits different. Therefore, the digitized data form still demonstrates the property that the spectrum magnitude process has better correlation-based consistence for the target responses with close aspect angles than time domain process.

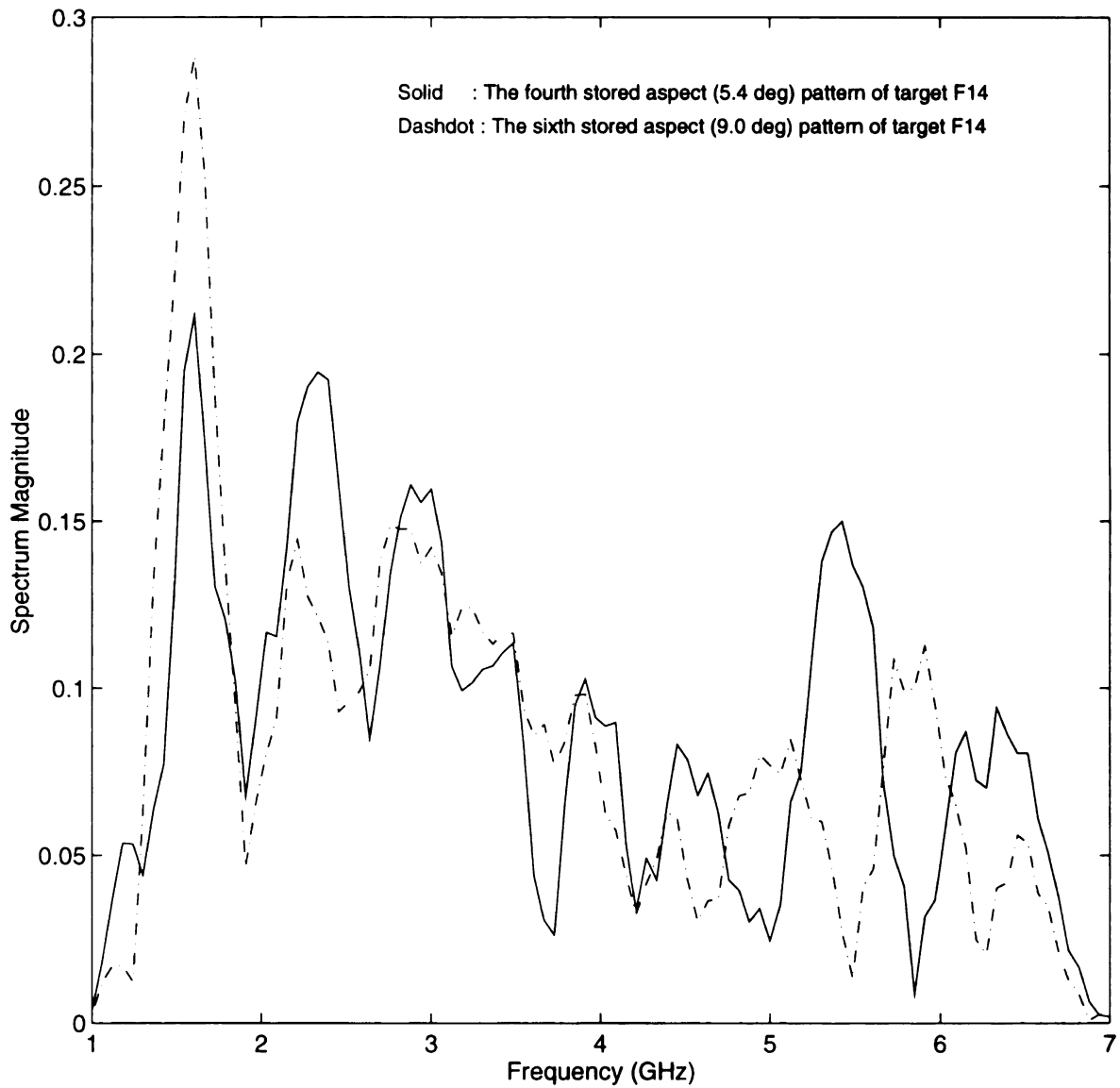


Figure 6.3 FFT spectrum magnitude patterns of two time responses from target F14 with respective aspect angle of 5.4° and 9° .

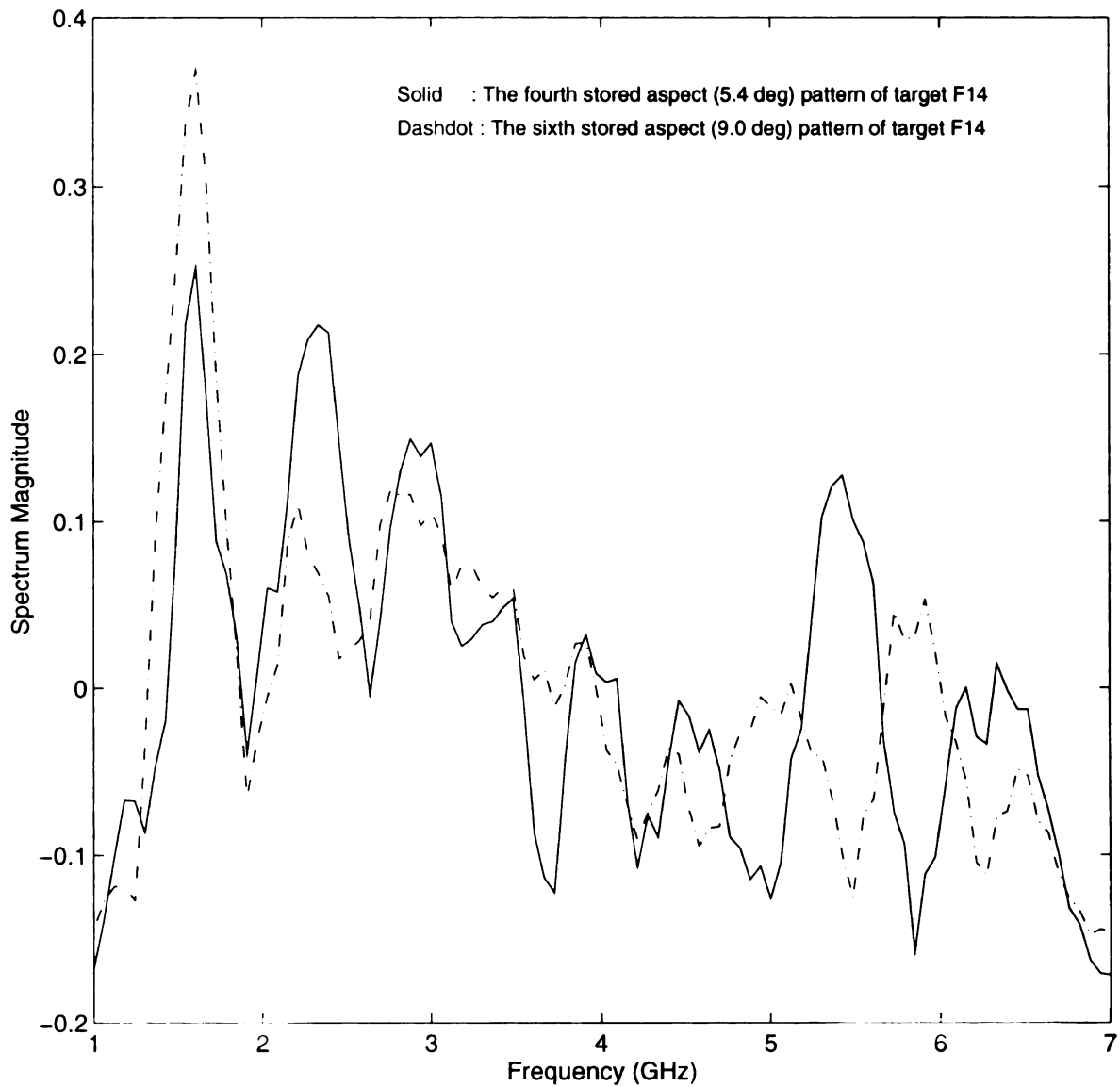


Figure 6.4 Normalized FFT spectrum magnitude patterns of two time responses from target F14 with respective aspect angle of 5.4° and 9° .

6.3 Manipulation of Spectrum Magnitude Data Before Network Process

In spectrum magnitude simulations, we use the same 4 targets, 68 stored/trained and 64 unstored/untrained aspect angles as we did in the time domain case. In time domain noise tolerance simulations, the desired noise based on the energy of the simulated pattern is directly added to the pattern, i.e. 100 samples or responses. Since we have assumed that the time responses measured from a time domain radar is only available in practical situation, we used IFFT to have time responses from lab-measured spectral responses and then used FFT to obtain the spectral responses as the network process information. To be consistent with this assumption, the noise in spectral simulations should be added to time domain signal and then use FFT to obtain the contaminated spectral responses.

Remember that there are 8192 time responses after IFFT from the measured spectral responses. Then we time-gated the 820 effective responses as time domain aspect responses measured by a time domain radar. Also recall that the time domain network only pick the first 100 samples among 820 responses as network process pattern, starting from the detected beginning response time. To emulate the practical situation, the simulated noise is added to the 820 time responses in spectral simulations. Given a time domain input with 820 responses, we calculate its energy, then add the desired simulated noise to the 820 responses. So, in our simulations, we calculate the energy of 820 samples for each pattern, then generate the corresponding noise with a SNR required for simulation. After noise is added to the 820 samples, extra 538 zeros from (1358-820) are attached to the 820 values to form a 1358-element signal. Then a 1358-point Discrete Fourier Transform (DFT) is used to obtain 1358 spectrum magnitudes. As described previously, the 1358-sample is designed to have the

available measurement band 1-7 GHz covered by 100 frequency samples, and no response outside the band. Finally we pick the 100 spectrum magnitude samples, sample 18 to sample 117, as spectral network process pattern. Then the further normalization is computed based on this spectral pattern before presenting the pattern to the network.

We have explained that the Signal To Noise (SNR) definitions for time domain and spectrum domain processes are different. We now show the noise contamination effects in time domain responses and then spectral responses after DFT. **Figure 6.5** shows the 820 time responses used as measured responses by a time domain radar. The solid line stands for the true responses, while the dotted line denotes the contaminated signal with a SNR of 0 dB. Then **Figure 6.6** presents the 1358 spectrum magnitudes obtained from the above 820 time responses plus 538 zeros by using 1358-point Fourier transform. It's clear to see that the true spectrum magnitude, solid line, only has values in two sample intervals, samples 18 to 117 and samples 1242 to 1341, corresponding to the frequency band 1-7 GHz. After noise is added to time responses, the transformed spectral responses have magnitudes spreading over the whole band. **Figure 6.7** shows the 100 time responses used as the time domain network process aspect pattern and its contaminated signal, dash-dot line, with a SNR of 0 dB. In time domain simulations, the desired noise calculation is based on the 100 responses, i.e. aspect pattern itself. Finally, **Figure 6.8** presents the 100 spectrum magnitudes used as spectrum network process pattern. They are truncated from the previous 1358 spectrum magnitudes, so only the noise spectrum occurred in the measure band, 1-7 GHz, will contaminate the true spectral pattern. Again the calculation of 0 dB is based on the 820 time responses. It is noted that the noise amplitudes become large, compared to the spectrum magnitude, if the spectrum

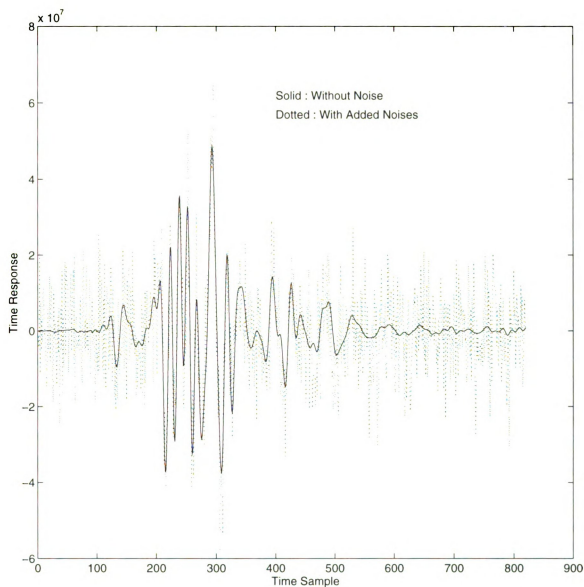


Figure 6.5 820 time responses of target B52 at aspect angle 0° used as measured responses by a time domain radar. The solid line shows the true responses, while the dotted line denotes the contaminated signal with a SNR of 0 dB.

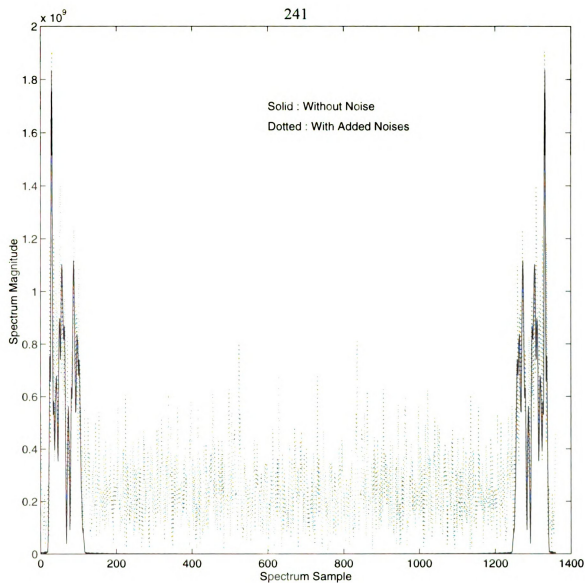


Figure 6.6 1358 spectrum magnitudes transformed from the previous two time responses by using a 1358-point DFT. The solid line shows the true responses, while the dotted line denotes the contaminated signal with a SNR of 0 dB. The evaluation of 0 dB is based on the 820 time responses.

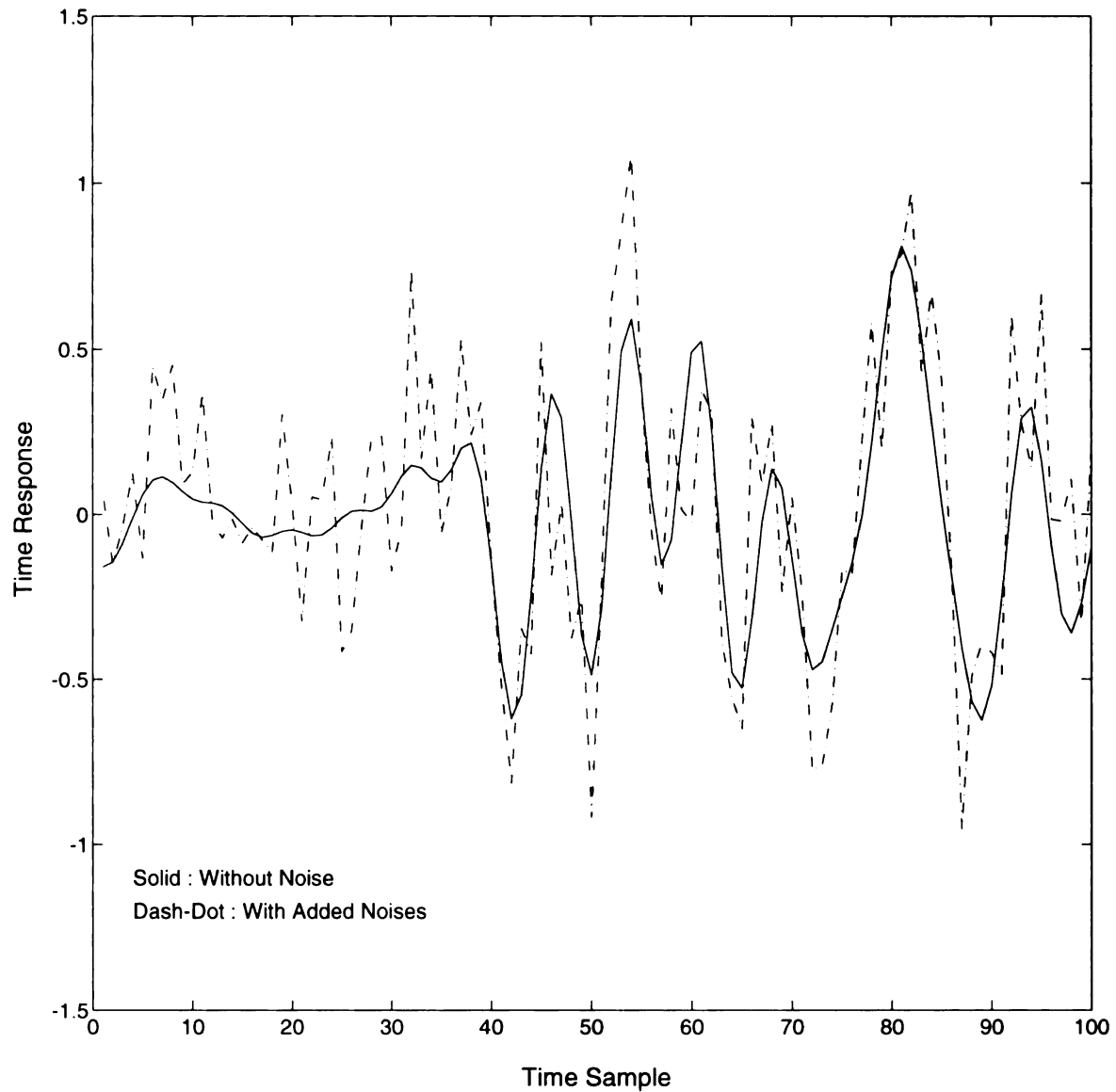


Figure 6.7 100 time responses truncated from the previous 820 responses used as the time domain network process aspect pattern. The solid line shows the true responses, while the dash-dot line denotes the contaminated signal with a SNR of 0 dB. Here the evaluation of 0 dB is based on the 100 time responses.

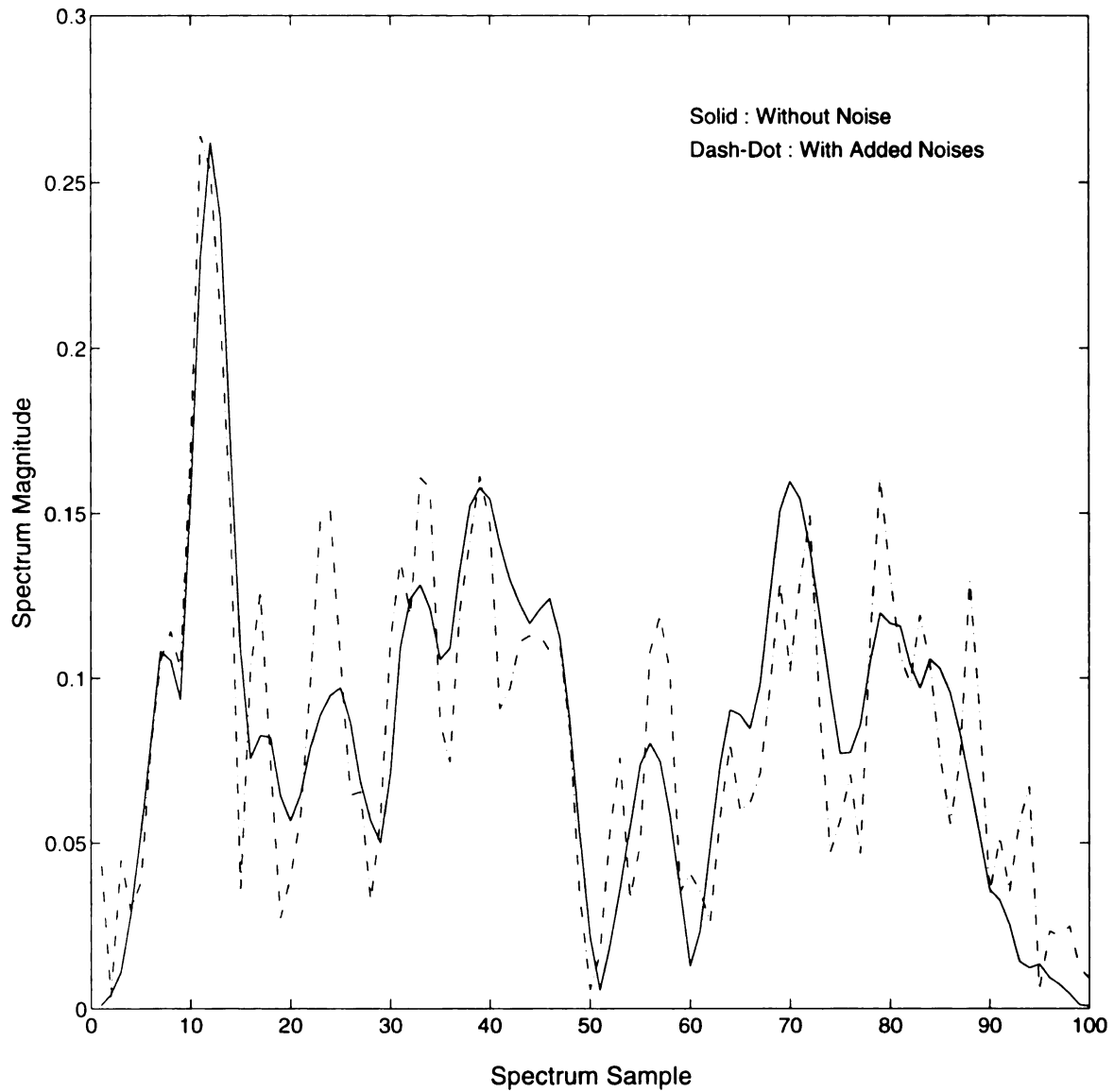


Figure 6.8 100 spectrum magnitudes truncated from the previous 1358 spectral responses used as the spectrum network process pattern. The solid line shows the true responses, while the dash-dot line denotes the contaminated signal with a SNR of 0 dB. The evaluation of 0 dB is based on the 820 time responses.

magnitude pattern is normalized, i.e. magnitudes will oscillate around the mean value.

6.4 Comparisons of Correlation-based Discrimination Resolutions for Different Data Formats

We have used different information forms for network processes, 7 levels encoded by 3 bits in time domain bipolar networks, analog valued responses in time domain RCAAM/ad, 5 and 7 levels encoded by 3 bits in spectral process bipolar networks, and analog valued spectrum magnitude in spectrum magnitude RCAAM/ad. Given different information forms, a RCAAM network will have different convergent iterations and performances. With the same network structure and stored aspect patterns, the convergent iterations indicate how easy a given information can be discriminated, therefore, we realize that the information form can affect the correlation-based discrimination resolution and then performances. Then we may ask which information form has high correlation-based discrimination resolution ? This section will answer (or analyze) this question.

Since the discrimination medium for most networks used in this thesis are based on correlation gains, we will focus on the relationship between the information form and its correlation-based discrimination resolution. What kind of correlation distribution indicates better discrimination resolution ? To systematically analyze both analog and digital data forms, let's normalize the maximum correlation gain to 1, then the normalized correlation gain of 1 will indicate a gain of 300 to our 300-bit bipolar patterns and 1 to the normalized analog patterns. Since the network output prior threshold function is sum of the stored patterns weighted by their individual correlation gains, the dominant patterns ought to have apparently larger correlation gains than the others. Our output stages always use the bipolar form, i.e.

a nonlinear threshold function. Then it will be difficult to find any dominant pattern after the threshold function, if great populations have large correlation gains. Thus the better correlation gain distribution, to a stored pattern input, should have few gains (or patterns) in high values and most gains in small values, then the stored pattern, the input, won't be interfered by those small gain patterns and can dominate the network convergence. What kind of transition curve, expressing gain population change, is better? If the population transition from small gain to high gain is smooth, then it's unclear to have the dominant patterns. Therefore, a better transition should be steep.

Another concern is that high consistencies among adjacent aspect patterns will help to enhance the target group clustering, and then the group consistent dominant force will become greater than a distinct pattern and easier to relieve the interferences from other patterns or groups. Therefore, the high gain population transition boundary is better to have small populations which are aspect consistent. Consistent patterns are 3 for one aspect spacing consistencies, plus and minus aspect spacing, and they are 5 for two aspect spacing consistencies, i.e. plus one, plus two, minus one and minus two aspect spacing. In this study, the aspect spacing for stored aspect patterns is 1.8° . Given a stored pattern, we may say the adjacent aspect patterns are aspect consistent, if the smallest correlation gain between the pattern and its adjacent aspect patterns is greater than an aspect consistency criterion. Therefore, if an aspect consistency criterion is assumed to 40% of the maximum gain, then the aspect consistent bipolar patterns should have at least $(50+40/2)\%=70\%$ or $300*70\%=210$ bits in common, or at most 30% or 90 bits different. Thus if a bipolar gain population curve has a steep transition and its high gain with distinctly small population starts

around 0.4 and has patterns about 3 or 5, then we can expect this bipolar data form has high correlation-based discrimination resolution. Although the number of the same response samples between two aspect consistent patterns can not be given for analog patterns from an assumed aspect consistency criterion, the same gain population will affect the discrimination output in the same way.

Figure 6.9 shows the correlation gain population distribution for the 68 stored time domain patterns with bipolar and analog forms, while **Figure 6.10** presents the gain scale population distribution, ignoring the signs of correlations. Again the maximum correlation gains have been normalized to 1 for both analog and bipolar patterns. First we assume an active threshold gain, then a stored pattern will be active if it has correlation gain with the given stored pattern input greater than the active threshold. The 'active pattern' means the pattern has the capability to affect the output. The correlation gain population is statistically estimated for all stored patterns by calculating the number of stored patterns which have correlation gains greater than the assumed active threshold. Comparison between two figures shows that the analog patterns have a lot of negative correlation gains with scales less than 30% of the maximum gain. Therefore, high negative gain population below the gain scale of 0.3 indicates that these negative correlations won't bother dominant patterns for a high active threshold but may become a problem if the active threshold falls below 30% of the maximum gain like heavy contamination situations. According to the previous analysis, the time domain bipolar patterns have a great gain population distribution, since it has steep transition curve and small population, about 4, at the high gain transition corner with a gain of 0.4. From the population distribution, presented a stored pattern as an input, the time domain bipolar data

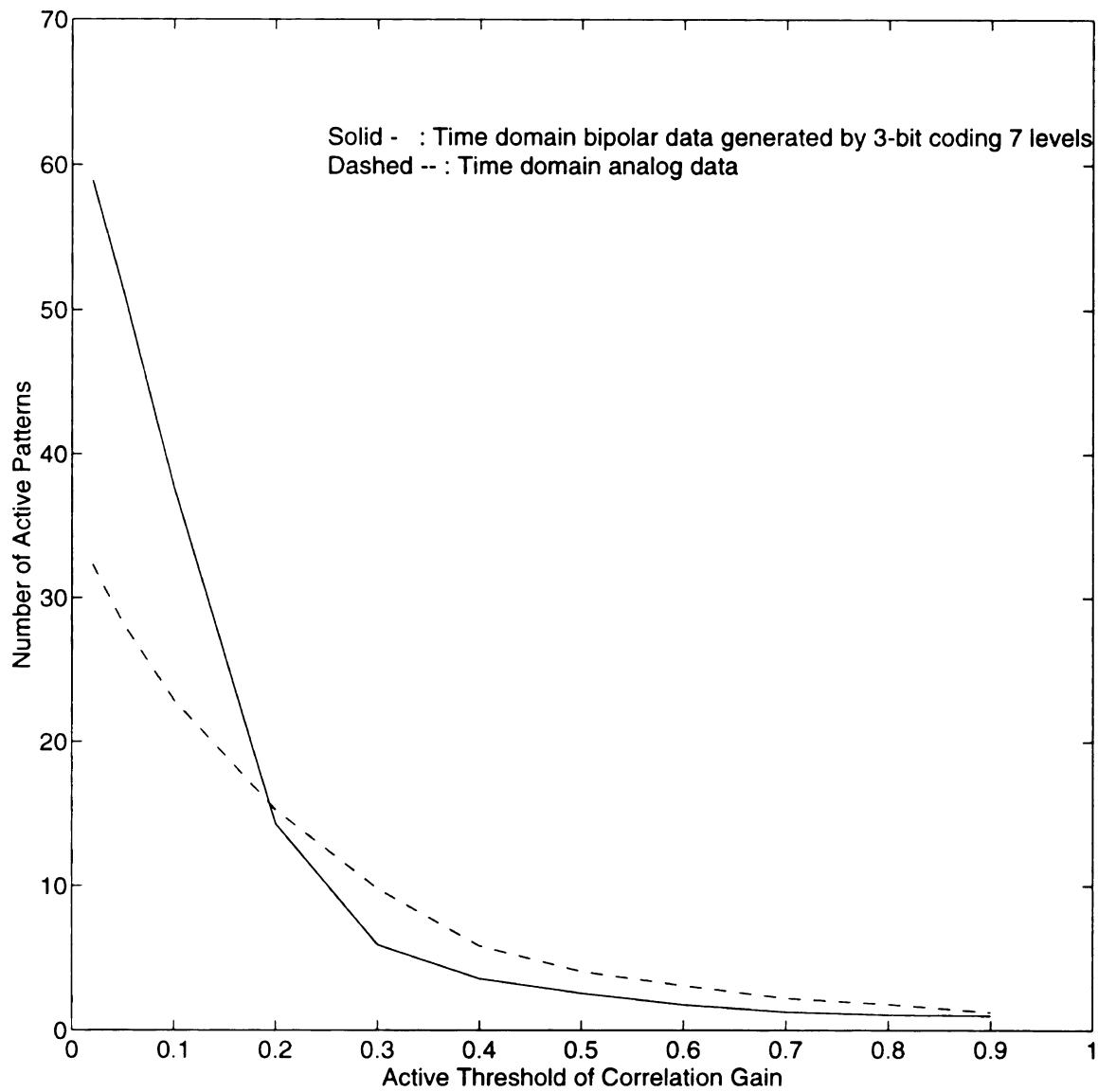


Figure 6.9 Correlation gain population distribution for 68 stored time domain patterns with different data forms.

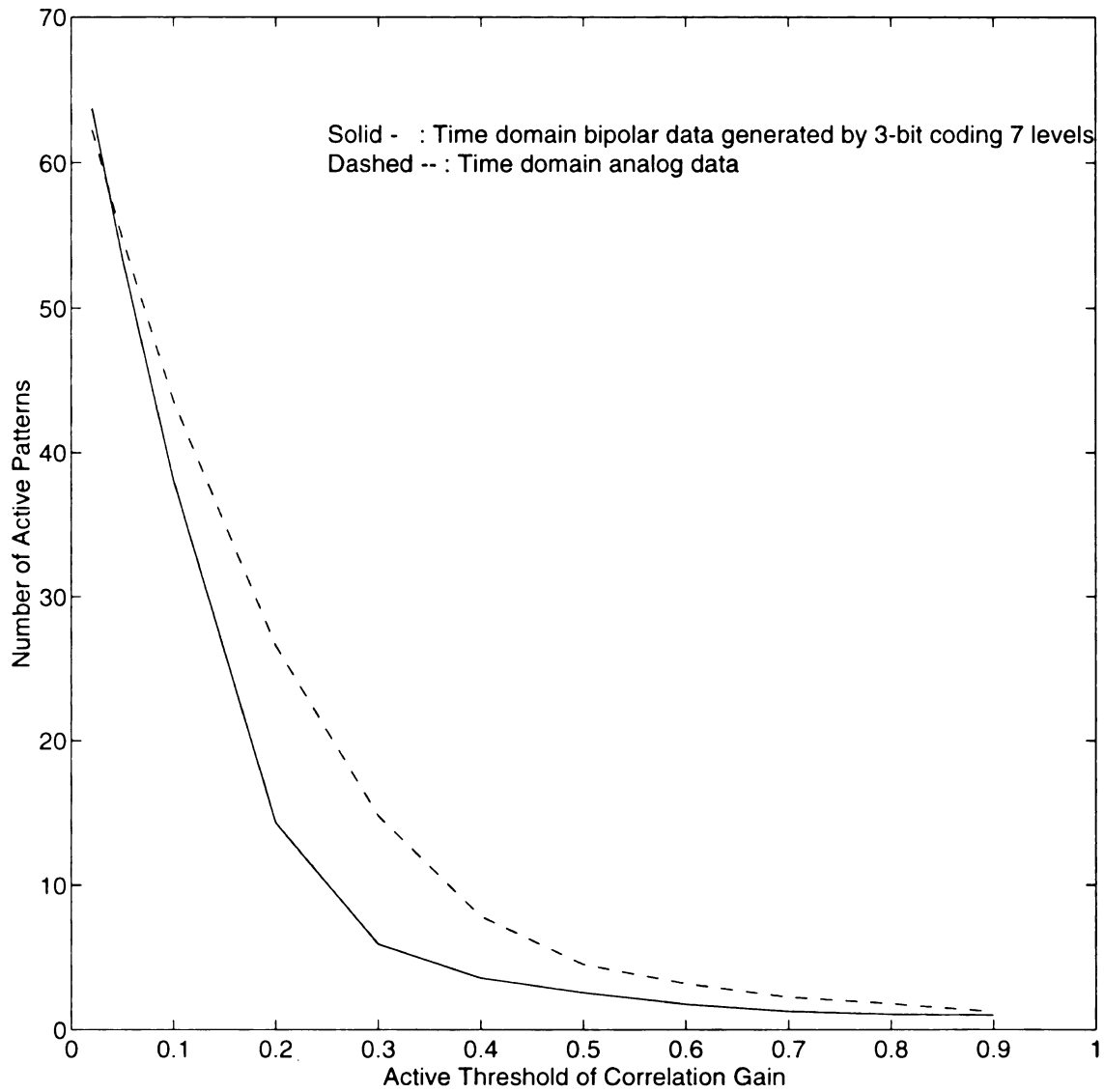


Figure 6.10 Correlation gain population distribution, only considering gain scale, for 68 stored time domain patterns with different data forms.

will only have 4 patterns with their correlation gains greater than 40% of the maximum gain, 300. Therefore, at most 4 stored patterns can dominate the network converging, if a dominant pattern should have its correlation gain greater than 40% of the maximum gain. The small population at the high gain side after the transition region may be regarded as an estimate for the number of aspect consistent patterns. Thus about 4 adjacent aspect patterns are in aspect consistency with respect to time domain bipolar data, and then the consistent aspect spacing is $(4-1) \times 1.8^\circ = 5.4^\circ$. It is apparent that in time domain the bipolar format has better correlation-based discrimination resolution than analog data, and this is consistent with the results of the previous chapter.

Figure 6.11 shows the correlation gain population distribution for 68 stored spectrum magnitude patterns with different data forms, while **Figure 6.12** presents the gain scale population distribution, ignoring the signs of correlations. Before normalization, the gain population distribution for analog spectral patterns is very poor, and there are 97% of the stored patterns with their gains greater than 70% of the maximum gain. Therefore, with most of the stored patterns at high gains all stored patterns will be confused together and bring no dominant winner, and then networks has no discrimination ability at all to these analog patterns. Now it is evident why we need to normalize the analog spectrum magnitude data before network processing. After the normalization, the analog correlation population distribution has been greatly improved so that the population with gains greater than 70% of the maximum gain has dropped to 7%, or about 5 patterns. Although the gain population transition curve is smooth, the new distribution has appeared the dominant patterns at gains greater than 0.7. The normalization greatly reduces the high gain population, then the aspect

consistency population greatly decreases. Therefore, we may say that the normalization nonlinearly increases the pattern space and the differences among adjacent aspect patterns, thus the aspect consistent patterns under the same consistency criterion decrease.

The offsetting to the normalized analog data makes the gain population distribution curve shift toward the left side, therefore the small population transition gain is shifted to 0.4 from 0.7. There are two deficiencies from this offsetting, the first is that the gain scale dynamic range is reduced from 1 to $(1-\text{offset})$ and the second is that the shifting or subtraction by the offset reflects a lot of negative gain population at low gain area. The first deficiency will reduce discrimination resolution, while the second one can cause negative gain interfering problem if the active gain drops to a low value like heavy contamination situation. Therefore, the correlation-based discrimination resolution for normalized and offset analog data is worse than the 7-level bipolar data and even worse than the 5-level bipolar data although the shifted population distribution curve is close to the one for the 7-level bipolar data.

To systematically compare with the other data forms, the normalized and offset correlation gains are again normalized to its own maximum gain, $(1-\text{offset})$. Then the normalized and offset analog data format still has great improvement compared to the normalized analog data, although the transition curve is still not as steep as the 5-level and 7-level bipolar data. The small population transition gain is shifted to a lower value and the negative gain population is again increased for correlation gain scale less than 0.4, compared to the normalized analog data. The smooth transition curve is a characteristic for analog data, since analog data format has full linearity and then the aspect consistency is high and smooth.

Therefore, this high aspect consistency, with more population at high gain, will make analog patterns struggle in self-competition at high gain area, and then delays converging but won't bother the target discrimination correct rate, since those aspect consistent patterns belong to the same target.

The 7-level bipolar spectral data format apparently has the best gain population distribution and the negative gain population at low gains is few. The real small population transition gain should occur at the gain of 0.4, the population with gains greater than 0.4 is less than 1.5 patterns. The dominant pattern is almost unique when the considered active gain is larger than 0.4. Therefore, the 7-level bipolar data format has a small number of aspect consistent patterns, then the target group idea becomes faint. It is equivalent that the 7-level bipolar patterns are distinct from each other, and then they have stronger distinct pattern idea than target group idea. Thus the discrimination with 7-level bipolar patterns almost depends on distinguishing distinct patterns and has few self-competition phenomena, and then saves converging iterations.

The 5-level bipolar spectral data format also has a good gain population distribution, and few negative gain population at low gains. Its few population transition gain occurs at the gain of 0.5, and its aspect consistency is higher than the 7-level bipolar data. Since the 5-level bipolar coding has higher linearity than the 7-level bipolar coding, the 5-level bipolar data has more patterns in aspect consistency than the 7-level bipolar data. It is an interesting phenomenon that the 5-level bipolar gain distribution curve is crossed around the gain of 0.43 by the distribution curve of the normalized and offset analog data. The 5-level bipolar data format has lower population than the normalized and offset data for the gains larger than

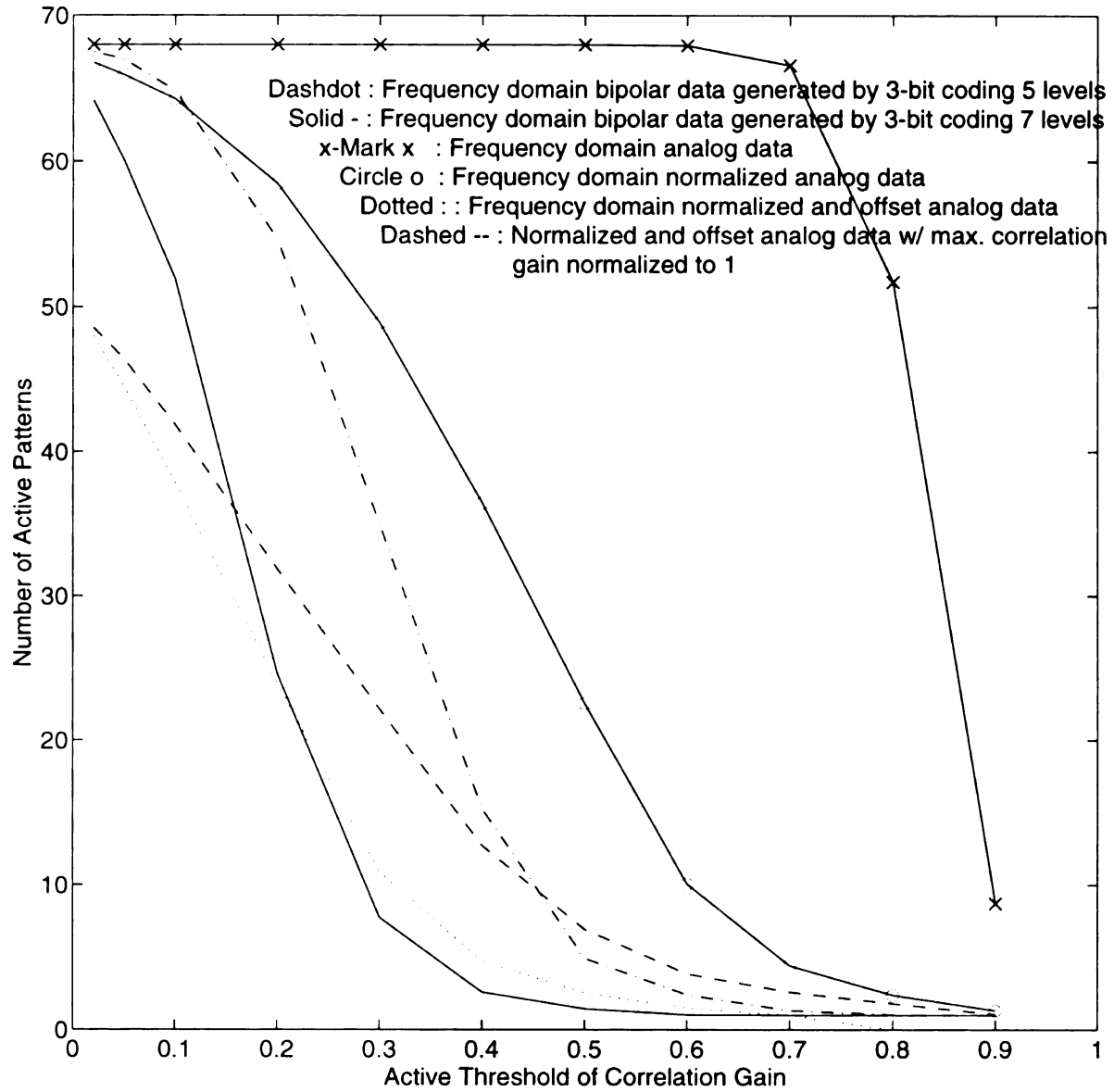


Figure 6.11 Correlation gain population distribution for 68 stored spectral process patterns with different data forms.

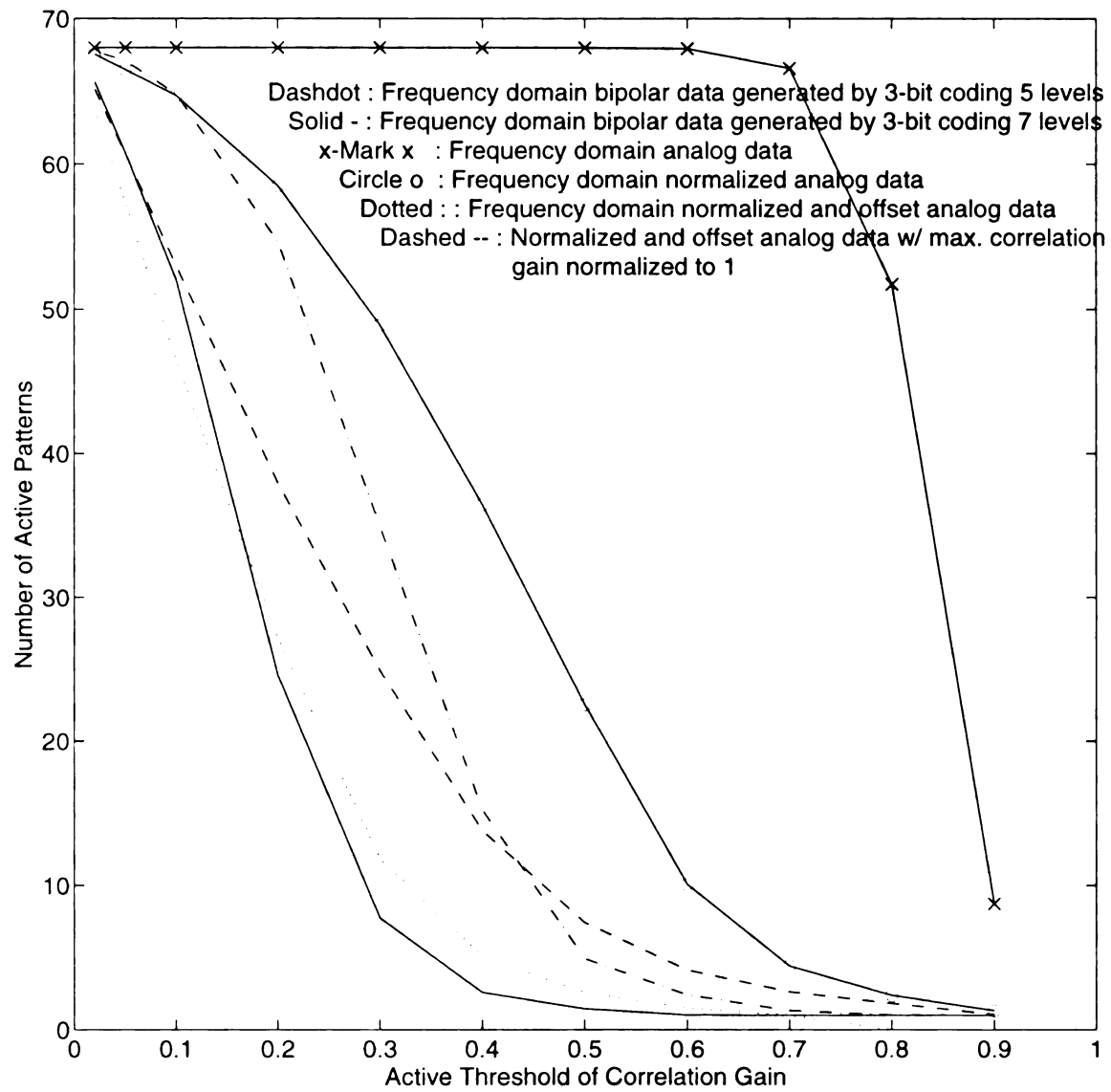


Figure 6.12 Correlation gain population distributions, only considering gain scale, for 68 stored spectral process patterns with different data forms.

0.043 and has higher population for the gains lower than 0.043. It indicates that the 5-level bipolar form has better correlation-based discrimination resolution if dominant patterns needs to have correlation gain larger than 0.5. Practically, it can be expected that the 5-level bipolar data format has better correlation-based discrimination resolution and fewer converging iterations than the analog data, since the few population transition for 5-level bipolar data apparently occur at the gain of 0.5 while the analog data has a smooth population transition curve.

Now let's use the normalized target (cross)correlation gain estimate formula, developed in the previous chapter, to quantitatively estimate the correlation-based discrimination resolutions for different data forms. In the previous chapter, the estimation calculation was based on two assumptions as follows :

- (1). Only active patterns, whose correlation gains greater than the assumed active threshold gain, can participate in generating the next state in the output stage.
- (2). For the active patterns, their associative output patterns are consistent.

From the previous correlation gain population distributions for different spectral data, the analog data format has its small population transition at high gain, therefore we use higher active threshold gains for the estimate formula to keep consistent with above two assumptions. The following three tables show the estimated normalized target correlation gains for different spectral data forms. The estimated normalized target correlation gains for time domain data were presented in the previous chapter.

In Table 6.2, an active threshold gain of 0.4, 40% of the maximum gain, is assumed. The normalized analog data have normalized target crosscorrelation gains comparable to 1,

Spectral Process Input Data Form	No. (%) of Active Patterns w.r.t. a Threshold of 40%	Input Target	Estimated Normalized Target Correlation Gain w.r.t. Input Target			
			B52	B58	F14	TR1
Normalized Analog Data	36.44 (53.59%)	B52	1.0000	1.0808	0.9665	1.1579
		B58	0.5369	1.0000	0.7341	0.7888
		F14	0.5569	0.8495	1.0000	0.7881
		TR1	0.5552	0.7580	0.6558	1.0000
Normalized and Offset Analog Data	13.79 (20.28%)	B52	1.0000	0.5595	0.3881	0.4797
		B58	0.2172	1.0000	0.3418	0.3815
		F14	0.1952	0.4433	1.0000	0.4513
		TR1	0.2157	0.4422	0.4044	1.0000
Bipolar Data with 5 Qantization Levels	15.24 (22.41%)	B52	1.0000	0.6480	0.3326	0.5785
		B58	0.2446	1.0000	0.3748	0.4790
		F14	0.1387	0.4142	1.0000	0.5901
		TR1	0.1915	0.4192	0.4684	1.0000
Bipolar Data with 7 Qantization Levels	2.59 (3.81%)	B52	1.0000	0	0	0
		B58	0	1.0000	0.0412	0.1039
		F14	0	0.0453	1.0000	0.0476
		TR1	0	0.1107	0.0460	1.0000

Table 6.2 Estimated normalized target correlation gains subject to a active threshold of 40% for different spectral process data forms.

therefore its correlation-based discrimination will be greatly interfered by these undesired patterns and then has low correlation-based discrimination resolution. After offsetting the normalized analog data, the normalized target crosscorrelation gains for the normalized and offset data have been apparently decreased and the active pattern number also decreases. Compared to the 5-level bipolar data, both normalized target crosscorrelation gains are very close, and this quantitative similarity is consistent with the crossing gain around 0.4 for the two gain population distribution curves in the previous figure. The 7-level bipolar data have normalized target crosscorrelation gains near or equal to 0, thus at this fair active threshold the 7-level bipolar data format has much greatly prevailed the others.

In Table 6.3 an active threshold gain of 50% is assumed. The normalized analog data have slightly improved their normalized target crosscorrelation gains, while the normalized and offset analog data apparently reduce their normalized target crosscorrelation gains. Compared to the normalized and offset analog data, the 5-level bipolar data have further noticeably improved their normalized target crosscorrelation gains, and again this quantitative variation is consistent with the previous figure. The 7-level bipolar data have all normalized target crosscorrelation gains 0 except two terms. Finally an active threshold 60% of the maximum gain is assumed in Table 6.4. This time the normalized analog data have greatly reduced its normalized target crosscorrelation gains. The 5-level has 0's in all its normalized target crosscorrelation terms except four terms, while the normalized analog and offset analog data continue improving their target crosscorrelation gains. The 7-level bipolar data finally have 0's in all their normalized target crosscorrelation terms.

Spectral Process Input Data Form	No. (%) of Active Patterns w.r.t. a Threshold of 50%	Input Target	Estimated Normalized Target Correlation Gain w.r.t. Input Target			
			B52	B58	F14	TR1
Normalized Analog Data	22.53 (33.13%)	B52	1.0000	0.7998	0.6356	0.9655
		B58	0.3372	1.0000	0.5211	0.6763
		F14	0.3047	0.5923	1.0000	0.6368
		TR1	0.3802	0.6310	0.5233	1.0000
Normalized and Offset Analog Data	7.41 (10.90%)	B52	1.0000	0.3229	0.2058	0.1930
		B58	0.1161	1.0000	0.1782	0.2693
		F14	0.0991	0.2386	1.0000	0.2371
		TR1	0.0818	0.3177	0.2087	1.0000
Bipolar Data with 5 Qantization Levels	4.91 (7.22%)	B52	1.0000	0.1762	0.0239	0.0242
		B58	0.0790	1.0000	0.0776	0.1954
		F14	0.0125	0.0906	1.0000	0.2853
		TR1	0.0095	0.1720	0.2153	1.0000
Bipolar Data with 7 Qantization Levels	1.44 (2.12%)	B52	1.0000	0	0	0
		B58	0	1.0000	0	0.0229
		F14	0	0	1.0000	0
		TR1	0	0.0238	0	1.0000

Table 6.3 Estimated normalized target correlation gains subject to a active threshold of 50% for different spectral process data forms.

Spectral Process Input Data Form	No. (%) of Active Patterns w.r.t. a Threshold of 60%	Input Target	Estimated Normalized Target Correlation Gain w.r.t. Input Target			
			B52	B58	F14	TR1
Normalized Analog Data	10.06 (14.79%)	B52	1.0000	0.4897	0.2111	0.4572
		B58	0.1968	1.0000	0.2027	0.3782
		F14	0.1048	0.2507	1.0000	0.4205
		TR1	0.1993	0.4105	0.3689	1.0000
Normalized and Offset Analog Data	4.12 (6.06%)	B52	1.0000	0.1274	0.0988	0.0317
		B58	0.0455	1.0000	0.0718	0.1512
		F14	0.0406	0.0822	1.0000	0.1060
		TR1	0.0124	0.1649	0.1008	1.0000
Bipolar Data with 5 Quantization Levels	2.41 (3.55%)	B52	1.0000	0	0	0
		B58	0	1.0000	0	0.0900
		F14	0	0	1.0000	0.0408
		TR1	0	0.0867	0.0343	1.0000
Bipolar Data with 7 Quantization Levels	1.03 (1.51%)	B52	1.0000	0	0	0
		B58	0	1.0000	0	0
		F14	0	0	1.0000	0
		TR1	0	0	0	1.0000

Table 6.4 Estimated normalized target correlation gains subject to a active threshold of 60% for different spectral process data forms.

Information Process Form	RCAAM Discrimination Iterations for Contaminated Stored Patterns Under 40 dB / 0 dB ; Average.		
	RCAAM/di22	RCAAM/fi12	RCAAM/ad12
Time Domain Process (Bipolar Data Coding 7 Levels)	3.17 / 5.33 ; 4.35	3.98 / 8.54 ; 6.76	5.54 / 6.04 ; 6.09
Spectral Process (Bipolar Data Coding 5 Levels)	5.05 / 8.69 ; 7.12	5.49 / 7.94 ; 7.12	6.49 / 7.40 ; 9.71
Spectral Process (Bipolar Data Coding 7 Levels)	2.61 / 4.71 ; 4.06	3.95 / 6.15 ; 5.61	6.41 / 7.51 ; 9.70

Table 6.5 Comparisons of network discrimination iterations among different data forms by testing networks with contaminated stored patterns.

Let's check if the above gain population distributions and the estimated normalized target correlation gains are consistent with network discrimination iterations for different data forms. To compare between analog and bipolar data, the RCAAM/fi and RCAAM/ad should be used, since they have the same discrimination algorithm but different input data forms. In Table 6.5, the p denotes the initial order and the q represents the stability criterion in network RCAAM/dipq, RCAAM/fipq and RCAAM/adpq. The comparisons of discrimination iterations between the time domain networks and spectrum magnitude networks are inappropriate, since the time domain responses have less consistencies resulted from the

inconsistent detections of the start beginning responses. In Table 6.5, only the discrimination iterations for 68 contaminated stored patterns under 40 db, 0 db and the average among 10 SNR's are listed here. The 0 db case may approximate the contaminated unstored pattern case. From Table 6.5, Under 40 dB the order of discrimination iterations from small to high are : the 7-level bipolar spectral data < the 7-level bipolar time data < the 5-level bipolar spectral data $\approx$ the analog time data < the normalized and offset analog spectral data. This order is consistent with the correlation gain population distributions in the previous four figures and the quantitatively estimated normalized target correlation gains in the previous three tables.

Under 0 dB the order of discrimination iterations from small to high are : the 7-level bipolar spectral data < the analog time data < the 5-level bipolar spectral data < the 7-level bipolar time data < the normalized and offset analog spectral data. For spectral data forms, the consistencies are still kept. The 7-level bipolar time data format becomes worse than the analog time data, since the 7-level code will suffer severe nonlinear distortion but the analog data still have the linearity when the contamination scale is statistically greater than 1.5 levels. Under serious contamination, the correlation gain population distribution curve should be shifted toward the left side, since the dominant patterns with high correlation gain will become confused by serious distortion and have their high gains drop to low gains. The analog time data format has smooth population distribution curve, while the 5-level bipolar spectral data format has apparent population transition around the gain of 0.5. Therefore, under severe contamination the 5-level bipolar time data may drop its high gains below the transition gain, while the analog time data smoothly decrease their high gains. Then the analog time data will

have better correlation-based discrimination than the 5-level bipolar spectral data.

Since the normalized and offset analog spectral data have a lot of negative gain population resulted from the offsetting around low gain scale, the heavy contamination can make its high gain population move toward low gains and then struggle with the negative gain population.

6.5 Network Simulations with Spectrum Magnitude Response

Here we simulate the major networks used in this study, GI, HCAM, ECAM, RCAAM and GI cascaded networks, with bipolar data coding 5 and 7 quantization levels. As described before, we use the bipolar data with 5 quantization levels due to their high linearity and less oscillations in spectrum magnitude response. The 7-level data format has a high correlation-based discrimination resolution and can better express the detailed response variations, as shown in the previous section, although it has worse linearity. **Figure 6.13** shows the GI network performances vs. SNR for the 68 contaminated stored patterns, while **Figure 6.14** presents the network generalization performances for the 64 contaminated unstored patterns. The 5-level bipolar data format demonstrates its higher noise tolerance than the 7-level data format for both stored and unstored patterns. If a GI net, trained with patterns of few quantization levels, can converge with zero error, then it means that the well trained GI net can discriminate all training patterns under such a low resolution. Thus the GI net trained with low resolution patterns can still work very well, while the GI net trained with high resolution patterns suffers from noise sensitivity. The spectrum magnitude GI network still prefers to leave ambiguous inputs unknown rather than discriminate wrongly.

Figure 6.15 shows the HCAM and HCAM-GI cascade network performances vs.

SNR for the 68 contaminated stored patterns, while **Figure 6.16** presents the network generalization performances for the 64 contaminated unstored patterns. For 5-level bipolar patterns, the HCAM of order 5 is used, since an order of 3 has very poor performances. For the 7-level bipolar patterns, the HCAM of order 3 is used, since it already shows better performances than the 5-level bipolar form. Again, this is another proof analyzed in the previous section that the 7-level data format has better correlation-based discrimination resolution. Both HCAM's, order of 5 for the 5-level code and order of 3 for the 7-level code, have poor performances for contaminated unstored patterns. With SNR higher than 0 dB, both HCAM-GI cascade networks have improved the noise tolerance performances for the stored patterns. For the unstored patterns, it also raises the wrong discrimination rate with almost the same amount as the correct one, while the HCAM-GI cascade networks improve the correct discrimination rate.

Figure 6.17 shows the ECAM and ECAM-GI cascade network performances vs. SNR for the 68 contaminated stored patterns, while **Figure 6.18** presents the network generalization performances for the 64 contaminated unstored patterns. Similar to the time domain ECAM performances, spectrum magnitude ECAM has excellent performances for both 5-level and 7-level bipolar data. Comparing the performances with SNR's less than 0 dB, the 5-level bipolar patterns have a higher correct discrimination rate and a lower wrong discrimination rate than the 7-level bipolar patterns. It again proves our analysis that the 5-level data have better linearity than the 7-level data, and then the 5-level data can still have their linearity under severe contamination where the linearity of the 7-level data has no longer survived. The correlation-based discrimination resolution becomes trivial and plays an

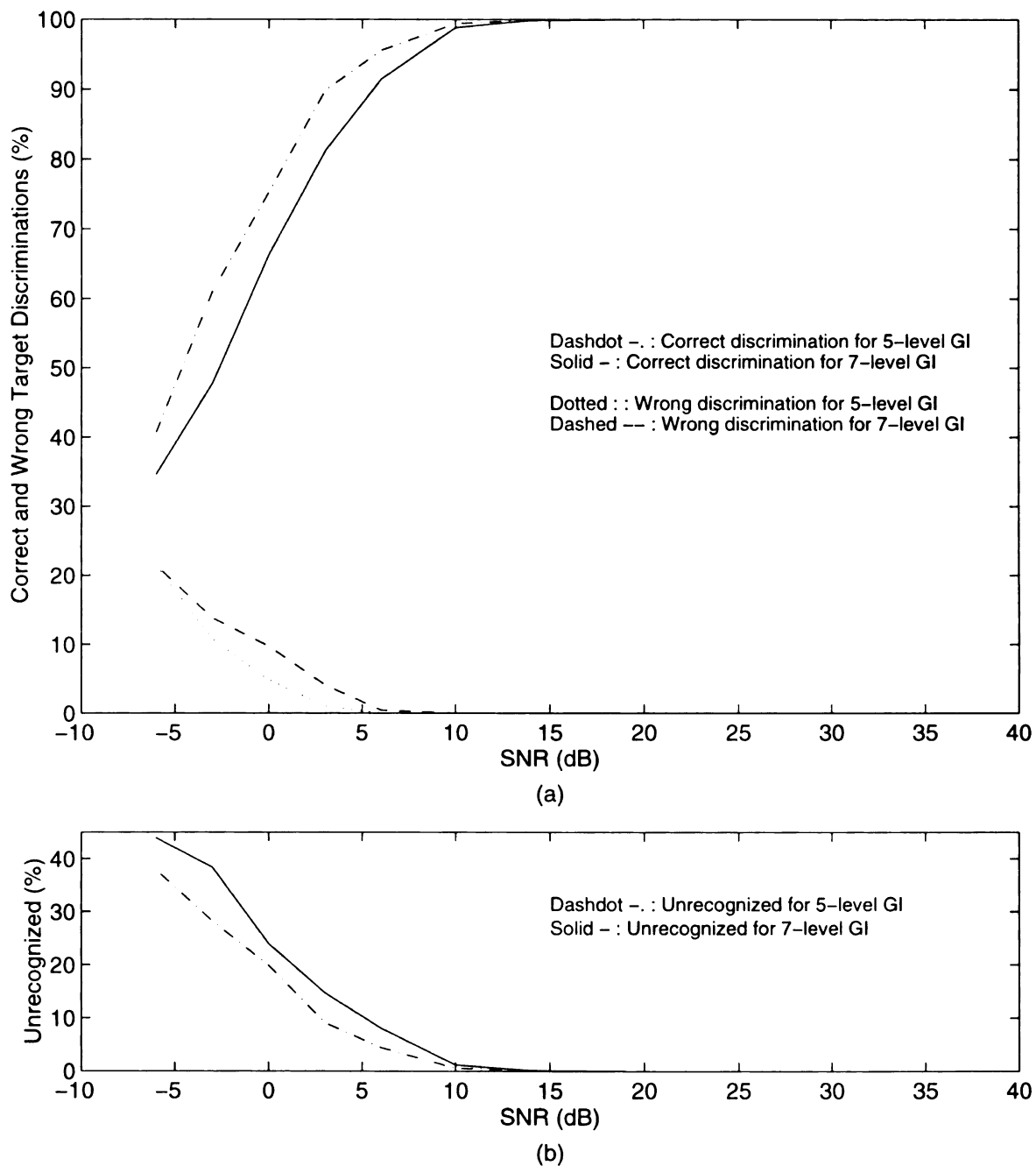


Figure 6.13 Spectrum magnitude GI network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.

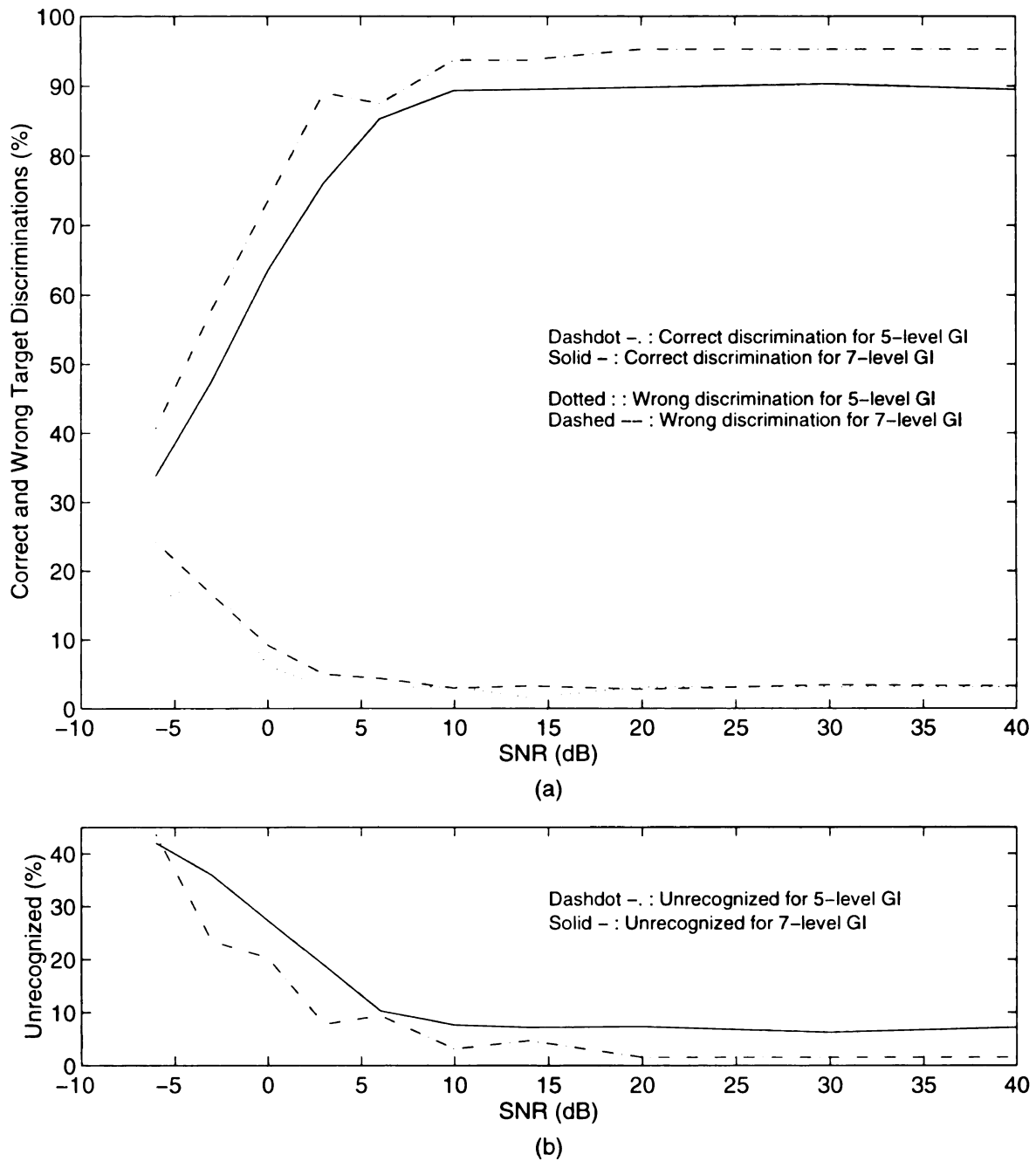


Figure 6.14 Spectrum magnitude GI network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.

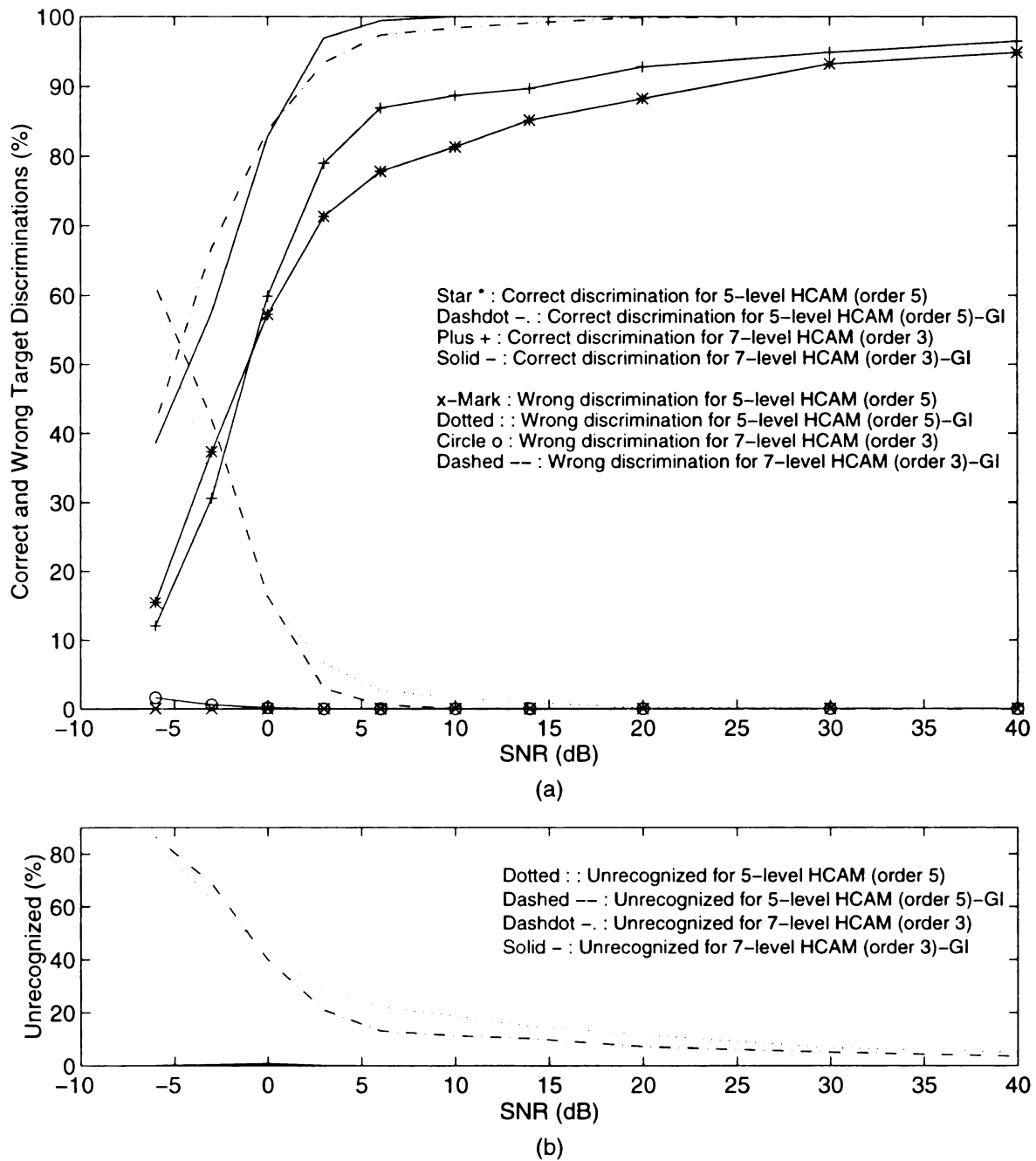


Figure 6.15 Spectrum magnitude HCAM and HCAM-GI cascade network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.

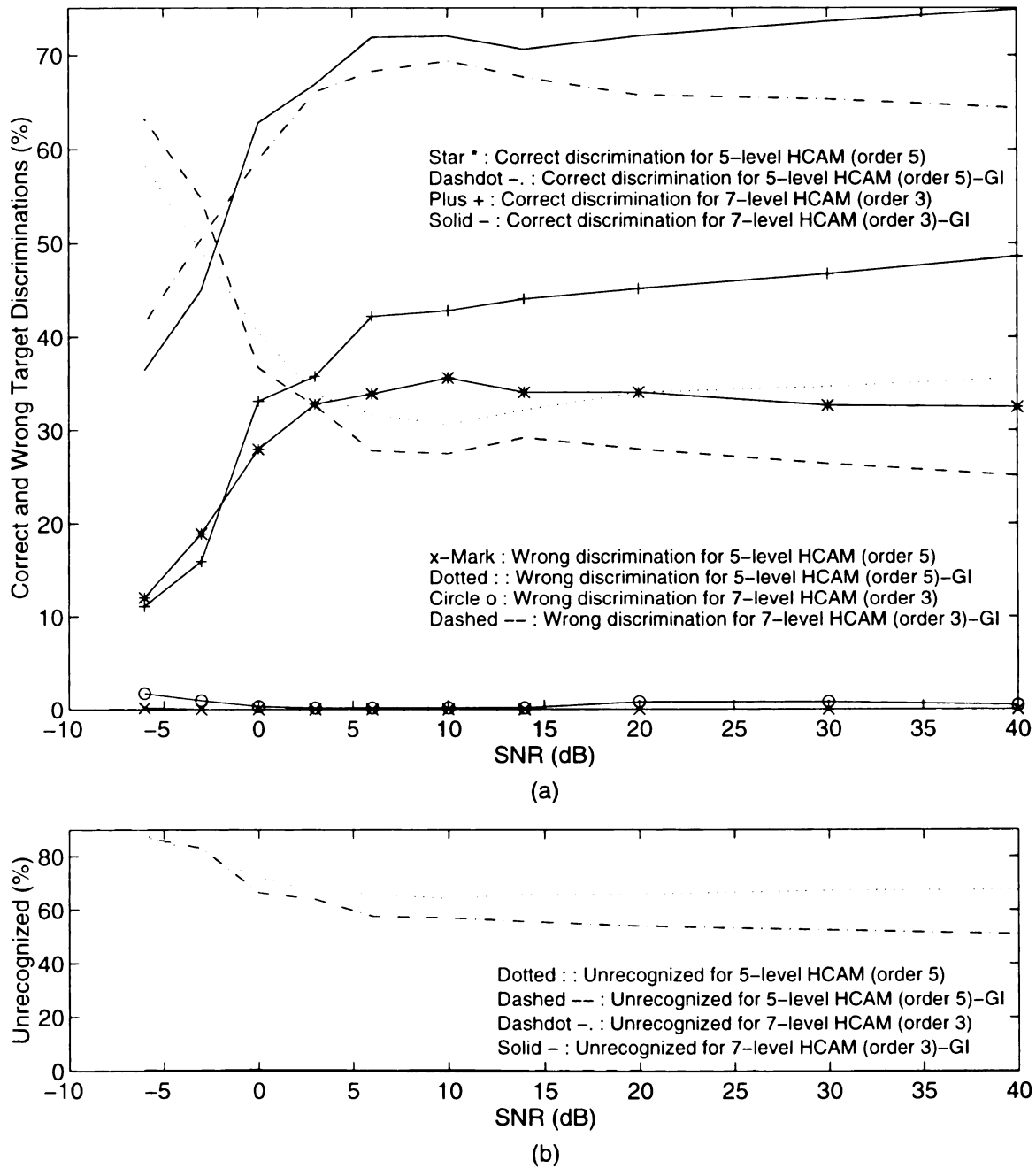


Figure 6.16 Spectrum magnitude HCAM and HCAM-GI cascade network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.

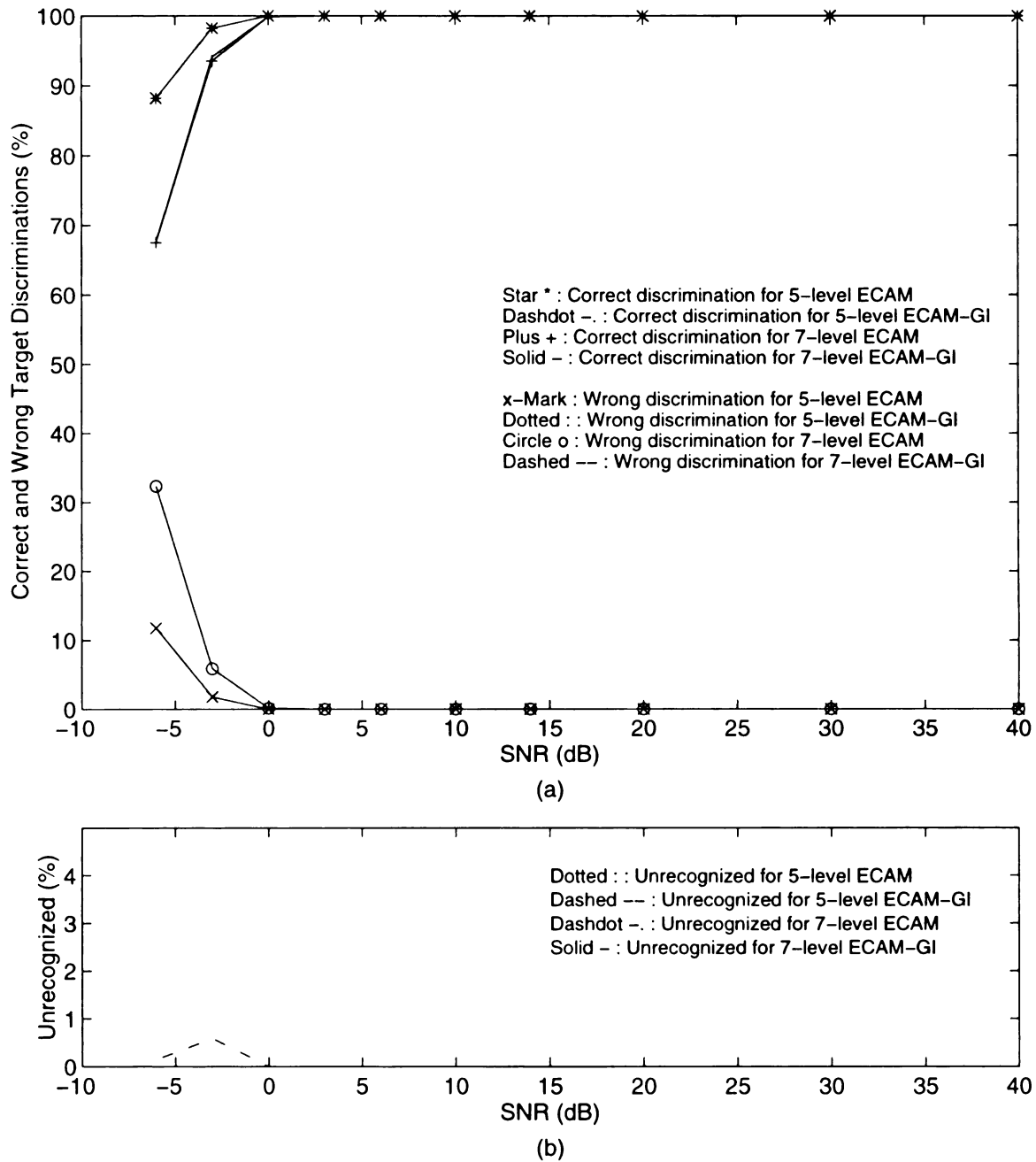


Figure 6.17 Spectrum magnitude ECAM and ECAM-GI cascade network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.

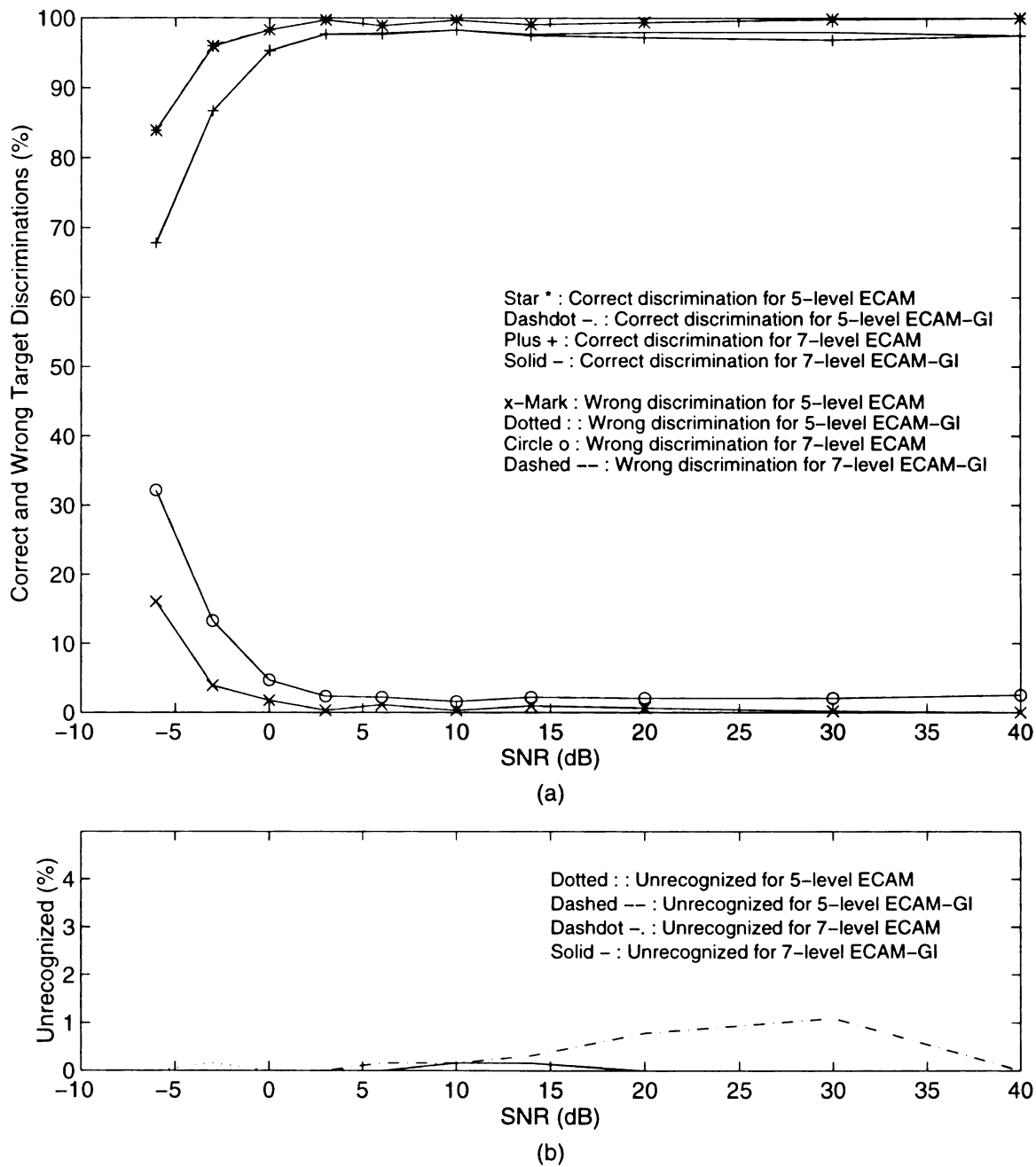


Figure 6.18 Spectrum magnitude ECAM and ECAM-GI cascade network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.

unimportant role to the ECAM, since the ECAM already exponentially expands the discrimination space by using the exponentially weighted correlation gain. Therefore, only the linearity part of bipolar codes will affect the ECAM performances. The ECAM still aggressively converges all inputs into two categories, correct or wrong patterns, and leaves no unknown behind for both stored and unstored patterns. Therefore, the cascaded GI network nearly couldn't improve nothing.

Figure 6.19 shows the RCAAM/di22 and RCAAM/di22-GI cascade network performances vs. SNR for the 68 contaminated stored patterns, while **Figure 6.20** presents the network generalization performances for the 64 contaminated unstored patterns. The first digit behind the 'di' stands for the initial order, and the second digit denotes the stability criterion (**sc**) used by the network. The 7-level bipolar data format prevails the 5-level bipolar data format for the RCAAM/di22 due to its high correlation-based discrimination resolution. Here we only use the low initial order and **sc** to show its advantage over the HCAM with higher orders, 5 for 5-level and 3 for 7-level. The low discriminations for contaminated unstored patterns can be improved by using a higher initial order. The RCAAM/di-GI cascade network has a chance to improve the performances for 5-level bipolar data, since the RCAAM/di22 using 5-level bipolar data leaves several unknowns behind due to its low discrimination resolution.

Figure 6.21 shows the RCAAM/fi12 and RCAAM/fi12-GI cascade network performances vs. SNR for the 68 contaminated stored patterns, while **Figure 6.22** presents the network generalization performances for the 64 contaminated unstored patterns. Comparing the RCAAM/fi12 performances between 5-level and 7-level bipolar data, the 7-

level patterns prevail for fair and slight contamination situations, while the 5-level patterns present better performances for serious distortion cases. It is a good example to express that the discrimination resolution factor will lead network performances where the code linearity still functions, and the linearity factor will dominate network performances where the code linearity fails. For the RCAAM/fi12-GI cascade networks, the 5-level bipolar data format always has better performances than the 7-level bipolar data format for both stored and unstored patterns, since the 5-level GI net trained with lower resolution data has higher noise tolerance. It is amazing that the RCAAM/fi12-GI cascade network, using such a small discrimination space, has the same performances as the ECAM does for the contaminated stored patterns.

From the above observations, we can expect that the high correlation-based discrimination resolution advantage for the 7-level bipolar data will fade when the stability criterion (**sc**) becomes higher, since a higher **sc** will delay to give the discrimination output until an output can stay at the same state for **sc** times correlation accumulations. During the **sc** correlation accumulations, the discrimination resolution is increased, but the code linearity is kept the same. Thus a lower resolution data, 5-level, will improve its discrimination resolution from a high **sc**, but a higher resolution data, 7-level, can not improve its linearity from the **sc**. Hence, the linearity advantage will gradually appear when the **sc** goes high.

Figure 6.21 shows the RCAAM/ad12 and RCAAM/ad12-GI cascade network performances vs. SNR for the 68 contaminated stored patterns, while **Figure 6.22** presents the network generalization performances for the 64 contaminated unstored patterns. Compared to the RCAAM/fi12, especially for unstored patterns, the analog spectrum

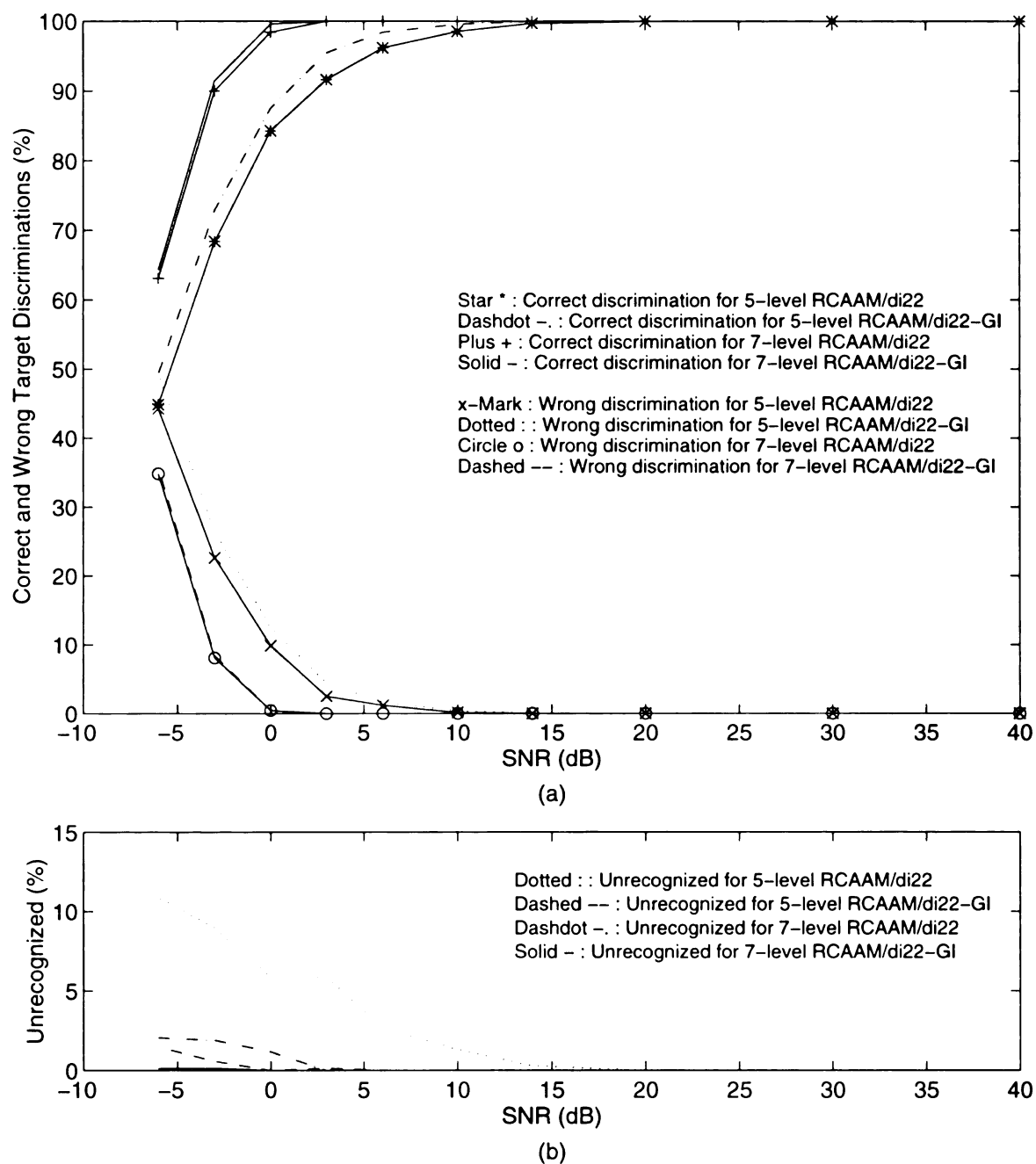


Figure 6.19 Spectrum magnitude RCAAM/di22 and RCAAM/di22-GI cascade network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.

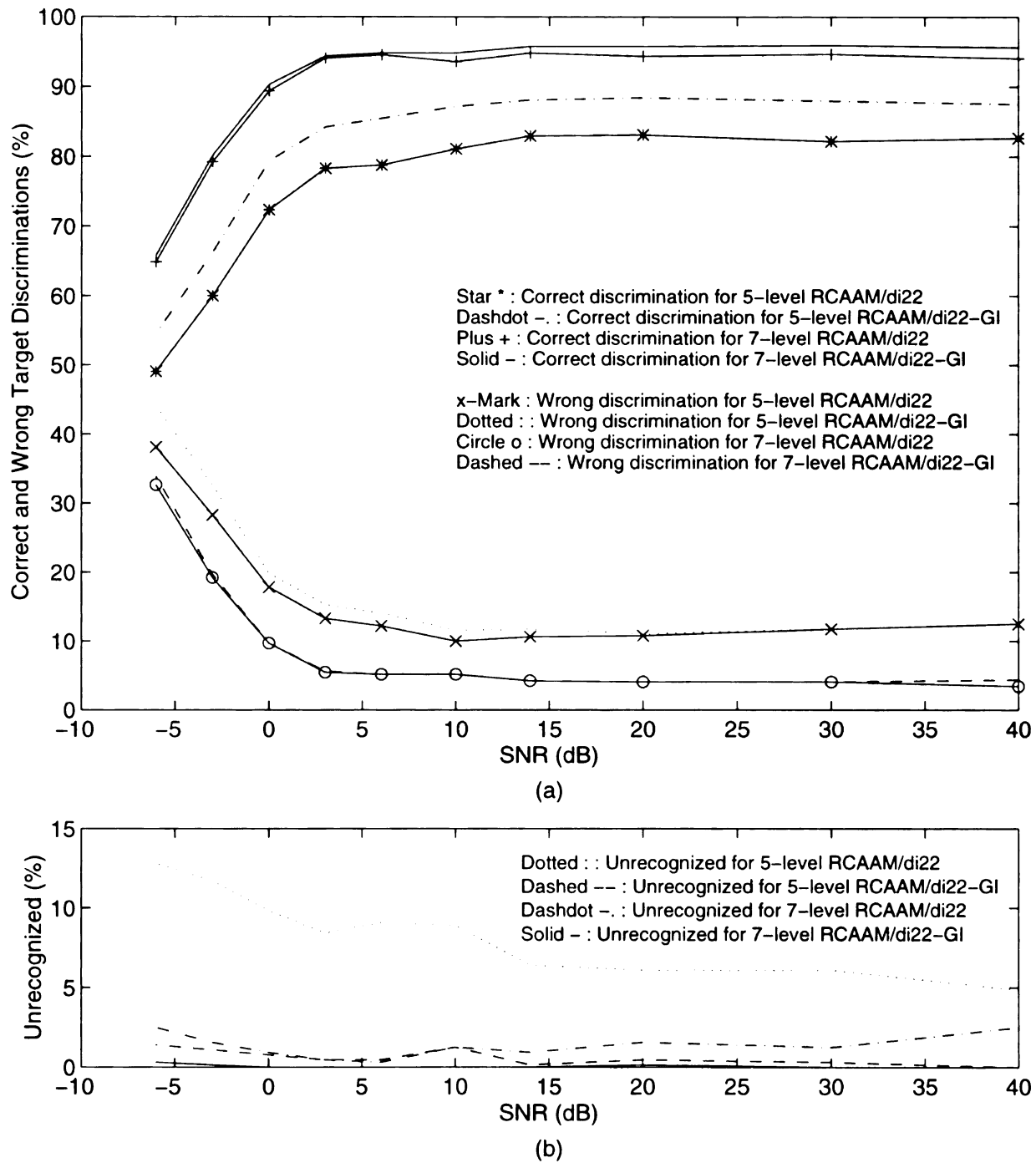


Figure 6.20 Spectrum magnitude RCAAM/di22 and RCAAM/di22-GI cascade network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.

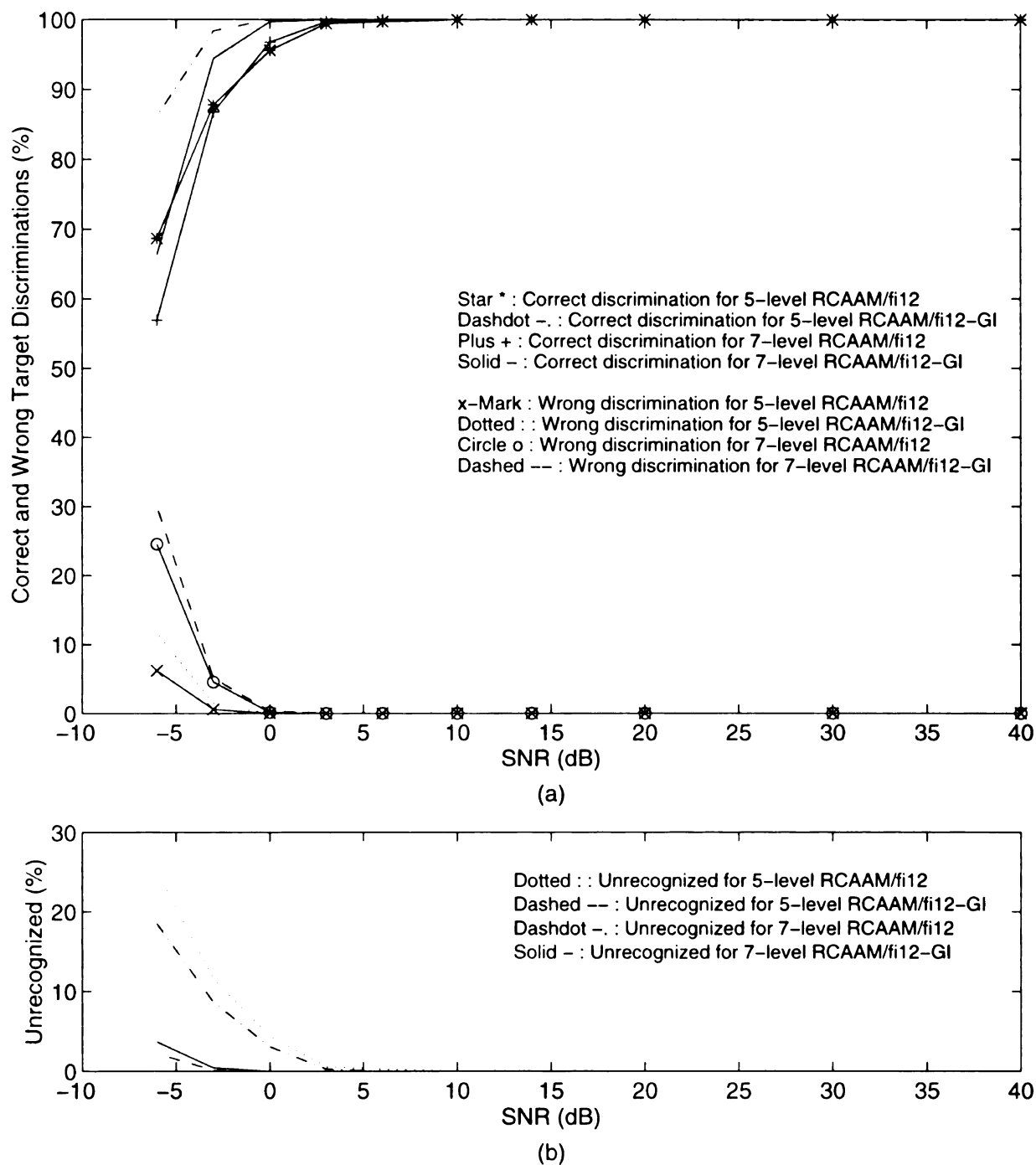


Figure 6.21 Spectrum magnitude RCAAM/fi12 and RCAAM/fi12-GI cascade network performances with bipolar data coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.

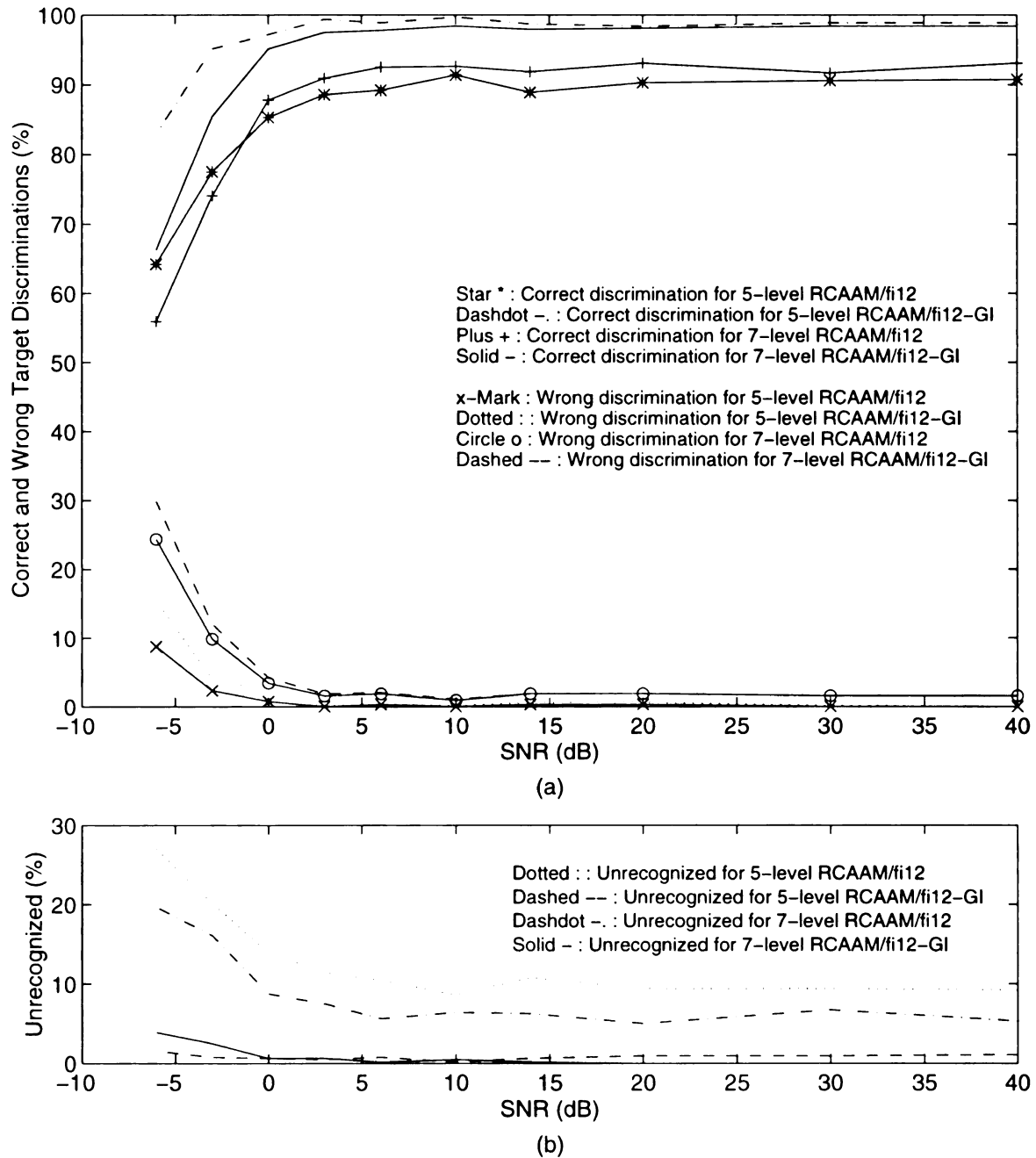


Figure 6.22 Spectrum magnitude RCAAM/fi12 and RCAAM/fi12-GI cascade network generalization performances with bipolar data coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.

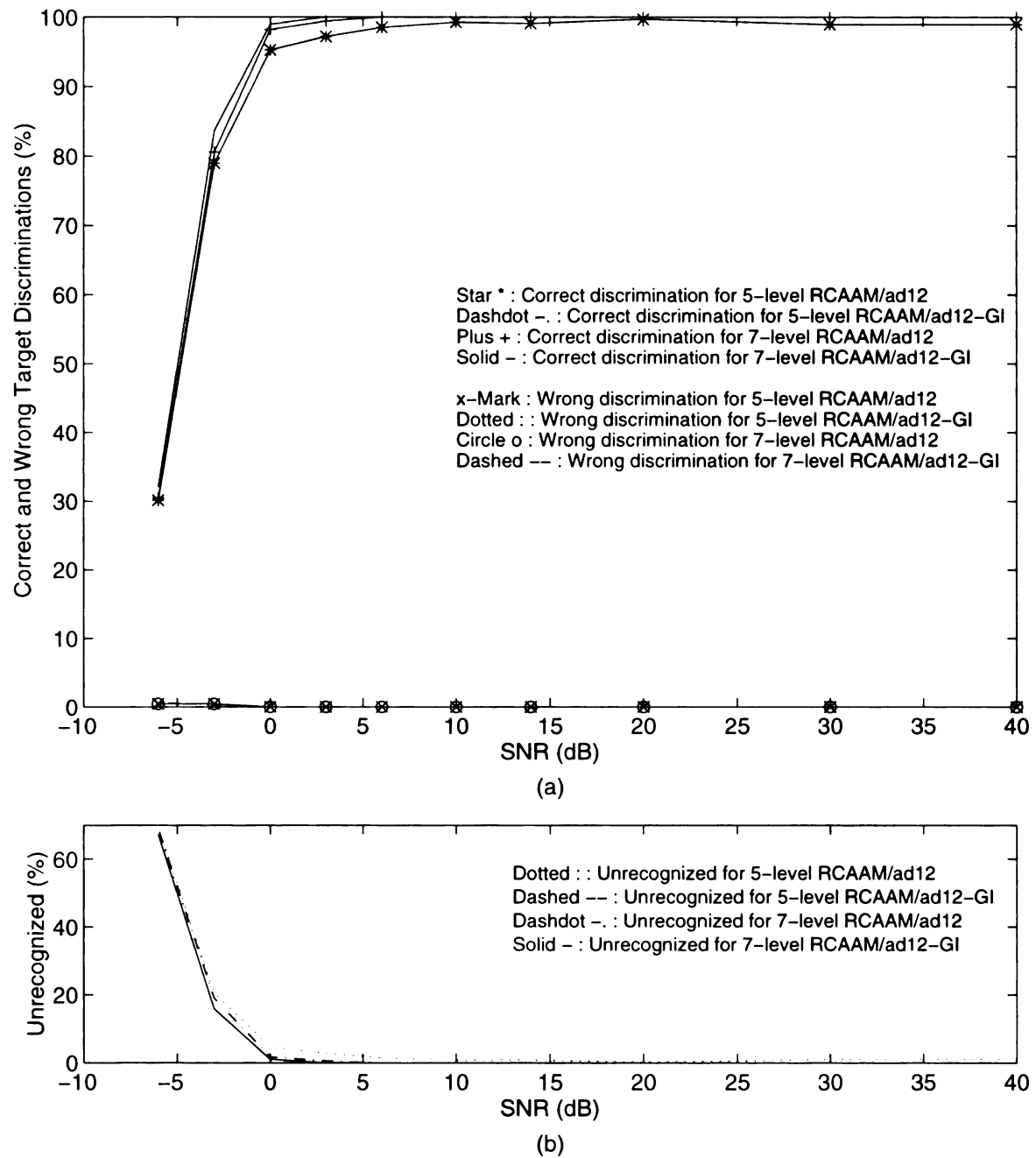


Figure 6.23 Spectrum magnitude RCAAM/ad12 and RCAAM/ad12-GI cascade network performances with bipolar output forms coding 5 and 7 levels vs. SNR for 68 contaminated stored patterns.

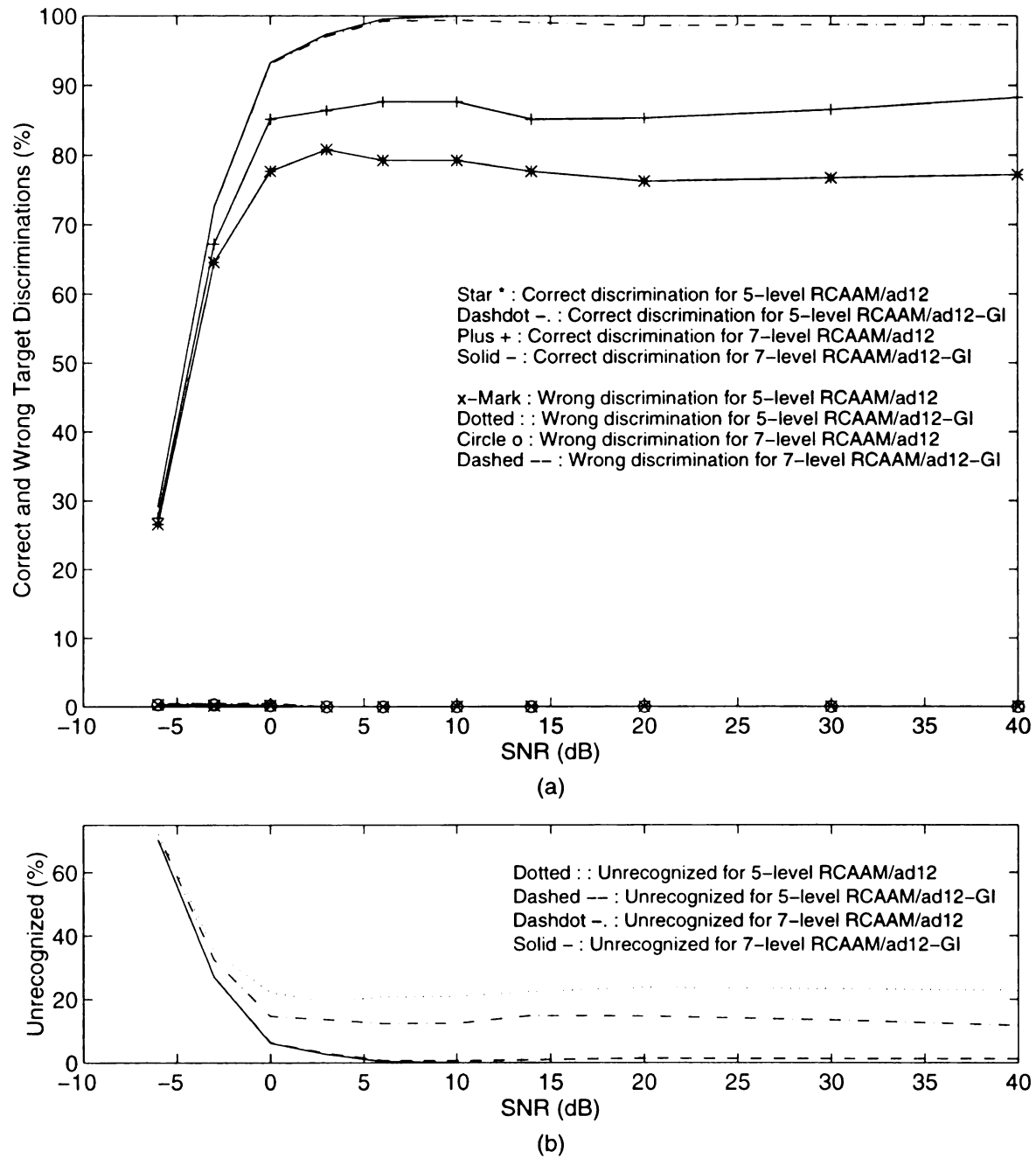


Figure 6.24 Spectrum magnitude RCAAM/ad12 and RCAAM/ad12-GI cascade network generalization performances with bipolar output forms coding 5 and 7 levels vs. SNR for 64 contaminated unstored patterns.

magnitude has low correlation-based discrimination resolutions. Since a low sc , 2, is set, the patterns with low correlation-based discrimination resolution will still remain confused together after residing at the same state for 2 times and then a lot of inputs are left unrecognized. From the RCAAM/ad12 performances for contaminated unstored patterns, we can realize that the 7-level bipolar data format has better pattern distinction resolution than the 5-level data format, since the weighted gains, the correlation gains between an analog input and the stored analog patterns, to both 5-level and 7-level bipolar data are the same. A better pattern distinction resolution indicates that the patterns (or codes) are highly distinct from each other, thus have high correlation-based discrimination resolution. The near zero wrong discrimination from the RCAAM/ad12 indicates that the normalized and offset analog spectrum magnitude patterns have very high linearity so that they only become ambiguous but not close to any wrong pattern under noisy situations. The cascaded GI net has greatly converged those ambiguous states to their true corresponding targets, thus the combination of the RCAAM/ad and GI networks becomes a very good scheme. Compared to the time domain RCAAM/ad12, the effect of correlation-based discrimination resolution of data forms to network performances is clear.

6.6 Network Comparisons

Several bipolar/binary networks, the Hopfield recurrent net and BAM's, have been simulated for the purpose of comparisons. To demonstrate the RCAAM's advantages over other popular networks, the following neural networks have been simulated : Hopfield recurrent network, Bidirectional Associative Memories (BAM) and Multi-Layer feedforward and error-BackPropagation (ML/BP) networks.

The autoassociative Hopfield net always leads to some undefined states and leaves none discriminated. The BAM also has poor performances and always converges to wrong heteroassociative partners. We have altered the BAM process strategy by using target group code in its heteroassociative partners, and then we only discriminate the target group code portion of the final stable state and ignore the rest of the code. This effort has greatly improved the correct recognition rate and also reduced the wrong rate in simulations. In our simulations, we design a set of 7-bit heteroassociative codes corresponding to the 68 300-bit stored patterns. The first two bits are designed as a target group code for four different targets, i.e. $[-1 \ -1]$ for B52, $[-1 \ 1]$ for B58, $[1 \ -1]$ for F14 and $[1 \ 1]$ for TR1, and then the next five bits are coded to represent 17 azimuthal responses of each target. Therefore, this altered BAM will discriminate an input as a correct target or wrong target, and leave none unknown. For example, if the BAM using group code has 36.2% and 8.7% correct discriminations respectively at 40 and 0 dB; then the remaining 63.8% and 91.3% all contribute to wrong discriminations.

We also simulate two ML/BPs, one without a hidden layer and another with one hidden layer containing 25 neurons, to compare their performances to the networks we used. We use the uncoded 100 analog responses as training inputs for each training pattern, and use the same output target set used by the GI net. Therefore, they have analog inputs and bipolar outputs. The initial weights are randomly initiated, and a momentum has been used to reduce the chance of being trapped in local minima. Compared to the GI net, ML/BPs require more training epochs and manipulations to converge all training patterns to their desired targets with zero error. Although a momentum has been used, the trainings are still sometimes

trapped in local minima a number of times. The ML/BP nets have a similar performance phenomenon to the GI net. As expected, they prefer to leave ambiguous inputs unknown rather than discriminate wrongly.

We have used two different discrete levels to quantize an analog spectrum magnitude, 5 levels and 7 levels. Less quantization levels give not only higher linearity but also lower noise sensitivity, although they may lose resolution. In a severe noisy condition, the coding's linearity and noise tolerance sometimes are more important. From figures 1 and 6, it is easy to see that the time responses oscillate with a much larger range than the frequency spectral magnitudes. Therefore 7 levels are used in the time responses for accurately representing its large oscillation range. Different factors prevail in different networks. The noise tolerance and linearity factor prevail in training-based convergent networks, i.e. the GI net, while the resolution factor prevails in correlation-based recurrent memories for moderate noisy conditions. For correlation-based recurrent memories, more quantization levels will enlarge the correlation gain of two similar patterns and reduce the gain of two unlike patterns. This two-side effect greatly increases the discrimination resolution, so the recurrent network performance will improve.

To illustrate in detail 9 different network performances under 10 different SNR levels, six figures are plotted. **Figure 6.25** shows the performance comparisons of time domain networks with 7-level bipolar data for 68 contaminated stored patterns, while **Figure 6.26** presents the network generalization performance comparisons for 64 contaminated unstored patterns. **Figure 6.27** shows the performance comparisons of spectrum magnitude networks with 5-level bipolar data for 68 contaminated stored patterns, while **Figure 6.28** presents the

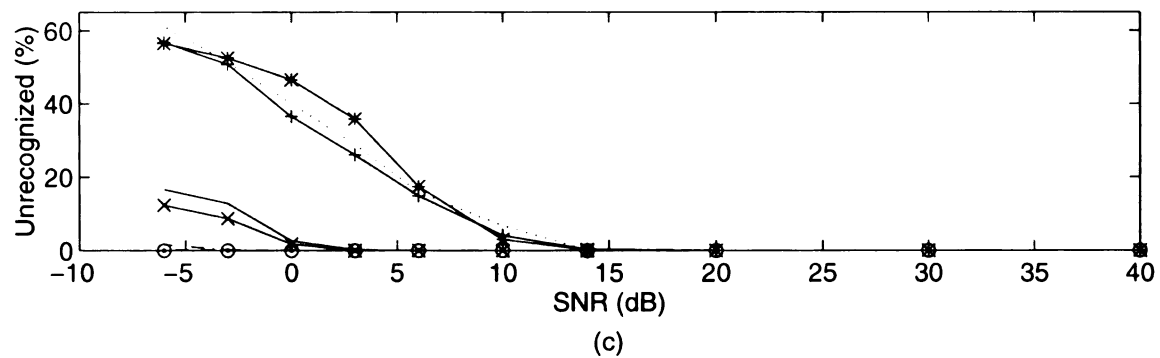
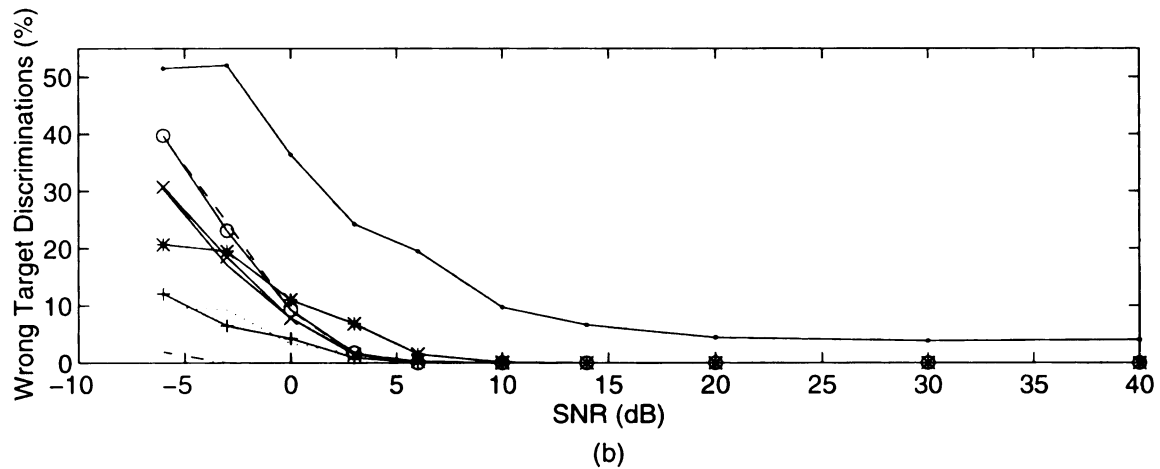
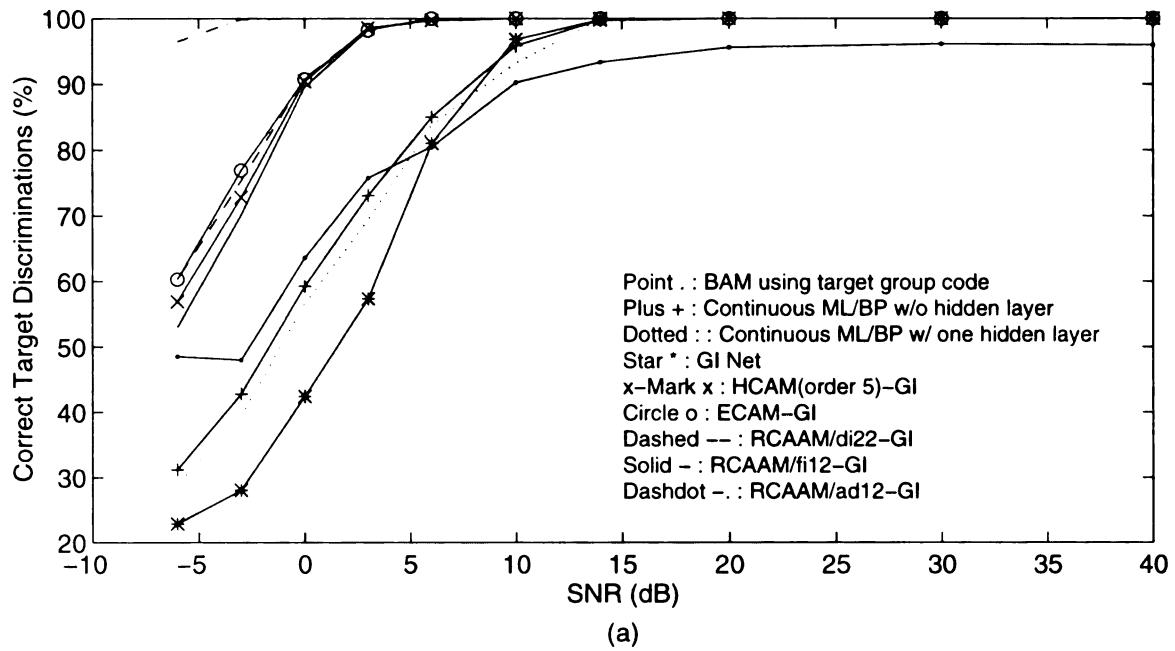


Figure 6.25 Performance Comparisons of different time domain networks (7-level bipolar data, if used) vs. SNR for 68 contaminated stored patterns.

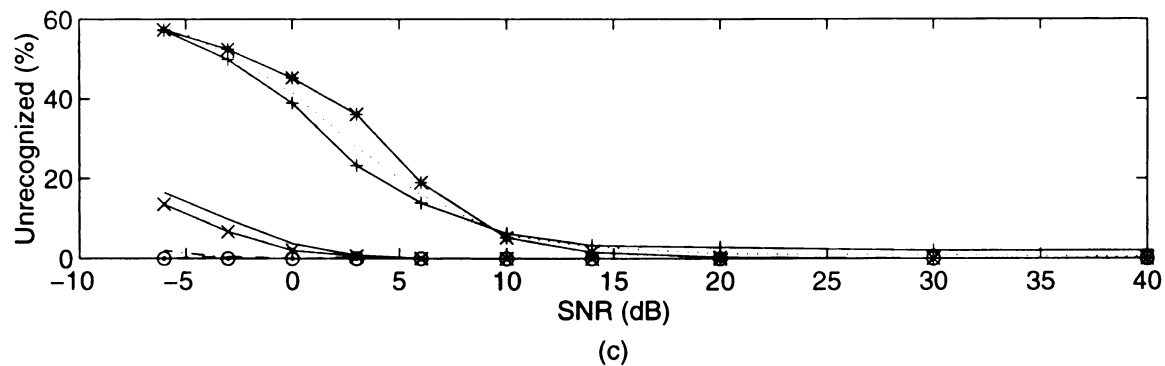
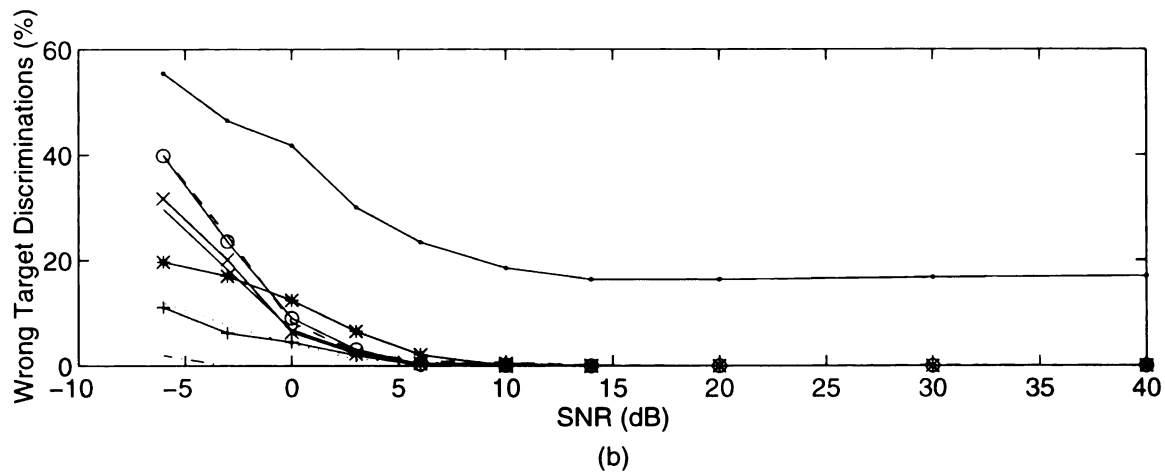
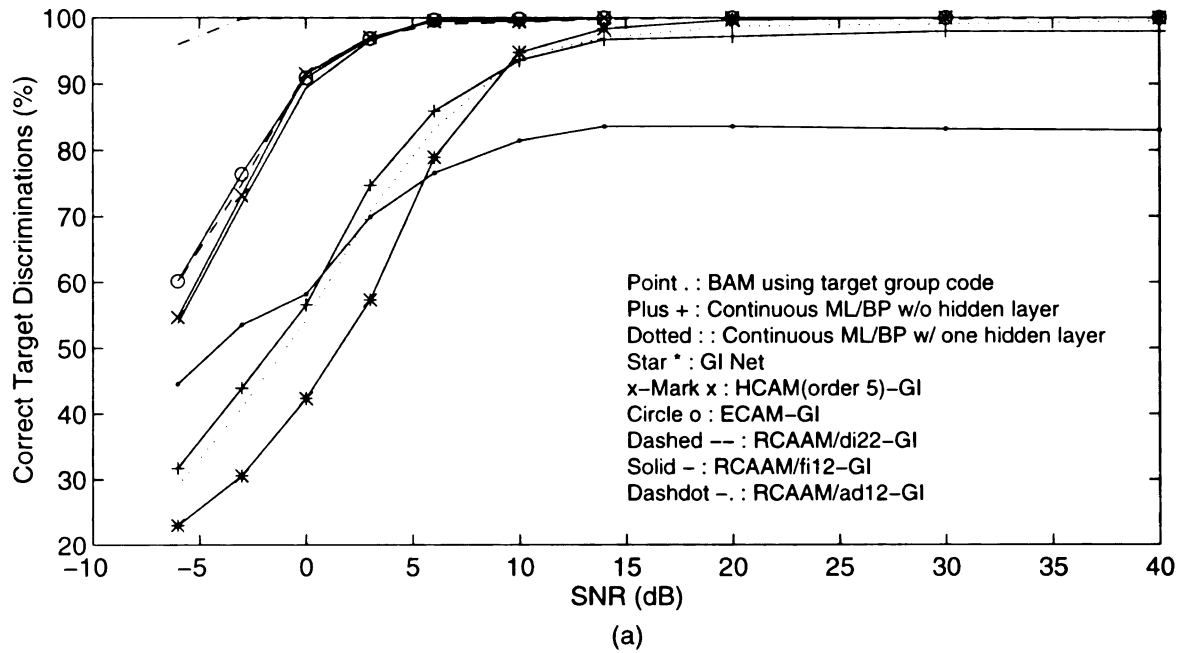


Figure 6.26 Generalization performance Comparisons of different time domain networks (7-level bipolar data, if used) vs. SNR for 64 contaminated unstored patterns.

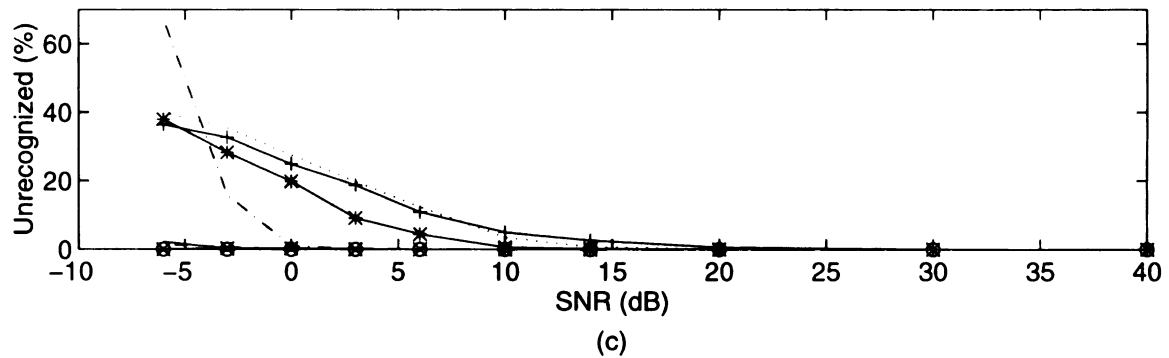
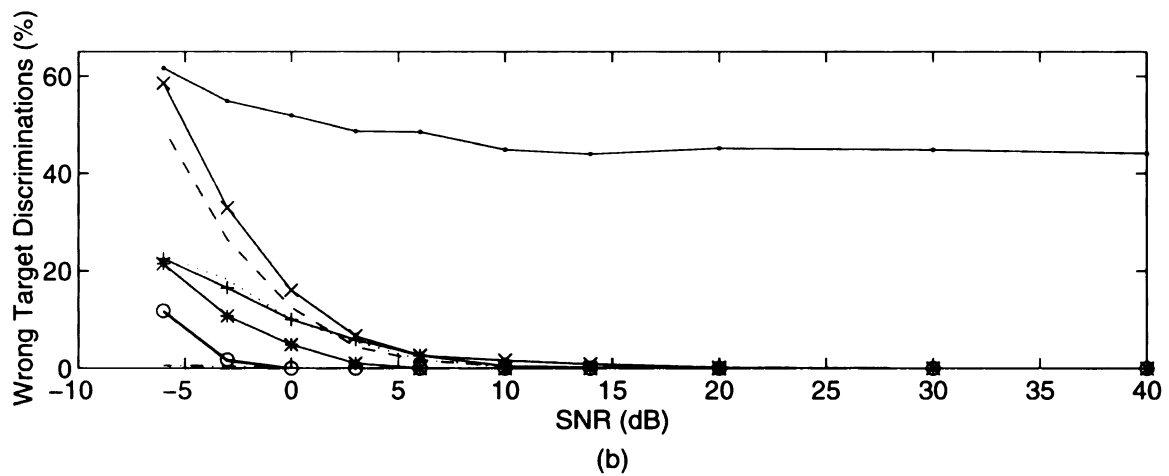
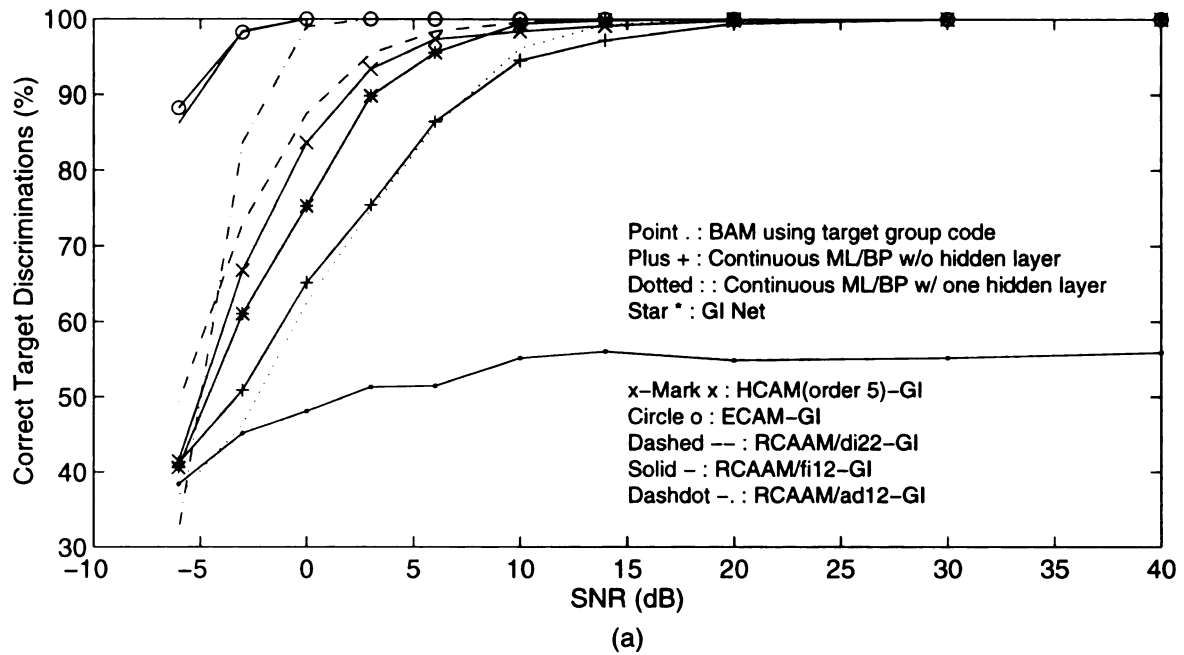


Figure 6.27 Performance Comparisons of different spectrum magnitude process networks (5-level bipolar data, if used) vs. SNR for 68 contaminated stored patterns.

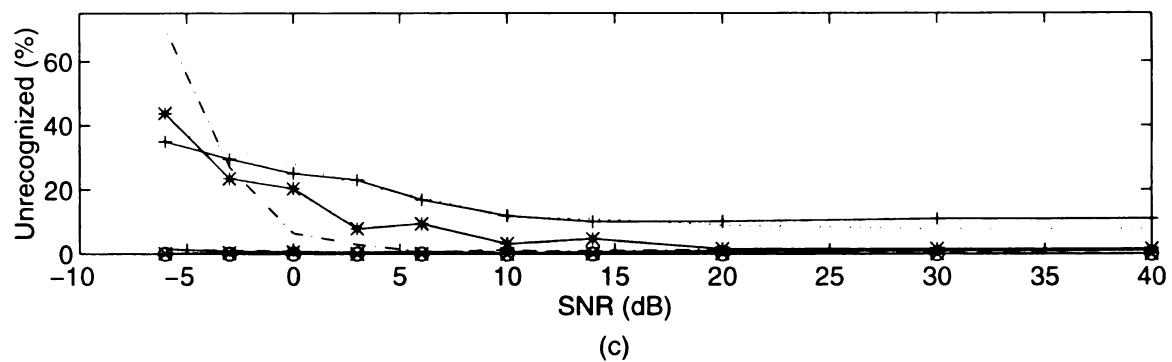
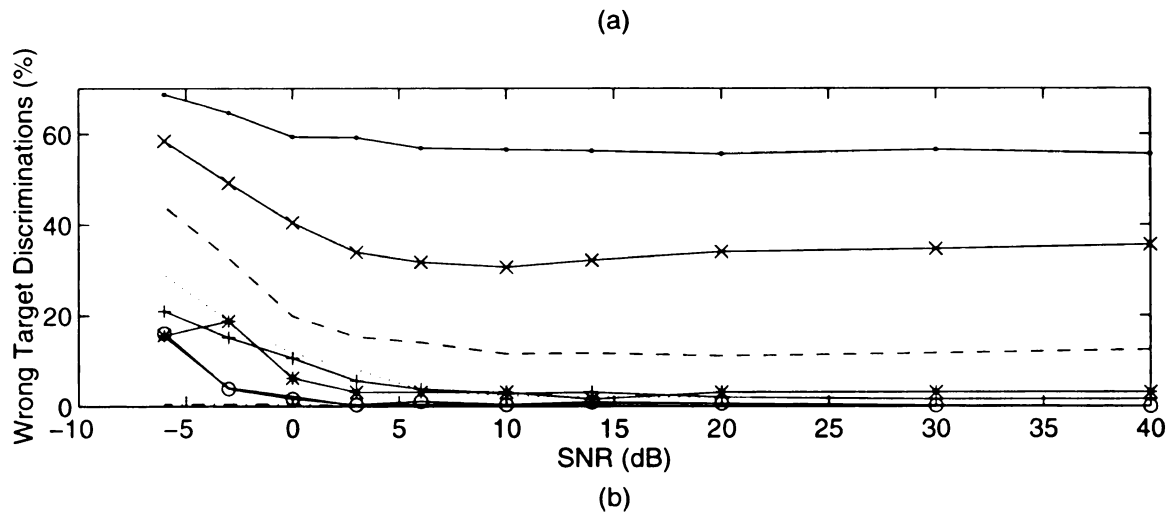
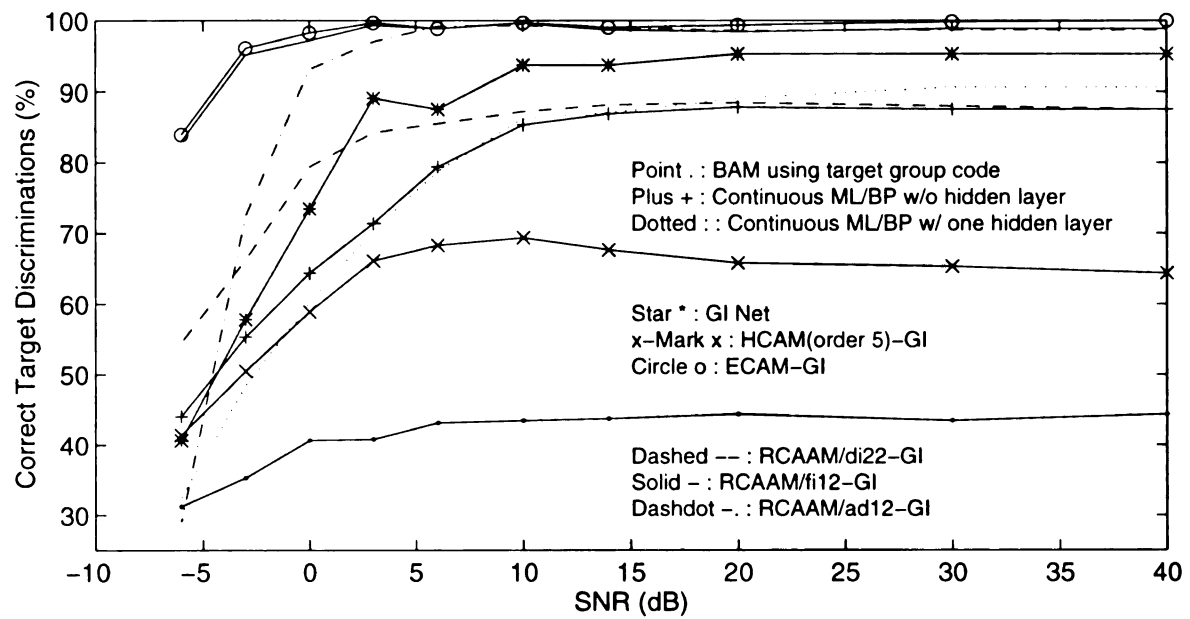
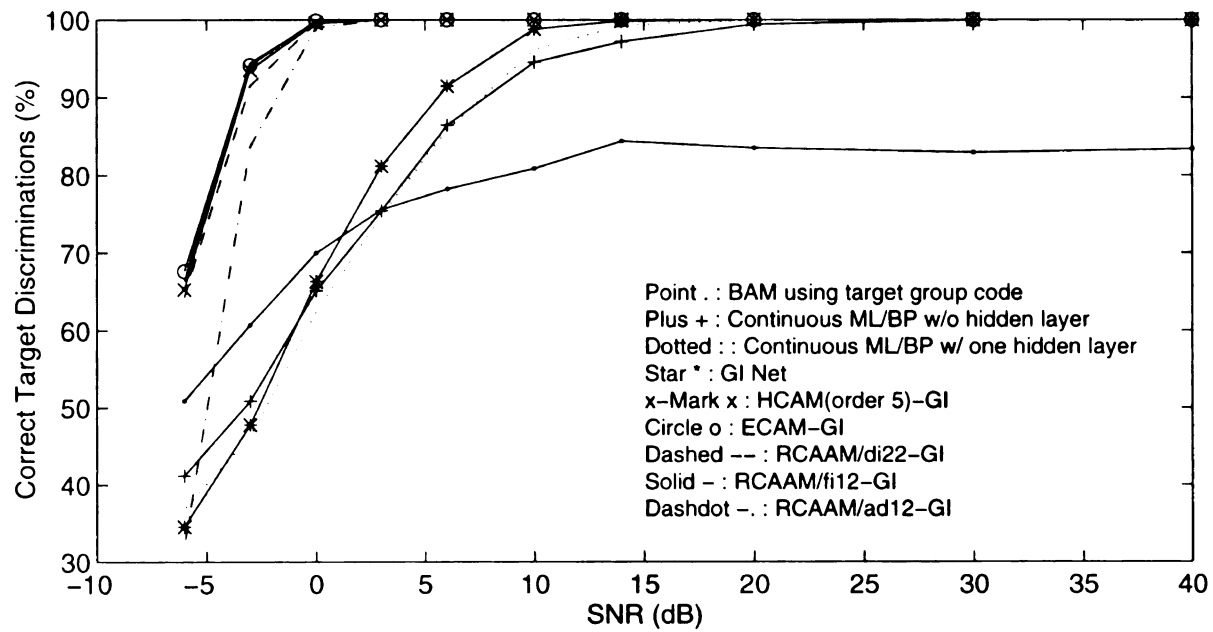
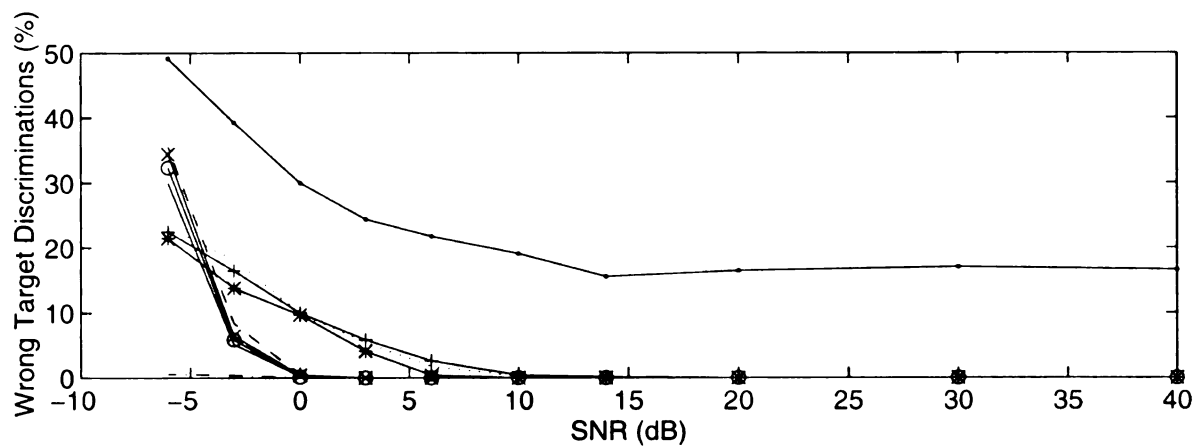


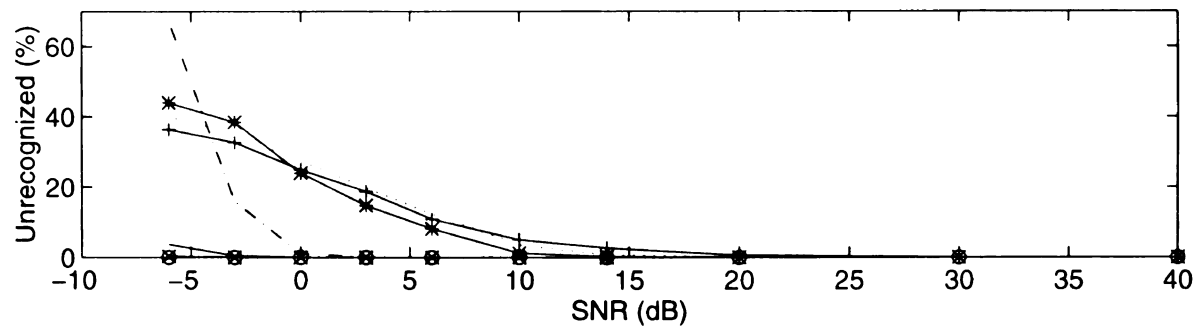
Figure 6.28 Generalization performance Comparisons of different spectrum magnitude process networks (5-level bipolar data, if used) vs. SNR for 64 contaminated unstored patterns.



(a)

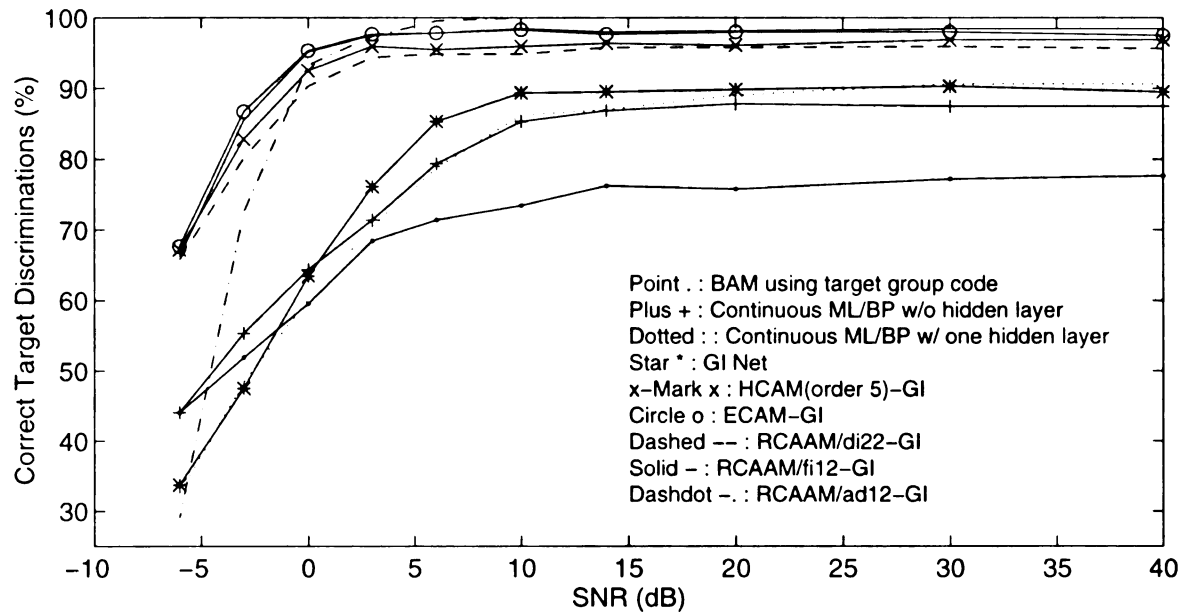


(b)

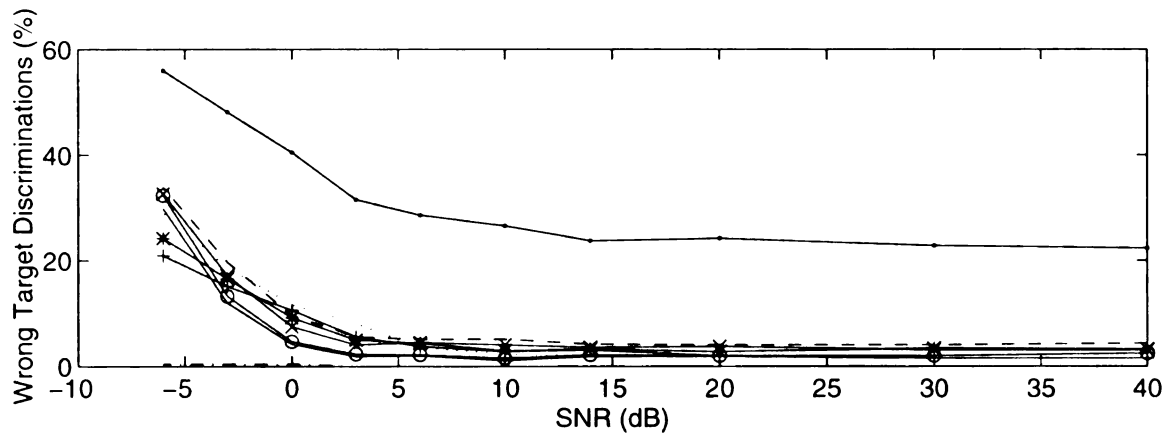


(c)

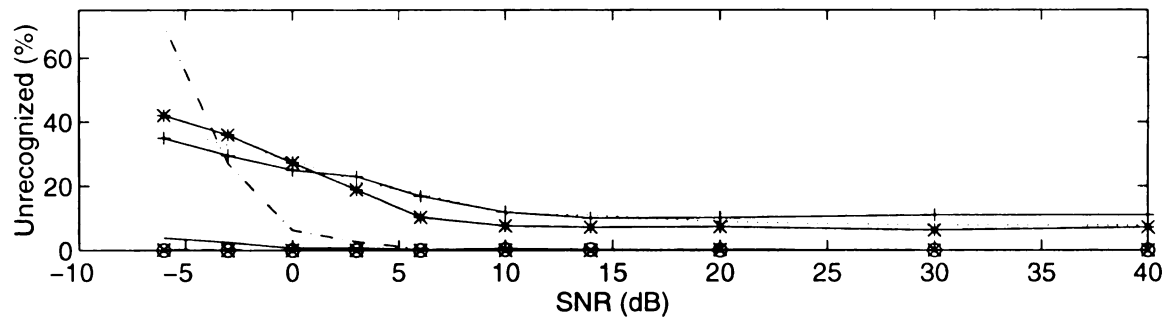
Figure 6.29 Performance Comparisons of different spectrum magnitude process networks (7-level bipolar data, if used) vs. SNR for 68 contaminated stored patterns.



(a)



(b)



(c)

Figure 6.30 Generalization performance Comparisons of different spectrum magnitude process networks (7-level bipolar data, if used) vs. SNR for 64 contaminated unstored patterns.

network generalization performance comparisons for 64 contaminated unstored patterns. **Figure 6.29** shows the performance comparisons of spectrum magnitude networks with 7-level bipolar data for 68 contaminated stored patterns, while **Figure 6.30** presents the network generalization performance comparisons for 64 contaminated unstored patterns. For correlation-based networks, we use their GI-cascaded networks for performance comparisons. All HCAM's use an order of 5 in comparisons, since we want the best performances of the HCAM to compare with the RCAAM. In time domain networks, it is clear that the correlation-based networks, except the BAM, have better performances than the training-based networks despite of the needed discrimination space. The RCAAM/ad12-GI cascade network has the best performance, and the ECAM-GI and the RCAAM/di22-GI have almost the same performances. The BAM has the worst results, since, to improve its discrimination, its discrimination philosophy is altered to the bi-state, correct or wrong.

In the spectrum magnitude networks, the RCAAM/ad12-GI cascade network no longer has the best performances, since the normalized and offset analog spectrum magnitude has the lower correlation-based discrimination resolution than the 5-level and 7-level bipolar data. The RCAAM/fi12-GI and ECAM-GI cascade networks using 5-level bipolar data have the best performances, since the linearity of 5-level bipolar code can survive through severe noise. Again, the BAM has the worst performances. The HCAM-GI cascade network using 5-level bipolar data has poor performance for contaminated unstored patterns, but the HCAM-GI cascade network has the performances similar to the RCAAM/di22-GI network's. But a HCAM with an order of 5 means that each recurrent operation needs to manipulate 5 multiplications to have a correlation gain for one stored pattern. Therefore, compared to the

RCAAM, the HCAM needs 5 times more hardware operations, or clocks in digital circuit, to finish the correlation evaluation.

We summarize the architectures of the recurrent correlation associative networks used in Table 6.6. The computation space for RCAAM's is iteratively adaptive. For lightly contaminated patterns, the computation space required for discrimination is small due to few iterations, while a larger space is required for highly distorted patterns. Row 5 presents the available knowledge observed from the network operations about contamination or similarity between input and the final stable output. The ECAM converges most distorted inputs to some stable states within 3 iterations since it extremely expands the discrimination space. Therefore, we are unable to determine contamination from the ECAM. The HCAM's with small order are usually trapped in some spurious stable states, while the RCAAM typically avoids that with its accumulatively dynamic memory and the adjustment of sc . The decisions are deterministic for both HCAM and ECAM, when their outputs don't change for one iteration, since their memories are fixed.

The architectures and numerical performances are summarized in Table 6.7 for time domain process networks, in Table 6.8 for spectrum magnitude process networks with 5 quantization levels coded by 3 bits and in Table 6.9 for spectrum magnitude process networks with 7 quantization levels coded by 3 bits. Column 4 in each table presents the training epochs and final errors for the ML/BPs and the GI networks. The first number in this column presents the training epochs with which network trainings converge to 0 error, while the second number denotes the extra training cycles made to ensure all training pattern outputs deeply enter the saturation regions of the sigmoid function. Theoretically, this will increase noise

Network	HCAM	ECAM	RCAAM
Input form	Binomial	Binomial	Binomial or Analog
Process structure/ Recurrent operation	Fixed memory/ Adaptive	Fixed memory/ Adaptive	Dynamic memory/ Accumulatively adaptive
Computation Space	Small for small orders	Extremely huge	Fit for discrimination
Observability about contamination or similarity between input and the final output	High for small orders ; low for high orders	Very little	High
Possibility trapped in unknown stable states	Very high for small orders	Very little	Adjustable
Decision strategy	Deterministic	Deterministic	Flexible
Hardware Realization	Capable for fair orders	Nearly incapable	Capable
Multiplication operation (clock) times required to calculate a gain weighting for each stored pattern at each update in a digital circuit	times of the order used; i.e. 5 times for the HCAM with order of 5	(1 + correlation gain) times; times of correlation gain are spent on calculating the power of exponential base	1 time

Table 6.6 Architecture summary of the recurrent correlation associative networks used.

tolerances and decrease the biased learnings. Column 5 presents the average iterations which the recurrent networks require to reach the stable state adopted for target discrimination. The upper number in every row corresponding to 'Trained' means the result obtained by testing the contaminated trained/stored patterns, while the lower number corresponding to

'Untrained' indicates the result obtained by testing the contaminated untrained/unstored patterns. Column 6 presents the minimum SNR in dB at which the network still performs 95% correct discrimination. Column 7 shows the correct and wrong discrimination rates in percentage respectively at 40 and 0 dB. The number pairs left to the ',' shows the correct discrimination for 40 and 0 dB, while the number pairs right to ',' presents the wrong discrimination. Finally, column 8 presents the maximum integer computation scale required for processing one stored pattern.

High resolution and linearity are the most important issues as long as the input data format is concerned in evaluating the network performance. The high resolution will expand the differences between any two trained/stored patterns and then make pattern discrimination easier, while the linearity between an input and the stored patterns can greatly increase the confidence in network performances. But analog data are hard to use in a recurrent associative update, therefore the analog networks can't iteratively adapt to the final stable state. Without using analog signal, we may not confidently determine the contamination or the similarity between an input and the network output. Also an analog network can't be cascaded to a high performance and high dimension recurrent autoassociative network as a group decoder. Unless an analog network is capable of hardware realization, it can't take advantage of today's digital computer technologies. A binomial data only has two states by which an artificial neuron model emulates the bi-state, activated and inactive, of a biological neuron. Therefore, the binomial data is well qualified for use in recurrent and cascade operations, like complicated biological neural nets. Binomial data uses much less space than continuous data in a digital computer process, and it can be further compressed. Quantizing

continuous data by finite discrete levels can also result in a tolerance of light contamination in a way. And the binary data are easy and safe to store for long periods of time.

Network	Input/ Output Form	Memory Size	Training Epoch /Error	Trained, Untrained			Max. Integer Comput. Scale
				Averg. Recog. Iterat.	Min. dB w/ 95% Correct Recog.	Percent (%) of Correct, Wrong Recog. at 40/0 dB	
Hopfield net	Bip/Bip	300x300	X	9.68; 9.71	None; None	0/0, 0/0; 0/0, 0/0	300 ²
BAM using group code	Bip/Bip	300x7	X	4.28; 4.25	20 dB; None	96.0/63.6, 4/36; 83/58.2, 17/42	300 ²
ML/BP w/o hidden layer	Analog /Bip	100x4	164+ 1012/0	X	10 dB; 14 dB	100/59.3, 0/4; 98.0/56.6, 0/5	Not Integer
ML/BP w/ 1 hidden layer	Analog /Bip	100x25 +25x4	624+ 414/0	X	14 dB; 14 dB	100/56.6, 0/3; 99.5/54.2, 0/4	Not Integer
GI Net	Bip/Bip	300x4	3+ 120/0	X	10 dB; 14 dB	100/42.4, 0/11; 100/42.3, 0/12	Not Integer
HCAM (order of 3); HCAM (order of 5)	Bip/Bip	68x300	X	5.02; 6.12	None; None	80.6/ 52.5, 0/ 2; 63/ 47.2, 0/ 3	300 ³
				3.44; 3.76	20 dB; None	97.5/84, 0/ 7; 92.0/85.2, 0/ 6	300 ⁵
ECAM	Bip/Bip	68x300	X	3.01; 3.03	3 dB; 3 dB	100/90.7, 0/ 9; 100/90.9, 0/ 9	2 ³⁰⁰
RCAAAM /di22-GI	Bip/Bip	68x300	X	4.35; 5.15	3 dB; 3 dB	100/90.9, 0/ 9; 100/91.7, 0/ 8	Adaptive
RCAAAM /fi12-GI	Bip/Bip	68x300	X	6.76; 7.88	3 dB; 3 dB	100/89.6, 0/ 8; 100/89.4, 0/ 7	Adaptive
RCAAAM /da12-GI	Analog /Bip	68x100 +68x300	X	6.09; 7.3	-6 dB; -6 dB	100/100, 0/ 0; 100/100, 0/ 0	Adaptive

Table 6.7 Network architectures and performances summary for time domain target discrimination with 7 quantization levels encoded by 3 bits.

Network	Input/ Output Form	Memory Size	Training Epoch /Error	Trained; Untrained			Max. Integer Comput. Scale
				Averg. Recog. Iterat.	Min. dB w/ 95% Correct Recog.	Percent (%) of Correct, Wrong Recog. at 40/0 dB	
Hopfield net	Bip/Bip	300x300	X	5.40; 5.29	None; None	0/0, 0/0; 0/0, 0/0	300 ²
BAM using group code	Bip/Bip	300x7	X	4.73; 4.71	None; None	55.9/48.1, 44/52; 44.4/40.6, 56/59	300 ²
ML/BP w/o hidden layer	Analog /Bip	100x4	1981+ 864/0	X	14 dB; None	100/65.2, 0/10; 87.5/64.4, 2/11	Not Integer
ML/BP w/ 1 hidden layer	Analog /Bip	100x25 +25x4	1337+ 678/0	X	10 dB; None	100/62.5, 0/10; 90.6/60.2, 2/12	Not Integer
GI Net	Bip/Bip	300x4	10+ 120/0	X	6 dB; 20 dB	100/75.3, 0/ 5; 95.3/73.4, 3/ 6	Not Integer
HCAM (order of 3);	Bip/Bip	68x300	X	6.56; 6.22	None; None	2.79/ 0, 0/ 0; 0/ 0, 0/ 0	300 ³
HCAM (order of 5)				4.51; 7.71	None; None	94.9/57.2, 0/ 0; 32.5/28, 0/ 0	300 ⁵
ECAM	Bip/Bip	68x300	X	2.95; 3.05	-3 dB; -3 dB	100/100, 0/ 0; 100/98.3, 0/ 2	2 ³⁰⁰
RCAAAM /di22-GI	Bip/Bip	68x300	X	7.12; 9.42	3 dB; None	100/87.5, 0/12.5; 87.5/79.4, 12/20	Adaptive
RCAAAM /fi12-GI	Bip/Bip	68x300	X	7.12; 10.06	-3 dB; -3 dB	100/100, 0/ 0; 98.9/97.2, 0/ 2	Adaptive
RCAAAM /da12-GI	Analog /Bip	68x100 +68x300	X	9.71; 12.85	0 dB; 3 dB	100/99, 0/ 0; 98.8/93.1, 0/ 0	Adaptive

Table 6.8 Network architectures and performances summary for spectrum magnitude target discrimination with 5 quantization levels encoded by 3 bits.

Network	Input/ Output Form	Memory Size	Training Epoch /Error	Trained; Untrained			Max. Integer Comput. Scale
				Averg. Recog. Iterat.	Min. dB w/ 95% Correct Recog.	Percent (%) of Correct, Wrong Recog. at 40/0 dB	
Hopfield net	Bip/Bip	300x300	X	6.37; 6.19	None; None	0/0, 0/0; 0/0, 0/0	300^2
BAM using group code	Bip/Bip	300x7	X	4.17; 4.13	None; None	83.4/70, 17/30; 77.7/59.5, 22/40	300^2
ML/BP w/o hidden layer	Analog /Bip	100x4	1981+ 864/0	X	14 dB; None	100/65.2, 0/10; 87.5/64.4, 2/11	Not Integer
ML/BP w/ 1 hidden layer	Analog /Bip	100x25 +25x4	1337+ 678/0	X	10 dB; None	100/62.5, 0/10; 90.6/60.2, 2/12	Not Integer
GI Net	Bip/Bip	300x4	10+ 120/0	X	10 dB; None	100/66.3, 0/10; 89.5/63.4, 3/9	Not Integer
HCAM (order of 3);	Bip/Bip	68x300	X	5.18; 9.38	None; None	96.5/ 59.9, 0/ 0; 48.6/ 33.1, 0/ 0	300^3
HCAM (order of 5)				3.28; 3.4	0 dB; 3 dB	100/99.1, 0/ 0; 96.6/92, 3/ 8	300^5
ECAM	Bip/Bip	68x300	X	2.96; 3.05	0 dB; 0 dB	100/99.9, 0/ 0; 97.5/95.3, 3/ 5	2^{300}
RCAAAM /di22-GI	Bip/Bip	68x300	X	4.06; 5.83	0 dB; 14 dB	100/99.6, 0/ 0; 95.6/90.3, 4/10	Adaptive
RCAAAM /fi12-GI	Bip/Bip	68x300	X	5.61; 8.08	0 dB; 0 dB	100/99.7, 0/ 0; 98.4/95.2, 2/ 4	Adaptive
RCAAAM /da12-GI	Analog /Bip	68x100 +68x300	X	9.7; 13.37	0 dB; 3 dB	100/99, 0/ 0; 100/93.3, 0/ 0	Adaptive

Table 6.9 Network architectures and performances summary for spectrum magnitude target discrimination with 7 quantization levels encoded by 3 bits.

CHAPTER 7

Conclusions

We have used several different neural network architectures to discriminate among radar targets at a wide variety of aspect angles. From the simulations, it appears that correlation-based neural networks have powerful and effective problem solving abilities. Comparing the GI network to the recurrent correlation-based associative memories, we find a well trained GI network has a quite good attraction basin within which the GI net can correctly converge from a state to its associative stored pattern. But the GI network noise tolerance is inferior to the correlation-based associative memories. Typical RCAM's have off-line predetermined fixed memory, limiting the network flexibility and adaptability. Therefore, some RCAM networks may blindly reserve a huge computation space to operate in regardless of inputs, and some may not work well if used with insufficient predetermined order.

We have proposed a flexible and highly adaptive real-time learning network, the RCAAM, which has dynamic memory to allow the given input to adapt in a parallel fashion to all stored patterns, and an adjustable stable criterion to observe contamination and semi-stable states. The adaptive mechanism can be expressed from two points of view. We may say that the dynamic memory learns to adapt to the input; therefore the adaptive (or changing) direction along which the dynamic memory changes is subject to the input presented. The dynamic memory seemingly tries to find a way to compromise with the given input at a low energy-like state. In contrast, we may also say, with the memory fixed, the input keeps changing (or updating) in order to adapt to the memory constructed by stored patterns through

recurrent learnings. Hence the resources required for discrimination are dynamic and just fit to discriminate the input presented, and is no longer deterministic.

A flexible decision strategy should be considered whenever noise contamination is involved in network performance. The contamination observability of the RCAAM through semi-stable states allows us to estimate the input contamination and the corresponding network reliability (or confidence) so that we can decide to accept or discard the network discrimination output based on that information. The network may have several semi-stable states corresponding to the same input if the stability criterion is set to a low value, say 1. The spurious (unknown) semi-stable states will further converge to one of the stored patterns if the stability criterion becomes high. Thus, the discrimination decision strategy is flexible : 3 phases, correct/wrong/unknown, or 2 phases, correct/wrong. We have simplified the RCAAM iteration to an easily realizable implementation form, which speeds up the dynamic memory computations. The GI-cascaded network usually improves the performances for fair contamination situations, and saves a great deal of discrimination space. A GI network cascaded to an RCAAM with a low initial order and stable criterion is an effective and economical combination.

We have also used spectrum magnitudes as the network processing patterns. The spectrum magnitude network simulations produce excellent results for the ECAM-GI, HCAM-GI with order higher than 5, and RCAAM-GI. We find that the linearity and resolution are the most important factors when the data format is considered in network performance. We have analyzed the correlation-based discrimination resolutions for different data formats, and find that the bipolar data format usually has a better correlation-based

discrimination resolution for both time and spectral domain data. The analog data format always has better linearity than bipolar data. A nonlinear normalization with additional offsetting can convert the useless analog spectrum magnitude into useful analog data with correlation-based discrimination resolution and a correlation gain population similar to the bipolar data expressing 5 quantization levels. We find that, with coding bits fixed, the code with less quantization levels has better linearity. The RCAAM has the same or better performances, but with great space savings compared to the ECAM or the same implementation complexity as the HCAM. The bipolar process network performances show the spectral processes producing results comparable to those using time responses, although half the information has been ignored. Hence, the spectrum magnitude process is an alternative, effective and realizable technique.

Speaking of hardware implementation, the ECAM processing high dimension patterns can't be realized since the known physical materials can't represent such a huge dynamic range needed for ECAM's pattern discrimination. Although the HCAM with order higher than 5 can perform very well, the high order will reduce its updating speed. Consider an HCAM with order of 7. Since the hardware multiplication in the HCAM will be repeated 7 times between the current input and each stored pattern on each update, the time spent on correlation-gain calculation for each stored pattern for the HCAM is 6 times more than the time for the RCAAM if a control clock is used. Therefore, the HCAM with higher order is less efficient and less flexible than the RCAAM. Typically, the RCAAM/fi requires one more pattern-dimension register than the RCAAM/di in hardware realization if the input and output registers can be triggered in different directions.

Some improvements can be developed to relieve the interference resulting from the negative correlation gains. This can be done simply by adding a positive-valued offset to the correlation gain between the input and each stored pattern. However, this offset will raise the correlation mean, so the correlation-based discrimination resolution will be decreased, and then the discrimination processing time will be increased. A better improvement in evaluating the correlation gain for the stored pattern i with bipolar format is given below :

$$Gain_i = (Input \otimes Weighted\ Pattern_i) \times [Sum (Input \odot Pattern_i)] \quad (1)$$

where the $\otimes$ denotes the inner product and the $\odot$ represents the Exclusive-NOR (or Equivalence) operation, which gives an active output only if two input bits are the same. In this improvement, the negative gains will be inhibited by the bracketed term, since the correlation gain with a large negative value indicates that the different bits between the input and the evaluated pattern are more than the same bits, and then the sum between these two bipolar vectors should be small. In contrast, the bracket term will benefit the positive gains, since the positive gains indicate that more corresponding bits between the input and the evaluated pattern have the same signs, and then the sum of the Exclusive-NOR outputs should be large. The hardware implementation for an Exclusive-NOR is not difficult at all. To improve the converging consistency, a weighted previous network sum can be added to the current network sum before the nonlinear threshold function. This weighted previous network sum can be considered as a momentum to keep the converging trace consistent, and the older momenta with weighting less than 1 will gradually decay when time goes on.

Some target discrimination methods combined with other techniques, the discrete

wavelet transform and the late-time natural frequency integration filter scheme, have also been used during this research period and demonstrated very pleasing results. A perspective study will be devoted to a complicated neural network synthesis which is expected to be able to predict sequential actions and estimate its own performance confidence (or error) to a causal system. Since the updated desired outputs used for network training are temporarily generated by the synthesis network itself, the network training can converge only if the self-consistence criterion is satisfied.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1]. P. Ilavarasan, J. E. Ross, E. J. Rothwell, K. M. Chen, and D. P. Nyquist, "Performance of an automated radar target discrimination scheme using E pulses and S pulses, "IEEE Trans. Antennas Propagat., vol. 41, pp. 582-588, May 1993.
- [2]. C. E. Baum, E. J. Rothwell, K. M. Chen , and D. P. Nyquist, "The singularity expansion method and its application to target identification, "Proc. IEEE, vol. 79, no. 10, pp.1481-1492, October 1991.
- [3]. E. J. Rothwell, K. M. Chen, D. P. Nyquist, and W. M. Sun, "Frequency domain E-pulse synthesis and target discrimination, "IEEE Trans. Antennas Propagat., vol. AP-35, pp. 426-434, April 1987.
- [4]. E.M. Kennaugh, "The K-pulse concept, "IEEE Trans. Antennas Propagat., vol. AP-29, no. 2, pp. 329-331, March 1981.
- [5]. F. Y. S. Fok, "K-pulse estimation for a right-angle bent wire using more than one impulse response, "IEEE Trans. Antennas Propagat., vol. 38, pp. 1092-1098, July 1990.
- [6]. J. P. Bayard and D. H. Schaubert, "Target identification using optimization techniques, "IEEE Trans. Antenna Propagat., vol. 38, pp. 450-456, April 1990.
- [7]. M. A. Morgan, "Scatterer discrimination based upon natural resonance annihilation, "J. Electromagnetic Waves Appl., vol. 2, no. 5/6, pp. 481-502, 1988.
- [8]. D. G. Dudley and D. M. Goodman, "Transient identification and target classification" in *Time-Domain Measurements in Electromagnetics*, E. K. Miller, Ed. New York : Van Nostrnad-Reinhold, 1986, pp. 456-497.
- [9]. C. W. Chuang and D. L. Moffatt, "Natural resonances of radar targets via Prony's method and target discrimination, "IEEE Trans. Aerosp. Electron. Syst., vol. AES-12, pp. 583-589, Sept. 1986.
- [10]. E. J. Rothwell, K. M. Chen, D. P. Nyquist, and J. E. Ross, "Time-domain imaging of airborne targets using ultra-wideband or short-pulse radar, "IEEE Trans. Antennas Propagat., vol. 43, no. 3, March 1995.

- [11]. C. L. Bennet, "Time-domain inverse scattering, "IEEE Trans. Antennas Propagat., vol. AP-29, no. 2, pp. 213-219, March 1981.
- [12]. J. D. Young, "Radar imaging from ramp response signatures, "IEEE Trans. Antennas Propagat., vol. AP-24, pp. 276-282, May 1976.
- [13]. M. C. Lin and Y. W. Kiang, "Target discrimination using multiple-frequency amplitude returns, "IEEE Trans. Antennas Propagat., vol. 38, pp. 1885-1889, Nov. 1990.
- [14]. M. C. Lin, Y. W. Kiang and H. J. Li, "Experimental discrimination of wire stick targets using multiple-frequency amplitude returns, "IEEE Trans. Antennas Propagat., vol. 40, pp. 1036-1040, Sept. 1992.
- [15]. H. J. Li and S. H. Yang, "Using range profiles as feature vectors to identify aerospace objects, "IEEE Trans. Antennas Propagat., vol. 41, pp. 261-268, March 1993.
- [16]. E. J. Rothwell , K. M. Chen, D. P. Nyquist, J. E. Ross, and R. Bebermeyer, "A radar target discrimination scheme using the discrete wavelet transform for reduced data storage, "IEEE Trans. Antennas Propagat., vol. 42, no. 7, pp. 1033-1037, July 1994.
- [17]. H. V. Poor, "An Introduction to Signal Detection and Estimation", Springer-Verlay, New York, 1988.
- [18]. T. Kohonen, "Self-Organization and Associative Memory", Springer Verlag, Berlin, 1989.
- [19]. Bart Kosko, "Neural Networks for Signal Processing", Pretice-Hall, Inc. 1992.
- [20]. Robert Schalkoff, "Pattern Recognition : Statistical, Structural and Neural Approaches", John Wiley & Sons, Inc. , 1992.
- [21]. John J. Craig, "Introduction to Robotics -mechanics and control", Addison-Wesley, 1989.
- [22]. R. G. Atkins, R. T. Shin and J. A. Kong, "Chapter 7 : A Neural Network Method for High Range Resolution Target Classification", Progress in Electromagnetics Research 4, J. A. Kong, Ed., Elsevier, New York, 1991.
- [23]. H. J. Li and V. Chiou, "Aerospace Target Identification-Comparison between the Matching Score Approach and the Neural Network Approach", Journal of Electromagnetic Waves and Applications, Vol. 7, No. 6, 873-893, 1993.
- [24]. John Edwin Ross III, "Chapter 5 Transient Measurements", PhD. Thesis : Application

of Transient Electromagnetic Fields to Radar Target Discrimination, 1992.

[25]. Minsky, M., and Papert, S. *Perceptrons : An Introduction to Computational Geometry*. MIT Press, Cambridge, 1969.

[26]. Widrow, B., and Hoff, M.E. *Adaptive Switching Circuits*. Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, part4, 1960, 96-104.

[27]. Werbos, P. J. *Beyond Regression : New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph.D. thesis, Harvard University, 1974.

[28]. Parker, D. B. *Learning Logic*. Invention Report, S81-64, File 1, Office of Technology Licensing, Stanford University, 1982.

[29]. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. *Learning Internal Representations by Error Propagation*. In Rumelhart, D. E., and McClelland, J. L. (eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. 1. MIT Press, Cambridge, 1986.

[30]. J. Hertz, A. Krogh and R. G. Palmer, *Introd. to the Theory of Neural Computation*, Addison-Wesley, 1991.

[31]. Y. Kamp and M. Hasler, "Recursive Neural Networks for Associative Memory" (John Willy & Sons Ltd.) 1990.

[32]. M. H. Hassoun, "Dynamic Associative Memories" in *Artificial neural networks and stochastic pattern recognition old and new connections*, pp.195-218, 1991.

[33]. M. H. Hassoun, "Associative Neural Memories, Theory and Implementation", Oxford University Press, 1993.

[34]. D. Hebb, *Organization of Behavior*, John Wiley & Sons., N.Y., 1949.

[35]. J.J. Hopfield, "Neural network and physical systems with emergent collective computational", *Proc. Nat. Acad. Sci. U.S.* vol.79, pp. 2554-2558, 1982.

[36]. J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons", *Proc. Nat. Acad. Sci. U.S.* vol.81, pp.3088-3092, 1984.

[37]. McEliece, R. J., Posner, E. C., Rodemich, E. R., and Venkatesh, S. S. (1987). "The Capacity of the Hopfield Associative Memory," *IEEE Trans. Info. Theory*, IT-33, 461-482.

[38]. B. Kosko, "Bidirectional Associative Memories", *IEEE Trans. Sys. Man Cybern.*,

SMC-18, pp.49-60, 1988.

[39]. Y. F. Wang, J. B. Cruz, Jr. and J. H. Mulligan, Jr., "Guaranteed Recall of All Training Pairs for Bidirectional Associative Memory" IEEE Trans. on Neural Networks vol.2 no.6 pp.559-567, 1991.

[40]. T. D. Chiueh and R. M. Goodman, "High-Capacity Exponential Associative Memory" in Proc. IEEE Int. Conf. Neural Networks (San Diego, CA), vol. 1, pp.153-160,1988.

[41]. T. D. Chiueh and R. M. Goodman, "Recurrent Correlation Associative Memories" IEEE Trans. on Neural Networks, vol.2 no. 2, pp.275-283, 1991.

MICHIGAN STATE UNIV. LIBRARIES



31293014027241